HISTO CRYPT 2026

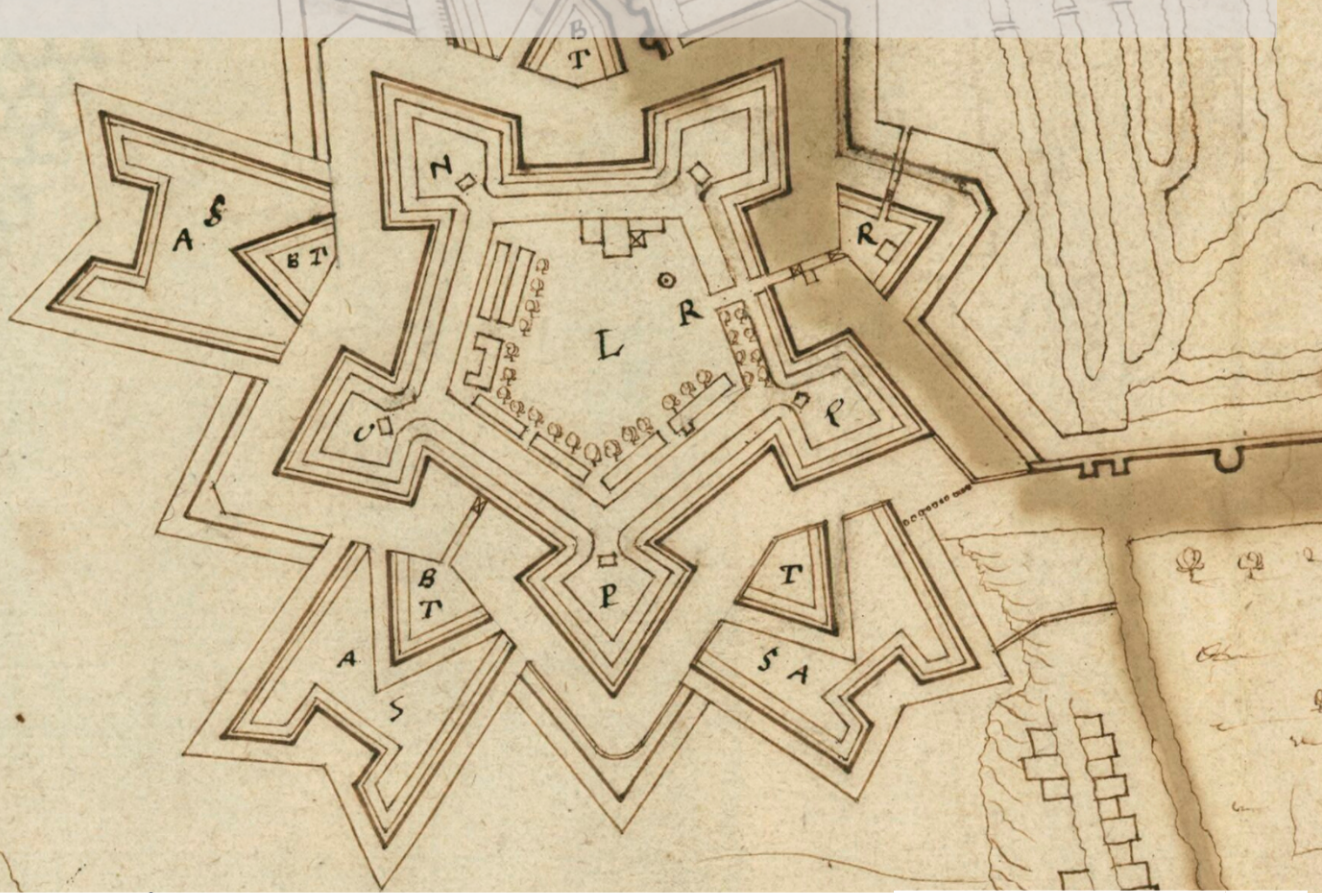
THE 9TH INTERNATIONAL CONFERENCE ON HISTORICAL CRYPTOLOGY

date of event

JUNE 22 - 24

event's location

AMIENS, FRANCE



MAP OF THE CITY OF AMIENS WITH ITS FORTIFICATIONS, 17TH CENTURY, BIBLIOTHÈQUE NATIONALE DE FRANCE, GAIGNIÈRES, 5931. SOURCE GALLICA.BNF.FR

UNIVERSITÉ de Picardie Jules Verne

CHSSC UR UPJV 4289
Centre d'histoire des sociétés, des sciences et des conflits

AMIENS MÉTROPOLE

Inria

PROCEEDINGS OF THE 9TH INTERNATIONAL CONFERENCE ON HISTORICAL CRYPTOLOGY

Histocrypt 2026

Editors:

Camille Desenclos

Cécile Pierrot

Published by

NEALT Proceedings Series Number 61

D-Space at Tartu University Library

ISSN: 1736-8197

ISSN: 1736- 6305

ISBN: 9789908539997

Preface

The programme committee is delighted to present the proceedings of the 9th International Conference on Historical Cryptology (HistoCrypt 2026). The conference will be held on June 22-24, 2026 in Amiens, France. It could not take place without the support of the Université de Picardie Jules Verne, of the CHSSC (Centre d'histoire des sociétés, des sciences et des conflits), of Inria, of Amiens Métropole and of the MESHS (Maison Européenne des Sciences Humaines et Sociales).

Following the tradition of previous HistoCrypt conferences, HistoCrypt 2026 addresses all aspects of historical cryptography and cryptanalysis and is inherently cross-disciplinary, incorporating work from various fields such as AI, computational linguistics, computer science, history, image processing, linguistics and mathematics. This year's topics demonstrate growth in the application of AI in historical cryptology and the vivid history of cryptology, with 11 papers on each topic. On the contrary, only a few papers dedicated to unsolved historical cryptograms. All in all it resulted in five different talk sessions: "From transcription to decryption: AI in action" ; "Early modern cryptology: practices" ; "Early modern cryptology: influences" ; "Processing cryptology" and "20th century cryptology". A poster session was organised for the presentation of tools and experiments.

The scientific programme was compiled by an international scientific programme committee consisting of researchers in computer science, computer vision, cryptology, digital humanities, history, language technology, and (computational) linguistics. The programme committee welcomed submissions in two tracks: regular papers up to 10 pages (excluding references and appendices) on substantial, original, and unpublished research, including evaluation results, where appropriate; and short papers up to 4 pages (excluding references and appendices) on smaller, focused contributions, work in progress, negative results, surveys, tutorials, or opinion pieces. An additional page was sometimes granted following the peer-review process to ensure that the reviewers' expectations were met. Along with HistoCrypt 2024, this year's conference received one of the highest number of submissions. The 39 submissions are a 60% increase compared to the average of 24 submissions in previous HistoCrypt conferences from 2018 to 2023 and in 2025. These submissions came from all over the world, including from Algeria, Canada, France, Germany, Hungary, Israel, Italy, Netherlands, Nigeria, Slovakia, Spain, Sweden, United Kingdom, United States.

The programme committee aimed to compile a high-quality programme with a wide variety of topics by conducting a double-blind review process. Each submission was evaluated by at least three expert reviewers in the corresponding field. The reviews were synchronized and, if necessary, thoroughly discussed among the reviewers and area chairs of the programme committee. The final selection was based on their recommendations and discussions. In the end, we accepted : 19 regular papers and 8 short papers for publication, resulting in a total of 27 papers (69% acceptance rate). All these 27 accepted submissions

are included in this volume, organised in line with the conference programme into five sessions of presentations, in addition to the posters. For the conference in Amiens, we have invited four keynote speakers who have graciously ´ accepted our invitation. With two keynotes on the topic and a full-day excursion related to it, the conference's focus this year is on World War I. The four keynote speakers are:

- **Agathe Couderc, "Swiftness, Accuracy, and Secrecy: the Entente Cipher Bureaus' cooperation during World War I"**. Agathe Couderc is a Lecturer in Intelligence at the Conservatoire national des arts et métiers (Cnam, Paris). She is a historian of contemporary intelligence, especially in the First World War.

  Abstract: During the First World War, signals intelligence (SIGINT) emerged as a significant tool for informed decision-making. It relied on three main fields: interception of messages, direction-finding, and cryptology, the latter being the speciality of the Cypher Bureaus. Such units were to break the codes and ciphers of the enemy and assumed responsibility for the communication security.

  In the early 1910s, France and the United Kingdom dealt with SIGINT in a defensive way: they produced common cryptographic systems to ensure the secrecy of their communications in the impending war. From the outset of the Great War, their Cypher Bureaus worked closely together in attacking the enemy encrypted messages, to ascertain the intentions of the Central Powers. Throughout the war, these Bureaus increased their workforce, their units, and their efficiency.

  This keynote aims at encapsulating the main objectives of such a cooperation. The Entente code-makers and cryptanalysts were engaged in a continuous race with the Central Powers, concerning both aggressive and defensive cryptographic issues. Precision was paramount when trying to reconstruct a code book or identify a cypher, in order to provide an accurate insight into the enemy's situation. Furthermore, discretion on their results was crucial, and they had to trust that their allies would safeguard their secrets. This five-year collaboration played a considerable part in the final victory of the Entente and its allies and illuminates the intricacies of intelligence collaboration within a coalition force.

- **Daniel Larsen, "Codes, Ciphers, and Cultures of Secrecy: The Dawn of US Diplomatic Secrecy during the First World War"**. Daniel Larsen is Lecturer of Intelligence and War Studies at the University of Glasgow. He is a historian of American and British foreign policy and intelligence in the first half of the 20th century.

  Abstract: As important as codes and codebreaking are in their own right, the history of cryptology can also provide us with a profoundly useful window into wider cultures of secrecy within government. American diplomatic codes were profoundly

vulnerable before and during the First World War – the British had easily solved the US codebooks by late 1915 – but this cryptographic vulnerability offers us a much more complex story than simple arrogance or incompetence, as one might initially suppose. Rather, American diplomats operated in a wider cultural context where secrecy was seen as necessary only for matters of "national defence", a concept that decidedly did not include diplomacy. Before 1914, Americans actually practiced an extraordinary diplomatic transparency, with the State Department routinely published its own communications for decades. The exigencies of the First World War brought about a new culture of secrecy in American diplomacy, which is well reflected in a shift in State Department attitudes towards their codes and ciphers, and this set the stage for the secrecy practices associated with "national security" today.

- **Nagwa Metwally, "Cryptographic Writing in Ancient Egypt: A Historical and Functional Study"**. Nagwa Metwally is director general at the General Administration of Scientific Publishing Suprem Council of Antiquities (Minister of Tourism and Antiquities, Egypt). She holds a PhD in Egyptian Archaeology and her fields of expertise include Ancient Egyptian language and scripts, cryptographic and enigmatic writing and epigraphy.

  Abstract: The ancient Egyptians developed a writing system aimed at obscuring meaning. The roots of this enigmatic writing can be traced to the early stages of the Egyptian language. Its use continued until the end of the Egyptian linguistic tradition and is attested in the last known hieroglyphic text dated to 394 A.D. Cryptographic writing was widely employed during the New Kingdom and later became a characteristic feature of temple inscriptions in the Graeco-Roman Period. This mysterious style of writing is referred to as "cryptography" or "enigmatic writing". In Arabic, it is known as al-kitābah al-muammah. The use of cryptography was not confined to temples; it also appeared in tombs and on various objects. The primary purpose of cryptographic writing was to emphasize the significance of specific texts. Certain meanings were rendered in unusual or complex forms within otherwise plain texts to highlight their sacred or symbolic value. Over time, cryptography served additional purposes, including the concealment of magical formulas and the restriction of specialized knowledge related to narcotics and toxic substances. Moreover, individuals employed cryptographic writing to secretly record personal wishes, and in some cases, to write their names in a concealed manner in the belief that this would grant immortality and protection from harm. By the 3rd–4th centuries A.D., this writing system had also been adopted by monks in their correspondence. In addition to the above-mentioned functions, cryptographic writing served a wide range of purposes.

  Egyptian cryptography introduced hundreds of new signs, numerous orthographic

variations, and unconventional renderings of personal names, place names, and pronouns. These elements operated according to specific cryptographic principles, including direct pictorial representation, the rebus principle, acrophony, the principle of consonants, the interchange of signs, changes in the order and placement of signs, and phonetic alteration. This keynote conference examines selected examples of cryptographic words and writing elements, demonstrating how their multiple phonetic values were transmitted across different periods of Egyptian writing, culminating in the Graeco-Roman era. Egyptian cryptography thus constitutes a significant historical source, offering valuable insights into the cultural, religious, intellectual, and political dimensions of ancient Egyptian civilization, and underscoring its importance for the field of Egyptology.

- **Peter Stokes, "Unsolved problems in HTR: Insights from eScriptorium"**. Peter Stokes is directeur d'études at the École Pratique des Hautes Études – Université PSL (Paris, France), specialising in digital and computational approaches to palaeography. He is the Co-Director of the eScriptorium platform for Automatic Text Recognition.

  Abstract: There is little question of the very significant advances that machine learning and other digital and computational methods have brought to manuscript studies, of which HTR (handwritten text recognition) is a clear example. Tools are now readily available that can successfully treat many different cases of historical writing, supported by rapidly increasing quantities of training data for different writing systems including Latin, Greek, Hebrew, Arabic, Cyrillic, Syriac, Georgian, Chinese, Japanese, and many others. However, this is by no means to claim that the problem is "solved", as there are still significant challenges that remain. In some cases the problem is one of training data, and therefore the solution seems clear: we simply need to produce more data. However, this "simple" solution hides a multitude of difficulties. How can we produce homogeneous standard-compliant data for all different types of documents? Is it possible or even desirable to have a single standard of transcription that can meet all the different reasons why one might transcribe a document? Other problems are more technical, such as how to encode all the variety of signs that one might want to transcribe in all the world's history of writing, how to manage different directionalities and non-linearity in writing. Other challenges go to the very foundation of what writing is, such as how to decide the boundary between the graphetic as the graphemic in transcription. In this keynote, I will therefore present these difficulties as they have arisen in the eScriptorium projects, considering how we have addressed them and what problems still remain.

Organizing a conference and a peer-review process always relies on the goodwill and support of many colleagues to take their valuable time and contribute to an interesting and fruitful programme. Our special thanks goes to all area chairs of the programme committee:

Eugen Antal, John F. Dooley, Alicia Fornés and Beáta Megyesi for their substantial support and for many constructive online-meetings. We also want to thank the 34 reviewers for their time, effort, and constructive feedback during the review process. In addition, we thank all the authors who have made these proceedings possible. Lastly, we would like to express our gratitude to Hubert Weikert for managing the conference website.

Nancy, 4 June 2026

Camille Desenclos and Cécile Pierrot
General Chairs of HistoCrypt 2026

Programme Committee

Eugen Antal: Slovak University of Technology in Bratislava, Slovakia.
Raphaël Baena: Ecole Nationale des Ponts et Chaussées, France.
Elisa H. Barney Smith: Luleå University of Technology , Sweden.
Corinne Bayerl: University of Oregon, USA.
Richard Bean: University of Queensland, Australia.
Meriem Beloucif: Uppsala University, Sweden.
Micaella Bruton: Stockholm University, Sweden.
Chris Christensen: Northern Kentucky University, USA.
Carola Dahlke: Deutsches Museum, Germany.
Camille Desenclos: Université de Picardie Jules Verne, France.
Jörgen Dinnissen: Independent researcher, The Netherlands.
John F. Dooley: Knox College, USA.
Magnus Ekhall: Independent researcher, Sweden.
Bernhard Esslinger: Universität Siegen, Germany.
Alicia Fornés: Universitat Autònoma de Barcelona, Spain.
Joachim von zur Gathen: Universität Bonn, Germany.
Raphaela Heil: Stockholm University, Sweden.
Benjamin Kiessling: Inria Paris, France.
Kevin Knight: Threeven Labs, USA.
Nils Kopal: Hochschule Niederrhein Krefeld, Germany.
Benedek Lang: Eötvös Loránd University, Budapest, Hungary.
George Lasry: Independent researcher, Israel.
Beáta Megyesi: Stockholm University, Sweden.
Vasily Mikhalev, University Mannheim, Germany.
Christian Millichap: Furman University, USA.
Jakub Mírka: State Regional Archives in Pilsen, Czech Republic.
Jessika Nowak: Bergische Universiät Wuppertal, Germany.
Eva Pettersson: Uppsala University, Sweden.
Eva Pich Ponce: Universidad de Sevilla, Spain.
Cécile Pierrot: Inria Nancy, France.
Léo Robert: Université de Picardie Jules Verne, France.
Yann Rotella: Université de Versailles, France.
Anne-Simone Rous: Saxon State Palaces, Castles and Gardens, Germany.
Klaus Schmeh: Independent researcher, Germany.
Dermot Turing: Visiting Fellow, Kellogg College, University of Oxford, UK.
Florentijn van Kampen: Radboud University, The Netherlands.
Marco Vito: University of Wien, Austria.

Michelle Waldispühl: Universitetet i Oslo, Norway.
Pavol Zajac: Slovak University of Technology in Bratislava, Slovakia
Paul Zimmermann: Inria Nancy, France.

Area Chairs for HistoCrypt 2026

Eugen Antal (area chair: cryptanalysis), Slovenská technická univerzita v Bratislave (Bratisvala, Slovakia)
Camille Desenclos (area chair: early modern history of cryptology), Université de Picardie Jules Verne (Amiens, France)
John F. Dooley (area chair: modern history of cryptology, security), Knox College (Galesburg, USA)
Alicia Fornés (area chair: computer vision), Universitat Autònoma de Barcelona (Barcelona, Spain)
Beáta Megyesi (area chair: computational linguistics), Stockholms universitet (Stockholm, Sweden)
Cécile Pierrot (area chair: cryptanalysis), Université de Lorraine, CNRS, Inria (Nancy, France)

Steering Committee

Carola Dahlke (chair), Deutsches Museum, Germany
Benedek Láng (vice chair), Budapest University of Technology and Economics, Hungary
Richard Bean (secretary), University of Queensland, Australia
Dermot Turing (member), Kellogg College Oxford, UK
Camille Desenclos (member), Université de Picardie Jules Verne, France

Local Organisation and General Chairs 2026

Camille Desenclos (programme co-chair), Université de Picardie Jules Verne (Amiens, France)
Cécile Pierrot (programme co-chair), Université de Lorraine, CNRS, Inria (Nancy, France)

Contents

Joint Transcription and Decryption of Images of Encrypted Handwritten Documents: A Comparison with the Traditional Pipeline

Marino Oliveros-Blanco
Lei Kang
Alicia Fornés
Computer Vision Center
Department of Computer Science
Universitat Autònoma de Barcelona, Spain
marino.oliverosblanco@gmail.com
{lkang,afornes}@cvc.uab.es

Beáta Megyesi
Department of Linguistics
Stockholm University, Sweden
beata.megyesi@ling.su.se

Abstract

Historical encrypted manuscripts present a challenging problem at the intersection of cryptology, linguistics, paleography, and computer vision. Current automatic decipherment approaches usually rely on a two-stage pipeline: transcription of cipher symbols from manuscript images, followed by decryption into plaintext. However, this design is sensitive to transcription errors, which propagate to the final output. We present Direct Image Decryption, an end-to-end approach that directly maps encrypted manuscript images to plaintext, bypassing the intermediate transcription stage. Using the Copiale cipher as a case study, we build a synthetic data generation pipeline to create large-scale cipher-like training data and compare the traditional pipeline with the proposed joint architecture. Results show that joint image-to-plaintext modeling is a promising alternative to traditional transcription-based pipelines.

1 Introduction

Historical ciphers constitute an important part of the written record of early modern and modern Europe, appearing in diplomatic correspondence, private letters, intelligence reports, and the papers of learned and secret societies. Many such documents remain only partially studied, not because they have been lost, but because their encrypted content is still difficult to access. Their analysis requires expertise from several fields, including cryptology, linguistics, paleography, history, and, increasingly, computer vision.

In practice, the decipherment of encrypted manuscripts usually follows a two-stage process. First, the cipher symbols are transcribed from the image into a textual representation. Second, the resulting sequence is subjected to cryptanalytic and linguistic analysis to recover the plaintext and reconstruct the underlying encryption system. However, this sequential pipeline has important limitations: Errors introduced during transcription propagate directly to the decryption stage, compromising the final output. In addition, transcription often requires considerable manual effort and specialist knowledge, which restricts the scalability of existing approaches.

Motivated by these limitations, this work investigates an alternative approach, which we call *Direct Image Decryption*. Rather than treating transcription and decryption as separate tasks, this approach learns a direct mapping from images of encrypted handwritten text to decrypted plaintext within a single end-to-end model. The goal is to reduce the impact of transcription errors, alleviate the manual bottleneck associated with symbol transcription, and exploit visual information that may be lost when manuscript images are first converted into discrete symbol sequences. To examine these possibilities, we implement and compare two deep learning architectures: a traditional two-stage transcription–decryption pipeline and a joint image-to-plaintext model.

We evaluate both approaches on real and synthetically generated data. As a case study, we focus on the well-known Copiale cipher. This 18th-century manuscript, discovered in Germany, consists of 105 pages written in a large set of symbols and abstract glyphs. The system is a homophonic substitution cipher in which individual plaintext letters are represented by multiple cipher symbols

drawn from an alphabet of approximately 100 distinct glyphs. The manuscript was deciphered in 2011 (Knight et al., 2011), showing that the text encoded German and described rituals associated with Freemasonry and a secret society known as the *Oculists*.

It must be noted that our model is trained to decrypt this specific substitution system rather than to perform fully cipher-agnostic decryption. However, our framework could be applied to other encrypted handwritten sources.

2 Related Work

Work on historical ciphers has increasingly combined traditional cryptanalytic scholarship with computational methods. In case of historical encrypted manuscripts, this often implies the transcription of cipher symbols from document images, and the subsequent decipherment of the resulting symbol sequences.

An example in computational historical cryptology is the decipherment of the Copiale manuscript by Knight et al. (2011), which showed how large historical ciphers can be approached through a combination of transcription, statistical language modeling, clustering, and cryptanalytic analysis. Similar workflows have also been applied to other historical ciphers. At the same time, these studies make clear that transcription remains a substantial practical challenge. For Copiale, manual transcription reportedly required around 35 minutes per page once the symbol inventory had been identified, and considerably longer otherwise (Knight et al., 2011). Dinnissen and Kopal (2021) likewise describe the significant effort required to transcribe the Ramanacoil manuscript before any decipherment could be attempted. This dependence on high-quality transcription continues to shape the feasibility of computational work on historical ciphers.

On the decipherment side, there was a growing interest in neural approaches. Kambhatla et al. (2018) introduced neural language-model-based methods with beam search for cipher decipherment. Aldarrab and May (2021) proposed transformer-based multilingual models that achieved strong results on both synthetic and handwritten ciphers. Aldarrab (2022) further explored neural end-to-end settings and highlighted the continuing difficulty of transferring models trained on synthetic data to real manuscripts.

In parallel, advances in handwritten text recognition have influenced the treatment of encrypted manuscripts. Convolutional Recurrent Neural Networks (CRNNs) trained with Connectionist Temporal Classification (CTC) loss (Graves et al., 2006) have become a standard architecture for sequence recognition from images (Shi et al., 2017). They eliminate the need for explicit character segmentation, useful when the segmentation of symbols is difficult. Yin et al. (2019) addressed automatic segmentation, glyph recognition and transcription for historical ciphers, emphasizing the challenges posed by non-standard alphabets, degraded images, and the absence of lexical constraints such as dictionary-based error correction. Bluche et al. (2017) introduced an end-to-end model combining LSTMs with attention mechanisms for paragraph-level recognition. More recently, the ICDAR Competition on Handwriting Recognition of Historical Ciphers (Fornés et al., 2024) has provided standardized shared datasets and evaluation protocols, enabling more systematic comparison of methods.

Despite the progress, current neural decryption methods still depend fundamentally on accurate transcription, which creates error propagation that directly affects decryption performance. More broadly, most approaches continue to treat transcription and decipherment as separate tasks. As a result, overall system performance remains closely tied to transcription quality, while the scarcity of annotated historical material continues to encourage heavy reliance on synthetic data.

We believe three fundamental limitations of current approaches persist:

1. Error propagation: Errors from transcription to decryption compound inaccuracies and limit overall system performance. A single transcription error can cascade through the decryption process and corrupt surrounding text interpretation.

2. Transcription bottleneck: This process requires substantial manual effort or large quantities of training data. For many historical ciphers, neither sufficient manpower nor annotated data is readily available. The scarcity of real manuscript data, rarely exceeding a few thousand lines, necessitates heavy reliance on synthetic data generation, which often fails to capture authentic manuscript complexity. This limitation makes decrypting documents such as the Voynich Manuscript (Clemens,

2016), part of the Zodiac Killer Ciphers (particularly the brief Z13 and Z32), or the Beale Ciphers (Gillogly, 1980) particularly challenging.

3. Synthetic-to-real gap: While synthetic data enables training of large-scale models, it typically fails to replicate exact characteristics of historical manuscripts, including aging effects, ink degradation, writing style variations, and document damage. As a result, models trained primarily on synthetic data often exhibit degraded performance on original historical ciphers.

These limitations motivate our Direct Image Decryption approach, which bypasses transcription entirely and directly maps visual features to decrypted plaintext, representing a fundamental departure from established methodologies.

3 Synthetic Data Generation

The scarcity of annotated historical encrypted manuscripts presents the most fundamental challenge for training deep learning models. With only approximately 2,000 segmented line images available from the original Copiale manuscript (Fornés et al., 2024), we developed a comprehensive synthetic data generation pipeline to produce training samples of sufficient quantity and quality for robust model development of both the transcription-decryption pipeline and the Direct Image Decryption model. This pipeline takes lines of text as input and generates augmented images of the original text encoded into Copiale, along with the transcriptions and original decrypted plaintext.

The present work builds on these developments by exploring a joint formulation in which encrypted manuscript images are mapped directly to plaintext, allowing transcription and decryption to be learned within a single model.

Our synthetic data must satisfy three requirements: visual similarity to the Copiale manuscript, including appropriate symbol shapes, spacing, and overall appearance; linguistic patterns reflecting 18th-century German, as this was the language encoded in the original cipher; and realistic degradation effects—including noise, ink variations, and aging marks—to reduce the gap between synthetic samples and authentic historical documents.

3.1 Text Source Selection

We selected historical German texts chronologically and stylistically aligned with Copiale. Our primary corpus contains four major works: Goethe's *Faust* (1808-1832), Kant's *Critique of Pure Reason* (1781), *the Lutheran Bible* (1760 revision), and Adalbert Stifter's *Nachsommer* (1857), providing over 115,000 lines of period-appropriate German text. The dataset generated from this corpus is called the "Faust" dataset.

Text preprocessing filtered the corpus to retain only the 106 characters present in the original cipher, removing modern punctuation and uncommon characters. Line segmentation ensured generated images contained 12-40 characters, matching the length distribution observed in the original manuscript. For additional evaluation, we generated datasets using English texts (*American Psycho* by Bret Easton Ellis, *East of Eden* by John Steinbeck) and Old German poetry (*Hymns of the Night* by Novalis, 1,300 images).

3.2 Visual Representation and Augmentation

The encoding employs the *"Copiale.ttf"* font file, which maps standard Unicode characters to cipher glyphs, enabling automatic generation of cipher-like text from plaintext input. This encoding follows the fixed substitution key of the Copiale cipher specifically; all models trained on this synthetic data learn to decrypt this particular key rather than performing general cipher-agnostic decryption. The mapping process utilizes the vocabulary file from the DECRYPT project (Megyesi et al., 2020), ensuring our synthetic data maintains the same symbol-to-meaning relationships as the authentic manuscript data.

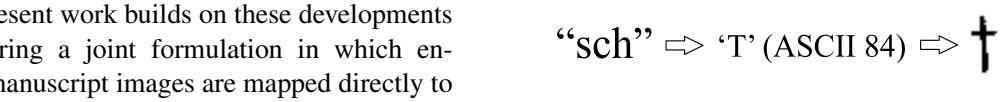


Figure 1: Input to Copiale encoding

Initial image generation produces clean renderings, just encoded text with an applied font. To obtain visual similarity to the aged manuscripts, we apply comprehensive augmentation: **Degradation effects** include Gaussian noise (paper texture, scanning artifacts), random erosion and dilation (ink spread, fading), gamma correction (brightness variations), and Kanungo noise patterns (spots, stains, fiber patterns). **Geometric transformations** apply random rotation (±3 degrees), shearing (perspective distortions), random scaling, and random cropping.

The augmentation parameters were carefully tuned to produce realistic variations as similar

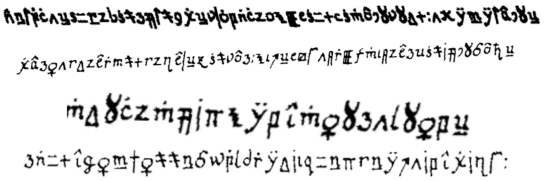


Figure 2: Different augmentation effects

as possible to the original manuscript, balancing degradation intensity to avoid over-distortion while capturing authentic aging characteristics. Figure 2 shows the range of augmentation effects obtainable through our pipeline.

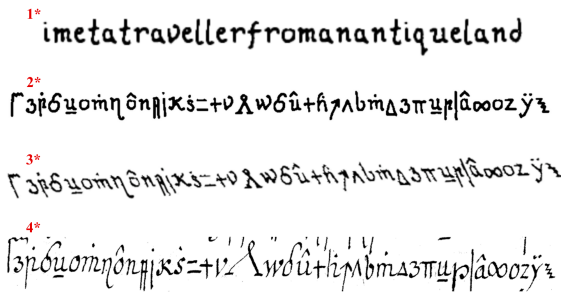


Figure 3: Comparison of 1* Plaintext in Copiale font, 2* Encoded text (non-augmented), 3* Encoded text (augmented), and 4* Original manuscript image.

3.3 Dataset Statistics

Our primary synthetic dataset, "Faust", comprises 115,000 line images with 80/10/10 split for training, validation, and testing. The Copiale dataset consists of 2,000 grayscale images with transcription and decrypted plaintext. Figure 3 shows that augmentation effectively achieves correct symbol morphology, spacing patterns, and degradation effects; although real manuscripts exhibit more pronounced historical wear, and paper-like effects. Vocabulary distribution in our synthetic data closely resembles the original manuscript (Figure 4), ensuring models encounter symbol patterns consistent with real manuscript images.

4 Transcription & Decryption Pipeline

The two-stage pipeline represents the traditional approach to the decipherment of encrypted manuscripts, consisting of first transcription followed by decryption, as shown in Figure 5.

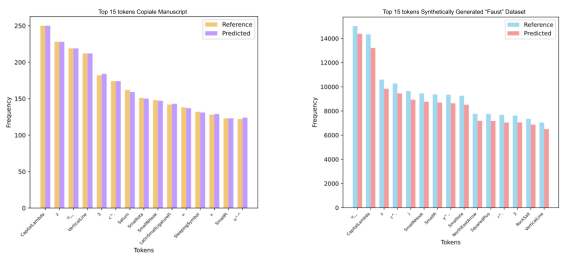


Figure 4: Token frequency: Copiale dataset vs. Synthetically generated "Faust" dataset.

4.1 Stage 1: Transcription

The transcription stage converts images of cipher manuscripts into sequences of symbol tokens. Obviously, achieving high-quality transcription is critical for the posterior decryption.

Model Architecture

The transcription model employs a Convolutional Recurrent Neural Network (CRNN) architecture with Connectionist Temporal Classification (CTC) loss (Shi et al., 2017). Input images are resized to 64 pixels height, padded/cropped to 800 pixels width, and normalized to [0,1].

CNN Feature Extractor: Four convolutional blocks (1→64→128→256→256 channels) progressively extract hierarchical features with 3×3 convolutions, batch normalization, ReLU activation, and pooling (2×2, 2×2, 2×1, none). Output feature maps are reshaped to (batch_size, width/4, 2048).

Bidirectional LSTM: 4 layers with 256 hidden units per direction process CNN features bidirectionally to disambiguate visually similar symbols. Dropout (0.5) provides regularization, producing 512-dimensional vectors per timestep.

CTC Classification: The vocabulary comprises 132 Copiale cipher tokens plus special tokens (blank, padding, UNK). Greedy CTC decoding selects the most probable token at each timestep, then collapses repetitions and removes blanks. The CTC loss (Graves et al., 2006) enables alignment-free training by marginalizing over all possible alignments:

$$\mathscr{L}_{\text{CTC}} = -\log P(y|x) = -\log \sum_{\pi \in \mathscr{B}^{-1}(y)} \prod_{t=1}^{T} p_t(\pi_t|x) \tag{1}$$

where y is the target symbol sequence, x is the

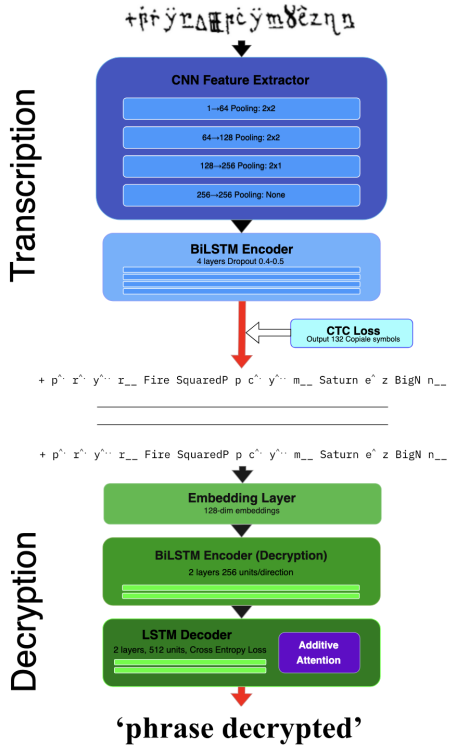


Figure 5: Two-stage pipeline: image → transcription + transcription → decryption → plaintext.

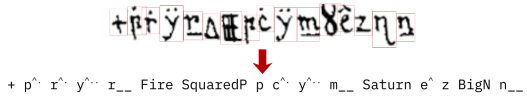


Figure 6: Example of successful transcription, in which the input image is converted into a cipher symbol token sequence.

input image, T is the sequence length, π is an alignment path, $\mathscr{B}^{-1}(y)$ is the set of valid alignments, and $p_t(\pi_t|x)$ is the probability of symbol π_t at timestep t.

Training Configuration

AdamW optimizer (Loshchilov and Hutter, 2019) with learning rate 3×10^{-4}, weight decay 1×10^{-5}, batch size 8, gradient clipping (max norm 1.0), and ReduceLROnPlateau scheduler (factor 0.1, patience 5 epochs). Training runs 100 epochs with early stopping.

4.2 Stage 2: Decryption

The decryption stage consists of taking transcribed symbol sequences and generating decrypted German plaintext using a sequence-to-sequence architecture with attention mechanism.

Model Architecture

It consists of an Encoder-decoder architecture with additive attention. The encoder uses bidirectional LSTM (2 layers, 256 units per direction) to create contextual representations. The decoder generates plaintext character-by-character through matching LSTM architecture, with attention focusing on relevant encoded cipher portions. Character-level embeddings of dimension 128 for both input and output. Output projection maps decoder states to German alphabet vocabulary (uppercase, lowercase, special characters, control tokens: SOS, EOS, PAD, UNK). The decryption model uses the same vocabulary structure as transcription for its input, ensuring seamless integration with approximately 100-200 cipher symbol tokens depending on the dataset.

Training Strategy

AdamW optimizer (learning rate 1×10^{-3}, weight decay 1×10^{-4}, dropout 0.4), batch size 16, 15-35 epochs with early stopping based on validation edit distance. The decryption model is trained from scratch on synthetic data with no external pretraining or pretrained language model weights. ReduceLROnPlateau scheduler reduces learning rate when validation plateaus. Teacher forcing is annealed linearly from 1.0 to 0.0 across epochs, and gradient clipping is applied at max norm 1.0.

5 Direct Image Decryption

Our joint architecture, namely, Direct Image Decryption, learns direct image-to-plaintext mapping in a single end-to-end model. By jointly optimizing visual feature extraction and decryption without intermediate discrete decisions, the model discovers which visual features are most relevant for decipherment, addressing the limitations outlined in Section 2: error propagation, information loss during symbolic conversion, and architectural complexity of maintaining separate models.

5.1 Architecture Overview

The architecture comprises a CRNN feature extractor producing sequential visual representations, and an attention-based decoder generating plaintext characters autoregressively. The critical difference from the two-stage pipeline: intermediate representations remain continuous—the

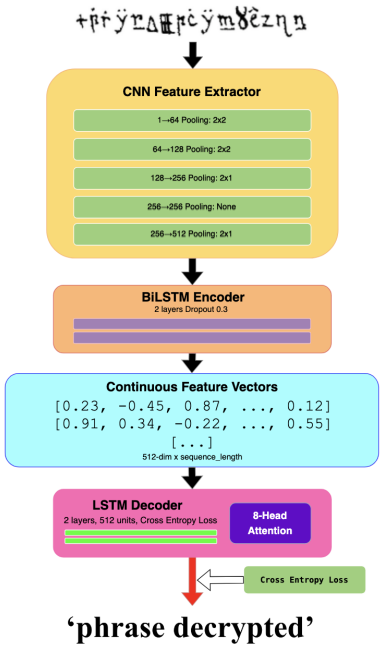


Figure 7: Direct Image Decryption architecture: images processed through CRNN feature extraction, decoded directly to plaintext through attention-based LSTM decoder.

model never commits to discrete cipher symbol decisions, allowing end-to-end gradient flow and joint optimization.

5.2 CRNN Feature Extractor

The feature extractor extends the CRNN architecture from Section 4 with modifications to support end-to-end training. It processes grayscale images through a deeper five-block CNN structure (adding a fifth block: 256→512 channels with 2×1 pooling), expanding upon the four-block design used in transcription.

Following CNN blocks, feature maps of dimension (batch_size, 512, height/16, width/4) are reshaped to dimension (batch_size, width/4, 512 × height/16). A 2-layer bidirectional LSTM (256 units per direction) produces contextualized visual representations of dimension (batch_size, sequence_length, 512). The CRNN can initialize with pretrained transcription model weights (Section 4.1.1), providing a strong starting point. However, unlike the transcription pipeline where CRNN weights remain fixed during decryption training, Direct Image Decryption allows fine-tuning during the end-to-end decryption, permitting optimization. The feature extractor output—sequences of 512-dimensional vectors—captures cipher symbols, spatial relationships, and visual characteristics without committing to discrete symbol decisions.

5.3 Attention-Based Decoder

The decoder generates German plaintext characters autoregressively, conditioning on encoded image features and previously generated characters. It employs 2-layer LSTM (512 hidden units) with 8-head multi-head attention mechanism.

At each timestep, the decoder receives embedding of the previous character (dimension 128) or SOS token. The LSTM produces a query vector, and multi-head attention (8 heads, per-head dimensionality 64) computes scores between query and encoded image features, determining which manuscript regions are most relevant for predicting the current character.

Attended features are concatenated with LSTM output, passed through linear projection (dimension 512), then final output projection to German alphabet vocabulary. Cross-entropy loss trains the model with gradients flowing backward through attention, decoder, and feature extractor. Teacher forcing during training stabilizes learning; inference uses autoregressive generation until an EOS token is found.

5.4 Training Configuration

End-to-end training uses AdamW (Loshchilov and Hutter, 2019) (learning rate 1×10^{-3}, weight decay 1×10^{-4}, dropout 0.3), batch size 16, 35-50 epochs with early stopping based on validation edit distance. The CRNN encoder is initialized with weights from the pretrained transcription model (Section 4.1), then the full architecture is fine-tuned end-to-end; the attention decoder is always trained from scratch. Gradient clipping (max norm 1.0) prevents instability. ReduceLROnPlateau scheduler (factor 0.1, patience 5 epochs) enables fine-grained optimization.

The model is trained end-to-end to directly generate plaintext from images:

$$\mathcal{L}_{\text{DID}} = -\sum_{t=1}^{T} \log P(w_t | w_{<t}, I; \theta) \qquad (2)$$

where I is the input manuscript image, w_t is the plaintext character at position t, $w_{<t}$ represents all previous characters, and θ represents all model parameters. This objective directly optimizes for

accurate plaintext generation without intermediate transcription objectives, allowing gradients to flow through the entire network.

6 Experiments

We compare both architectures across multiple scenarios: architectural components, performance on synthetic data, the Copiale manuscript, and sequence length variation.

6.1 Architecture Comparison

Both approaches employ CRNN-based visual encoding followed by attention-based decoding, but differ in key aspects. The transcription model uses a four-block CNN with 4-layer bidirectional LSTM, while Direct Image Decryption extends this with a deeper five-block CNN and 2-layer bidirectional LSTM. Both use 2-layer decoder LSTMs with 128-dimensional embeddings, though Direct Image Decryption employs 8-head multi-head attention compared to the simpler additive attention in the decryption stage.

The critical architectural distinction lies in the training paradigm. The two-stage pipeline trains two sequential models independently: transcription uses CTC loss to predict 132 cipher symbols, then decryption converts these discrete tokens to 62 plaintext characters using cross-entropy loss. This commits to discrete symbol decisions early, preventing gradient flow from decryption errors back to visual feature extraction. In contrast, Direct Image Decryption trains a single end-to-end model that directly predicts characters from images, maintaining continuous representations throughout and allowing visual features to adapt directly to decryption requirements through end-to-end gradient flow. Whether this end-to-end learning advantage outweighs architectural differences is the empirical question addressed in subsequent results. Comparison with the State-Of-The-Art could not be done, as currently baselines for a Direct Image Decryption approach do not exist. Work on this matter either delves into transcription or decryption separately; a conjoined baseline does not exist.

6.2 Evaluation Metrics

Models are evaluated using four case-insensitive metrics: **Token Accuracy** (percentage of exactly correct predictions, higher is better, [0.0–1.0]), **Normalized Edit Distance (NED)** (Levenshtein distance normalized by the length of the longer sequence, lower is better, [0.0–1.0]; equivalently, decryption success rate is reported as $1 - \text{NED}$), **Word Error Rate (WER)** (token-level error computed as $(S+D+I)/N$, lower is better, [0.0–1.0]), and **Character Error Rate (CER)** (character-level equivalent of WER, computed as $(S+D+I)/N$ where operations are counted over individual characters and N is the number of characters in the reference; unlike NED, CER normalizes by reference length and penalizes insertions and deletions independently, lower is better, [0.0–1.0]).

6.3 Results on Synthetic Data

We evaluate both approaches in our generated synthetic data. All models were trained on the 115,000-image "Faust" dataset and tested on two datasets: the held-out "Faust" test set and the out-of-distribution Novalis dataset (1,300 images of Old German poetry).

Transcription Performance

In the traditional pipeline (namely 2-stage), the transcription model achieves strong performance on the "Faust" test set (Table 1), with 91.5% token accuracy and 4.3% normalized edit distance. Figure 8 illustrates some of the best and worst case examples based on Edit Distance.

Table 1: Transcription performance on "Faust".

Metric	Score
Token Accuracy ↑	0.915
Edit Distance ↓	0.043
WER ↓	0.075
CER ↓	0.076

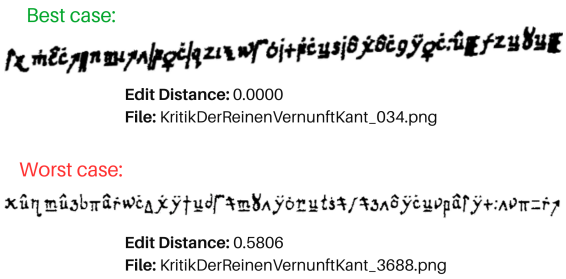


Figure 8: Best and worst case examples of transcription on synthetic data.

Comparison: Two-Stage vs. Direct Image Decryption

Table 2 shows the performance comparison of both approaches. On in-distribution data (Faust), Direct Image Decryption outperforms the two-stage pipeline across most metrics. The 1.1% improvement in token accuracy validates our hypothesis that eliminating the transcription bottleneck reduces error propagation. The 49% reduction in WER (0.206 $\rightarrow$ 0.105) demonstrates superior ability to generate coherent character sequences.

The performance gap widens substantially on the out-of-distribution Novalis dataset, which contains different vocabulary, sentence structures, and poetic phrasing compared to the training data. Direct Image Decryption achieves 75.8% token accuracy compared to 69.5%—a 6.3% absolute improvement, suggesting that end-to-end learning enables better generalization. The two-stage pipeline's performance degradation (91.3% $\rightarrow$ 69.5%) is more severe than Direct Image Decryption's (92.4% $\rightarrow$ 75.8%), indicating that error propagation from transcription compounds when encountering unfamiliar text patterns. The WER reduction (59.7% $\rightarrow$ 31.6%) demonstrates that Direct Image Decryption maintains better sequence-level coherence even when faced with novel vocabulary and grammatical structures. These results provide strong evidence that the end-to-end paradigm offers robustness advantages beyond simple accuracy improvements.

Table 2: End-to-end decryption on synthetic data.

Dataset	Metric	2-Stage	Direct
Faust	Token Acc. ↑	0.913	**0.924**
	Edit Dist. ↓	0.045	**0.038**
	WER ↓	0.206	**0.105**
	CER ↓	**0.056**	0.065
Novalis	Token Acc. ↑	0.695	**0.758**
	Edit Dist. ↓	0.190	**0.162**
	WER ↓	0.597	**0.316**
	CER ↓	0.204	**0.183**

6.4 Results on the Copiale Manuscript

Next, we evaluate both approaches on approximately 2,000 line images from the real 18th-century Copiale cipher (Fornés et al., 2024), probing model behavior under real conditions.

Transcription Performance

In the two-stage pipeline, the transcription component generalizes well to the original manuscript (Table 3), achieving 91.1% token accuracy—only 0.4 percentage points lower than on synthetic data, confirming that visual recognition of Copiale glyphs transfers effectively.

Table 3: Transcription performance on Copiale.

Metric	Score
Token Accuracy ↑	0.911
Edit Distance ↓	0.023
WER ↓	0.017
CER ↓	0.014

End-to-End Decryption Performance

In contrast to transcription, joint end-to-end decryption performance degrades substantially on the authentic manuscript (Table 4). As expected, both approaches perform significantly worse than on synthetic data, with absolute accuracies well below practical usability.

The two-stage pipeline achieves 39.6% token accuracy, while Direct Image Decryption reaches 51.4%—an absolute improvement of 11.8 percentage points (30% relative). Direct Image Decryption also reduces WER from 89.0% to 76.0% and CER from 43.0% to 39.3%, indicating improved sequence-level coherence and fewer catastrophic decoding failures. Despite this improvement, both models fail quite often on the Copiale manuscript, demonstrating that historical cipher decipherment remains a challenge under current data constraints. The fact that transcription succeeds (91.1%) while decryption fails (39.6-51.4%) could suggest that the bottleneck lies in linguistic modeling rather than in visual recognition.

Table 4: End-to-end decryption on Copiale.

Metric	2-Stage	Direct	Δ
Token Acc. ↑	0.396	**0.514**	+11.8%
Edit Dist. ↓	0.428	**0.303**	-12.5%
WER ↓	0.890	**0.760**	-13.0%
CER ↓	0.430	**0.393**	-3.7%

Analysis: The Data Scarcity Problem

The performance collapse from 91-92% (synthetic) to 40-51% (real) reveals a fundamental challenge: insufficient training data on real

manuscripts. Critically, when we train models on reduced synthetic data—20,000 images yield 53% accuracy, 8,000 images yield 31% (Table 5)—demonstrating that models require large-scale data to learn robust decipherment, regardless of whether data is synthetic or real.

This reframes our understanding of the synthetic-to-real gap. The problem is not primarily that synthetic data is qualitatively inadequate, but rather that we have **57 times less real data** (2,000 images) than synthetic training data (115,000 images). Models achieve 91-92% accuracy on synthetic data because they have sufficient examples to learn robust patterns. When applied to real manuscripts, they fail because there was not enough real examples. Performance degradation on limited data stems from:

Insufficient statistical coverage—with only 2,000 real manuscript images, models encounter symbol combinations and degradation patterns during testing never seen during training. Deep learning models require tens of thousands of examples to generalize robustly. Our ablation study presented in Table 5 confirms this: reducing synthetic training data by 82% (115k $\rightarrow$ 20k) causes a 38+ percentage point accuracy drop, comparable to synthetic-to-real degradation.

Compounding error propagation—when trained on limited data, both visual feature extraction and language modeling components underfit, leading to catastrophic failure cascades.

Linguistic domain specificity—while the Copiale manuscript's esoteric content differs from our Faust/Kant/Bible/Nachsommer corpus, this vocabulary mismatch would be learnable given sufficient real manuscript data.

Table 5: Impact of training corpus and scale.

Corpus	Images	Accuracy ↑
Faust	115,000	92.4%
American Psycho	20,000	53.7%
East of Eden	8,000	31.7%
Copiale Real	2,000	51.4%

Despite severe data scarcity, Direct Image Decryption's 11.8% improvement over the two-stage baseline remains meaningful, validating that end-to-end learning reduces error propagation regardless of training set size. Critically, the advantage is most pronounced on challenging real-world data: 1.1% improvement on abundant synthetic data versus 11.8% improvement on scarce real data, suggesting that end-to-end gradient flow provides robustness benefits that become more valuable precisely when training data is limited.

The core challenge is clear: we need more real manuscript data. The 51% token accuracy represents the performance ceiling achievable with 2,000 training examples. Scaling to 10,000-50,000 real manuscript images would likely yield higher accuracy needed for practical deployment. Until such data becomes available, automatic decipherment systems serve as tools for augmenting and aiding human expert analysis.

6.5 Sequence Length Experiment

How does sequence length affect decryption success? To answer this, we evaluated models on dataset variations categorized by length. For the "Faust" dataset: very short (3–12 characters), normal (12–40 characters), and long sequences (40–70 characters). As shown in Figure 9, decryption success rate (1-NED) remains high for shorter sequences but declines as text length increases, dropping from nearly 1.00 to approximately 0.84 for the 70-character range.

For the original Copiale manuscript, the model struggles significantly with extremely short inputs (below 10 tokens), where success rates hover around 0.35. However, performance peaks and stabilizes once the reference length exceeds 10 tokens, maintaining a success rate near 0.5 for standard manuscript line lengths.

The divergence suggests that while synthetic models may overfit to specific length patterns, real manuscript data requires a minimum threshold of linguistic context for reliable decryption. In synthetic data, increased complexity of longer strings drives errors up, whereas in Copiale, the primary hurdle is lack of sufficient information in very short snippets. This asymmetry reveals that the models learn different failure modes depending on their training distribution—synthetic models struggle with long-range dependencies, while real-data models need sufficient context to disambiguate the noisy visual features.

7 Conclusions and Future Work

This work has explored the transition from traditional multi-stage pipelines to end-to-end neural architectures for decipherment of historical encrypted manuscripts. By introducing Direct Image

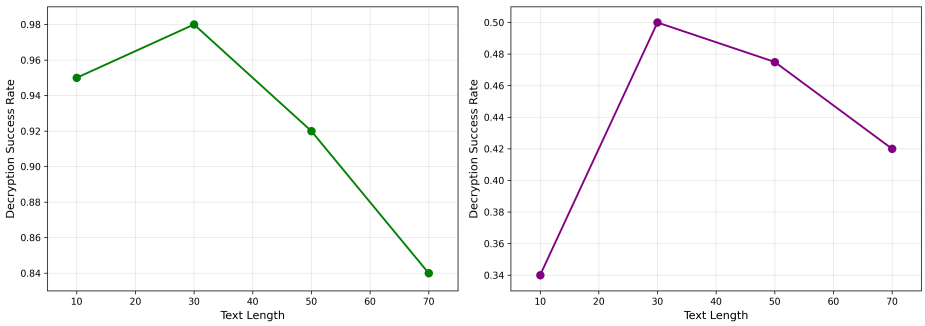


Figure 9: Decryption success rate relative to sequence length for synthetic “Faust” (left) and the original Copiale manuscript (right).

Decryption, we have demonstrated that mapping encrypted manuscript images directly to decrypted plaintext is both feasible and superior to the traditional transcription-decryption sequence.

For the evaluation, we have developed a comprehensive synthetic data generation pipeline producing over 115,000 realistic Copiale-like manuscript images from historical German texts. We implemented and evaluated a CRNN-based transcription model with CTC loss, achieving 91.5% token accuracy on synthetic data and 91.1% on the original Copiale manuscript. We then compared Direct Image Decryption against the traditional 2-step approach, demonstrating consistent improvements across all evaluated datasets.

Direct Image Decryption consistently outperformed the two-stage baseline by a mean of 6% token accuracy (Faust, Novalis, Copiale), with advantages becoming more pronounced under challenging conditions—11.8% improvement on the real Copiale versus 1.1% on synthetic data. This validates our hypothesis that eliminating the intermediate transcription step significantly reduces error propagation, thus improving performance.

While transcription models generalize effectively to real manuscripts, both pipelines’ decryption performance drops sharply, revealing a fundamental data scarcity challenge. Our analysis proves this is mainly quantitative rather than qualitative: models require many examples to learn robust linguistic patterns, yet we possess 57 times less real data (2,000 images) than synthetic data (115,000 images). The consistency of cipher glyphs allows visual recognition models trained on synthetic data to perform reliably on authentic 18th-century handwriting, suggesting the primary bottleneck lies in linguistic modeling rather than visual feature extraction.

These findings underscore that the path forward for automatic historical cipher decipherment lies not in algorithmic innovation alone, but in scaling real data. Until such data becomes available and the architecture is tested on other ciphers, systems like Direct Image Decryption serve as tools for augmenting human expert analysis, offering relative improvements that can meaningfully reduce the manual effort required for decipherment.

The main contribution of this work is to show that end-to-end image-to-plaintext modeling can help in the decipherment workflow, especially by reducing transcription-related error propagation and helping prioritize expert effort. However, our experiments remain restricted to a cipher with a known key, so the method should be understood not as a replacement for scholarly decipherment, but as a computational tool that may assist it.

Concerning future work, several research avenues remain to bridge the gap between experimental models and practical tools. First, one should explore generative or difussion models for more realistic synthetic data generation, able to capture subtle characteristics like ink flow variations and aging patterns that current augmentation approximates but does not fully replicate. Second, our Direct Image Decryption architecture should be evaluated on other historical ciphers, such as the Borg or Ramanacoil manuscripts, to determine whether end-to-end mapping advantages generalize beyond other historical substitution-based ciphers. Finally, self-supervised learning could allow models to learn from untranscribed manuscripts, while AI-in-the-loop systems integrating expert feedback would refine linguistic priors and create high-quality training data through interactive correction, accelerating decipherment of unsolved historical manuscripts.

Acknowledgments

This work has been partially supported by Riksbankens Jubileumsfond, grant M24-0028 (Echoes of History: Analysis and Decipherment of Historical Writings, DESCRYPT), the Spanish project PID2024-157778OB-I00 (SUKIDI) from the Ministerio de Ciencia e Innovación, the Departament de Cultura of the Generalitat de Catalunya, and the CERCA Program / Generalitat de Catalunya. Alicia Fornés acknowledges financial support for her general research activities from ICREA under the ICREA Academia (Departament de Recerca i Universitats de la Generalitat de Catalunya). Lei Kang acknowledges financial support from the Beatriu de Pinós del Departament de Recerca i Universitats de la Generalitat de Catalunya (2022 BP 00256).

References

Nada Aldarrab and Jonathan May. 2021. Can Sequence-to-Sequence Models Crack Substitution Ciphers? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7226–7235. Association for Computational Linguistics.

Nada Aldarrab. 2022. *Automatic Decipherment of Historical Manuscripts*. Ph.D. thesis, University of Southern California.

Théodore Bluche, Jérôme Louradour, and Ronaldo Messina. 2017. Scan, Attend and Read: End-to-End Handwritten Paragraph Recognition with MDLSTM Attention. In *Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pages 1050–1055. IEEE.

Raymond Clemens. 2016. The Voynich Manuscript. Beinecke Rare Book and Manuscript Library Digital Collections, Yale University.

Joachim Dinnissen and Nils Kopal. 2021. Island Ramanacoil a Bridge too Far: A Dutch Ciphertext from 1674. In *Proceedings of the 4th International Conference on Historical Cryptology (HistoCrypt 2021)*, pages 48–57. Linköping University Electronic Press.

Alicia Fornés, Jialuo Chen, Pau Torras, Carles Badal, Beäta Megyesi, Michelle Waldispühl, Nils Kopal, and George Lasry. 2024. Icdar 2024 competition on handwriting recognition of historical ciphers. In *International Conference on Document Analysis and Recognition*, pages 332–344. Springer.

Jim Gillogly. 1980. A Dissenting Opinion: The Beale Ciphers. *Cryptologia*, 4(2):116–119.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pages 369–376. ACM.

Nishant Kambhatla, Anahita Mansouri Bigvand, and Anoop Sarkar. 2018. Decipherment of Substitution Ciphers with Neural Language Models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 869–874. Association for Computational Linguistics.

Kevin Knight, Beáta Megyesi, and Christiane Schaefer. 2011. The Copiale Cipher. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora*, pages 2–9. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Beáta Megyesi, Bernhard Esslinger, Alicia Fornés, Nils Kopal, Benedek Láng, George Lasry, Karl de Leeuw, Eva Pettersson, Arno Wacker, and Michelle Waldispühl. 2020. Decryption of historical manuscripts: the DECRYPT project. *Cryptologia*, 44(6):545–559.

Baoguang Shi, Xiang Bai, and Cong Yao. 2017. An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(11):2298–2304.

Xusen Yin, Nada Aldarrab, Beáta Megyesi, and Kevin Knight. 2019. Decipherment of Historical Manuscript Images. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 78–85.

Location Matters: Accelerating Historical Cipher Transcription with Detection-Based Models

George Lasry
The CrypTool Project
george.lasry@gmail.com

Abstract

Transcribing historical ciphers is the first step toward decryption and analysis. Machine learning models proposed for this task often neither consume nor produce symbol locations. Such location-discarding approaches do not integrate naturally with transcription and analysis tools that rely on a visual feedback loop linking image regions to transcribed symbols. We argue for location-aware, detection-based deep learning models that preserve location information in both supervision and output, supporting an end-to-end visual workflow with tools such as CTTS (CrypTool Transcription and Solver). To this end, we present two detection-based models with distinct architectures, evaluate them quantitatively across diverse cipher collections, and illustrate the workflow through a case study. The results show that this approach is practical and sample-efficient: it performs well with limited training data and remains effective in a challenging yet typical bootstrap scenario for new cipher collections. It supports human-in-the-loop correction, significantly reduces manual work, and helps produce accurate transcriptions and training data.

1 Introduction

The field of HTR (Handwritten Text Recognition) has been dominated in the last 15 years by segmentation-free approaches, in which models receive weak supervision in the form of line-level transcripts and, at inference, produce text sequences without character locations. This shift from prior segmentation-based approaches, which required explicit character boundaries, was made possible by the advent of CTC loss (Graves et al., 2006), bidirectional LSTMs, and later transformers (Li et al., 2023). This shift also affected HTR for historical documents, and most automated transcription solutions for texts from the Middle Ages and later — including leading platforms such as Transkribus (Muehlberger et al., 2019) and Kraken (Kiessling, 2019) — are built around segmentation-free architectures. Such approaches are especially effective for plaintext historical documents, where character-level annotation is costly, scripts are often cursive and connected, corpora are relatively large, alphabets are small and stable, and language priors compensate for weaker per-character supervision.

Ancient-script transcription projects differ. Not only are most ancient scripts non-cursive, with spatially separate symbols, but corpora are often limited, and the material poorly preserved, requiring every visual and spatial cue (2D, 3D, spectral imagery) to distinguish the features of each symbol. The ancient script alphabet itself may still be under investigation. As a result, most of those projects are built around detection-based architectures, such as DeepScribe for cuneiform (Williams et al., 2025) and YOLOv8 ensembles for Greek papyri (Turnbull and Mannix, 2025).

With historical handwritten, enciphered documents dating from the Renaissance to the 19th century, the picture is more nuanced, but still with a tendency to rely on segmentation-free approaches. This includes submissions to the ICDAR 2024 Competition on Historical Ciphers (Fornés et al., 2024), most of which do not produce symbol locations. Recent evaluations of HTR methods for ciphers (Souibgui et al., 2023) focus on such location-discarding approaches. Notable exceptions include few-shot detection (Souibgui et al., 2021), a detection-based model for cipher digits (Antal et al., 2022), and DTLR (Baena et al., 2024), a contest participant that is detection-based but trained partially without location annotations.

This paper makes three concrete contributions. First, it presents a detection-based approach to historical cipher transcription, validated on real-world collections totaling tens of thousands of symbols. Second, it introduces and evaluates two models with distinct architectures — a TIMM+FPN CenterNet-style detector and an adapted DINO-DETR/DTLR model — both

trained with explicit bounding-box supervision across diverse cipher collections. Third, it shows how tight integration into a transcription GUI (CrypTool Transcription and Solver – CTTS) enables a human-in-the-loop workflow that simultaneously produces high-quality transcriptions and new training data efficiently and at scale.

2 Why Symbol Locations Matter

In this article, we argue that transcribing documents with historical cipher symbols is much closer to transcribing ancient scripts than to transcribing plaintext handwritten documents. Several properties of ciphers undermine the advantages of segmentation-free approaches:

Unknown symbol sets. Cipher scripts use dozens to several hundred distinct types, adding a new layer of complexity to both transcription and interpretation. The symbol inventory itself is a research output, not an input — compound symbols, diacritical marks, and scribal variants are often discovered progressively.

Unknown keys. The cipher key, which maps specific symbols to language elements, is also unknown in most cases, adding one more layer of complexity. Transcription and decryption are coupled: errors in transcription often become apparent only through failed decryption attempts. This iterative cycle requires a spatial link between each symbol and its image location.

No language priors. Unlike plaintext, encrypted text offers no linguistic redundancy or clues. A segmentation-free model cannot – a priori – rely on a language model to resolve visual ambiguity, e.g., similarly-looking symbols, small diacritics, damaged or partially visible symbols, common in historical material.

Annotation cost. With plaintext in a known language, line-level transcription is fast because the annotator reads the text. With cipher symbols, the annotator inspects each unfamiliar symbol individually — effectively doing character-level work. As a result, transcribing a historical cipher document "as text", unless digits are employed, is no faster than annotating locations, which is not only convenient but also more accurate with a visual tool like CTTS.

Because of those additional layers of complexity, maintaining the link between transcripts and symbols is critical. The goal of processing historical cryptograms is to produce a fully decoded, accurate text ready for analysis. This requires a complex workflow — transcription, decryption, error correction, and analysis — in which errors introduced at one stage often become apparent only at another, and where all relevant elements should be integrated, viewed, and edited side by side without switching between applications. This includes the image with the symbols, the transcription, the nomenclature of symbol types (classes) and their meanings (the decryption key), the raw decrypted text, and the edited decryption in readable form. Such an integrated and interactive view is supported by tools such as CTTS. Location-aware models naturally support this iterative visual workflow.

Furthermore, while location-discarding models rely on "weak supervision", detection models benefit from "strong supervision" via explicit bounding boxes, and in most cases, can learn effectively from far fewer pages.

3 A Detection-Based Workflow

We propose a human-in-the-loop workflow that addresses the challenges of transcribing historical ciphers, for the typical case of an unknown symbol set and key:

1. **Bootstrap**: Manually transcribe a small initial set (~1,500 symbols) using CTTS, and train a first detection model.
2. **Assist**: Use the trained model to annotate the next batch, and review and correct the transcription with CTTS.
3. **Refine**: Retrain the model on the additional data, then repeat the process, improving both the data and the model.

Each cycle improves the model and expands the training set. Inference can even be run on training data to spot inconsistencies in earlier annotations, further improving dataset quality. The goal of producing high-quality transcriptions (for cryptanalysis, decryption, and analysis) and the goal of building quality training data become one and the same. In Section 11, we describe an alternative bootstrap approach using generic models.

4 Detection-Based Models – Early Investigations

We explored various options for a model based on existing components that would not require extensive computational power or lengthy training.[1] This included:

- A custom CNN with 5-channel output (objectness + offsets + size);
- YOLOv11 (yolo11x – 57M);
- ArcFace classification – two-stage pipeline: detect first, then classify cropped symbols (~4M params).[2]

5 First Model (TIMM)

We first implemented a model that combines well-established deep learning components into an end-to-end pipeline that jointly optimizes detection and classification objectives. The output consists of bounding boxes with class labels and confidence scores for each detected symbol, as well as a feature vector extracted from the fused FPN representation at the symbol's center location, which CTTS uses for clustering and outlier detection. The architecture (see Figure 1) consists of:

- **Backbone (via TIMM)**: Pretrained image classification networks—ConvNeXt (Liu et al., 2022) or ResNet (He et al., 2016) variants, with strong feature extraction without requiring massive training data. The script supports arbitrary TIMM backbones. We used one lighter configuration for fast-turnaround experiments,[3] and one heavier configuration for the final model used to transcribe documents.[4]
- **Feature Pyramid Network (FPN)** (Lin et al., 2017a): Top-down architecture with lateral connections that fuses high-resolution spatial detail with semantically rich deeper features. Essential for detecting symbols of varying sizes within the same line.
- **Detection heads** (CenterNet-style) Predict object centers as heatmap peaks rather than anchor boxes. Separate heads output center probability (with focal loss (Lin et al., 2017b)), bounding box size, and sub-pixel offset for precise localization.
- **Classification head**: Parallel branch predicting symbol class at each spatial location, trained jointly with detection using cross-entropy loss.

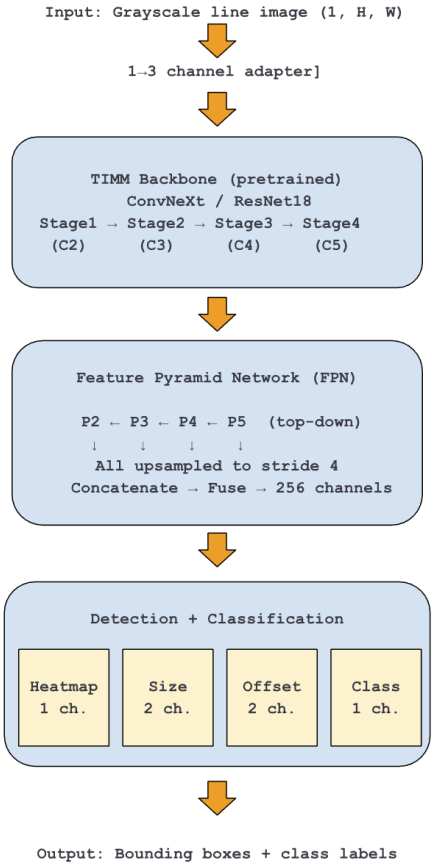


Figure 1. Model architecture with pretrained backbone, FPN decoder, and parallel detection/classification heads.

We trained using AdamW with a learning rate of 10^{-4} and weight decay of 10^{-4}, with cosine annealing. Input lines are cropped from page-level grayscale images, resized to 64 pixels

[1] With the assistance of LLM tools (Claude Code).

[2] Due to a lack of space, we do not reproduce here the results of those experiments. We discarded some options that could not produce embeddings or feature vectors needed by CTTS for clustering and outlier detection (see Section 9), like YOLOv11.

[3] *resnet18.fb_ssl_yfcc100m_ft_in1k*, ~11M parameters, pretrained self-supervised + ImageNet.

[4] *convnext_base.clip_laion2b_augreg_ft_in12k_in1k*, ~88M param., pretrained on ImageNet + CLIP. Reduces CER by a typical (relative) 20-30% compared to the lighter backbone (e.g., 3.5-4% vs. 5%)

height, width-capped, and padded to a fixed width for batching. We use focal loss (α=2.0, β=4.0) for the heatmap head. Training uses configurable probabilistic augmentation (including padding jitter, small rotations, brightness/contrast/gamma changes, blur, noise, and cutout), whereas validation is performed without augmentation. During inference, the default confidence threshold is 0.2, same-class NMS uses an IoU threshold of 0.3, cross-class NMS is disabled by default, and local peak detection uses a 3×3 kernel. Performance is evaluated using CER as the primary metric for model selection,[5] with precision, recall, and classification accuracy reported alongside. Training stops after 100 epochs or after 20 epochs without improvement in CER.

6 Case Study: The Borg Cipher

We illustrate the proposed workflow through a case study on the Borg cipher collection, a bootstrap scenario with no initial training data. This case is particularly challenging due to the irregular handwriting and intertwined symbols. Line segmentation was performed with CTTS and an algorithm based on horizontal projections.

6.1 Bootstrap – Manual transcription of an initial batch

We manually transcribed 5 pages using CTTS, totaling 1,519 symbols across 50 distinct classes.[6] This effort took 3.5 hours. This phase was also essential for understanding the cipher symbols' peculiarities, e.g., compound symbols.

6.2 First trained TIMM model

We trained the detection model on these initial data, using 40% of the material for validation.[7] Training was completed in 6 minutes. Results:

- Precision: 96.4%, Recall: 95.0%
- Classification accuracy: 96.1%
- CER: 11.02%

Figure 2 shows sample results on a test image: false negatives in red, false positives in orange, and misclassifications in purple. The relatively high segmentation recall and precision already make this early model useful and time-saving.

Figure 2. Sample results with the first model trained on Borg.

6.3 Batch with semi-automated transcription

Using this model, we processed a new batch of 5 pages. We manually fixed 37 false negatives, 2 false positives, and 38 misclassifications. Correcting the model's output took 45 minutes.

Figure 3 shows CTTS used to review and correct the transcription of a relevant page of the Borg cipher. All symbols of a given type can be viewed in one place, highlighting outliers.

[5] CER Character Error Rate is calculated based on the minimum number of character-level edits – substitutions (S), deletions (D), and insertions (I) required to transform the predicted text into the original reference text, divided by the total number of characters (N). CER = (S + D + I) / N. To qualify for a true positive, we employ loose IoU thresholds: 0.15 for the same class and 0.3 for a different class (misclassification).

[6] Only 50 of the 70 classes in the collection were observed in the first 5 pages.

[7] In this bootstrap scenario, we did not allocate holdout material for the final evaluation. The metrics are for the validation set.

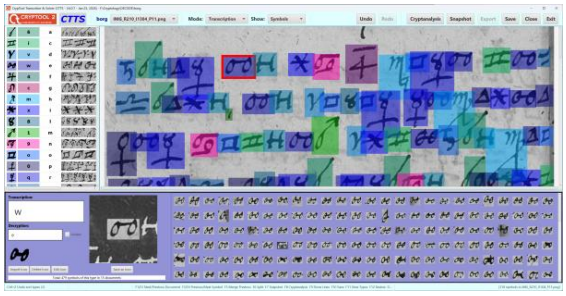


Figure 3. Working with CTTS to verify and improve the automated transcription.

6.4 Second trained TIMM model

We retrained the model on the extended dataset:

- Precision: 97.1%, Recall: 97.1%
- Classification accuracy: 96.9%
- CER: 7.06%

6.5 Additional batch with semi-automated transcription

With the improved model, we processed another 5 pages. Correction time dropped to 20 minutes, with 20 false negatives, a couple of false positives, and 35 misclassifications.

6.6 Third trained TIMM model

We retrained the model on the extended dataset:

- Precision: 98.3%, Recall: 97.4%
- Classification accuracy: 98.5%
- CER: 5.00%

7 Second Model – Adapted DINO-DETR/DTLR Model

In addition to the TIMM+FPN detector described in Section 5, we adapted the DTLR model of Baena et al. (2024) to be fine-tuned with explicit symbol bounding boxes and to predict both the class and position of detected symbols.

DTLR is a text line recognition model built on the DINO-DETR object detection architecture (Zhang et al., 2022), trained in two stages: pretraining on synthetic data with full detection supervision, then fine-tuning on real data with weaker sequence-level supervision. A ResNet-50 backbone extracts multi-scale feature maps from a cropped line image, which are refined by a transformer encoder (6 layers of multi-scale deformable self-attention) and decoded by a transformer decoder (6 layers). The decoder uses 900 learned content queries, each paired with a positional anchor initialized from the encoder output; after self-attention and deformable cross-attention to the encoder features, each query predicts a 4-coordinate bounding box and a probability vector over the symbol alphabet. During pretraining, the Hungarian algorithm matches predictions to ground truth, and the loss combines focal loss for classification with L1 and generalized IoU losses for box regression. The model produces all detections for a line in parallel, unlike autoregressive approaches that generate one symbol at a time. In the original DTLR pipeline (Figure 4, left side), this architecture is pretrained on synthetic line images of Latin characters with exact positions and identities. The resulting pretrained "Latin" model is then fine-tuned on real ciphertext data using only line images and transcribed symbol sequences, without bounding boxes, via an adapted CTC loss: the predicted boxes order the character queries spatially, and CTC aligns this ordered sequence with the ground-truth transcription. At inference, the fine-tuned DTLR model primarily outputs an ordered sequence of symbols.

Our adaptation (Figure 4, right side) consumes and produces symbol locations. We start from the same pretrained Latin DINO-DETR model released by Baena et al., but replace the classification heads with new layers sized to the target cipher's symbol inventory. Their rows are initialized by copying weights and biases from randomly selected rows of the pretrained Latin classifier, rather than from a fully random initialization, following Baena et al.'s transfer strategy. Instead of fine-tuning with CTC loss on transcribed sequences, we fine-tune with explicit symbol-class and bounding-box supervision, using the same detection loss as in DTLR pretraining: focal loss for classification and L1 plus GIoU losses for box regression. At inference, the adapted model outputs both symbol classes and bounding boxes.

There are two additional practical differences. First, DTLR fine-tunes on binarized line crops, whereas we use grayscale line crops extracted from manuscript pages. Second, our implementation splits long lines into overlapping segments for both training and inference to accommodate the variable, often extended line lengths typical of historical ciphers; the original DTLR processes each line as a single input.

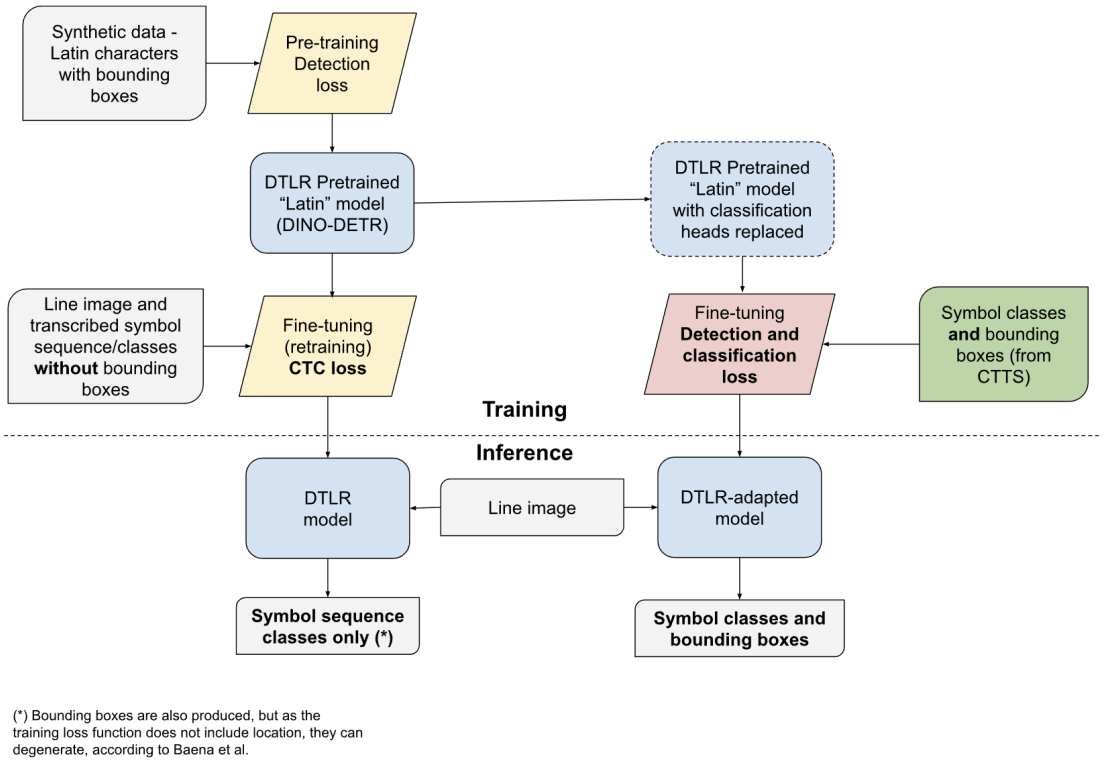


Figure 4. Comparison of the original DTLR pipeline (left) and our adaptation (right). Both start from the same pretrained Latin DINO-DETR/DTLR model. The original pipeline fine-tunes with CTC loss using only transcribed sequences without bounding boxes, producing primarily ordered class sequences at inference. Our adaptation fine-tunes with the full detection-and-classification loss, producing both symbol identities and locations at inference.

8 Performance Evaluation

Training sessions on collections of encrypted documents, transcribed with CTTS, were conducted using an NVIDIA RTX 3090 and took between 6 and 25 minutes for the TIMM model, and up to 90 minutes for the DINO model. We trained models on increasing amounts of training materials, pre-allocating a fixed holdout set of 20,000 symbols (10,000 for Borg) for evaluation. For validation and selecting the best epoch, we used 20% of the training material. As shown in Table 1, the models performed well across multiple collections of ciphertexts, including a subset of the recently deciphered letters from Mary Stuart (Lasry et al., 2023), comprising 157 distinct classes.[8]

Four of the collections were included in the ICDAR contest (Copiale, Borg, Ramanacoil, and BnF). The CER figures are not directly comparable to those of the contest, because our models were trained with strong supervision on grayscale images with explicit symbol boxes and class labels, whereas the contest systems used weaker supervision on binarized segmented lines with transcripts only.[9] We do include the ICDAR data in Table 1 because they provide useful context on training scale: the contest used 2,594 training lines for Borg, 1,502 for Copiale,[10] 1,291 for Ramanacoil, and 788 for BnF, whereas our evaluation emphasizes bootstrap settings with much smaller training sets under strong supervision.[11]

[8] A similar model is currently being used to transcribe another collection of ciphertexts related to Mary Queen of Scots, with more than 100,000 symbols.

[9] Our models were trained on datasets generated using CTTS, which we curated and refined.

[10] The Copiale's thin vertical symbols (bars, iotas) were a significant source of errors in earlier versions of the TIMM model. Targeted augmentation strategies substantially reduced these errors in later versions.

[11] The results on the BnF dataset stand out as a case where strong supervision does not provide a substantial sample-efficiency advantage. The BnF dataset has a small symbol set (36) and clean, well-separated symbols.

Dataset	Training Lines	Training Symbols	Number of Classes	Average Sym./Class	Model	CER (%)	Prec. (%)	Recall (%)	Class. (%)
Borg	165	2500	70	35.7	TIMM	6.6	98.6	98.5	96
					DINO	6.75	98.9	97.9	96.2
Borg	311	5000	70	71.4	TIMM	5.55	98.9	98.7	96.7
					DINO	5.54	99.1	97.7	97.4
Borg	640	10000	70	142.8	TIMM	4.26	99	99	97.6
					DINO	4.96	98.6	98.6	97.7
In the ICDAR contest, the best CER on the Borg dataset was 6.76% with **2594** training lines and weak supervision (sequence-only)									
Copiale	67	2500	113	22.1	TIMM	10.97	99.3	98.3	91
					DINO	4.46	99.7	98.1	97.6
Copiale	119	5000	113	44.2	TIMM	4.12	99.7	98.9	97.2
					DINO	2.57	99.7	99.1	98.6
Copiale	239	10000	113	88.5	TIMM	2.37	99.7	99.3	98.6
					DINO	1.61	99.8	99.4	99.2
Copiale	480	20000	113	177	TIMM	1.32	99.9	99.5	99.3
					DINO	1.21	99.7	99.6	99.4
In the ICDAR contest, the best CER on the Copiale dataset was 1.62% with **1502** lines for training and weak supervision									
Ramanacoil	68	2828	71	39.8	TIMM	6.84	98.7	98.4	95.6
					DINO	16.16	97.9	91.7	93.1
Ramanacoil	84	4990	71	70.3	TIMM	3.18	99.1	99.1	98.2
					DINO	6.37	98.5	97.1	97.7
Ramanacoil	201	10000	71	140.8	TIMM	1.85	99.7	99.4	99
					DINO	4.99	98.6	98	98.2
In the ICDAR contest, the best CER on the Ramanacoil dataset was 5.61% with **1291** training lines with weak supervision									
BnF Fr. 3029	323	10000	36	277.8	TIMM	1.21	99.8	99.8	99.1
					DINO	1.73	99.6	99.7	98.9
In the ICDAR contest, the best CER for the BnF Fr., 3029 dataset was 0.89% with **788** lines for training and weak supervision									
Mary Stuart	270	20004	157	127.4	TIMM	2.46	99.9	99.7	97.9
					DINO	4.13	98.8	99.2	97.8

Table 1. Evaluation results across cipher collections with our two detection-based models trained with strong supervision. ICDAR 2024 data are included as contextual reference points for weak-supervision settings, to illustrate the amount of training data used in the contest. The CER figures are not directly comparable.

Table 1 also shows that the two architectures have complementary strengths. In these experiments, TIMM is generally more robust in low-data bootstrap settings and performs better on Borg, Ramanacoil,[12] and BnF, whereas the adapted DINO model becomes competitive or superior on some datasets, especially Copiale. In practice, this makes TIMM a strong default model, while DINO provides a useful alternative whose strengths may depend on the symbol inventory and visual characteristics of the documents in a collection. As described in Section 12, the availability of a second model with a different architecture has several additional advantages, e.g., enabling an independent review of the results of the first model.

We analyzed the results of 80 training runs with the TIMM model, with datasets from various origins, sizes, and types, transcribed with CTTS. We found a strong correlation between the best CER achieved and the ratio of the total number of symbols used for training to the number of distinct symbol classes (the average number of times a given class appears in the training data), as shown in Figure 5.[13] Recall and precision reach 90%+ in most cases with fewer than 1000 symbols, as shown in Figure 6.

9 Integrating the Models into CTTS and Transcription Workflows

To integrate the models into CTTS, we use ONNX (Open Neural Network Exchange), an open format for representing trained machine learning models, allowing them to be exported from one framework (e.g., PyTorch) and run in another runtime environment without requiring the original training code. An ONNX model stores both the model's architecture (as a computation graph) and its learned weights in a single portable file. CTTS with an ONNX model loaded can detect symbols at a rate of 5 to 50 symbols per second without a GPU, depending on the number of CPU cores and the model (detection with TIMM models is faster). A GPU can accelerate detection by a factor of 5-10.

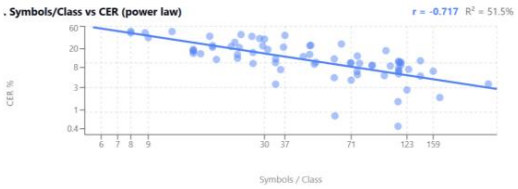


Figure 5. TIMM model – CER as a function of (number of symbols/number of classes).

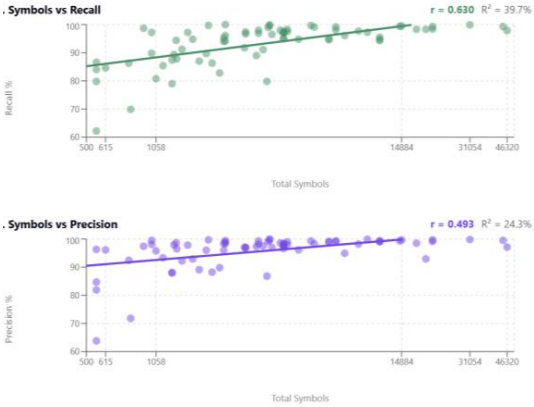


Figure 6. TIMM model – precision and recall as a function of training data size.

In addition to the location and identity of the symbols in a document, the deep learning models also produce symbol detection features (TIMM models), or symbol embeddings (DINO models), which are used by CTTS for clustering similar symbols into tentative classes, auto-classifying newly detected symbols by comparing their features/embeddings with those of previously transcribed symbols, or identifying outliers and detecting potential classification errors.[14]

10 Towards a Universal Detection Model for Cipher Symbols

So far, we have described a bootstrap workflow in which some material is manually transcribed, and an initial model is trained on it; additional pages are transcribed with this model and reviewed manually, and so on. We are also evaluating

[12] The source of DINO's weaker performance on Ramanacoil is not fully understood; the collection's exceptionally long lines, not consistently straight, may play a role, although our tiling strategy is designed to handle such cases.

[13] Some of the correlations can be partially explained by symbols not seen or rarely seen in training data.

[14] Such detection features or embeddings are not produced by non-detection-based models, obviously. Preliminary results indicate that the embeddings produced by our DINO models are more discriminative than those from TIMM models. We are also experimenting with a Real-Time DETR architecture, which is much faster while still producing high-quality embeddings.

generic models trained on a variety of historical cipher documents and symbols (~400 classes). Those models yield locations with high precision and recall (90-95%) with ciphertexts with symbol types unseen during training. Classification is partially useful as symbols of the same type tend to be assigned to the same class, and similarly, symbols assigned to a given class tend to be similar. Generic models can accelerate the end-to-end transcription process compared to manual transcription, and are being integrated into CTTS.

11 An Alternative Bootstrap Workflow

With the tight integration of generic models with CTTS, we can propose an alternative bootstrap scenario that requires even less manual work, no model training or retraining, and no GPU resources, all the steps performed with CTTS:

- Apply a **generic model** to automatically transcribe a few initial documents.
- Apply the **CTTS clustering** function to improve the initial classifications.
- Review and manually correct this initial transcription and classification, leveraging **CTTS's built-in tools to detect outliers**.
- Apply the generic model again to additional documents.
- **Auto-classify the symbols with CTTS** (which compares their embeddings to those of already transcribed symbols).
- Review the transcription and classifications and repeat those steps. CTTS autoclassification becomes more accurate as more data is transcribed.

12 Working with Multiple Models

If we can train two fine-tuned models on the same symbol set, it is possible (with CTTS) to apply one model (e.g., TIMM) to detect and classify symbols in documents, and then apply the second model (e.g., DINO) to generate a list of edit suggestions. This feature is powerful, enabling high-quality transcriptions with minimal effort. Since the TIMM and DINO architectures described here differ, they often produce different types of errors, and the probability that both will make the same error is relatively low.[15]

13 Conclusion

In this article, we highlighted the advantages of working with detection-based models. First, by outputting bounding boxes alongside class predictions, deep learning models integrate directly into tools like CTTS, maintaining the visual feedback loop required for efficient correction and ultimately producing rigorous, publication-quality transcriptions and decryptions. Second, precise supervision via explicit bounding boxes enables models with diverse architectures to generalize from training sets of limited size, which is the general case with historical ciphers.[16] Third, detection models achieve high segmentation precision and recall early, often before they can achieve high classification accuracy. This allows collections to be processed incrementally, with early versions already relieving the transcriber of the most laborious task — drawing thousands of bounding boxes from scratch—and leaving only the faster work of verification and relabeling.

Our case study on the Borg cipher demonstrates these advantages across an entire bootstrap workflow, where initially, no training data was available. Current work on generic models for detecting cipher symbols will enable such bootstrap scenarios to be implemented at even lower cost (in terms of manual effort and GPU resources). Overall, this approach, together with tight integration with CTTS, significantly accelerates transcription, including large-scale collections, and improves its quality. Higher quality transcriptions enable more accurate decryption and key recovery, which are essential for historical research. This approach also improves the quality of the training data.

We are planning to release the scripts, the models, and the integrated version of CTTS.

[15] With this approach, we have transcribed hundreds of thousands of symbols from multiple collections, including large ones, with minimal effort. We are also evaluating the option of producing ensemble predictions from several models.

[16] The two models we presented already leverage pretrained models. Further research may include vision foundation models such as SAM2 for segmentation, or DINOv2 for feature extraction, pretraining on synthetic data consisting of cipher symbols, pseudo-labeling, and other detection and classification architectures. In case the cipher key is known, language models may also help improve classification accuracy.

Acknowledgments

We would like to thank Raphael Baena and his co-authors for publishing their code and models and making them available. We also thank the anonymous reviewers for their valuable feedback.

References

Eugen Antal and Pavol Marák. 2022. Automated Transcription of Historical Encrypted Manuscripts. *Tatra Mountains Mathematical Publications,* 82(2): 65–86.

Raphael Baena, Syrine Kalleli, and Mathieu Aubry. 2024. General Detection-Based Text Line Recognition. In *NeurIPS 2024*.

George Lasry. 2026. *CTTS – CrypTool Transcriber & Solver.* In *Proceedings of the 7th International Conference on Historical Cryptology (HistoCrypt 2026)*, Linköping University Electronic Press.

Alicia Fornés, Jialuo Chen, Pau Torras, Carles Badal, Beáta Megyesi, Michelle Waldispühl, Nils Kopal, and George Lasry. 2024. ICDAR 2024 Competition on Handwriting Recognition of Historical Ciphers. In *ICDAR 2024*. Springer.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *ICML 2006*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR 2016.*

Benjamin Kiessling. 2019. Kraken – a Universal Text Recognizer for the Humanities. *DH 2019, Book of Abstracts.*

George Lasry, Norbert Biermann, and Satoshi Tomokiyo. 2023. Deciphering Mary Stuart's Lost Letters from 1578–1584. *Cryptologia,* 47(2): 101–202.

Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. 2023. TrOCR: Transformer Based Optical Character Recognition with Pretrained Models. In *AAAI 2023*.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017a. Feature Pyramid Networks for Object Detection. In *CVPR 2017*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017b. Focal Loss for Dense Object Detection. In *ICCV 2017*.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. 2022. A ConvNet for the 2020s. In *CVPR 2022.*

Guenter Muehlberger, Louise Seaward, Melissa Terras, Sofia Ares Oliveira, Vicente Bosch, Maximilian Bryan, Sebastian Colber, et al. 2019. Transforming Scholarship in the Archives Through Handwritten Text Recognition: Transkribus as a Case Study. *Journal of Documentation,* 75(5): 954–976.

Mohamed Ali Souibgui, Alicia Fornés, Yousri Kessentini, and Crina Tudor. 2021. A Few-Shot Learning Approach for Historical Ciphered Manuscript Recognition. In *ICPR 2020*.

Mohamed Ali Souibgui, Pau Torras, Jialuo Chen, and Alicia Fornés. 2023. An Evaluation of Handwritten Text Recognition Methods for Historical Ciphered Manuscripts. In *HIP '23*. ACM.

Robert Turnbull and Evelyn Mannix. 2025. Detecting and Recognizing Characters in Greek Papyri with YOLOv8, DeiT and SimCLR. *International Journal on Document Analysis and Recognition (IJDAR),* 28: 277–285.

Kathryn Williams, Yingjie Su, David Schloen, Sandra Schloen, Miller Prosser, Susanne Paulus, and Sanjay Krishnan. 2025. Localization and Classification of Elamite Cuneiform Signs on Persepolis Fortification Tablets. *Journal on Computing and Cultural Heritage,* 18(1): 1–22.

Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. 2022. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. In *ICLR 2022*

Learning to Decipher from Pixels — A Case Study of Copiale

Lei Kang
Giuseppe De Gregorio
Computer Vision Center
Universitat Autònoma de Barcelona
{lkang, gdegregorio}@cvc.uab.es

Raphaela Heil
Department of Linguistics
Stockholm University, Sweden
raphaela.heil@ling.su.se

Alicia Fornés
Computer Vision Center
Universitat Autònoma de Barcelona
afornes@cvc.uab.es

Beáta Megyesi
Department of Linguistics
Stockholm University, Sweden
beata.megyesi@ling.su.se

Abstract

Historical encrypted manuscripts require both paleographic interpretation of cipher symbols and cryptanalytic recovery of plaintext. Most existing computational workflows rely on a transcription-first paradigm, in which handwritten symbols are transcribed prior to decipherment. This intermediate step is labor-intensive, error-prone, and not always aligned with the goal of direct plaintext recovery. We propose an end-to-end, transcription-free approach that directly maps handwritten cipher images to plaintext. Using the Copiale cipher as a case study, we introduce the first text-line-level dataset pairing cipher images with German plaintext. We show that pretraining on generic handwriting data followed by cipher-specific fine-tuning substantially improves decipherment accuracy. Our results demonstrate that transcription-free image-to-plaintext decipherment is both feasible and effective for historical substitution ciphers, offering a simplified and scalable alternative to traditional pipelines. https://github.com/leitro/Decipher-from-Pixels-Copiale.

1 Introduction

Historical encrypted manuscripts pose a dual challenge at the intersection of document analysis and cryptology (Yin et al., 2019; Megyesi et al., 2020). As handwritten artifacts with encoded content, they require both visual interpretation and cryptanalytic decoding. Consequently, most computational approaches adopt a transcription-first workflow in which symbols are transcribed into machine-readable ciphertext and then analyzed to recover plaintext (Megyesi, 2020; Méndez et al., 2024). This paradigm has shaped research practices in historical cryptology and digital humanities.

Despite its success, transcription-first processing presents substantial limitations. Historical ciphers often exhibit large and irregular symbol inventories, inconsistent spacing, and idiosyncratic writing conventions, making segmentation and transcription labor-intensive and error-prone. Errors propagate into downstream decipherment and require frequent reference to manuscript images. Moreover, ciphertext is rarely the final scholarly objective; researchers primarily seek readable plaintext, raising the question of whether explicit transcription is always necessary.

Early computational work assumed reliable ciphertext and focused on cryptanalytic modeling. Aldarrab introduced a probabilistic noisy-channel framework (Aldarrab, 2017) and explored early image-based decipherment using segmentation and clustering, identifying character segmentation as a major bottleneck. Yin et al. later formulated decipherment from manuscript images as an integrated task combining visual processing and statistical cryptanalysis (Yin et al., 2019), demonstrating the feasibility of image-based approaches while retaining explicit symbol transcription. Together, these studies reflect a shift toward image-aware pipelines that nonetheless preserve staged processing.

Meanwhile, handwritten text recognition has advanced substantially through LSTM-based methods (Bluche et al., 2017), neural encoder–decoder models (Kang et al., 2018; Kang et al., 2021), transformer-based architectures (Kang

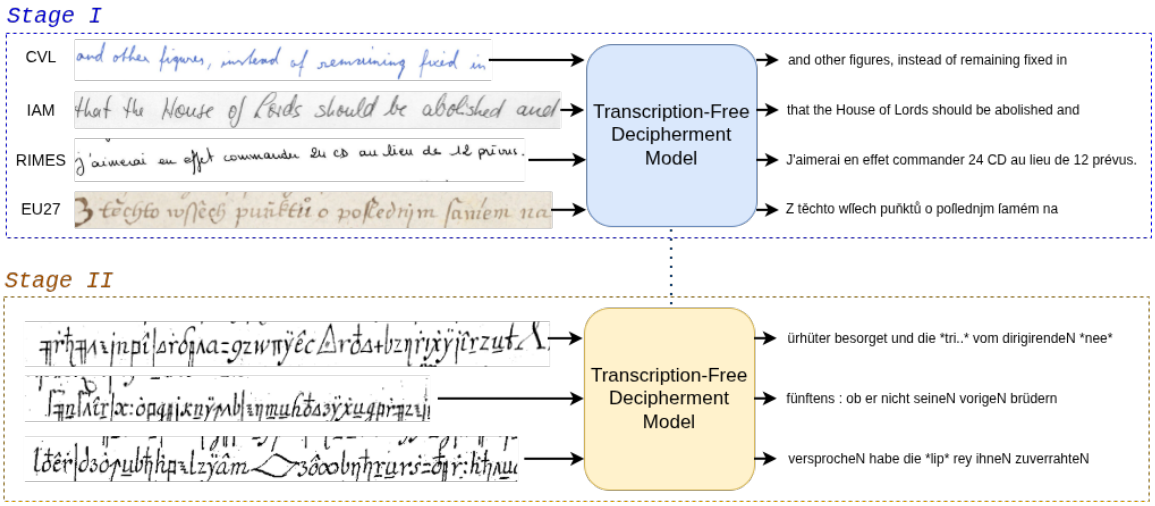


Figure 1: Overview of the training pipeline for our proposed Transcription-Free Decipherment paradigm. Stage I involves pretraining on a unified corpus of publicly available handwritten text-line datasets, followed by Stage II, where the model is fine-tuned on our curated Copiale image-to-plaintext text-line dataset.

et al., 2022), and large-scale systems such as TrOCR (Li et al., 2023). These developments enable direct image-to-text modeling and support end-to-end document understanding without explicit symbolic intermediates.

Building on these advances, we propose an end-to-end, transcription-free decipherment paradigm that directly maps handwritten cipher images to plaintext. Rather than supervising models with cipher symbols, we train them to generate decrypted natural-language text from pixel-level input. We evaluate this approach on the Copiale cipher (Knight et al., 2011; Knight et al., 2012), a well-studied eighteenth-century German manuscript.

Our contributions are threefold. First, we release the first publicly available text-line-level dataset pairing Copiale cipher images with aligned German plaintext. Second, we demonstrate the feasibility of end-to-end image-to-plaintext decipherment for historical manuscripts. Third, we show that pretraining on generic handwriting data followed by cipher-specific fine-tuning substantially improves performance. These results indicate that transcription-free pipelines provide a scalable alternative to traditional multi-stage workflows in computational historical cryptology.

2 Data

2.1 Handwriting Pretraining Data

Pretraining is conducted on a merged corpus comprising 66,492 handwritten text lines drawn from four widely used and complementary benchmarks: IAM (Marti and Bunke, 2002), CVL (Kleber et al., 2013), RIMES (Grosicki and El-Abed, 2011), and EU27 (Kohút and Hradiš, 2025). This combination is designed to maximize diversity in writing styles, languages, scripts, and acquisition conditions, thereby improving the robustness and generalization of the learned representations.

The IAM dataset provides a large collection of English handwritten text lines produced by multiple writers under controlled scanning conditions, and is commonly used as a standard benchmark for offline handwriting recognition. CVL complements IAM by offering high-resolution handwritten documents from a distinct writer population, with variations in writing instruments and stroke dynamics that enrich intra-writer and inter-writer variability. RIMES contributes French handwritten text collected in a realistic mail-processing scenario, introducing linguistic diversity as well as challenges such as cursive writing, ligatures, and noise typical of real-world document workflows. Finally, EU27 extends coverage to a multilingual European setting, incorporating handwriting samples across multiple scripts and orthographic conventions, which is particularly valuable for learn-

ing script-agnostic and language-robust features.

The combined corpus is randomly split into training, validation, and test sets using an 80/10/10 ratio, while preserving the overall distribution of datasets and handwriting styles. Detailed statistics on text-line contributions from each dataset are reported in Table 1.

Table 1: Pretraining data for Stage I.

Dataset	Textline Counts
CVL	13,440
IAM	8,873
RIMES	12,104
EU27	32,075
Total	66,492

2.2 Copiale Image-to-Plaintext Dataset

For cipher fine-tuning, we align handwritten Copiale text-line images with their corresponding German plaintext lines, producing the first dataset that supports direct image-to-plaintext decipherment of the manuscript. The number of lines, minimal and maximal number of characters and words per line for the training, validation and tests sets are described in Table 2. The entire dataset is released publicly to support future research.

Table 2: Fine-tuning data for Stage II: total number of lines, as well as the minimum and maximum number of characters and words per line, for the training, validation, and test sets of the Copiale.

Set	Count	Char Length		Word Length	
		Min	Max	Min	Max
Train	1,269	1	67	1	14
Validation	175	1	65	1	14
Test	370	1	64	1	13

3 Method

3.1 Transcription-Free Decipherment Formulation

We formulate decipherment as a sequence-to-sequence learning problem. In this paradigm, a model learns to transform one ordered sequence into another, without requiring explicit intermediate representations. Given an input image I representing a handwritten cipher text line, the model predicts a plaintext string P in natural language. No explicit ciphertext representation is produced or supervised during training.

3.2 Model Architecture

We adopt TrOCR (Li et al., 2023), a transformer-based encoder–decoder model for handwritten text recognition, and repurpose it for image-to-plaintext decipherment. A vision transformer encoder maps input images to latent visual embeddings, which are autoregressively decoded into plaintext tokens by a transformer decoder. Apart from adapting the output vocabulary, no architectural modifications are required. The overall training pipeline is shown in Figure 1.

3.3 Two-Stage Training Pipeline

We adopt a two-stage training strategy consisting of handwriting pretraining followed by cipher-specific fine-tuning. In Stage I, the model is pretrained on a unified corpus of publicly available handwritten text-line datasets (see Section 2.1), to learn robust and largely style-invariant handwriting representations. In Stage II, the pretrained model is fine-tuned on the Copiale image-to-plaintext dataset using German plaintext supervision (see Section 2.2). This strategy separates handwriting acquisition from task-specific learning: pretraining yields general visual representations independent of cipher symbols, while fine-tuning adapts these features for end-to-end decipherment.

4 Experiments

4.1 Implementation Details

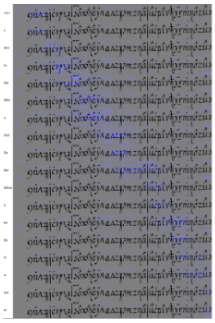
In Stage I pretraining, the model is trained for 5 epochs with a learning rate of 5×10^{-5} and a batch size of 64. For Stage II, we adopt a learning rate of 2×10^{-5} and a batch size of 64, and apply early stopping with a patience of 20 epochs based on validation performance, resulting in a total of 21 training epochs. AdamW is used as the optimizer, and the backbone model is `microsoft/trocr-base-handwritten`. All experiments are conducted on a single NVIDIA 4090 GPU.

4.2 Evaluation Metrics

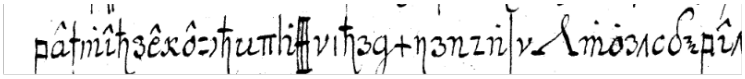
We report Character Error Rate (CER) and Word Error Rate (WER), which are standard metrics for handwriting recognition and sequence prediction. CER is defined as the normalized Levenshtein

Table 3: End-to-end Copiale cipher decoding performance evaluated with CER (%) and WER (%), where lower is better.

Method	Stage I	Stage II	Train Set		Validation Set		Test Set	
			CER↓	WER↓	CER↓	WER↓	CER↓	WER↓
Baseline	–	✓	42.58	93.06	45.85	101.81	46.10	98.48
Ours	✓	✓	**0.19**	**0.28**	**11.02**	**34.14**	**11.03**	**33.03**

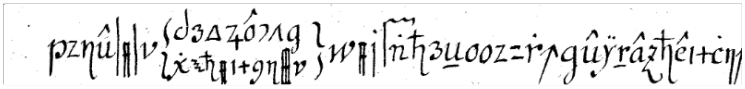


(a)



GT: beschwehre euch dass ihr mir das *nee* wort gebet
Pred: beschwehre euch dass ihr mir das *nee* wort gebet

(b)



GT: dieses { project geheimniss } erfahreN durch eine heimliche
Pred: dieses *bigx* *bigx* neugier *tri..* erfahreN durch eine heimlich

(c)

Figure 2: (a) Attention visualization illustrating alignment between handwritten cipher regions and decoded plaintext tokens. (b) Successful prediction example, where the model output (blue) matches the ground truth (black). (c) Failure case, where predictions (red) deviate from the ground truth (black), illustrating typical error patterns.

distance at the character level, CER $= \frac{S_c+D_c+I_c}{N_c}$, where S_c, D_c, and I_c denote the numbers of character substitutions, deletions, and insertions, respectively, and N_c is the total number of characters in the reference text. Similarly, WER measures error at the word level and is defined as WER $= \frac{S_w+D_w+I_w}{N_w}$, where S_w, D_w, and I_w denote the numbers of word substitutions, deletions, and insertions, and N_w is the total number of words in the reference text.

4.3 Results

The results are shown in Table 3. Handwriting pretraining yields a CER of 5.93% and WER of 17.99% on held-out handwriting data, indicating strong general visual representations. Table 3 compares direct fine-tuning on Copiale against our two-stage approach. The pretrained model dramatically outperforms the baseline, reducing test-set CER from 46.10% to 11.03% and WER from 98.48% to 33.03%. These results show that non-cipher handwriting data substantially improves decipherment accuracy, even though it contains no cryptographic structure.

5 Qualitative Analysis

Qualitative evaluation indicates that the model learns semantically meaningful alignments between handwritten cipher symbols and corresponding plaintext segments, even without explicit ciphertext-level supervision. As shown in Figure 2(a), the attention maps reveal coherent correspondences between spatial regions of the cipher input and decoded plaintext tokens, suggesting that the model captures the underlying structure of the cipher.

Figures 2(b) and (c) present representative successful and failure cases, respectively. In Figure 2(b), the predicted plaintext closely matches the ground truth, demonstrating reliable decoding performance across diverse inputs. In contrast, Figure 2(c) illustrates typical failure patterns, where predictions deviate from the ground truth, highlighting the limitations of the current model.

6 Conclusion

We proposed Transcription-Free Decipherment, an end-to-end approach to historical cipher decipherment that directly maps handwritten cipher images to plaintext. Using the Copiale cipher, we introduced a new publicly available image-to-

plaintext dataset and demonstrated that handwriting pretraining on non-cipher data dramatically improves decipherment performance. By collapsing transcription and decipherment into a single learnable mapping, our approach reduces annotation effort, simplifies processing pipelines, and opens new directions for scalable analysis of historical encrypted manuscripts.

Acknowledgments

This work has been supported by Riksbankens Jubileumsfond, grant M24-0028: Echoes of History: Analysis and Decipherment of Historical Writings (DESCRYPT); the Beatriu de Pinós del Departament de Recerca i Universitats de la Generalitat de Catalunya (2022 BP 00256); European Lighthouse on Safe and Secure AI (ELSA) from the European Union's Horizon Europe programme under grant agreement No 101070617; the Spanish projects CNS2022-135947 (DOLORES), PID2021-126808OB-I00 (GRAIL) and PID2024-157778OB-I00 (SUKIDI), the Consolidated Research Group 2021 SGR 01559 from the Research and University Department of the Catalan Government, and PID2023-146426NB-100 funded by MCIU/AEI/10.13039/501100011033 and FSE+. Alicia Fornés acknowledges financial support for her general research activities from ICREA under the ICREA Academia (Departament de Recerca i Universitats de la Generalitat de Catalunya).

References

Nada Aldarrab. 2017. Decipherment of historical manuscripts. Master's thesis, University of Southern California.

Théodore Bluche, Jérôome Louradour, and Ronaldo Messina. 2017. Scan, attend and read: End-to-end handwritten paragraph recognition with mdlstm attention. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, volume 1, pages 1050–1055. IEEE.

Emmanuele Grosicki and Haikal El-Abed. 2011. Icdar 2011-french handwriting recognition competition. In *2011 International Conference on Document Analysis and Recognition*, pages 1459–1463. IEEE.

Lei Kang, J. Ignacio Toledo, Pau Riba, Mauricio Villegas, Alicia Fornés, and Marçal Rusinol. 2018. Convolve, attend and spell: An attention-based sequence-to-sequence model for handwritten word recognition. In *German Conference on Pattern Recognition*, pages 459–472. Springer.

Lei Kang, Pau Riba, Mauricio Villegas, Alicia Fornés, and Marçal Rusiñol. 2021. Candidate fusion: Integrating language modelling into a sequence-to-sequence handwritten word recognition architecture. *Pattern Recognition*, 112:107790.

Lei Kang, Pau Riba, Marçal Rusiñol, Alicia Fornés, and Mauricio Villegas. 2022. Pay attention to what you read: non-recurrent handwritten text-line recognition. *Pattern Recognition*, 129:108766.

Florian Kleber, Stefan Fiel, Markus Diem, and Robert Sablatnig. 2013. Cvl-database: An off-line database for writer retrieval, writer identification and word spotting. In *2013 12th international conference on document analysis and recognition*, pages 560–564. IEEE.

Kevin Knight, Beáta Megyesi, and Christiane Schaefer. 2011. The copiale cipher. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, pages 2–9.

Kevin Knight, Beáta Megyesi, and Christiane Schaefer. 2012. The secrets of the copiale cipher. *Journal for Research into Freemasonry and Fraternalism*, 2(2):314.

Jan Kohút and Michal Hradiš. 2025. Practical finetuning of autoregressive models on limited handwritten texts. In *International Conference on Document Analysis and Recognition*, pages 22–39. Springer.

Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. 2023. Trocr: Transformer-based optical character recognition with pre-trained models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 13094–13102.

U-V. Marti and Horst Bunke. 2002. The IAM-database: an english sentence database for offline handwriting recognition. *International journal on document analysis and recognition*, 5(1):39–46.

Beáta Megyesi, Bernhard Esslinger, Alicia Fornés, Nils Kopal, Benedek Láng, George Lasry, Karl de Leeuw, Eva Pettersson, Arno Wacker, and Michelle Waldispühl. 2020. Decryption of historical manuscripts: the DECRYPT project. *Cryptologia*, 44(6):545–559.

Beáta Megyesi. 2020. Transcription of historical ciphers and keys. In *3rd International Conference on Historical Cryptology, Histocrypt 2020*, pages 106–115. Linköping University Electronic Press.

Martín Méndez, Pau Torras, Adrià Molina, Jialuo Chen, Oriol Ramos-Terrades, and Alicia Fornés.

2024. Structured analysis and comparison of alphabets in historical handwritten ciphers. In *European Conference on Computer Vision*, pages 330–344. Springer.

Xusen Yin, Nada Aldarrab, Beáta Megyesi, and Kevin Knight. 2019. Decipherment of historical manuscript images. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 78–85. IEEE.

The Secret Writing of Michele Zoppello: An Introduction

Marco Vito
marcovito001@gmail.com

Abstract

Michele Zoppello is an insufficiently studied figure within the late medieval cryptographic landscape. Although his biography remains largely uncertain, it is enriched by the existence of a treatise on cryptography, which provides valuable insights into a rapidly evolving intellectual milieu.

This paper takes a case study from his treatise, *Litterarum Simulationis Liber*, to examine the cipher system he devised, with particular attention to its strengths and limitations, thereby clarifying its role in the history of cryptography and its implications in the period. This study provides a basis for future research and further analysis of both Zoppello's figure and his treatise, representing an initial step toward introducing this subject into scholarly debate on the history of cryptography.

1. Introduction

This study aims to provide an initial framework for further analysis. Michele Zoppello's work fits within the evolving context of secret writing and represents an early example of a cryptographic treatise of the fifteenth century. The *Litterarum Simulationis Liber* is dedicated to the highest figure of authority: the pope.

It was primarily intellectuals who engaged in the creation of these ingenious forms of secret writing, both to demonstrate their own skill and to provide a service to those who could benefit from it, whether clerics, ambassadors, or other figures of power involved in encrypted communication. Indeed, it was power itself, and its political role, that drove the adoption of ever-new systems to protect messages, as the information they contained could determine the success or failure of a given strategy.

Another fundamental aspect was military use, which Zoppello himself mentions in parts of his treatise, providing example letters that could plausibly be employed in various contexts.

The method adopted by Zoppello appears to rely on a nomenclator, which could be modified as needed. He organized it alphabetically to make the operation of the system clearer. Nomenclators had already been in use previously. When a sender composed a letter using the nomenclator, they had to replace the key words with those listed in the nomenclator; conversely, to reconstruct the message, even without rewriting the letter, the recipient needed to substitute the corresponding words from the nomenclator with the appropriate current equivalents.

Between 1325 and 1327 (Meister, 1906), during the conflicts between the Guelphs and Ghibellines, some letters from Pope John XXII Duèse to his Angevin ally Robert of Naples were disguised: names, titles, and places were replaced with fictitious words. Despite this, enemies

sometimes managed to break the ciphers and decrypt the papal correspondence.

Consequently, it was essential to continually update the system to prevent adversaries from understanding the messages. The oldest surviving document using this encryption method, dating to 1326–1327, concerns the battles fought between Guelphs and Ghibellines at Orvieto and Viterbo, fueled by Ludwig of Bavaria's Roman campaign. The document mentions several prominent figures, representative of the political realities of the early fourteenth century.

For illustrative purposes, only two examples of this cipher are presented here:

> "*Item pro Viterbio: Civitas nova:, eo quod de novo fuerit in civitatem et sedem episcopalem erect[a]*. […]
> *Item per Gebellinis predictorum latorum vel aliorum: egyptii: Et per Guelfis: filii israel;*"

As can be observed, the toponym Viterbo is concealed through the expression '*Civitas Nova*', accompanied by an etymological explanation linked to its recent elevation to an episcopal see. Similarly, the Ghibellines are identified as the 'Egyptians', while the Guelphs are referred to as the 'sons of Israel'.

A similar principle of substitution with alternative words is also present in Zoppello's treatise. However, a closer reading reveals more complex substitution dynamics, which are more closely aligned with the use of nomenclators that, from the fifteenth century onward, characterized monoalphabetic substitution ciphers.

2. Historical context of the Treatise

The figure of Zoppello remains insufficiently known, with only a few fragmentary references attested. Future research will aim to reconstruct his biography and provide a more comprehensive analysis of his treatise, complementing the work presented here.

Michele Zoppello was a Venetian from Sacile and served as secretary to Duke Ludovico of Savoy. He is mentioned in several diplomatic letters, which allow for the reconstruction of brief episodes from his life, including his imprisonment and subsequent release by order of Francesco Sforza (Herman, 2019; Soffietti, 2011).

His cryptographic treatise is entitled *Liber Litterarum Simulationis et Dezifris*, also known as *Litterarum Simulationis Liber*[1]. The variation in the title arises from the wording on the verso of the cover and that recorded in the incipit of the work, where the second formulation appears:

> "Ad Beatissimu*m* sanctissimu*m* et dominum nostrum D*omini* C*allisti* P*apae* III mei Michaelis Zopello Sacilensis Litterarum Simulationis Liber Incipit"[2]

The book is dedicated to Pope Callixtus III Borgia[3], pontiff from 1455 to 1458, a

[1] This treatise has been digitalised from University of Pennsylvania, Kislak Center for Special Collections, Rare Books and Manuscripts. Url: https://bibliophilly.library.upenn.edu/viewer.php?id=LJS%20225#page/1/mode/2up.

[2] English translation: *To the Most Blessed, Most Holy, and our Lord, Lord Callistus Pope III, me Michaelis Zopello of Sacile - book of the Simulation of Letters - incipit.*

[3] In 1455, Alonso de Borja was elected pope, assuming the name Callixtus III. His pontificate (1455–1458) was largely dominated by the crusade against the Turks

circumstance that allows the treatise to be chronologically placed within the few years of his pontificate.

The historical context follows the Ottoman capture of Constantinople in 1453, the Peace of Lodi the following year, and the subsequent creation of the Italian League in 1455, established to counter the foreign threat.

Callixtus III's main objective was the implementation of an anti-Turkish crusade, promoted throughout Europe, but which received limited support from the major political powers of the time. With the exception of Hungary, which was directly threatened by invasion, most European powers were primarily concerned with defense, consolidation, and the expansion of their own authority.

For the Papal States, the main source of instability remained the Italian peninsula, where Alfonso of Aragon attempted, through the condottiero Jacopo Piccinino, to besiege Siena.

This action contributed to weakening the balance of the Italian League and, indirectly, to obstructing the Pope's anti-Turkish ambitions, forcing him to concentrate his efforts on stabilizing the political situation in Italy, ultimately neutralizing the pro-Aragonese condottiero through an economic agreement.

Within this context, Zoppello's treatise appears to have been conceived for a specific purpose: not so much to confront the Turkish threat, but to address internal tensions within the Italian peninsula and preserve the system of alliances.

This interpretation is primarily supported by a linguistic factor. The idiomatic difference between the Latin West (characterized by the coexistence of Latin and vernacular languages) and the Ottoman world, associated with Arabic, already constituted a significant communicative barrier. Consequently, the use of ciphers was less necessary in contacts outside territories where language represented the primary obstacle to understanding. By contrast, in the Italian peninsula, where secret writing was already widespread and conflicts were constant, interest in cipher systems was predominantly directed toward internal use, even though the papacy's political attention was chiefly focused on the East.

Zoppello dedicates his work to the Pope both to demonstrate his own expertise and to provide the papacy with a tool to support the defense of the political integrity of Italian alliances, through a secret communication system that was innovative for the time and capable of ensuring safer message transmission.

3. Practice and method of cipher use

Zoppello's work is a short treatise consisting of 20 folios, the first of which is richly illuminated. In the initial section, the author presents the work, introducing its methodology and the various implementations. The following section comprises a nomenclator arranged alphabetically, listing words and their substitutions. The subsequent part provides several worked examples; one of these is the case examined here, as it adopts one of the different

following the fall of Constantinople, achieving limited political success but marked by intense diplomatic and military activity. See Rendina, 2006; Navarro Sorni, 2006.

substitution schemes employed by Zoppello. Finally, the treatise includes a monoalphabetic cipher with homophones, as well as a nomenclator in which specific words are not replaced by other words, but by signs, symbols, letters, and numbers, following the model of monoalphabetic ciphers that became widespread during the fifteenth century.

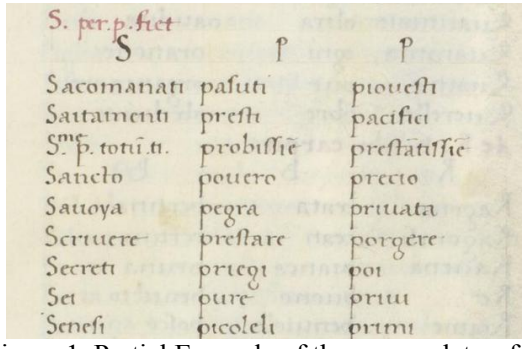


Figure 1. Partial Example of the nomenclator, f. 12v.

Zoppello's work focuses primarily on the use of the nomenclator, which consists of a sample of words for each letter of the alphabet, from 'a' to 'z'.

The list follows an alphabetical order, where the first row corresponds to the plaintext words, while the two subsequent rows indicate the two possible substitute words. The presence of two alternatives allows, depending on the context of the message, the selection of the most appropriate word to conceal the message.

This aspect is crucial because, if the text were to appear suspicious, it would become evident that the seemingly plain message actually conceals a code:

> "Per alphabetum ordo notatur, mutatur que primo .a. in .c. et sic successive mutantur et aliae figuris aliis. diplicantur .n. mutatae, ut si uno vocabulo non bene sonaret simultatio alio concordari queat in melius"[4]

The nomenclator is then used in different ways with letters invented by Zoppello, of a diplomatic-military nature, whose main themes concern prisoners or women whose husbands are imprisoned. This choice was intended to avoid raising suspicion about the possible falsification of messages and to make them appear to be used in the most credible context possible.

Since a full treatment would require too much space, this study focuses on a single example, encrypted according to the rules of the treatise. The complete text is provided in the appendix.

Before proceeding, it is necessary to address a primary limitation of the system. Being a nomenclator that essentially concerns common words and occasionally notable figures, and that uses two ciphertext words for each plaintext word, using the nomenclator continuously limits the possibility of encrypting the entire text. For example, if the ciphertext word 'poi' (after) is used to encrypt '*secreti*' (secrets), it could not be used elsewhere; if it were needed in the cleartext, it would be impossible, as it would correspond to the same ciphertext word. To make the system function, the nomenclator can only be applied selectively.

Zoppello proposes several strategies. One of these is to place a full stop at the end of each

[4] English translation: *Using the alphabet, one can observe a fixed order: first .a. is changed into .c., and thereafter other words are altered in the same way; some are doubled, with further changes of .n., so that if the simulation does not work well with one word, it can be better matched to another*.

word that is to be encrypted. In this way, punctuation is used to indicate where the nomenclator should be applied and where the word should be left in plaintext. Returning to the previous example, the word 'poi' can thus be used either as a cipher word or in plaintext, depending on the presence of the full stop as a recognition mark.

The reference rule is as follows:

> "Atqui ut teneat qui script*arum* erit cuius sit partis partium […] si principio cuius vis line*a*e: post primam habitam partis prim*a*e: qu*a*e verax est: erit quod scribetur: in fine pr*a*ecedentis. line*a*e, virgula […] ponatur hoc modo (,). Si vero secund*a*e; […] qu*a*e mutatur, punctis subponatur sic (.) firmus. si superflu*a*e per iadum isto modo (:)"[5]

Therefore, if the text is in plaintext, a comma ',' is used; if the text is meaningless and written only to mislead, a colon ':' is used; if, instead, the text - or more precisely the word - is to be encrypted, it is preceded by a full stop '.'.

In the treatise, Michele Zoppello does not present fully encrypted examples, but rather models on which messages can be encrypted according to different types of rules. One of these, the use of the full stop, applies to the text presented here.

[5] English translation: *However, whoever is recording must keep track of which part belongs to whom for each of the previously [...] If at the start of a line you wish to indicate, after noting the first part of the first line - which is true - what will be written will appear at the end of the previous line; a comma should be placed in order like this: ','. If it is instead the second part; [...] the one that is changed, dots should be placed beneath it like this: '.' (full stop). If there are superfluous parts, they should be indicated per iadum like this: ':'.*

It is a confidential message expressing concerns about the situation in Italy. The writer worries about the stability of the Italian League, as the various armies have not been disbanded, and adds two reasons. The first is that mercenary captains and condottieri have many men under their command. These leaders need war to maintain their armies, because without it they cannot pay their men, and the captains themselves cannot survive without it, since they have no interest in retiring from arms to devote themselves to agriculture. The second reason is that without a unified militia in Italy, the very principle of territorial defense would be lost, which would be extremely damaging given the invading forces beyond the mountains (oltramontani).

The letter concludes with the idea of hiring and paying a salary in place of the condotte, while those who have many men under their command should gradually be deprived of part of them, engaging the soldiers in tournaments and exercises to avoid idleness.

The words to be encrypted in the letter are marked with a full stop, according to Zoppello's rule.

In the text, full stops appear in twelve instances corresponding to as many words to be written in cipher.

The essential passages are reported below:

> "et non havendo provisto intuto ale zentedarme. per diversi modi e da dubitare non surga più exterminio di guera de quela e stata. […] como è el S*ignore* Sigismondo[6].

[6] Sigismondo Pandolfo Malatesta, born in Brescia in 1417 and the illegitimate son of Pandolfo III Malatesta, was

conte Iacomo Pic*cinino*[7]. Guido Ro*n*gone[8] […] et suscitare nova guera. ni credere se vole gli voglieno gire ala zapa. […] che fora pezor male del continuo stato da molti anni in qua. […] a Ro*man*i acadete havendo loro in deditione tuta Italia. […] intendeti per no*n* ve atediare più in longo lezere. […] et agli utili de condegno stipendio provedere. […] et cossi gradatim. et agli omini darme et ale fanterie."

The words preceding the full stops and their corresponding ciphertext equivalents are:

1. zentedarme = hora o honeste
2. Sigismondo = Padre, Pietro o piacevole
3. Jacomo Piccinino = nostro o novello
4. guera = erba elemosina
5. zapa = harena hami
6. Italia = nunciata nominata
7. fanterie = leze o lieve

While four other words have no corresponding cipher equivalents and are the following:

8. stata
9. qua
10. lezere
11. provedere
12. gradatim

By substituting the words, the sentences would read as follows:

"et non havendo provisto intuto ale honeste. per diversi modi e da dubitare non surga più exterminio di guera de quela e stata. […] como è el S*ignore* Pietro. conte nostro. Guido Ro*n*gone […] et suscitare nova erba. ni credere se vole gli voglieno gire ala harena. […] che fora pezor male del continuo stato da molti anni in qua. […] a Ro*man*i acadete havendo loro in deditione tuta nominata. […] intendeti per no*n* ve atediare più in longo lezere. […] et agli utili de condegno stipendio provedere. […] et cossi gradatim. et agliomini darme et ale leze."

Following Zoppello's criterion, one immediately observes limited effectiveness, both in terms of the number of encrypted words used and the risk that some full stops may, accidentally or otherwise, correspond to words not present in the nomenclator. In addition, the choice to encrypt only a few words allows the simulation of the true content of the message, which is nonetheless primarily contained in the sensitive words addressed to key figures.

a condottiero of fifteenth-century Italy. Between 1447 and 1461 he came into conflict with Alfonso of Aragon and later with Pope Pius II, who excommunicated him in 1461 and waged military action against him. Defeated, he progressively lost his dominions and died in Rimini in 1468. See Rendina, 1985; Falcioni, 1998.

[7] A condottiero born in Perugia in 1423 and the son of Niccolò Piccinino, he was active in the wars between Italian states in the fifteenth century. In 1450 he commanded Milanese troops, and later transferred his service to the Venetians against Francesco Sforza following the fall of the Ambrosian Republic. After 1454 he formed a mercenary company, and between 1456 and 1462 he fought in southern Italy, shifting his allegiance from Alfonso of Aragon to John of Anjou. In 1463 he aligned himself with Ferrante of Aragon, but, suspected of treason, he was arrested and executed in Naples in 1465. See Rendina, 1985; Ferente, 2005.

[8] Guido Rangoni 'il Vecchio', was born in the early years of the fifteenth century. He belonged to a family linked to the Este and began his military career in 1431 in the service of Bologna. From 1434 he entered the permanent service of the Republic of Venice. After the Peace of Lodi (1454), he remained formally in Venetian service, consolidating his feudal and political positions. He died in 1467. See Andenna, 2018.

Encrypting the word ‘zapa’, while altering the final meaning, carries less weight than encrypting the name of Jacopo Piccinino.

The encrypted phrase ‘como el Signore Pietro. conte nostro’ perfectly simulates the original, rendering it equally credible despite being encrypted. Likewise, the expressions ‘et suscitare nova erba. ni credere se vole gli voglieno gire ala harena’ allude to an agricultural context rather than a war-related expression.

Only from this analysis is it possible to present certain considerations.

4. Utility and analysis of the Cipher

Zoppello’s cipher exhibits several limitations, although these are confined to the single example presented, but it also has aspects that are original for the fifteenth century.

First of all, the use of punctuation is an innovative idea for the period, as it introduces a component that is usually excluded from ciphered writings into the encryption system (Alberti, 1998; Natale, 1961). Punctuation simplifies decryption, marks the beginning and end of sentences, and facilitates comprehension. In this case, punctuation is an integral part of the cipher, as well as necessary to indicate where words should be substituted.

Nonetheless, it is not particularly effective, since ink smudges or misplaced marks can confuse the reader in possession of the nomenclator when attempting to decrypt the message. Furthermore, the marks are easily identifiable and may reveal the system for choosing which words to encrypt and which to leave in cleartext, once the rule is understood. Consequently, the use of punctuation, while original, exposes the cipher to a risk of systematic recognisability.

However, a strength remains the nomenclator itself, because even if the punctuation system were understood, it would remain almost impossible to identify the cleartext words.

The weakness, however, is that Zoppello does not place any mark after the name of the mercenary captain Guido Rongone; indeed, he places a comma to indicate that the text should be left as it is. Yet leaving the mercenary captain name in plaintext compromises the message, since the topic becomes easily identifiable through reference to his figure.

Moreover, the limited number of encrypted words means that only some passages can be effectively hidden, while still preserving originality in word choice. This aspect limits the overall effectiveness of the cipher, without undermining its conceptual originality.

Zoppello’s system intends to use a nomenclator in a new and dynamic way. Since he explains the rule but does not want the work to be immutable, it must instead adapt to the needs of the message, provided that the commissioners are familiar with the rules and the words to be used in cipher.

The choice of such a system, apparently detached from the contemporary practice of monoalphabetic substitution ciphers with homophones, is in fact fully consistent with the cryptography of the late fourteenth and fifteenth centuries, because at the end of the treatise, in the last two written pages, Michele Zoppello presents precisely a monoalphabetic cipher that follows the rules of monoalphabetic ciphers:

"Et nota q*ue* scribitur diffuse absq*ue* dista*n*tia sillabar*um* et sine puncto vel virgula"[9]

This means that Zoppello, although aware of contemporary cipher systems, deliberately focused on a system different from the one that was becoming widespread. The two systems could not be used simultaneously: integrating the monoalphabetic system with the nomenclator would have been impossible, since punctuation is not employed in the former, whereas it is required in the latter.

For this reason, Zoppello places the monoalphabetic system at the end of the treatise, to demonstrate his knowledge of it; this does not reflect a lack of expertise, but rather indicates that it is not the principal focus of his work.

From a broader perspective, Zoppello's treatise provides a fixed point from which one can analyse later examples that adopt the idea of substitution via entire words—most notably the so-called 'Ave Maria' cipher of Trithemius (1518), presented in his *Polygraphiae*, as discussed by Gamer (2022), although the name of the system was not given by Trithemius himself. In this cipher, each word of the prayer corresponds to a single letter of the alphabet. Trithemius devotes 190 pages to the system, proposing 380 possible Latin alphabets (*minutiae*), with each letter assigned a corresponding word. This yields a total of 9,120 words, after which the sequence begins again with a new set of words.

[9] Zoppello, 1455-1458, 20r). English translation: *And it is a notation that is written in a continuous manner, without spacing between syllables and without pictograms or commas.*

5. Conclusions

The present work has shown that, upon initial analysis, Zoppello's system differs from other contemporary approaches to secret writing and that he sought to employ cryptography in a distinctive manner compared to figures who, between the fifteenth and early sixteenth centuries, devised new methods of secret writing.

An example of this is the collection of Milanese ciphers compiled by Francesco Tranchedino, preserved in the Austrian National Library in Vienna (ÖNB, cod. 2398) and in the *Codex Urbinate* held in the Vatican Apostolic Archive (AAV, Urb. lat. 998) which contains monoalphabetic substitution ciphers. These include sections related to the nomenclator system (mostly with words substituted by pairs of letters and numbers, and not always by other words), but for the most part rely on alphabetical substitution.

Through this contribution, Michele Zoppello's work is situated within the broader landscape of historical cryptography studies, highlighting an early use of nomenclator-based techniques.

This study, far from being considered conclusive, represents a first step toward a more comprehensive examination of Michele Zoppello's work and his role in the development of fifteenth-century cryptographic practices.

References

Archivio Apostolico Vaticano (AAV), *Codex Urbinate*, Urb. lat. 998.

Leon Battista Alberti. 1994. *Dello Scrivere in cifra*, Buonafalce Augusto (curated by). Galimberti, Torino.

Leon Battista Alberti. 1998. *De Componendis Cyfris*, Buonafalce Augusto (curated by). Galimberti, Torino.

Andenna Giancarlo. 2018. Rangoni, Guido il Vecchio. In *Dizionario Biografico degli Italiani (DBI)*, 86. Url: https://www.treccani.it/enciclopedia/guido-il-vecchio-rangoni_(Dizionario-Biografico)/.

Anna Falcioni. 1998. *La signoria di Sigismondo Pandolfo Malatesti*. B. Ghigi, Rimini.

Serena Ferente. 2005. *La sfortuna di Jacopo Piccinino: storia dei bracceschi in Italia (1423-1465)*. L. S. Olschki, Firenze.

Maximilian Gamer. 2022. *Die Polygraphia des Johannes Trithemius nach der handschriftlichen Fassung: Edition, Übersetzung und Kommentar*. Brill, Leiden, Boston.

Nicholas Herman. 2019. Who was Michele Zopello?. In *Fifty-two discoveries from the BiblioPhilly project*, No. 35/52, (on-line): https://bibliophilly.pacscl.org/who-was-michele-zopello/.

Aloys Meister. 1906. *Die Geheimschrift im Dienste der päpstlichen Kurie*. Druck und Verlag von Ferdinand Schöningh, Paderborn.

Alfio Rosario Natale (curated by). 1961. *I diari di Cicco Simonetta*. A. Giuffre, Milano.

Miguel Navarro Sorni. 2006. *Callisto III: Alfonso Borgia e Alfonso il Magnanimo*. Oliva Anna Maria and Chiabò, Maria (curated by). Roma nel Rinascimento, Roma.

Paul Michel Perret. 1890. Les règles de Cicco Simonetta pour le déchiffrement des écritures secrètes (4 juillet 1474). *Bibliothèque de l'École des chartes*, 51: 516-525.

Claudio Rendina. 1985. *I capitani di ventura. Storia e segreti: le affascinanti biografie dei condottieri italiani nell'età delle Signorie e dei Principati. I protagonisti di una grande epopea mercenaria tra battaglie e congiure, tradimenti e violente passioni*. Newton compton editori, Roma.

Claudio Rendina. 2006. *I papi. Storia e segreti: dalle biografie dei 265 romani pontefici rivivono retroscena e misteri della cattedra di Pietro tra antipapi, giubilei, conclavi e concili ecumenici*. Newton compton editori, Roma.

Isidoro Soffietti. 2011. A proposito di un segretario veneto del duca Ludovico di Savoia. *Bollettino storico-bibliografico subalpino*, 109: 607-614.

Francesco Tranchedino. 1475-1496. *Furtivae litterarum notae, quibus usus fuisse videtur in cancellaria Vicecomitum Mediolanensium 1450-1496*. Österreichische Nationalbibliothek (ÖNB), cod. 2398.

Johannes Trithemius. 1518. *Polygraphiae libri sex Ioannis Trithemii abbatis Peapolitani, quondam Spanheimensis ad Maximilianum Caesarem*. Ioannis Haselbergi de Aia, Basel.

Michele Zoppello. 1455. *Litterarum Simulationis Liber*. Roma.

Appendix A. Example of a message to be encrypted, as presented

- Abbreviations have been expanded and are shown in italics.
- The original punctuation has been maintained, as intentionally applied by the author Michele Zoppello.
- Diacritical marks have been added to facilitate the reading of the Italian text, although they are not present in the manuscript.
- Many words appear without spacing and are written continuously; for the purposes of transcription, they have been separated to enhance readability.
- Letters enclosed in square brackets [ ] indicate characters omitted in the original text and supplied for completeness and comprehension.

Ali*a*e litter*a*e simultand*a*e, emanand*a*e q*ue* e locis sociorum vel amicorum

Havendo fata questa pace et liga univ*er*sale de la Italia, et non havendo provisto intuto a le zente darme. per diversi modi e da dubitare non surga più exterminio di guera de quela è stata. primo che quigli sono cassi et da cavalo et da piedi como è el S*ignore* Sigismondo. conte Iacomo Pic*cinin*o. Guido Ro*n*gone, et infiniti altri quali hano le conduc*c*te grande, et qual di loro non le vora tenir*e* a so[e] spese ni cassar le vorano, et lo [s]forzo di loro non hano il modo de tenire epse gente se*n*cia soldo, el che li farà mistiero far qualche + novitate, et suscitare nova guera. ni credere se vole gli voglieno gire ala zapa. Secundo che mancando la militia in Italia, et dandosi a l'ocio, non fia Italia da oltramontani invasa, che fora pezor male del continuo stato da molti anni in qua. como più volte a Ro*man*i acadete havendo loro in deditione tuta Italia. Las[c]io molte altre ragione quale per suma sapientia V*ostra*. intendeti per no*n* ve a tediare più in longo lezere. et conchiudo che al tuto se voria fare che quisti tali et tuti altri soldati sono de reputatione fosseno proveduti, non dico con le conduc*c*te usate, ma con gli homini stimati: et tal famigli desutili: apti alopre mecanice, cassare: et agli utili de condegno stipendio[10] provedere. q*ue*gli haveano mille cavagli ne tenisseno seicento, et cossi gradatim. et agli omini d'arme et a le fanterie. et quisti hogni mese havesseno fare lor mostre armate et giostre con simili exercitii: acio non ma*r*cisseno in ocio.

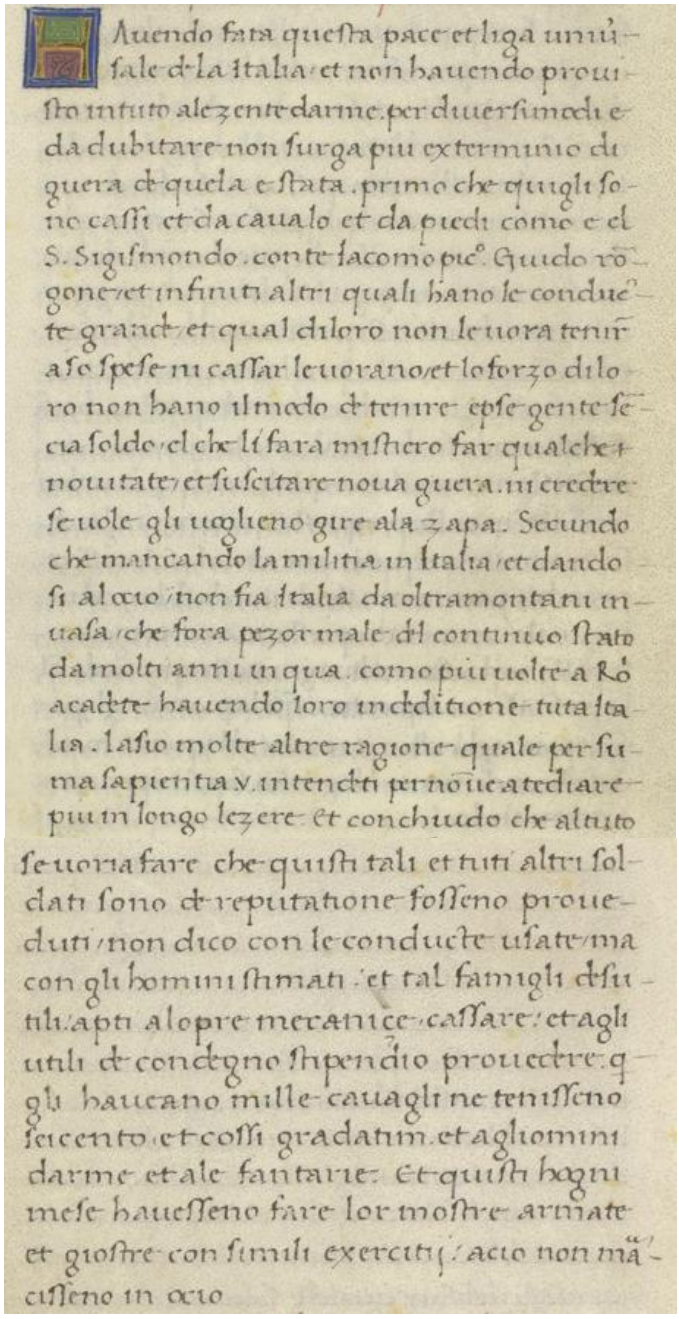
English translation:

Having established this peace and a universal league in Italy, and having made no provision to dismantle the armies, there is reason to fear in various ways that a new and even greater outbreak of war may arise than what has already occurred.
First: because there are captains of cavalry and infantry, such as Lord Sigismondo, Count Jacopo Piccinino, Guido Rangoni, and countless others, who hold large commands. None of them can maintain these forces at their own expense, nor do they wish to disband them; and their efforts alone cannot sustain troops without pay, which will force them to seek new ventures and potentially provoke a new war. One should not imagine they are willing to turn to farming.
Second: because if Italy lacks a proper militia, and men give themselves over to idleness, Italy shall not be invaded by foreigners, which would be a great misfortune for the ongoing stability of the state, as had often happened to the Romans, when they held all of Italy under their dominion.
I leave many other reasons unspoken, which Your Wisdom can understand, so as not to weary you with a long reading. I conclude that it would be wise to ensure that these men and all other soldiers of reputation are properly provided for—not necessarily through the usual commands, but by assigning trustworthy men, removing unsuitable families, and providing competent soldiers with adequate pay.

[10] Mark written above the letter 'd'.

Those who had a thousand horses should keep six hundred, and so gradually. Likewise, provisions should be made for men-at-arms and infantry. These forces should also conduct monthly demonstrations, armed exercises, and jousts, so that they do not fall into idleness.

Appendix B. The letter examined in this article



Hauendo fata questa pace et liga unu-
sale de la Italia / et non hauendo prou-
sto intuto alezente darme per diuersi modi e
da dubitare non surga piu exterminio di
guera de quela e stata. primo che quigli so-
no cassi et da caualo et da piedi come e el
S. Sigismondo. conte Iacomo pic. Guido rō-
gone / et infiniti altri quali hano le conduc-
te grande / et qual di loro non le uora tenir
a so spese ni cassar le uorano / et lo forzo di lo-
ro non hano il modo de tenire epse gente se-
cia soldo / el che li fara mistiero far qualche
nouitate / et susitare noua guera. ni credre
se uole gli uoglieno gire ala zapa. Secundo
che mancando la militia in Italia / et dando
si al ocio / non sia Italia da oltramontani in-
uasa / che fora pezor male de l continuo stato
da molti anni in qua. como piu uolte a Rō
acadete hauendo loro in deditione tuta Ita-
lia. lasio molte altre ragione quale per su-
ma sapientia v. intendti per nō ue atediare
piu in longo lezere. et conchiudo che al tuto
se uoria fare che quisti tali et tuti altri sol-
dati sono de reputatione fosseno proue-
duti / non dico con le conducte usate / ma
con gli homini stimati / et tal famigli de su-
tili / apti a lopre meranice / cassare / et agli
utili de condegno shpendio prouedre. q
gli haueano mille cauagli ne tenisseno
seicento / et cossi gradatim. et agli homini
darme et ale fantarie. Et quisti hogni
mese hauesseno fare lor mostre armate
et giostre con simili exercitij / acio non mā-
cisseno in ocio

Cryptographic Practices within Jacobite Diplomatic Networks (1715–1745): A Typology of Ciphers and Keys in Wartime

Camille Rocher
Université Bordeaux Montaigne
19 esplanade des Antilles
33607 Pessac, France
Camille.rocher@u-bordeaux-montaigne.fr

Abstract

This paper examines cryptographic practices within Jacobite diplomatic networks between 1715 and 1745 through the systematic analysis of a corpus of 57 cipher keys among the 318 ones preserved in the Stuart Papers. While Jacobite correspondence has been studied from political perspectives, the cipher keys themselves have rarely been considered as historical objects deserving systematic analysis. Focusing on periods of intense military and diplomatic activity, this study investigates the structure, typology, and circulation of cipher keys used by James Francis Edward Stuart and his agents across Europe. This article proposes a typology based on the combination of cryptographic features observed in the corpus and hybrid systems. A frequency-based analysis highlights recurrent patterns in key design and reveals a strong preference for hybrid systems combining jargon with monoalphabetic or homophonic substitution. The results show no clear correlation between the complexity of a cipher key and the status of the correspondents involved, suggesting that pragmatic constraints, such as usability, availability, and key reuse, played a decisive role in cryptographic choices. By foregrounding cipher keys as artefacts of communication, this study contributes to a better understanding of early modern cryptographic practices and sheds new light on the operational dynamics of Jacobite diplomatic networks during wartime.

1 Introduction

The study of early modern cryptography has increasingly shifted from modest-size ciphers toward broader analyses of cryptographic practices and systematic studies on the development of ciphers and the way they were encrypted (Megyesi, 2021; Ulbert, 2024). Among the materials preserved from this period, cipher keys occupy a central position, as they define not only the technical mechanisms of encryption but also reflect concrete practices of communication, secrecy, and risk management within political and diplomatic networks.

The Stuart Papers (c. 1689–1800), preserved at the Royal Archives in Windsor, contain an exceptional collection of cipher keys associated with Jacobite correspondence. These keys were used by James Francis Edward Stuart and members of his political and diplomatic networks operating across Europe. While Jacobite correspondence has been studied from political and diplomatic perspectives, the cryptographic dimension of these materials, and particularly the cipher keys themselves remains largely unexplored.

This article examines a corpus of 57 Jacobite cipher keys dating from 1715 to 1745, focusing on their structure, typology, and usage. In doing so, the study contributes to ongoing discussions in historical cryptology regarding the practical and contextual dimensions of cipher key design.

2 Historical Context

In 1689, following what is known in British historiography as the “Glorious Revolution” of 1688, James II and VII (1633–1701), who had been deposed from the thrones of England and Scotland, sought refuge in France at the court of Louis XIV, accompanied by approximately 40,000 supporters. This event marked the beginning of a prolonged struggle to recover the English and Scottish crowns, successively led by James II and VII and later by his descendants:

first his son James Francis Edward Stuart (1688–1766), known as the Old Pretender, and then his grandson Charles Edward Stuart (1720–1788), the Young Pretender. They were supported in their efforts by those who remained loyal to the Stuart dynasty, commonly referred to as the Jacobites (Pittock, 1998).

In order to advance their claims, the Jacobites needed to secure financial resources, troops, and arms. One of the primary means of achieving this was by seeking international support from foreign monarchies, particularly during periods of heightened geopolitical tension. As a result, several invasion attempts were launched in the context of major European conflicts, including the rising of 1715, shortly after the War of the Spanish Succession (1701–1714) and during the Great Northern War (1701–1721), as well as the projected invasion of 1717 and the failed landing of 1719, which coincided with the War of the Quadruple Alliance (1718–1720). The final and most significant attempt took place in 1745, during the War of the Austrian Succession (1740–1748) (Szechi, 2019).

3 Sources and Corpus

This work is based on the *Stuart Papers*, both the volume dedicated to cipher keys[1] and the correspondence between the Jacobite agents across Europe and the central power in periods of conflicts, meaning between September 15, 1715 and June 29, 1719[2], and January 1st, 1744 to August 8, 1745[3]. This period represents a sample of 57 keys preserved in the dedicated volume. This choice was made to be able to compare the keys given to the different agents, their use, and the possible keys missing from the register.

4 Analysis of Cipher Keys

4.1 Terminology and Typology

A cipher key defines the process by which plaintext elements are replaced with ciphertext equivalents in order to produce encrypted messages. The following terminology refers exclusively to the encryption systems identified in the sample of keys analysed in this study. Jargon (J) is a specific type of nomenclor in which a plaintext word is replaced with a ciphertext word rather than with a ciphertext character. Such substitutions usually concern personal names, geographical locations, or politically sensitive terms (Figure 1). Such substitutions usually concern nomenclator elements, most commonly personal names, geographical locations, or politically sensitive terms.

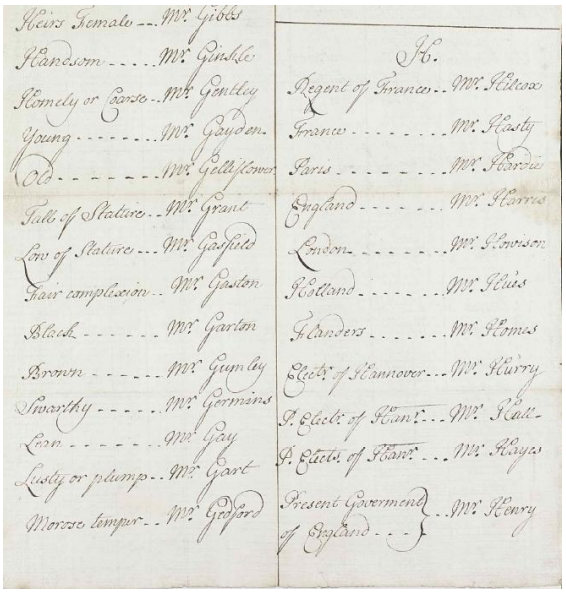


Figure 1. Example of jargon (RA SP/MAIN/5, f. 41)

Cipher keys that rely solely on nomenclators (N) encrypt plaintext elements using two- or three-digit codes assigned to a fixed list of words (Figure 2). Monoalphabetic substitution (M) assigns to each plaintext character a single ciphertext element. Homophonic substitution (H) allows a single plaintext element to be represented by several distinct ciphertext elements, in order to flatten frequency distributions and resist statistical attack. Polyalphabetic substitution (P) uses several substitution alphabets in alternation, typically governed by a keyword or a fixed pattern, so that the same plaintext character can be encrypted differently depending on its position in the message (Kahn, 1996). To complexify decryption, the most frequently occurring plaintext characters or words in a language might have several corresponding ciphertext characters. Cipher elements may be either fixed-length, in which all regular ciphertext elements consist of the same number of digits, or variable-length. While most of the surviving cipher keys used 3-digits ciphertext elements, some used ciphertext

[1] Royal Archives, Windsor, Stuart Papers, BOX/5, 290 keys, 992 folios.

[2] RA SP/MAIN/5 to RA/SP/MAIN/43.

[3] RA SP/MAIN/255 to RA/SP/MAIN/266.

elements that were between one and four digits long. Within the present corpus, all ciphertext elements are composed exclusively of words or numbers; no graphic symbols are used. Some keys also include dedicated ciphertext elements for punctuation marks. A final category concerns the use of nulls (Figure 2), that is, cipher elements without semantic value, which may serve either to introduce meaningless content or to cancel neighboring ciphertext elements. Those elements usually come under specific instructions of use. Since the creation of a new key was both tedious and time-consuming, already-existing ones were sometimes reused; either in full or with minor adaptations.

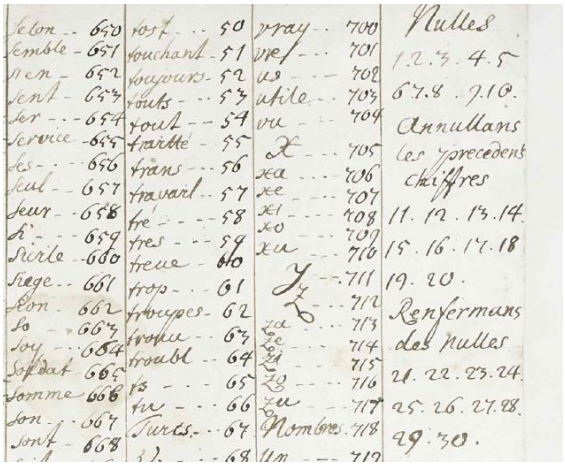


Figure 2. Example of a display of 2 and 3-digits code followed by nulls (RA SP/BOX/5, f. 45).

Rather than treating these techniques as mutually exclusive, this study adopts a combinatorial typology, in which each key is classified according to the set of cryptographic features it incorporates. This approach reflects the hybrid nature of early modern cipher keys and allows for meaningful comparison between systems of varying complexity while preserving their internal structure. This is similar to what is found in other early modern times keys (Megyesi et al., 2024).

4.2 Statistical analysis

A frequency-based analysis was conducted to identify recurring patterns and usage of cipher keys within the corpus. Rather than aiming at statistical exhaustiveness, this approach seeks to highlight tendencies in the distribution of cryptographic features and the relative complexity of the keys employed (Figure 3).

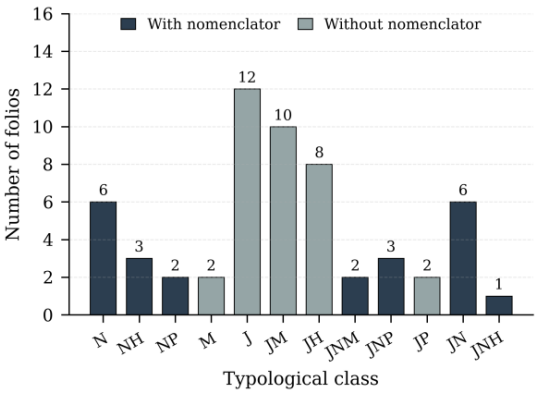


Figure 3. Distribution of cipher key types

The corpus is dominated by cipher keys relying partially or fully on substitution-based systems. Jargon encryption appears as the most frequent technique. However, in the majority of cases, this system is combined with two or more cryptographic features, such as monoalphabetic substitution, or nomenclator cipher, resulting in hybrid cipher keys rather than strictly homogeneous designs. Although sole nomenclators are less frequent than other substitutions, it appears in keys with a significantly higher number of entries (Figure 4).

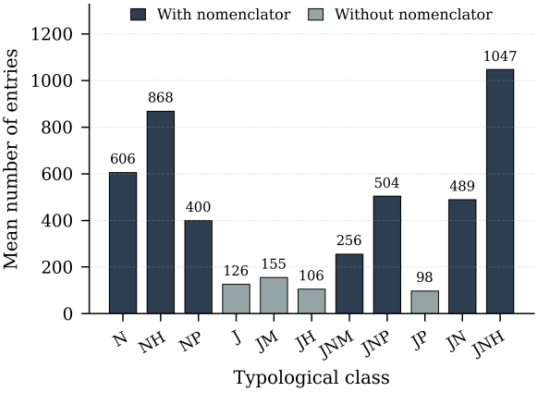


Figure 4. Number of entries per cypher key

This association indicates a tendency to reinforce encryption strength when the volume or importance of encrypted information increased. Conversely, keys with very limited lexical content tend to rely on substitution schemes, likely reflecting constraints of usability and the need for rapid deployment.

The most found system is a combination of homophonic substitution and nomenclator. This hybridization allows the agents greater autonomy and flexibility in the information they were allowed to transmit and the way they could

encrypt it, while allowing for greater complexity in encryption and thus limiting the risk of decryption by adversaries.

At the current stage of analysis, no clear correlation can be established between the structural complexity of a cipher key and the specific individuals or correspondents between whom it was used. While some highly elaborate keys are associated with prominent figures or sensitive exchanges (e. g. James Stuart or his secretary the duke of Mar), similarly complex systems also appear in more routine correspondence. This absence of a straightforward relationship suggests that factors other than individual status, such as practical constraints, availability of keys, or habits of reuse may have played a determining role in the choice of encryption methods.

4.3 Key Reuse and Circulation

Beyond their internal structure, cipher keys must also be understood as objects in circulation within communication networks. The analysis of the Jacobite corpus reveals that the production of cipher keys did not necessarily follow a strict logic of uniqueness. Instead, pragmatic considerations often governed key management.

At least three explicit cases of cipher key reuse have been identified within the corpus. In these instances, the same key appears to have been used over extended periods or shared between multiple correspondents. Such practices suggest that the creation of new keys was constrained by material, logistical, or cognitive factors. Copying, distributing, and memorising a key required time and effort, particularly within transnational networks operating under conditions of secrecy.

Beyond these clear cases, structural similarities observed across several keys raise the possibility of partial reuse or borrowing of cryptographic components, such as identical substitution alphabets or closely related nomenclator entries. While the present analysis does not allow for a definitive identification of systematic reuse, these patterns point toward a repertoire-based approach to cryptographic practice, in which pre-existing and trusted models were adapted rather than replaced.

Nine keys were addressed to two or more correspondents simultaneously, indicating that they were designed from the outset for collective use within sub-networks. This dimension of cryptographic practice deserves further investigation, particularly with regard to the geography and social composition of these sub-networks.

5 Conclusion & Future Works

Future research could include the transcription of all the cipher keys already identified, following the methodology developed by Tudor (2019), in order to enable a comprehensive analysis of the topics encrypted in Jacobite ciphers, such as the distribution and prominence of place names, personal names, and sensitive terms. Patterns of cipher reuse could also be explored using statistical methods, while the frequency of key usage in surviving correspondence represents another promising line of inquiry. Finally, the work of Tudor, Megyesi, and Láng (2020) on the automatic extraction of cipher key structures may offer valuable perspectives for further development.

References

David Kahn. 1996. *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet*. Scribner, New York.

Beáta Megyesi, Benedek Láng, Nils Kopal, Vasily Mikhalev, Crina Tudor, and Michelle Waldispühl. 2024. A Typology for Cipher Key Instructions in Early Modern Times. In *Proceedings of the 7th International Conference on Historical Cryptology,* HistoCrypt24, pages 183-193. Linköping University Electronic Press.

Beáta Megyesi, Crina Tudor, Benedek Láng, Anna Lehofer, Nils Kopal, Karl de Leeuw, and Michelle Waldispühl. 2024. Keys with Nomenclatures in the Early Modern Europe. In *Cryptologia*, 48(2), pages 97-139. Linköping University Electronic Press.

Beáta Megyesi, Crina Tudor, Benedek Láng, Anna Lehofer, Nils Kopal, and Michelle Waldispühl. 2022. What Was Encoded in Historical Cipher Keys in the Early Modern Era? In *Proceedings of the 5th International Conference on Historical Cryptology*, HistoCrypt22, pages 159-167. Linköping University Electronic Press.

Beáta Megyesi, Crina Tudor, and Benedek Láng, Anna Lehofer. 2021. Key Design in the Early Modern Era in Europe. In *Proceedings of the 4th International Conference on Historical Cryptology*, HistoCrypt21, pages 121-130. Linköping University Electronic Press.

Beáta Megyesi. 2020. Transcription of Historical Ciphers and Keys. In *Proceedings of the 3rd International Conference on Historical Cryptology*, HistoCrypt20, pages 106-115. Linköping University Electronic Press.

Murray Pittock. 1998. *Jacobitism.* Macmillan Press Ltd, Basingstoke, UK.

Daniel Szechi. 2019. *The Jacobites: Britain and Europe.* Manchester University Press, Manchester.

Crina Tudor, Beáta Megyesi, and Benedek Láng. 2020. Automatic Key Structure Extraction. In Proceedings *of the 3rd International Conference on Historical Cryptology*, HistoCrypt20, pages 146-152. Linköping University Electronic Press.

Crina Tudor. 2019. *Studies of Cipher Keys from the 16th Century: Transcription, Systematisation and Analysis*. Master thesis in Language Technology, Uppsala University, Sweden.

Jörg Ulbert. 2024. Le chiffre diplomatique français au dix-huitième siècle. In Sébastien Drouin and Sébastien Côté (ed.), *Secrets et surveillance épistolaires dans l'Europe du dix-huitième siècle*, pages 1-17. Voltaire Foundation in association with Liverpool University Press, Liverpool, UK.

The Codebook of Willem Six van Oterleek: Dutch Diplomatic Intelligence from Saint Petersburg between 1806-1810

Florentijn van Kampen
iHub, Radboud University
Nijmegen – The Netherlands
florentijn.vankampen@ru.nl

Abstract

Between 1806 and 1810, Willem Six van Oterleek served as diplomatic representative of the Netherlands in Saint Petersburg, Russia. During these years, van Oterleek kept his Minister of Foreign Affairs informed as fully as possible about political developments, information he received from other diplomats and the *couleur locale* from his posting in Saint Petersburg. This sensitive communication was to be kept secret and was therefore sent in code. These coded messages and the accompanying codebook that was used to protect them survived in the Dutch National Archives. This article will explore this codebook, analyse code usage and decode the secret messages to present a unique peek behind the curtain of the diplomatic developments and intrigues of those days.

1 Introduction

Van Oterleek's posting began during turbulent times for Europe. Napoleon Bonaparte had crowned himself Emperor of the French in 1804, and in June 1806, he installed his younger brother Louis as King of Holland. In Russia, Tsar Alexander I had come to power in 1801 after his father was murdered. Right at the start of van Oterleek's posting in 1806, the Holy Roman Empire was dissolved during the Napoleonic Wars.

The Dutch Republic had been dispatching special missions to Moscow since 1615. Relations between the Netherlands and Russia intensified during the reign of Tsar Peter the Great. The Batavian Revolution of 1795 formally ended diplomatic relations with Russia. The Danish *chargé d'affaires* was entrusted with the legation's three chests of correspondence and chancery papers. The Danes managed Dutch affairs, without explicit authorization, until the arrival of a new envoy, in 1803.

The Dutch National Archive in The Hague holds the archive of the Dutch Legation in Russia between the years 1720 and 1810[1]. This archive contains, among other paperwork, the correspondence of the Dutch diplomatic representation in Russia with the Dutch Parliament and the Minister of Foreign Affairs from 1720 until 1810.

Willem Six van Oterleek (1761 – 1811) was a Dutch diplomat who served as "Extraordinary envoy and minister plenipotentiary" from 1806 to 1810. During his posting in Saint Petersburg, the capital of the Russian Empire at the time, he used a codebook to send secret diplomatic messages to the Minister of Foreign Affairs in the Netherlands, Willem Frederik baron Röell (1767 – 1835). The archive

[1] https://www.nationaalarchief.nl/onderzoeken/archief/1.02.13

material from this envoy forms the heart of this article.[2]

This material was previously noted by Karl de Leeuw, who was a pioneer in the research of historical cryptology in the Netherlands. He presented this archive material at the HistoCrypt conference of 2018 (De Leeuw, 2018). At that conference, he described the codebook and some of the encoded messages. De Leeuw's final conclusion at that time was that to decode the messages "(...) the codebook (...) is a likely candidate, but we are not sure (...)". No further references in academic literature to this material exist. Apparently, he never was able to apply the codebook successfully to decode the messages.

This article will explore this codebook and decode the messages. It will show how the codebook worked and what can be learned from the decoded messages. Section 2 will introduce the codebook together with its author, its design and how it was supposed to be used. Section 3 will use an example of an actual encoded message to demonstrate the usage of the codebook in practice. Section 4 will discuss the cryptographic properties of the codebook supported by statistical observations from the encoded messages. Section 5 presents the conclusions. Appendix A provides an overview of all the messages in the archive after their successful decoding.

1.1 Open source material

All the raw material, including the original scans, the transcriptions of the messages, the codebook, and the Python code to decode and analyse the messages is publicly available in a GitLab repository[3]. Readers are welcome to download the material and verify the observations and results from this article.

[2]Both the codebook (1.02.13.226) and the encoded messages (1.02.13.228) are in the archive.

[3]`https://gitlab.science.ru.nl/fvankampen/histocrypt-2026-oterleek-codes`

The codebook, messages and transcription are also available in the Decode database (Megyesi et al., 2019) (Megyesi et al., 2020) (Héder and Megyesi, 2022) for further integration into the broader research into historical cryptology. [4]

2 The Diplomatic Codebook

From a terminological perspective, it is important to note that the term *codebook* in this article is used in a somewhat colloquial fashion. The official designation for this type of system is *nomenclator*. This follows Mikhalev et al. (2023) which provides an overview of terminology for the field of historical cryptology. But since the term "codebook" more naturally conveys "a book containing codes for encoding and decoding messages", this article uses both terms. When analysing the code system in a systematic way, the more specific and correct term *nomenclator* is used.

The entire codebook consists of two parts. The first part is a two-page introduction that explains the usage of the system and provides some general guidance on how to write coded messages. The second part is the actual nomenclator and is 18 pages long and provides the mapping between numbers and *nomenclature elements*. Mikhalev et al. (2023) gives a precise definition:

> **Nomenclature element**: A plaintext element which is above the alphabet level. A nomenclature element can be a syllable, a name, a function and a content word as well as a phrase.

The following section will explore the codebook's introduction.

[4]Codebook `https://de-crypt.org/decrypt-web/RecordsView/1035` and messages `https://de-crypt.org/decrypt-web/RecordsView/1033` for the messages.

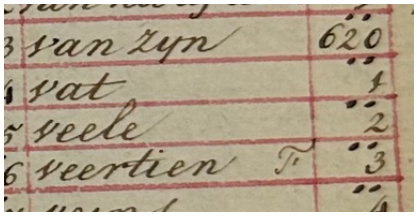


Figure 1: The word *veertien* (fourteen) with a *F* marker

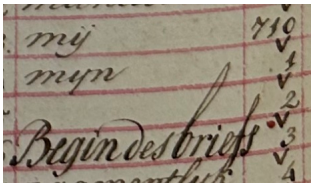


Figure 2: Entry for 7ĭ3 *Begin des briefs*

2.1 Author and Usage

The two-page preamble to the codebook provides an introduction on how to use the nomenclator and some guidelines on the security aspects of encoding messages. The codebook was designed to support both Dutch and French as plaintext languages. Most entries were specifically assigned for either Dutch or French words or phrases. In some cases however, a special marker in the form of a letter "F" or "D" indicated that a word could be used in both languages, but that the end user was responsible for its translation. The introduction gives the example of the entry for "fourteen", as shown in figure 1. This entry can be used for messages in both languages even though the entry only lists its Dutch original "veertien". In a French message, this should be translated to "quatorze".

According to the introduction, users were allowed to freely switch between these two languages while encoding the plaintext (although in the actual messages from the archive, there is no sign of this). The start and end of messages however, should be marked with specific Dutch headers. The letter should start with the phrase "Begin des briefs" (start of letter) and end with "Einde des briefs" (end of letter). These phrases were encoded by *single* numbers (see Figure 2 for an example). For security purposes, which will be explored in section 4, the codebook provided for multiple encoding alternatives for these very specific header phrases. Apparently, this demand was sometimes ignored, since there are messages in the archive that leave out these headers and start with the actual content right away.

The introduction very specifically forbids users to mix cleartext and ciphertext. It explicitly prohibits "(...) even the addition of a single plaintext letter". The codebook employs a system of "null" numbers. These numbers are present in the codebook, but have no associated plaintext entry. They can be used freely, to obscure the structure of the message. The introduction states that they should specifically be used at the beginning or end of a message, to obscure message boundaries. Besides these special "null" characters, the user was also allowed to use "gibberish" words before or after the special headers. In that way, it was obvious that these words were not part of the message and that they could safely be used to obscure the actual message even further.

The system also provided a grammar modification, in the form of a single dash mark after a code number. This dash mark would transform a single noun to a plural noun and a verb to its past participle form.

The introduction is signed by its designer "S.E. Croiset", The Hague, August 5, 1803 (see Figure 3). De Leeuw (2000) introduces this Dutch codebreaker and designer Samuel Egbert Croiset (1734 – 1816).

From 1738 the Dutch diplomatic nomenclators were designed by "the secretary of ciphers" Pieter Lyonet (1706 - 1789). Lyonet

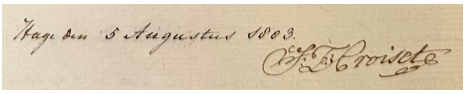


Figure 3: The signature of the author: S.E.Croiset

incorporated the lessons learned from code-breaking in the continuing evolution of new versions of Dutch diplomatic codebooks (De Leeuw, 2000). Croiset was a nephew of Lyonet and from 1756 onwards involved in designing Dutch diplomatic codebooks as well as in the cryptanalysis of other nations' codes.

De Leeuw (2000, p 27) gives a detailed description of some of the earlier codebooks, like the one designed by Lyonet in 1756: "The book still contained about 4000 items but each page was now divided into five columns in stead of three and would have a box left blank at the top or the bottom of the page. The code-groups would follow the columns, their meanings would follow the rows (...)". This description clearly resembles the design of the codebook of Croiset which will be presented in the following section.

2.2 Design

As stated, the nomenclator of Croiset consists of 18 pages, each page with the same layout. Every page is divided into 5 columns of 71 rows. Each column of the top 6 rows is split into two smaller columns. This results in a total of 385 entries per page: 6 rows with 10 columns (60 entries) and 65 rows with 5 columns (325 entries). A total of 18 pages results in a maximum capacity of 6930 entries. Some entries are left empty for the purpose of *null* codes.

The entries are filled in alphabetically in row order while the numbers are incremented per column. The smaller entries at the top of each page, with each column split in two, contain entries for single and double letters,

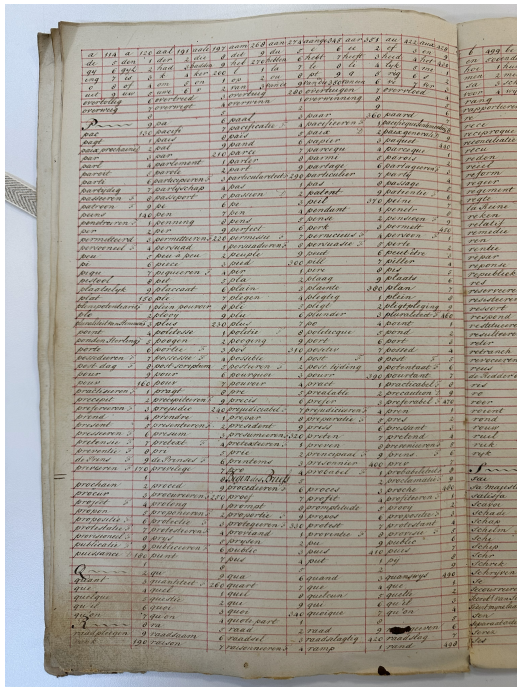


Figure 4: Example page from the codebook

frequently occurring syllables, particles and small prepositions. These split columns have their own alphabetical ordering from A to Z on each page. An example page can be seen in Figure 4 and a schematic overview in Figure 5.

The codebook is divided into seven different sections, each with numbers in the range of 1 to 999. Each section has a specific, distinguishing marker written above the number, with one section having no marker. The first four sections, and approximately half of the fifth section, contain alphabetically ordered letters, syllables, words and phrases. The remaining part of the sections contains more specific entries such as geographical names, diplomatic phrases, and treaty terminology. Although the words are ordered alphabetically row-wise and the numbers are increased column-wise, the different sections still main-

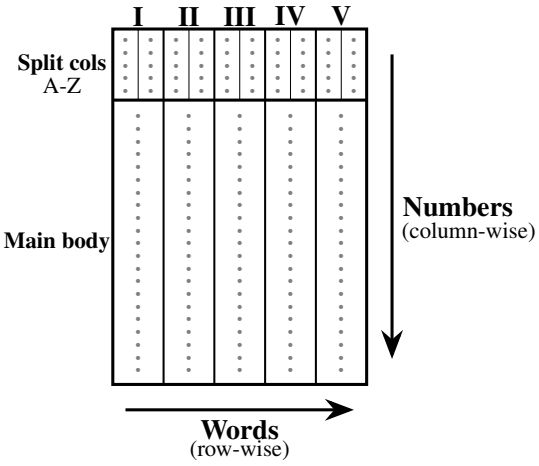


Figure 5: Schematic overview of codepage

Nr	Marker	(Alphabetical) Content
1	Tilde	General entries *A-D* and often used words and syllables
2	Quote	General entries *D-J* and often used words and syllables
3	Caret	General entries *J-O* and often used words and syllables
4	(no marker)	General entries *O-S* and often used words and syllables
5	Colon	General entries *S-W* and often used words and syllables
6	Plus	General entries *W-Z* and Geographic locations, titles, and international relations
7	Equals	Specific names and dates

Table 1: Markers and sections

tain an overall alphabetical order on a page level. This can be seen in table 1 where an overview is given of the seven sections of the codebook, the marker for each section and the specific flavour of words.

3 The codebook in practice

This section will use an example message to show the practical workings of the codebook. It demonstrates choices made during the encoding process. This message was also chosen as a typical example of the kind of diplomatic intelligence messages that were sent from Saint Petersburg to the Netherlands.

Figure 6 shows the first half of encoded message Nr. 4063. The first line starts with the following codes: `861 785 212+ 709" 96+ 17+`. To find the plaintext, we have to look up each code in the right section of the codebook. The first code has no special marker. From Table 1 we can see that this is the fourth section of the codebook. When we look at the entry at number `861`, it is empty. This means that this is a so called *Null* code. Remember that the introduction to the codebook (see section 2.1) specifically advises the

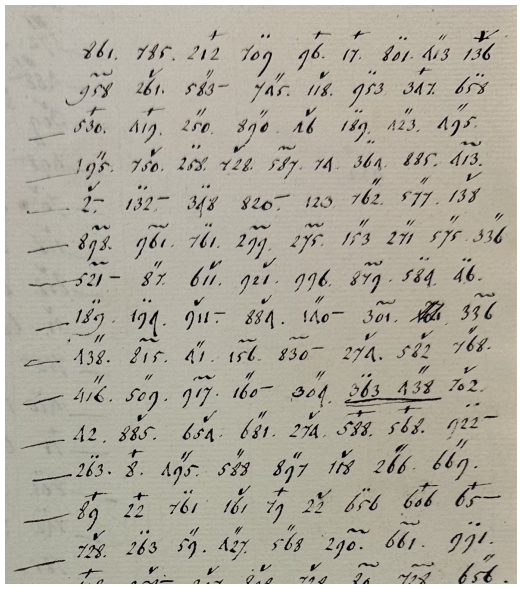


Figure 6: First half of encoded message 4063

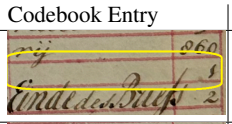
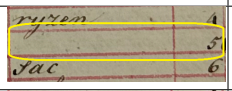
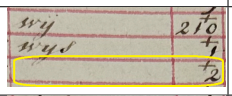
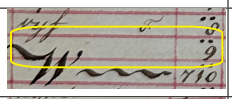
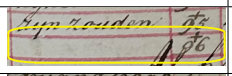
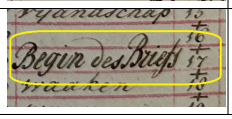
Code	Codebook Entry	Plaintext
861		Null
785		Null
212+		Null
709:		Null
96+		Null
17+		*Start of Letter*

Table 2: Header of message

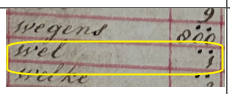
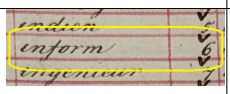
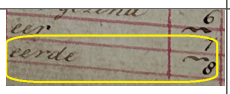
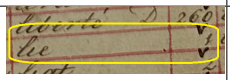
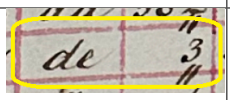
Code	Codebook Entry	Plaintext
801:		Wel *(well)*
413¨		ge
136^		inform
958~		eerde *(informed)*
261^		lie
583¨		de + plural *(persons)*

Table 3: Start of message content

user to use these *Null* at the start of the message "to obscure message boundaries". We continue this process with the next codes: code 785 is in the same "without marker" section, code 212+ is in the section with marker +, etc. The message header ends with code 17+ which has the Dutch meaning "Begin des Briefs" or "Start of Letter" in English. After this, the actual message begins. The summary of the message header can be seen in table 2.

We will now examine the first codes from the actual message to see how plaintext Dutch was encoded with this codebook. The codes we are going to look at are 801: 413¨ 136^ 958~ 261^ 583¨ -. Table 3 shows the result of the decoding. The message starts with the Dutch sentence "Wel geïnformeerde lieden" or "Well informed persons" in English.

The Dutch words "geinformeerde" and "lieden" are split up into smaller parts, each part encoded separately. The word *geïnformeerde* becomes: ge-inform-eerde and the word *lieden* becomes *lie-de-n*. The last encoding makes use of a special character, the dash ("-"), to show that the plural form of the plaintext word is to be used. The syllable "lie" is appended with the word "de" (the same as the Dutch definite article) the result made plural by the addition of the dash after the code number. The final result: "lie-de-n", or "persons" in English.

This process demonstrates how the plaintext Dutch was processed *syntactically* in such a way that it was supported by the available entries of the codebook, i.e., words, syllables and letters.

To conclude this example, a transcription of the message from figure 6 is provided below, together with the decoded entries from the codebook and an English translation.

```
[NULL:861] [NULL:785] [NULL:212+]
[NULL:709:] [NULL:96+]
Begin des Briefs[17+]
wel[801:] ge[413"] inform[136^]
eerde[958~] lie[261^] de-en[583"]
gelooven[745"] te[118^] weeten[953:]
dat[347+] in[658:] Saxen[530+]
```

```
een[419+] form[250"] ee[890"] l[46^]
Tractaat[189:] het[423"] geen[495"]
tot[195"] nu[750^] toe[258:]
niet[728^] best[587~] ond[74]
tusschen[364:] de[885] beide[413~]
Keizer-en[2^] sluiten-en[132:]
en[348:] ratificeren-en[820] is[123]
geworden[762"] de[577"] inhoud[138^]
daar van[898~] is[961~] voor[761:]
alle[299~] een[275~] die[153"]
p[271"] geheim[575"] ge[336"]
blijven-en[521~] doch[87"] men[611^]
onder[921^] steld[996]
eensdeels[879~] dat[584:] dit[46:]
Tractaat[189:] eene[194:]
offensif-en[911^] en[884^]
defensief-en[140"] alliantie[301~]
beh[336~] el[438"] s[815~] t[41"]
ander[156~] deel-en[830~]
dat[274^] een[582^] onderdaan[768^]
gedeelte[416"] van[509"]
deszelven[917~] despositie-en[160"]
even[304"] tue[363:] el[438"]
moeten[702^] zijn[42] in[885^]
het[654^] geval[681"] dat[274^]
Engeland[588+] aan de[568+]
voorslag-en[922:] tot[263:]
vrede[8+] geen[495"] gehor[588"]
geliefd[897"] te[118^] geeven[266"]
gelijk[669"] zulks[89+]
waarschijnlijk[22+] voor[761:]
komt[161^] zo[79+] lang[22^] de[656:]
Spaansche[606+] zaak-en[65+] niet[728^]
tot[263:] decisie[59"] zijn[427"]
gebr[568"] ach[290~] t[661~]
```

The English translation:

> Well-informed persons believe to know that in Saxony[5] a formal Treaty, which until now did not exist between the two Emperors, has been concluded and ratified. The contents thereof have remained a deep secret for everyone, but it is supposed on the one hand that this Treaty contains an offensive and defensive alliance.
>
> On the other hand, a subordinate part of its dispositions must be conditional in the event that England does not deign to give ear to the proposals for peace, as appears probable as long as the Spanish affairs[6] have not been brought to a decision. (...)"

[5] Reference to the Conference of Erfurt (September 27 - October 14, 1808), where Napoleon and Tsar Alexander I met to renew their alliance from the Treaty of Tilsit (1807). The meeting took place in Erfurt, the capital of the Principality of Erfurt in Saxony.

[6] The Peninsular War (1808-1814)

4 Cryptographic Observations

As with all systems based on a codebook without additional encryption, the book itself is the secret key. Once (part of) that book is reconstructed by a third party, all communications – past, present and future – are compromised. This particular codebook has some additional security functionality in its design that is worth exploring. This design by Croiset and Lyonet seemed to have been fairly common during the late 18th century (Kahn, 1996).

4.1 The security of the codebook

First, the codebook has been designed in such a way that some words have multiple encoding options. More formally, some plaintext elements have multiple ciphertext equivalents. This property is called *homophony* and is used to counter basic frequency analysis of often-used words and characters. The effectiveness of this security feature in practice is of course fully dependent on the discipline and craftsmanship of the encoder. The next section will provide some insights into how this was done in this particular situation.

Second, the codebook provides mechanisms to choose different ways of encoding for the same plaintext element. Since the system provides ways to encode single characters or syllables, some words can either be encoded using a number for the whole word, or combined by using building blocks on a smaller level. Again, the use of this possibility is dependent on the routine and craftsmanship of the person performing the encoding.

Third, the system is composed in such a way that the words are filled in alphabetically row-wise and the numbers are increased per column as shown in Figure 5. This means that the strict relation between the alphabetical order of the nomenclature elements and the number is not a simple one-to-one mapping. This guards against (very) elementary

codebook cryptanalysis where a codebreaker could know in what alphabetical range a word would fall into, only depending on the number and some knowledge of reconstructed words. Once this property becomes clear to the codebreaker, its additional layer of security is removed.

4.2 Homophony statistics

The archive contains 35 encoded messages from the posting of van Oterleek, all encoded with the same codebook. This makes it possible to gather statistics on how the system was used in practice. In this section we will examine the use of *homophony*. The messages have a total of 5359 code numbers with 1767 unique codes.

Often-used words, like prepositions or articles, have multiple entries in the codebook, giving the encoder a choice of which specific code number to choose in a particular sentence. To make maximum use of this security feature, the encoder would have to choose as many different numbers for a specific word as possible. This is a manual process with the quality fully dependent on the discipline and creativity of the person encoding the message.

To measure the use of this technique in this case, the following statistics were gathered: Take the top 10 words with the most encodings in the codebook that are used in the messages. For each of these words, the number of different encodings in the codebook is counted and the number of different encodings in the messages. The number of different encodings in the codebook is the theoretical maximum the encoder can use. The number of different encodings in the messages shows the actual choices of the encoder. The results of these statistics can be seen in Table 4. The fact that the observed numbers in the messages are very close, or even equal to the theoretical maximum, is a sign of true professionalism and craftsmanship of the encoder.

Dutch entry	**Nr codes in codebook**	**Nr codes in messages**	**English**
de	14	13	the
en	14	13	and
te	13	12	to
ge	11	11	(past participle prefix)
is	11	11	is
in	11	9	in
het	10	10	it
met	9	9	with
zijn	9	9	are
op	8	8	on

Table 4: Statistics on top 10 homophones

5 Conclusions

Between 1806 and 1810 Willem Six van Oterleek served as the special Dutch envoy in Saint Petersburg. During his posting, he used a Dutch diplomatic codebook to send encoded messages about political developments to his Minister of Foreign Affairs in the Netherlands.

This codebook and a collection of 35 messages have been preserved in the Dutch National Archive. Analysis of this codebook, combined with historical research, shows its position in a lineage of Dutch codebooks that started in the 18th century (De Leeuw, 2000). The codebook was specifically designed for diplomatic communications and supported both Dutch and French plaintext. The messages show that Dutch plaintext was *syntactically* processed in such a way that it could be supported by the available entries of the codebook, i.e., words, syllables and letters.

The messages and codebook also provide insight into how such a system was used in real life. Analysis of the encoded messages shows the use of *homophonic* encoding to add security of the system. Statistics show that the

people using the system were aware of the security properties of the codebook and that they were properly trained for this task.

Finally, the decoded messages provide a unique insight, from an eyewitness standpoint, into Dutch diplomatic intelligence in a very turbulent era. Van Oterleek operated as an intelligence professional cultivating local sympathetic contacts. His reports on the Erfurt Conference, in the messages referred to as "the Saxon Conference", show this. He relied on conversations with the French Ambassador to Russia, Caulaincourt, salon gossip, and trusted contacts to piece together what had been secretly agreed between the two emperors. Together with the codebook, the decoded messages provide a valuable insight into how diplomatic intelligence actually flowed through early nineteenth-century Europe.

Acknowledgments

I would like to thank Prof. Dr. Bart Jacobs and Dr. Rowin Jansen of the Radboud University for their valuable ideas, critical readings, and feedback on earlier drafts of this article.

References

Karl De Leeuw. 2000. *Cryptology and statecraft in the Dutch Republic*. Thesis. Universiteit van Amsterdam.

Karl De Leeuw. 2018. De codes van Willem Six van Oterleek. Presentation at HistoCrypt. `https://www2.lingfil.uu.se/histocrypt2018/De-codes-van-Willen-Six-van-Oterleek.pptx`.

Mihály Héder and Beáta Megyesi. 2022. The DECODE database of historical ciphers and keys: Version 2. In *Proceedings of the 5th International Conference on Historical Cryptology, HistoCrypt 2022*, pages 111–114. Linköping University Electronic Press.

David Kahn. 1996. *The Codebreakers : The Story of Secret Writing. Revised edition*. Scribner, New York.

Beáta Megyesi, Nils Blomqvist, and Eva Pettersson. 2019. The DECODE database collection of historical ciphers and keys. In *Proceedings of the 2nd International Conference on Historical Cryptology, HistoCrypt 2019*, pages 69–78. Linköping University Electronic Press.

Beáta Megyesi, Bernhard Esslinger, Alicia Fornés, Nils Kopal, Benedek Láng, George Lasry, Karl de Leeuw, Eva Pettersson, Arno Wacker, and Michelle Waldispühl. 2020. Decryption of historical manuscripts: the DECRYPT project. *Cryptologia*, 44(6):545–559.

Vasily Mikhalev, Nils Kopal, Bernhard Esslinger, Michelle Waldispühl, Benedek Láng, and Béata Megyesi. 2023. What is the Code for the Code? Historical Cryptology Terminology. In *Proceedings of the 6th International Conference on Historical Cryptology, HistoCrypt 2023*, pages 130–138. Linköping University Electronic Press.

Appendix A. Overview of the messages

This appendix gives an overview of all the decoded messages with a short summary. The full (Dutch) decoding can be reproduced using the provided open source material in the GitLab repository. The numerical identifier of each message is derived from the filename of the original digital scan. This numbering scheme has been kept to preserve a consistent workflow during analysis and reporting. The messages are numbered between 4037 and 4074. The numbers 4061, 4062 and 4066 are not included, since they represent plaintext messages.

Msg.	Summary
4037	Expectation that the French Emperor will achieve his objectives at the forthcoming meeting; orders from the 7 July dispatch being followed.

4038 Reports widespread rumours that an Austrian Archduchess will marry Napoleon; assurances that Franco-Russian relations will remain unaffected.

4039 Russian emperor received news of the King's marriage favourably; hopes for reducing family complications and guarantee against continental wars; Russian cabinet wants to keep good relation with France; Russian nobility weary for more wars due to taxation burden.

4040 The alliance between the three empires is not yet formally negotiated; Russo-Turkish negotiations progressing with French mediation; territorial arrangements discussed for Wallachia, Moldavia, Bessarabia, and Serbia.

4041 Brief report on conversation with Count Romanoff regarding Dutch affairs. Specific notes on extreme caution and reserve.

4042 Confidential conversation with Count Romanoff confirms Russia, France, and Austria are cooperating to bring England to peace.

4043 Courier arrived from Moldavia; military operations suspended; Russia demanding financial compensation; Austria may receive territorial advantages.

4044 News of King Louis Bonaparte's abdication received via Russian legation courier; Ministerial communication with Count Romanoff will be avoided pending further instructions.

4045–4046 Dutch affairs viewed with sympathy by Russian government, during King's stay in Paris; Russia longs for stability and is committed to neutrality; discussion of financial reforms and administrative challenges.

4047 Reports of abdication of Swedish crown by reigning king; new crown prince arriving in Stockholm; Franco-Austrian discussions described as amicable.

4048 Important information received from French courier; speculation about imperial marriage negotiations; strict secrecy maintained among only four or five persons.

4049 Assessment that no formal diplomatic démarches have yet occurred; only the Ambassador's views being solicited.

4050–4051 Discussion of territorial redistributions in Germany; detailed analysis of Russia's internal financial crisis: currency devaluation, excessive paper money emissions, and hardship among lower ranks and military personnel.

4052 Emperor reportedly considering declaration to halt paper money emissions; financial difficulties persist requiring new taxation; Counts Romanoff and Kotschubey consulted on these matters.

4053 Turkish front stagnating; siege of Silistra abandoned due to supply shortages; orders dispatched to Prince Bagration to resume advance.

4054 Peace conditions between Austria and France detailed; cession of part of New Galicia to Russia; rumoured Polish territorial demands proven unfounded; Russian friendship valued in secret negotiations.

4055 Intelligence that Napoleon, upon concluding peace with Austria, intends to concentrate doubled military force against Spain.

4056 Report that the young Prince of Orange, studying at Oxford, is destined to marry Princess Charlotte, daughter of the Prince of Wales.

4057 Prince Czertorinsky returned to Saint Petersburg; his Austrian sympathies noted; Polish uprisings condemned; assessment of Russian forces in Galicia; commercial shipping news.

4058 Reports of Czertorinsky's audience with the Emperor were incorrect; Orange party spread false information; Swedish negotiations proceeding slowly.

4059 Report about suggestion to Alopeus in Stockholm to place a Holstein prince on Swedish throne in exchange for Finland; rejected as Russia will not return Finland.

4060 Urgent report: Emperor issued orders to Prince Gallitzin to attack Austrian forces in the Grand Duchy of Warsaw.

4063–4064 Formal treaty concluded between Napoleon and Alexander in Saxony; suspected offensive and defensive alliance; internal Russian discontent is not diminishing; Emperor isolated but retains sufficient power to enforce compliance.

4065–4067 Caulaincourt confirmed agreement between the two Emperors; Alexander expressed sympathy for Dutch affairs and friendship for the King; frontier negotiations postponed; potential sources of discord between allied powers prevented.

4068–4069 Secret details about the Saxen Conference; Relay of Caulaincourt intentions to restore European peace; France agrees Russia to retain Finland. Connection between two empires. Turkish acquisitions. Emperor Alexander recognition of King of Spain; Prussian peace conditions moderated at Alexander's insistence.

4070 Despite widespread discontent, no disturbances anticipated during the Emperor's absence from the capital.

4071 Rumours of exchanging Grand Duchy of Berg for Dutch territory discussed with Caulaincourt; Treaty of Paris (1806) guarantees invoked; hopes that influential persons at the conference will protect Dutch interests.

4072–4073 Note about the secret evacuation of Prussian territory; démarches made but with moderate instructions; author exercising caution; trusted local contact warned against excessive inquiry.

4074 Expression of Emperor Alexander's gratitude for the reception of the Prince as conveyed via the French Ambassador; reflects the consideration shown towards the Dutch King.

Combinatorial Wheels and Movable Alphabets: from Ramon Llull to Leon Battista Alberti

Benedek Láng
Eötvös Loránd University (ELTE),
Hungary
lang.benedek@gtk.elte.hu

Abstract

This article reconstructs an alternative pre-history of late medieval cryptography by situating letter-based cipher devices within a broader tradition of combinatorial wheels, volvelles, and alphabetic diagrams. Starting from Ramon Llull's *Ars Magna*, it analyses how rotating systems of letters functioned as engines of combination designed to generate knowledge, and traces how similar visual–mechanical principles reappeared in divinatory practices (onomancy, *sortes* literature, and divinatory volvelles), ritual magic, and mnemotechnics. The core cryptographic contribution lies in the discussion of alphabetic wheels that no longer encode divine attributes but human secrets, culminating in Leon Battista Alberti's polyalphabetic cipher disk and Giovanni Fontana's hybrid mnemonic–cipher machines. By comparing these devices structurally rather than doctrinally, the article argues that medieval cryptography emerged within a shared manuscript culture of movable alphabets, permutation, and controlled randomness. While direct lines of influence can rarely be demonstrated, the persistence of concentric letter wheels across divination, magic, mnemotechnics, and cryptology suggests a common visual and combinatorial grammar that shaped the earliest mechanical thinking about encryption.

1 Introduction

Circular diagrams were widespread members of the group of visual representation schemes in medieval manuscripts and early modern printed books. They served for classificatory, representational, and pedagogical purposes. They visualized, among others, the universe's structure, the earthy elements' relations, and the analogies between the macrocosm and microcosm. Wheels were also widely used in computus texts; they helped calculate the moveable feasts and other dates.

Within the large family of such circles and wheels, which has deserved serious scholarly attention (Wickersheimer, 1914; Evans, 1980; Friedman, 1985; Murdoch, 1984), this article concentrates on a particular sub-group. This sub-group contains movable concentric wheels on which changeable combinations are represented by letters, and by combining the letters, the user has access to secret knowledge, let it be knowledge about God or knowledge only accessible to a few selected persons. By tracing the history of such wheels, the article seeks to reconstruct the intellectual origins of Leon Battista Alberti's famous cipher disk.

2 Wheels and alphabets

One can undoubtedly trace back the ancestry of circular combinatorics beyond the Catalan philosopher Ramon Llull (ca. 1232-ca. 1315). Yet, his oeuvre was so decisive in this history that later authors rarely failed to note his contribution. He was a Christian mystic receiving a vision, after which he dedicated his missionary activity to convert Jews and Muslims by demonstrating Christian truths.

In his *Ars Magna* (and in many of his other works bearing somewhat similar titles), Llull constructed a sophisticated combinatorial wheel system to lead the user to know God and prove the reality of universal Christian truths. In this system, letters of the alphabet were placed on revolving wheels to denote God's attributes and a wide range of knowledge elements. These letters could be combined with each other and other data to solve problems in all knowledge fields. This was a universalist system.

Ramon Llull's importance could hardly be overestimated. It is well known that he became the target of Jonathan Swift's sarcasm. Swift's thinking "engine" that its inventor intends to enable anyone to "write books in philosophy, poetry, politics, laws, mathematics, and theology,

without the least assistance from genius or study" is a mockery of Llull's system. But this was undeserved. Llull's intellectual influence was so significant that only its 14th-century French history filled five hundred pages (Hillgarth, 1971). And 14th-century France was not the most crucial chapter in the history of Lullism. Instead, its golden age was the late medieval and early modern era when such authors were concerned with the *Ars Magna* as Nicolaus Cusanus, Pico della Mirandola, Agrippa von Nettesheim, Giordano Bruno, Juan de Herrera (the architect of Escorial), Athanasius Kircher, and even René Descartes and Gottfried Wilhelm Leibniz. Frances Yates claimed that "The European search for method . . . began with Ramon Lull" (Yates, 1982: 7), while others see him as a pioneer of computer science. The intellectual movement of encyclopedism exemplified by the Lullist author Johann Heinrich Alsted also deserves to be mentioned.

Applied Lullism had a great tradition in poetry and music. In the 17th century, Georg Philipp Harsdörrfer and Quirinius Kuhlmann experimented with permutational poem writing. In the 20th century, Raymond Queneau published his *Cent mille milliards de poèmes* (Queneau, 1961), which is nothing more than ten sonnets printed on ten consecutive pages, each consisting of fourteen lines. Once the reader cuts the pages along the lines and opens the book at random, each time they open the book, they will read a new combination of lines, giving birth to one of the 100,000,000,000,000 possible sonnets. Parallelly, Johann Philipp Kirnberger experimented with combinatoric composition using random combinations of musical elements in the eighteenth century.

The tradition of pseudo-Lullian alchemy should also be mentioned even though it was entirely fictitious, but the fact that Llull was chosen as an attributed author also shows his fame (Pereira, 1989). While Llull was a common inspiration, what his method really consisted of was a subject of debates and reconstructions. Beyond doubt, his *Ars Magna* was meant as a thinking machine, a debating tool to prove knowledge claims and generate new ideas. By combining religious and philosophical attributes selected from a fixed set of preliminary concepts and represented on rotating and concentrically arranged circles, the conclusions were meant to be automatically deducted. His A figure listed God's attributes (in Llull's word: dignities): Goodness, Greatness, Eternity, Power, Wisdom, Will, Virtue, Truth, and Glory, all of which he meant as commonly acceptable categories in each monotheistic religion (Figure 1). His T figure, another sophisticated circle, represented the so-called relative principles, including Majority, Minority, Equality, Concordance, etc. The 3rd figure, a series of rubrics arranged triangularly, listed 36 possible two-letter combinations of the dignities and the relative principles taken from the two previous figures. The system was flexible: the double letters could signify principles from both the first and the second figure, such as Goodness is great, Goodness is eternal, etc. The real thinking machine was the next figure, three movable concentric wheels, each representing nine letters of the alphabet, generating all the possible 3-digit combinations of these letters.

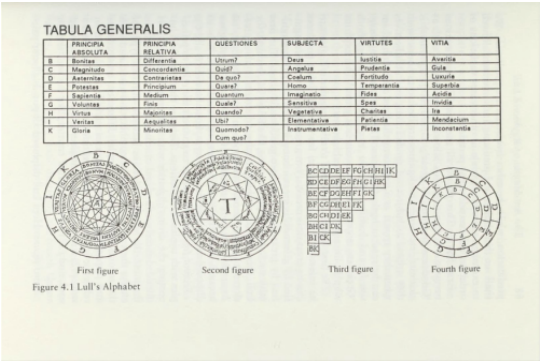


Figure 1. Llull's dignities and four figures

Another figure listed the same combinations in a tabular arrangement with a T inserted among the three letters, thus rendering BCD as BCTB, BCTC, and so on. Here T merely marks that the preceding letters belong to the first, and the following letters to the second figure. Altogether 1,680 possibilities appear here, many of which are, of course, repeated.

Once the user has a religious or philosophical doubt, they are to translate his question into the language of the Ars. Such a question is, for example, BDTB, meaning: *Utrum in aeternitate sit differentia*? Is there difference in eternity? Another question is BDTD. *Utrum inter divinam bonitatem sit contrarietas*? Is there contrast within divine Goodness?

While Llull was confident that his method could interpret and answer questions, his

followers were often uncertain about how to translate a scientific, logical, or theological question into the language of the *Ars* and how it should be answered. Do the combinations really generate new thoughts? Do they really provide answers? Or do they rather facilitate thinking, and the *Ars* is rather a mnemotechnic tool? Can it be rather used as a rhetorical aid? Agrippa, Bruno, and Alsted aimed to explain these issues in their lengthy commentaries to the *Ars*.

3 Sources of the *Ars Magna*

Before turning to the impact of the Lullian thought, it is worth reviewing shortly what can be known about its sources. That the Kabbalah might have played some role in the generation of the idea of combining alphabets, always sounded plausible. The golden age of this form of Jewish mysticism (including the Zohar's writing by Mose ben Sém Tóv and Abraham Abulafia's ecstatic Kaballah) took place close to Llull, both temporally and geographically. Frances Yates and Gershom Scholem argued that Llull's *Ars* could be traced back to the ecstatic Kabbalah of Abraham Abulafia and the Kabbalah of the Sefirot. There was even a work attributed to Lull entitled *De auditu cabbalistico*, which however turned out to be a forgery. It was written by a certain Pietro Mainardi (1456-1529), who tried to confirm the Renaissance philosopher Pico della Mirandola's theory on the relations of Lull and the Kabbalah.

The starting point of Moshe Idel, who found decisive evidence for the link between the two areas in 1988, was also a similar claim by Pico della Mirandola (Idel, 1988). However, he argued that it was not so much the Neoplatonic emanations of the Sepher Yetsira, but rather a particular type of Kabbalah, close to but not identical to Abraham Abulafia's method. This existed in the 13th century, and Pico perceived it as especially similar to the art of Ramon Lull. A close reading of Pico reveals that he may have thought of these combinations and revolutions of the alphabet when seeing similarities between the *Ars Magna* and the Kabbalah. Idel quotes a 13th-century commentary on the liturgy, an anonymous work which had not attracted the attention of Kabbalah scholars. This commentary contains a figure that particularly corresponds to Pico's reference to the "*revolutio alphabetorum*" since the concentric circles were intended to revolve to generate all possible combinations of the letters of the alphabet. Another figure, a triangular one, is strikingly close to Llull's third figure. And the ideas combined by the letters were indeed not very far from those of Llull, as the letter alef symbolizes Primeval Light, God, Lord, One, Truth–quite similarly to the references of Llull's letter A. Further research since then confirmed Idel's hypothesis that Pico must have had access to the Latin translation of this Cabbalistic work: the *Commentum Sefer Iesire* was part of his library, and he plausibly supposed that it could have been a crucial source for Llull (Idel, 2007: 280; Buzzetta, 2012; Mantovani, 2018).

4 Divinatory wheels

The following part of the article will review three interconnected fields, where circles and wheels play a central role. A common feature is that these fields–just like Llull's method–were concerned about godly privileges, but in a different way than the Catalan philosopher's work. And on at least one of them, Lullism exercised an implicit (never really proved and rarely pointed out) but significant impact.

The first is the general application of circular diagrams in a particular type of divinatory text, the so-called "Sphere of life and death". This was a simple practice of onomancy, name magic, where the user might learn about a given illness's outcome. As will be argued below, despite the seeming similarities, there is no accurate analogy with Llull here, as wheels are not capable of turning, there is no randomization procedure involved, not even letter (or other) combinations, and there is no alphabet applied. Last, the earliest Spheres of life and death considerably predate Lull's life.

The second example, however, is closer to the mechanism of the *Ars Magna.* We will analyze the mechanism of randomization devices in the so-called *Sortes* literature, another category of divination. This practice aims at reading signs that denote specific details of the knowledge only accessible to God. We have fixed combinations here, a specific randomization procedure, and numbers and letters are involved, just as in the *Ars Magna*.

Third and finally, we will see the volvelles, the movable disks, that were not only tools of the calendar (as most readers of medieval codices certainly know them) but also of divination. Here combinations of letters correspond to specific

parts of divine knowledge, which are randomized with dynamic turning methods using numbers and letters.

All three of these fall within the category of divination, a general term denoting the procedure of foretelling the future and discovering hidden knowledge through the interpretation of signs (Láng, 2008; Burnett, 1977; Burnett, 1998-99; Skeat, 1954; Savorelli, 1959). These signs might be written somewhere in the natural world (on various body zones such as in chiromancy), or they might be generated artificially by the users themselves, as in the so-called geomantic practices. Halfway between the natural and the artificially generated signs, there is a specific form of divination that operates with the numerical value of human names, called onomancy. A common–and easily recognizable–feature of divinatory texts is the ample use of diagrams, circular, tabular, quadrangular, and some of these combinatoric.

Strictly speaking, divination is not magic. Even though it shares the fate of being prohibited by theologians, the reasons for that were different. Acquiring foreknowledge of the future by human procedures–the theologians argued–is an abuse of God's privilege and contradicts the concepts of human free will and divine omnipotence. The users of divination, even when they are unaware of that, and even when this is not obvious in the seemingly non-demonic text, implicitly cooperate with demons. The repeated prohibitions and argumentations based on the demonic nature of this practice were undoubtedly motivated by the fact that palmistry (divination by the signs of the hand), geomancy (using randomly generated dots in the soil, and later on the parchment), scapulimancy (interpretation of the signs occurring in a sheep's shoulder blades), crystallomancy (gazing into crystal balls), catoptromancy (divination employing a mirror) and other similar texts remained popular among medieval readers.

4.1 The Sphere of Life and Death

The Sphere of Pythagoras, sometimes attributed to Apollonius, but most often named the Sphere of life and death is a much more frequent companion of medieval manuscripts than catalog entries suggest (Figure 2). As it is usually not longer than half a page and is often inserted between longer tracts, it does not always get the attention it deserves. This divination method takes the numerical value of the user's name as a starting point, and indicates whether the person inquiring about his fate will die or recover from his illness (Edge, 2015; Wickersheimer, 1914). Numbers are arranged in a circle (or less frequently in a rectangle), where the numerical values of the name of the client can be found. This value is then subjected to a series of mathematical operations, the result of which we should find either in the upper or in the lower hemisphere of the inner part of the circle. If it is in the upper part, the patient will survive; if in the lower, they are to die. Among our three examples, this diagrammatic device of onomancy shows the most minor traces of an analogy with the Lullian impact, as the circles are not turning, there are neither combinations nor permutations in the system, and no randomization takes place apart from the fact that the numerical value of names is substituted.

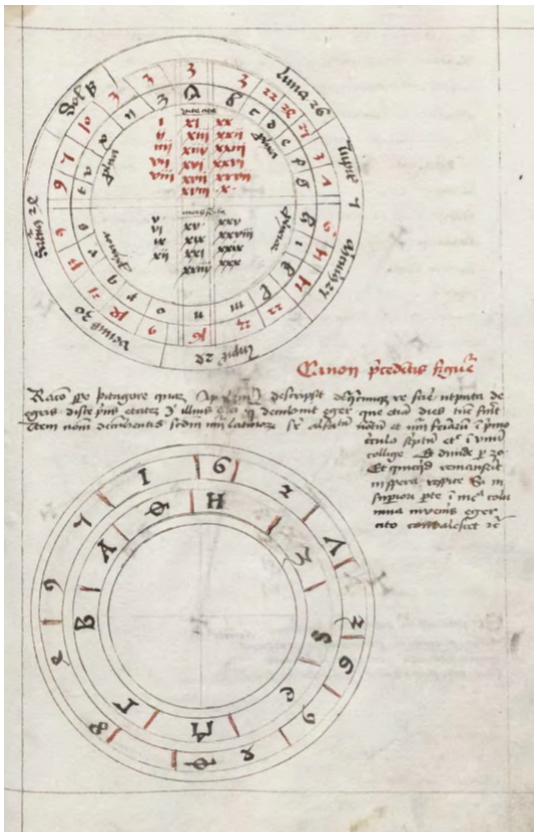


Figure 2. Biblioteka Jagiellonska (BJ), MS 793, f. 87r above: Sphere of life and death, below: a circular figure of unknown function

4.2 The *Sortes* literature

Sortes literature is a particular subclass of divination. The sorsbooks (in other words: books

of fate) are usually (but not always) shorter, and they are also usually (but not always) less sophisticated than other divinatory texts. Many of them bear a close resemblance to geomancy; that is, they take patterns of dots as a starting point, and then they go through a complicated procedure, a set of tables and circular diagrams, to arrive at an answer to the question of the user. Some texts are anonymous, and others are attributed to Albedatus, Socrates Basileus, Pythagoras, King Amalricus, and other real or imaginary authors (Burnett, 1996; Savorelli, 1959).

Sorsbooks–in contrast to most of the standard texts–are not supposed to be read linearly: readers (more precisely: users) go through them intermittently and in sections, and random methods define the direction of this non-linear reading. They are also visually organized, and thus they are easily recognized in the manuscripts by the often colorful tables and wheels they contain (Heiles 2018a, Láng 2008).

These late antique and early medieval texts, to which the 11th century added many translations from Arabic, were relatively early translated to the vernacular (e.g., German), which fact testifies that they were seen as helpful tools that the medieval people used on an everyday basis (Heiles, 2018b). While books of fates were usually listed in theologians' discourses on superstitions as forbidden and sanctioned by the Church (Johannes Hartlieb and Thomas Aquinas, among others, considered the act of reading them as a violation of the First Commandment), the average users, in contrast, may have rather perceived it as an innocent game that helped them orient in everyday life, and copied and used them extensively.

Books of fate give the impression of being combinatorial systems and able to provide a large set of answers to the questions of the user. Still, in reality they are, as T. C. Skeat, their early historian put it "systems comprising a fixed table of specific questions with a fixed number of alternative answers to each question" (Skeat, 1954). The primary means of the procedure were the full-page tables, but an important–though not necessary–accessory of books of fate were also the large, often multicolor, concentric diagrams that may remind the reader of Llull's figures. However, most of them were not composed of movable wheels but were merely circular diagrams used as static tables indicating the correspondences of numbers and geomantic figures. They play the same role in the procedure as the tables, the digits and letters appearing on them refer to divine knowledge (foreknowledge of the future, after all), but these wheels do not provide new permutational possibilities. Such a diagram can be found in Biblioteka Jagiellonska MS 793, f. 67r (shown and analyzed in Láng 2008: 86, 123-143), where it follows and complements a *Sortilegium Geomanticum* text (Figure 3).

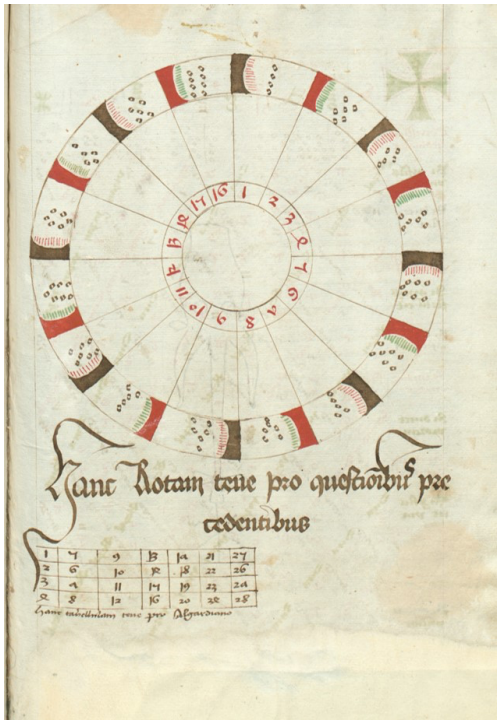


Figure 3. Biblioteka Jagiellonska (BJ), MS 793, f. 67r: Geomantic divination

4.3 Divinatory volvelles

Volvelles are best known as astronomical tools. They are concentric parchment (or paper) wheels constructed of several overlapping layers. With the help of a short rope or pin connecting them in the center, the middle one(s) was (were) able to turn independently around its central axis. Volvelles helped form new and new permutations of the numbers and letters written on the wheels. They were particularly useful for astronomical purposes and in computations of the calendar. As the volvelles could rotate, the reader did not need to turn the whole book as in the case of the traditional non-moveable figures in computus texts (Heiles, 2018a: 56; Crupi, 2016). The beginning of the history of

astronomical volvelles slightly predate Llull's circular figures, Matthew of Paris (1200-1259) already used them (Connolly, 2009; Gingerich, 1994; Crupi, 2019).

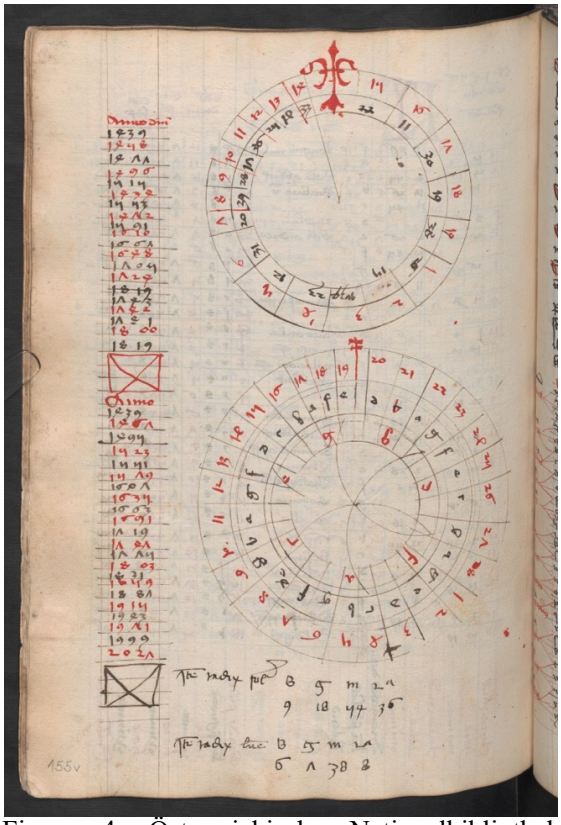


Figure 4. Österreichische Nationalbibliothek (ÖNB), 5327, f. 155v.

Such wheels with movable parts were equally prevalent in divinatory texts as dynamic randomization devices. In some cases, they are integral parts of a given sortes text. In an extensive multicolor divinatory handbook, the MS BJ 793 (Láng, 2008), for example, such a volvelle (BJ 793, f. 87v) belongs to the *Sortes regis Amalrici* text following it in the codex, and it provides the number necessary for the consultation of the *sortes*. Another example is in ÖNB 5327, f. 155v (Figure 4), where it is part of a sortes text, the "Prenostica Socratis Basile" (f. 155v–175r).[1] The oldest exemplar of this text can be found in Oxford, Bodleian Library, MS Ashmole 304 (Heiles personal consultation, see also Guardo, 2015; Iafrate, 2015). An interesting sortes volvelle can be found in Wolfenbüttel, Herzog August Bibliothek, Cod. 75.10 Aug. 2°, where a short alphabet is written along its arch.

[1] https://viewer.onb.ac.at/10027927/ (accessed: 2026.04.26).

Elizabeth Wade explicitly names the Lullain tradition as a possible inspiration source when presenting this source (Wade, 1998). However, like the earliest astronomical volvelles, the earliest divinatory volvelles predate Lull. The "Presnostica Socratis Basile" in the Arabic version first appeared before 1250 (MS. Bodleian Library, Ashmole 304) and in the Christian version at the end of the 13th century (British Library, MS Add. 15236) (Marco Heiles, personal communication).

Other volvelles serving for divinatory purposes can be found in several medieval manuscripts; they were not extremely widespread nor infrequent.[2]

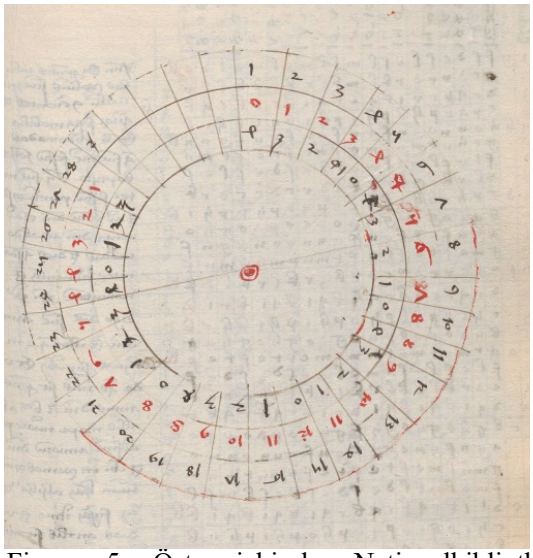


Figure 5. Österreichische Nationalbibliothek (ÖNB), 5327, f. 158

In many cases, the wheels–with or without moving accessories–are inserted between the *sortes* texts, but they are not strictly speaking parts of the text tradition. However, one may plausibly suppose that they were meant to complement the divinatory method, served either as a kind of illustration of the text or a mnemotechnic device serving to recall the method's details or as a tool of the randomization procedure. Even when the construction is not

[2] Berlin, Staatsbibl., mgf 642, f. 1r/v (pointer is missing); Heidelberg, Universitätsbibl., Cpg 552, front cover; London, British Libr., MS Add. 25435, front cover; München, Staatsbibl., Cgm 472; front cover (pointer missing), Olmütz/Olomouc, Heimatkundliches Museum, K-14905, front cover (Heiles, personal consultation).

movable, these drawings depict a real movable volvelle (Figure 5).

5 Ritual magic

Texts of ritual magic belong to an increasingly researched branch of medieval and early modern intellectual magic, the common feature of which is invocation of spirits of malign or benign nature. Users of this type of magic address the spiritual–demonic or angelic–powers through prayers and conjurations (Fanger, 1998; Fanger, 2012). Due to its apparent similarity with the Church's rituals, the expression "magic in a Christian framework" was also introduced as a helpful and rather precise description of ritual magic (Page, 2016). The best-known texts of ritual magic are the *Ars Notoria* (Boudet 2000, Veronese, 2007), the *Liber visionum* by John of Morigny (Fanger, 2015), the *Liber iuratus Honorii* (*Sworn Book of Honorius*) (Boudet, 2006), all of which survived in several copies and testify to a significant intellectual interest towards this kind of magic. Explicitly demonic types of handbooks, such as the Munich necromantic handbook published by Richard Kieckhefer (1997), much more rarely survived but still existed.

5.1 The book of the runes

Within this category, we find a text that only survived in four copies, the *Liber runarum* (Lucentini, 2001). It is specific because it mixes ritual magic elements, Scandinavian runes, and the talismans of astral magic. This fairly short text explains how to write the names of the spiritual forces of certain planets in a cryptic alphabet, the letters of which are called *runae*. The names constructed from these recognizable runes carry magical power. The text delves into the field of astral magic and gives detailed directions on how to inscribe the angelic names on specific metals and stones attributed to every planet. The runes corresponding to the planets and angels appear next to the text's main body. The reason why this text is included in the present typology of circular diagrams is that in one of the surviving copies held presently in the Biblioteca Apostolica Vaticana (BAV), the runes also appear on a *Rota runarum*, which stands separately, one hundred and fifty folios before the main text (Pal. lat. 1439 f. 348r-v: text; and 199r: rota, Figure 6). Apparently, in the unknown scribe's eyes, the wheel seemed to be an adequate representative form to represent the correspondences of letters and divine entities, the spirits.

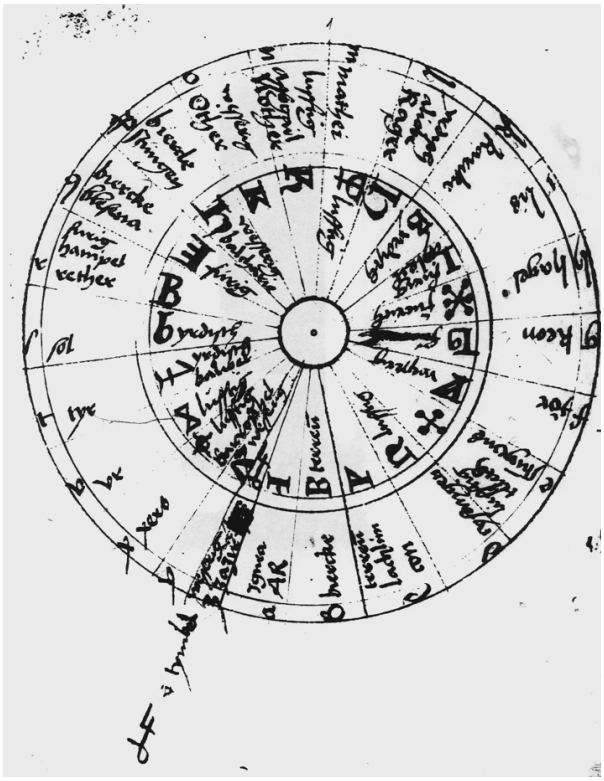


Figure 6. Biblioteca apostolica vaticana (BAV), MS Pal. lat 1439, f. 199r. *Rota runarum*

5.2 The Mirror of Floron

Another circular form representing spirit names and corresponding characters is the Mirror of Floron which survived both as a real object and in medieval textual depictions (Kieckhefer, 1997: 104-106; magical mirror, Mathematisch-Physicalischer Salon in Dresden, Figure 7). Here the–otherwise deliberately ambiguous–spirits can be rather called demons: texts on the Floron mirror appear in a demonic context. According to the description and the actual object, particular figures appear on a metal disc in a circular arrangement alongside names that seem to refer to the spirits. "Floron" is usually written in the inner circle of the object. The Munich necromantic handbook specifies that the mirror is to be prepared in the name of the spirit Floron according to detailed rituals involving suffumigations, clean clothes, and virgin boys. If appropriately prepared, an armed knight sitting on a horse will appear in the mirror, and then the master might ask him about the past, the present, and the future. Although the secret characters and the spirit names of the Dresden metal disc

are not identical to those given by the Munich manuscript, their number is the same (ten), and the arrangement of the elements and placement of the inscription "Floron" are analogous. This is one more example–besides the *Rota runarum*– where divine entities (demons, this time) are signified by letters or other characters, and represented in a circular form.

Figure 7. Magical mirror, Mathematisch-Physicalischer Salon in Dresden

6 The Cryptographic Application of Combinatoric Wheels

While they do not combine divine attributes or pieces of superhuman knowledge, the following two examples are nonetheless relevant for the present history, because of the combinatoric element, the circular arrangement of the letters, and the Lullian spirit are apparent in them. It is no surprise that historiography discussed a possible Lullian impact in the case of both of them. The following wheels combine letters that denote other letters in cryptography and pieces of earthly knowledge in mnemotechnics.

6.1 Leon Battista Alberti's cipher disk

Leon Battista Alberti (1404-1472) is not only famous as a Renaissance architect (Grafton, 2000) but also as the inventor of an early version of a cryptographic method that seemed unbreakable for centuries: the polyalphabetic cipher. In this encryption method the cipher alphabet (i.e., the alphabet, the letters of which substitute the letters of the readable text) can be changed. In other words, letters of the plaintext are not replaced by letters from one single code alphabet, but several ones in order to increase the safety of the ciphers.

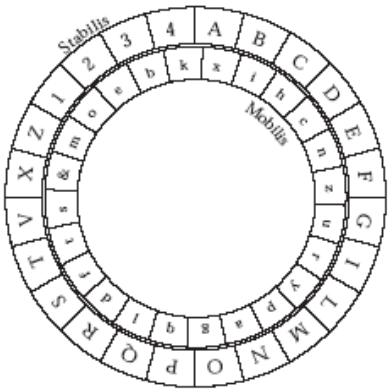


Figure 8. Alberti's moving circles, *De Componendis Cyfris*, 1446

When we start working with the plaintext with the help of Alberti's invention, we fix the inner ring and write down the lowercase letter that is opposite B. Then we start encrypting and do not rotate the ring until we wish to change code alphabets. However, if we do, we insert one of the four numerals 1, 2, 3, or 4 in the plaintext and the letter corresponding to these in the ciphertext. Then we rotate the rings so this lower-case letter is opposite B and continue the process until we change the code alphabet again. The method's strength lies in the fact that the code alphabet is constantly changed, and how it is changed has been long seen unpredictable.

With Alberti's method, one can shift the cipher alphabet after enciphering every letter of the plaintext, even though the Italian master designed his method so that alphabets are shifted less frequently, and not necessarily periodically. Periodicity was introduced by later authors, as a result of which, after a given number of such shifts, the user gets back to the already used

alphabets. This feature eventually led to the decryption of the polyalphabetic method, but only in the 19th century. For four hundred years, no one managed to break this method, true, it was rarely used for three centuries after its invention because it was thought to be too complicated and somewhat impractical. The practicability of such ciphers depends on the combinatoric "machine" that makes enciphering possible, the most famous of which was the enigma in World War II, and the first of which was Alberti's concentric circle-system (Alberti, 1906, 1997).

The movable disks represent the letters of the alphabet quite similarly to Llull's fourth figure. The functional similarity is so striking that David Kahn, historian of European cryptography, proposed the medieval Catalan mystic as a source of inspiration. "Although it cannot be proved that Lull's device inspired Alberti, there are grounds for suspecting that it did" (Kahn, 1980: 124). I find Kahn's suggestion convincing, but I would propose a more indirect way of impact: the Lullian circular diagrams may have influenced Alberti through the divinatory volvelles so widespread in the manuscript tradition (see above) and the mnemotechnic works of Giovanni Fontana so close to Alberti (see below). Alberti probably got acquainted with Fontana's work when they were simultaneously living in Northern Italy (Grafton, 2000: 84).

6.2 Giovanni Fontana's Mnemotechnics

Among the many exciting multitalented Renaissance engineers, Giovanni Fontana (c.1395?-1455) was one of the most exciting and multitalented ones, and compared to Leonardo da Vinci, and Leon Battista Alberti, he remained a relatively little-studied subject until recently (Battisti and Saccaro Battisti, 1984; Kranz, 2009). Fontana studied as a physician, and became a university professor and a dean, but left the academic career and served as a practicing doctor. His oeuvre forms an early chapter of many different histories: that of military engineering, that of mnemotechnics, and that of cryptology.

In his *Tractatus de instrumentis artis memorie,* mnemotechnics and cryptology are side-by-side. The reader is introduced into the functioning of memory devices suitable for storing and combining pieces of information and into enciphering methods convenient for secrecy practices. Bound together, under the title *Opera iuvenalia de rotis horologiis et mensuris* we find four Latin works on mechanical devices, clockworks, automates and rockets. The *Liber instrumentorum iconographicus,* the author's most famous book, describes complex military machinery in which illustrations play an increasingly important role. And *Secretum de thesauro experimentorum ymaginationis hominum* is entirely written on mnemotechnic devices. This last work constitutes a chapter in the surprisingly rich history of late medieval North Italian treatises on memory. Fontana was not without predecessors or companions; mnemotechnics was a major preoccupation in a period when the ability to store and recall information was crucial and when the automatization of these activities was only a futuristic dream. In several cases, Fontana's texts were written almost entirely in a simple substitution cipher using graphic symbols.

Both the *Secretum de thesauro* and the *Liber instrumentorum* are richly illustrated, text and image are closely related in them. Many of these illustrations are depictions of memory instruments that help memorize words. Some of them are mobile mnemotechnic machines. However, the present reader has difficulties imagining how they work as memorizing tools. It is much easier to imagine many of them as machines of encryption. This is particularly true for two figures. One is called *Speculum* (Kranz 2016, 98-99), which is composed of five concentric wheels meant to be movable by the author, on each of which the letters of the alphabet are listed, and that reminds us of Alberti's ciphering machine (Figure 9). And the second is the object called *Columna* (Kranz, 2016: 109), which looks like a famous cryptology device, the cryptex made of movable wheel containing the alphabet (Figure 10). And this is precisely the argument of the editor of the texts, Horst Kranz, (Kranz and Oberschelp, 2009). While Fontana does not refer to the possibility of using his machines for encryption, and apart from the fact that the book is written in cipher, there is very little reference to cryptology, these machines can be conveniently used for encryption, and for a most advanced type, the so-called polyalphabetic method, that was first documented a generation later, by Alberti, and then decades later by Johannes Trithemius (1462-1516) and Blaise Vigenère

(1523-1596) (Bayerl, 2024). Looking at the memory devices, the editors are certainly right that the early history of polyalphabetic cryptology still needs to be explored and that *ars memoriae* and the crypto-machines have a more intimate relationship than one would think.

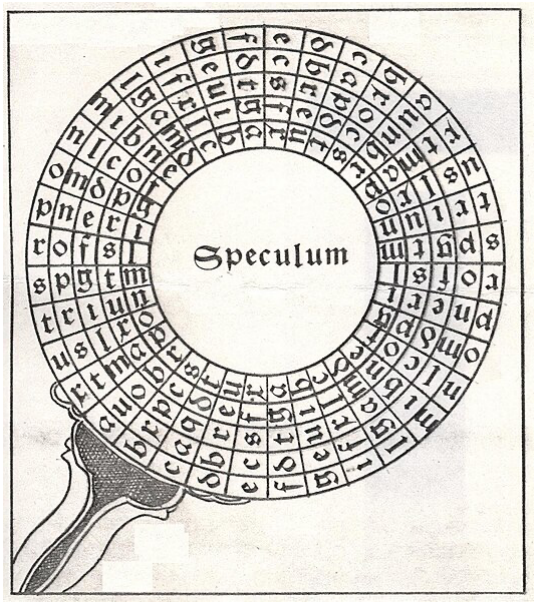


Figure 9. Giovanni Fontana, *Speculum*

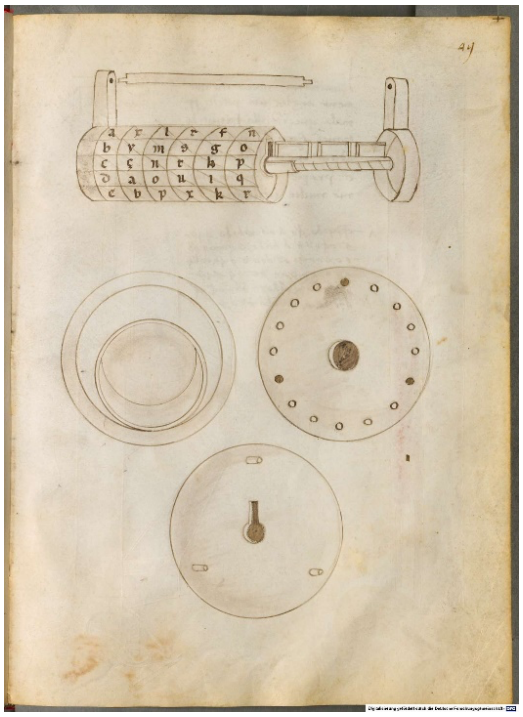


Figure 10. Giovanni Fontana, *Columna*

How far are these wheels analogous with the Lullian Ars? The disks are movable, and they contain letter combinations. However, what the letters refer to, are not divine but human secrets.

Conclusions

This article provided a catalog of medieval concentric wheel systems with the help of which letters of the alphabet or other characters could be combined in order to reveal divine or human secrets. Pieces of this confinable genre within the medieval codex illustrations might predate or follow the famous *Ars Magna* of Ramon Llull, some might have exercised an impact on his system, others were probably influenced by his heritage. Regardless of whether the letters signified divine traits, parts of divine knowledge, spiritual entities, human knowledge claims, or secret messages, the structural and visual similarities are striking. No tangible connection can be pointed out between Llull and the divinatory texts, nor can it be documented on textual grounds between Alberti's concentric wheels, Fontana's mnemotechnic devices and the Catalan philosopher's *Ars*. The links suggested by Elisabeth Wade, David Kahn and myself are just speculations.

In a particularly rich article, Gianfranco Crupi reconstructs the historical development of volvelles from the thirteenth to the seventeenth centuries and argues that, from Ramon Lull's combinatorial logic to their widespread use in astronomy, navigation, cryptography, and divination, volvelles functioned as material and cognitive devices that integrated manual manipulation with intellectual processes. While not asserting direct causal relationships, the article presents a tradition in which cross-fertilisation appears probable (Crupi 2019).

It can be argued that these figures belong to the same subclass of rotatable wheels in the Middle Ages, and also, that these visual representations might have influenced one another in the manuscript tradition. Llull was particularly influential through his own works and also through the Pseudo-Lullian alchemical corpus; thus it is highly possible that scribes of divinatory, magical, cryptologic, and mnemotechnic were influenced by his tradition, and that, in turn, they influenced each other. Astronomical, Kabbalistic, Lullian, divinatory and cryptographic volvelles might have mutually fertilized each other, while all the medieval authors lived in the middle of a much larger, in fact, cosmic volvelle: the universe.

Acknowledgement

This work has been supported by Riksbankens Jubileumsfond, grant M24-0028: Echoes of History: Analysis and Decipherment of Historical Writings (DESCRYPT).

References

Leon Battista Alberti. 1906. De Componendis Cyfris. In Aloys Meister, *Die Geheimschrift im Dienste der päpstlichen Kurie von ihren Anfängen bis zum Ende des 16. Jahrhunderts,* pages 125-141. Ferdinand Schoningh, Paderborn.

Leon Battista Alberti. 1997. *A Treatise on Ciphers.* Torino: Galimberti.

Eugenio Battisti and Giuseppa Saccaro Battisti, eds. 1984. *Le macchine cifrate di Giovanni Fontana*. Arcadia, Milan.

Corinne Bayerl. 2024. The Use of Volvelles in Two Early Modern Cryptography Manuals. In *Proceedings of the 7th International Conference on Historical Cryptology, HistoCrypt 2024*. pages 11-16. Linköping University Electronic Press.

Jean-Patrice Boudet. 2000. L'*Ars notoria* au Moyen Âge : une résurgence de la théurgie antique? In *La Magie: Actes du colloque internatonal de Montpellier 25-27 Mars 1999,* vol. 3, pages 173-191. Université Paul-Valéry, Montpellier.

Jean-Patrice Boudet. 2006. *Entre science et nigromance: Astrologie, divination et magie dans l'Occident médiéval (XII^{e}–XV^{e} siècle)*. Publications de la Sorbonne, Paris.

Charles Burnett. 1977. What Is the Experimentarius of Bernardus Silvestris? A Preliminary Survey of the Material. *AHDLMA,* 44: 62-108.

Charles Burnett. 1983. Scandinavian Runes in a Latin Magical Treatise. Postscript by M. Stoklund. *Speculum,* 58: 419-429.

Charles Burnett. 1996. *Magic and Divination in the Middle Ages: Texts and Techniques in the Islamic and Christian Worlds*. Variorum, Aldershot.

Charles Burnett. 1998-99. The *Sortes Regis Amalrici*: An Arabic Divinatory Work in the Latin Kingdom of Jerusalem? *Scripta Mediterranea,* 19-20: 229-237.

Flavia Buzzetta. 2012. Variazioni semantiche del Golem in alcune traduzioni latine del primo Rinascimento italiano. *Accademia, Revue de la société Marsile Ficin,* 14: 65-78.

Daniel K Connolly. 2009. *The Maps of Matthew Paris. Medieval Journeys through Space, Time and Liturgy*. The Boydell Press, Woodbridge.

Gianfranco Crupi. 2016. 'Mirabili visioni': from movable books to movable texts. *Italian Journal of Library and Information Science,* 7: 25-87.

Gianfranco Crupi. 2019. Volvelles of knowledge. Origin and development of an instrument of scientific imagination (13th-17th centuries). *JLIS.it*, 10(2): 1-27.

Joanne Theresa Edge. Nomen omen*: the 'Sphere of Life and Death' in England, c. 1200 – c. 1500*. PhD thesis, Royal Holloway, University of London.

Michael Evans. 1980. The Geometry of the Mind. *Architectural Association Quarterly,* 12: 32-55.

Claire Fanger, ed. 1998. *Conjuring Spirits: Texts and Traditions of Medieval Ritual Magic*. Pennsylvania State University Press, University Park.

Claire Fanger, ed. 2012. *Invoking Angels: Theurgic Ideas and Practices, Thirteenth to Sixteenth Centuries.* Pennsylvania State University Press, University Park.

Claire Fanger. 2015. *Rewriting Magic: An Exegesis of the Visionary Autobiography of a Fourteenth-Century French Monk*. Pennsylvania State University Press, University Park.

John B. Friedman. 1985. Les images mnémotechniques dans les manuscrits de l'époque gothique. In Bruno Roy and Paul Zumthor, eds., *Jeux de mémoire: aspects de la mnémotechnic médiévale*, pages 169-183. Librairie philosophique J. Vrin, Paris.

Owen Gingerich. 1994. Early astronomical Books with Moving Parts. In Bill Katz, ed., *A History of Book Illustration,* pages 288-296. Scarecrow Press, London.

Anthony Grafton. 2000. *Leon Battista Alberti: Master Builder of the Italian Renaissance.* Hill and Wang, New York.

Alberto Alonso Guardo, ed. 2015. *Prenostica Socratis Basilei: Étude, Édition Critique et Traduction*, Classiques Garnier, Paris.

Marco Heiles. 2018a. *Das Losbuch: Manuskriptologie einer Textsorte des 14. bis 16. Jahrhunderts.* Böhlau Verlag, Cologne.

Marco Heiles. 2018b. Deutschsprachige Losbücher. *Archivalia*, https://archivalia.hypotheses.org/74451.

J. N. Hillgarth. 1971. *Ramon Lull and Lullism in Fourteenth-Century, France*. Oxford University Press, Oxford.

Allegra Iafrate, ed. 2015. *Matthieu Paris, Le moine et le hazard: Bodleian Library, MS Ashmole 304.* Classiques Garnier, Paris.

Moshe Idel. 1988. Ramon Lull and Ecstatic Kabbalah: A Preliminary Observation. *Journal of the Warburg and Courtauld Institutes,* 51:170-174.

Moshe Idel. 2007. *La Cabbalà in Italia (1280-1510)*. Giuntina, Florence.

David Kahn. 1980. On the origin of polyalphabetic substitution. *Isis,* 71: 122-127.

Richard Kieckhefer. 1997. *Forbidden Rites: A Necromancer's Manual of the Fifteenth Century*. Sutton, Stroud.

Horst Kranz and W. Oberschelp eds., 2009. *Mechanisches Memorieren und Chiffrieren um 1430: Johannes Fontanas Tractatus de instrumentis artis memorie.* Franz Steiner Verlag, Stuttgart.

Horst Kranz, ed., 2011. *Johannes Fontana. Opera iuvenalia de rotis horologiis et mensuris/Jugendwerke über Räder, Uhren und Messungen*. Franz Steiner Verlag, Stuttgart.

Horst Kranz, ed., 2014. *Johannes Fontana. Liber instrumentorum iconographicus: Ein illustriertes Maschinenbuch*. Franz Steiner Verlag, Stuttgart.

Horst Kranz, ed., 2016. *Methoden des Erinnerns und Vergessens. Johannes Fontanas Secretum de thesauro experimentorum ymaginationis hominum.* Franz Steiner Verlag, Stuttgart.

Benedek Láng. 2008. *Unlocked Books, Manuscripts of Learned Magic in the Medieval Libraries of Central Europe*. Penn State University Press, University Park.

Paolo Lucentini, ed. 2001. *Liber runarum.* In Gerrit Bos, Charles Burnett, Thérèse Charmasson, Paul Kunitzsch, Fabrizio Lelli, and Paolo Lucentini, eds., *Hermes Trismegistus: Astrologica et divinatoria.* pages 401-449. Brepols, Turnhout.

Margherita Mantovani. 2018. Il Cristo Di Reuchlin. In Flavia Buzzetta, ed. *Cabbala I. Cahier de Accademia. Révue de la Société Marsil Ficin,* 11:43-60.

John Murdoch, ed. 1984. *Album of Science: Antiquity and the Middle Ages.* Charles Scribner's Sons, New York.

Sophie Page. 2013. *Magic in the Cloister: Pious Motives, Illicit Interests, and Occult Approaches to the Medieval Universe*. Penn State University Press, University Park.

Michalea Pereira. 1989. *The Alchemical Corpus Attributed to Raymond Lull*. Warburg Institute, London.

Raymond Queneau. 1961. *Cent mille milliards de poèmes.* Editions Gallimard, Paris.

Maria Brini Savorelli. 1959. Un manuale di geomanzia presentato da Bernardo Silvestre da Tours (XII secolo): l'Experimentarius. *Rivista Critica di Storia della Filosofia,* 14:282-342.

T. C. Skeat, 1954. An Early Mediaeval 'Book of Fate': The Sortes XII Patriarcharum, with a Note on 'Books of Fate' in General. *Mediaeval and Renaissance Studies,* 3:41-54.

Julien Véronèse. 2007. *L'*Ars notoria *au Moyen Âge. Introduction et édition critique*, SISMEL-Edizioni del Galluzzo, Florence.

Elisabeth I. Wade. 1998. A Fragmentary German Divination Device. Medieval Analogues and Pseudo-Lullian Tradition. In Claire Fanger, ed. *Conjuring Spirits: texts and traditions of medieval ritual magic,* pages 87–109. Pennsylvania State University Press, University Park.

Ernest Wickersheimer. 1914. Figures Médico-Astrologiques des IX^{e}, X^{e}, XI^{e} siècles. *Janus,* 19:1-21.

Paolo Yates. 1982. *Lull and Bruno. Collected Essays.* Vol. 1. Routledge, London.

What Counts as a Cipher?
The Evolving Role of Shakespearean Paratexts in Cryptographic History

Lyle Jennings Colombo
Tulane University
New Orleans, LA, USA
lcolombo@tulane.edu

Abstract

What counts as a cipher today has been shaped by nineteenth- and twentieth-century claims that Shakespearean texts contain ciphers proving Francis Bacon's authorship of the works. These claims formed the immediate institutional context for the founding of Riverbank Laboratories, where William and Elizebeth Friedman were engaged in evaluating the "Shakespearean Ciphers." Their systematic critique of Baconian methods played a formative role in establishing the methodological standards of modern cryptanalysis and ultimately contributed to the formation of the SIS and the NSA. This paper argues that the professionalization of cryptography prioritized algorithmic communication systems over historically attested practices of symbolic, diagrammatic, and concealment-based encryption. It proposes a historically grounded framework for reassessing Shakespearean paratexts under explicit methodological constraints and calls for a renewed expansion of cryptography's historical and methodological scope.

1 Introduction

For nearly 175 years, scholars and other writers have proposed that Shakespearean texts contain concealed meanings that establish the identity of the true author. These claims relied on unconstrained methods that failed to meet even minimal standards of falsifiability, rendering them incompatible with both literary and cryptographic analysis. The resulting dismissal of such claims hardened into a durable scholarly consensus that not only rejected the haphazard methods employed but also discouraged further inquiry into Shakespearean ciphers altogether.

This dismissal, which has left a 75-year gap in peer-reviewed research, is not based solely on the shortcomings of those early Baconian efforts. It also reflects a narrowing of what counts as a genuine encryption. As cryptography professionalized during the twentieth century, especially under military and intelligence imperatives, legitimate encryption came to be defined almost exclusively in terms of algorithmic, rule-based systems designed for efficient and secure transmission of information between correspondents (Kahn, 1967; Láng, 2018). Forms of secrecy that were symbolic, ritualized, diagrammatic, or non-communicative in intent were relegated to the margins, if not dismissed outright (Ellison, 2016; Láng, 2018). The history of the "Shakespearean ciphers" thus intersects with a larger history: the consolidation of modern cryptography itself and the disciplinary boundaries it has enforced.

It is within this contested disciplinary landscape that the Shakespearean paratexts deserve renewed attention. "Paratext" here refers to the dedicatory poems, epistles, and title-page material of early modern printed drama (Berger and Massai, 2014; Blair, 2021). Some Shakespearean paratexts exhibit multiple features historically associated with secret writing, such as acrostics, typographic anomalies, irregular syntax, non-variant misspellings, and conspicuous symbols. Given that these texts were produced within a dense early modern culture of secrecy, they merit serious cryptanalytic examination as structured sites of early modern concealment. These paratexts occupy a boundary zone between cipher, steganography, and symbolic access control, categories that early modern practitioners did not sharply distinguish. As such, they complicate the assumption that all meaningful encryption must conform to modern algorithmic models.

To understand the present state of research into Shakespearean paratexts, it is necessary to begin not with recent approaches, but with the episode that decisively shaped scholarly attitudes toward Shakespearean ciphers, namely, the nineteenth-

and early twentieth-century Baconian authorship movement and its repudiation.

First, however, a clarification of the status of the current work is needed. With few exceptions, research on Shakespearean paratextual grids and related concealment structures has not appeared in peer-reviewed literary or cryptographic journals, but has circulated instead through conference presentations, newsletter articles, and invited lectures. Within this context, the most consequential reframing of the problem was initiated by Alexander Waugh (2018), who applied early modern Cabalistic techniques and Hermetic symbology to the texts. While much subsequent work inspired by Waugh's approach has been pursued in the context of Shakespeare authorship debates, the methodological questions raised are independent of authorship attribution and call for assessment under the standards of historical scholarship and cryptographic analysis, rather than continuing to be pursued largely outside peer-reviewed academic venues.

2 The Baconian Moment and its Aftermath

The Baconian authorship movement began in the mid-nineteenth century and marked the first sustained attempt to formalize Shakespearean texts as cipher-bearing objects. Proponents such as Ignatius Donnelly and Elizabeth Wells Gallup argued that Francis Bacon had concealed proof of his authorship within the works using cryptographic techniques, most famously his own biliteral cipher.

The appeal of such claims was heightened by the failure to locate documentation of Shakespeare's literary career, despite sustained archival efforts. Such documentation is extant even for minor Elizabethan playwrights. Authorship doubts were also reinforced by the cryptic and often atypical features of the Shakespeare paratexts. For example, over 60% of early quarto editions either omit the author's name or hyphenate it as "Shake-speare," a practice commonly associated with pseudonyms in early modern publishing.

The 1609 title page of *Shake-Speare's Sonnets* is notable in that the space where an author's name would normally appear, between horizontal lines above the publisher, is left blank (Figure 1). Within a print culture that regularly employed pseudonyms and symbolic forms of attribution, such typographic features encouraged inquiry into the nature of the Shakespeare name.

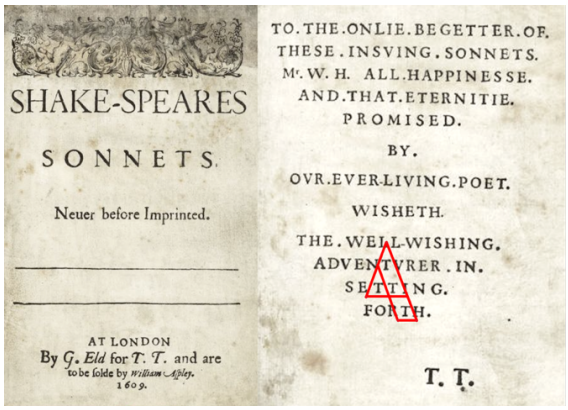

SHAKE-SPEARES
SONNETS
Neuer before Imprinted.
AT LONDON
By G. Eld for T. T. and are
to be solde by William Aspley.
1609.
TO.THE.ONLIE.BEGETTER.OF.
THESE.INSVING.SONNETS.
Mr. W. H. ALL.HAPPINESSE.
AND.THAT.ETERNITIE.
PROMISED.
BY.
OVR.EVER-LIVING.POET.
WISHETH.
THE.WELL-WISHING.
ADVENTVRER.IN.
SETTING.
FORTH.
T. T.

Figure 1. The *Sonnets'* title page and dedication. Source: Folger Shakespeare Library.

Donnelly's book *The Great Cryptogram* (1888) exemplifies the movement's ambitions and its weaknesses. He proposed that Shakespeare's texts contained elaborate multi-layered encodings, recoverable through arbitrary selection rules and flexible interpretive strategies. Gallup (1899) later refined these claims by asserting that typographic variations in early printed editions encoded Bacon's messages in binary form. These methods relied on unfalsifiable procedures: the analyst could continually adjust parameters, discard inconvenient results, and privilege coincidental alignments.

From a cryptographic perspective, the Baconian movement suffered from several critical flaws: (1) Lack of fixed rules: Decipherments depended on ad hoc decisions rather than invariant procedures; (2) Absence of controls: No systematic comparison with non-Shakespearean texts was performed; and (3) Confirmation bias: Solutions were known in advance and "found" retrospectively.

Yet despite these shortcomings, the Baconian episode was not trivial. It represented one of the earliest large-scale public engagements with cryptanalysis in Anglophone culture. Cipher breaking became a popular intellectual pursuit. It was dramatized in books, lectures, and newspapers. This cultural moment, dismissed today as methodologically undisciplined, nevertheless functioned as an incubation phase for modern cryptography, generating public fascination with secret writing and helping to precipitate later professionalization as the need to distinguish disciplined cryptanalysis from

imaginative pattern-seeking became increasingly urgent (Kahn, 1967).

3 The Professionalization of Cryptography

In 1913, industrialist and Baconian cipher enthusiast George Fabyan founded Riverbank Laboratories as a research enterprise devoted in large part to investigating claims of hidden ciphers in Shakespearean texts. He recruited William and Elizebeth Friedman, whose initial cryptographic training took place within this Baconian context. Through a systematic examination of Baconian cipher claims, the Friedmans decisively demonstrated that the interpretive freedom built into these systems allowed any desired conclusion to be extracted from the text.

This work culminated in their 1957 book *The Shakespearean Ciphers Examined*, in which they presented these cases to the public with exacting rigor and wry humor. For example, they expose the arbitrariness of unconstrained anagramming in their critique of Walter H. Begley's reading of the closing couplet of *The Tempest*, from which Begley extracts the following:

> Tempest of Francis Bacon, Lord Verulam,
> Do ye ne'er divulge me ye words.

Applying Begley's own method as described, the Friedmans, in turn, derive the following by rearranging the same letters:

> I wrote every line myself. Pursue no code.
> E. told me Bacon's a G. D. fraud.

In other words, a system that permits unrestricted rearrangement of letters can be made to affirm or deny any claim and is therefore unfalsifiable.

The Friedmans concluded that a genuine encryption must exhibit a clear encryption rule, a defined key, demonstrable resistance to chance, and an intention to transmit information covertly. Any system that failed to meet these criteria was not an encryption at all. This categorical move proved decisive. The "Shakespearean ciphers" were no longer treated as an open research problem but as a closed case.

From a historical standpoint, the Friedmans' intervention accomplished two things simultaneously: (1) They discredited Baconian cipher claims by demonstrating that proposed decipherment rules lacked statistical and logical rigor. (2) They articulated standards for cryptanalysis that would become foundational for modern cryptography: reproducibility, fixed rules, statistical validation, and independence from semantic foreknowledge.

The significance of the Friedmans' work extends well beyond Shakespeare studies. Riverbank Laboratories, where they worked on the Shakespeare ciphers, became an informal training ground for American codebreakers, and many techniques developed there were later institutionalized in the U.S. Army's Signal Intelligence Service (SIS). The SIS, under Friedman's leadership, eventually evolved into the National Security Agency. David Kahn (1967) links William Friedman's early confrontation with the Shakespeare cipher claims to his later insistence on strict methodological constraint.

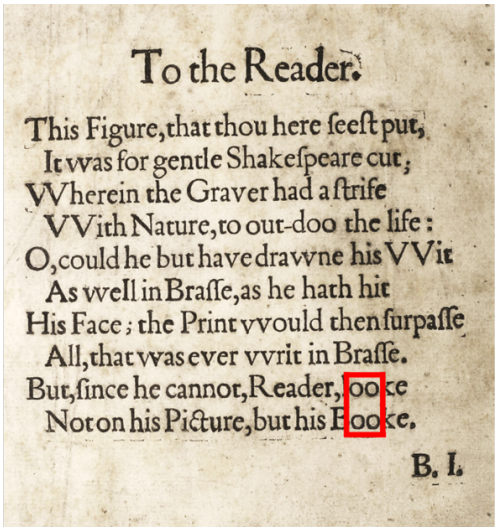
As cryptography aligned itself with military, diplomatic, and technological applications, literary and symbolic forms of secrecy were left behind. To suggest that Shakespeare's works contained ciphers was to invite association with discredited Baconian methods. Academic caution hardened into taboo, until even scholars sympathetic to authorship skepticism avoided cryptographic arguments, aware that such claims would likely undermine their credibility.

This cryptographic consensus, however, was not based on a comprehensive survey of all possible forms of concealment. It rested on an implicit assumption: that all meaningful encryptions are algorithmic, quantitative, and communicative in intent. Symbolic, ritualized, or concealment-based forms of secrecy were treated as either pre-cryptographic or non-cryptographic. Only recently has this assumption been questioned by scholars of early modern secrecy who study non-algorithmic forms of concealment.

4 Early Modern Cultures of Concealment

To assess forms of secrecy excluded by the modern cryptographic consensus, it is necessary to return to the early modern intellectual contexts in which the Shakespeare paratexts were produced. Among Shakespeare's contemporaries, Ben Jonson is particularly relevant as the generally accepted author of the First Folio poem "To the Reader," a paratext frequently examined for concealed structure (Figure 2). Any evaluation of symbolic features in that poem

should therefore take account of the intellectual environment in which Jonson was trained.



To the Reader.

This Figure, that thou here ſeeſt put,
It was for gentle Shakeſpeare cut;
Wherein the Graver had a ſtrife
With Nature, to out-doo the life:
O, could he but have drawne his Wit
As well in Braſſe, as he hath hit
His Face; the Print would then ſurpaſſe
All, that was ever writ in Braſſe.
But, ſince he cannot, Reader, Looke
Not on his Picture, but his Booke.

B. I.

Figure 2. Ben Jonson's Prefatory Poem, Image from Second Folio. Source: Miami University.

Jonson's education under William Camden at Westminster School is unusually well documented. Camden's scholarly method relied heavily on inscriptions, mottoes, coins, and fragmentary textual remains, requiring readers to reconstruct meaning from distributed, symbolic, and spatial evidence rather than from continuous narrative exposition (Camden, 1586; Watson, 1908; Parry, 1995; Woolf, 2003). Such practices cultivated habits of interpretation attuned to pattern, arrangement, and emblematic compression.

Camden's proximity to symbolic and mathematical modes of concealment is further evidenced by his correspondence with John Dee, mathematician and court astrologer to Elizabeth I. In one letter, Dee describes his use of the Greek letter *delta* as a symbolic figure combining astrological, mathematical, linguistic, and theological meanings (Reid, 2017). The delta functions as an emblem linking the material realm, which included Dee himself, to the spiritual realm of the Trinity. Dee regularly used the *delta* in place of his signature (Forshaw, 2017). Camden's correspondence with Dee places him within a network where symbolic compression, letter-based meaning, and mathematical structure were actively discussed.

Jonson is known to have employed such practices as a writer. His *Epigrams* include poems labeled as anagrams, demonstrating that he treated formal letterplay as an accepted mode of learned composition. Jonson is also explicit that literary meaning is not addressed to all readers but is reserved for those worthy readers who are capable of discerning it (1641).

This intellectual context is one reason why historians of cryptography increasingly question the assumption that genuine encryptions must be algorithmic, quantitative, and optimized for information transfer (Ellison, 2016; Láng, 2018). Katherine Ellison has shown that early modern texts described as cryptographic often presented families of techniques embedded in moral, theological, or courtly instruction, relying on creative problem-solving by the reader rather than fixed, universally applicable rules. Benedek Láng similarly urges an externalist approach attentive to why secrecy was practiced, demonstrating that many early modern ciphers were embedded in social ritual, patronage, and symbolic systems rather than designed solely for secure transmission. Sarah Lang has extended this critique into alchemical contexts, arguing that such ciphers were frequently designed to withhold comprehension through symbolic density rather than to enable straightforward decoding, functioning more as systems of concealment than communication (Lang, 2023).

Early modern cryptographic manuals reinforce this broader picture. Works such as Johannes Trithemius' *Steganographia* (1499) and Blaise de Vigenère's *Traicté des chiffres* (1586) do not insist on a strict separation between cryptography and other symbolic arts. Numerical correspondences, biblical allusions, and emblematic imagery frequently coexist with cipher alphabets. As Ellison has argued, this hybridity reflects an epistemic environment in which secrecy was oriented less toward secure transmission than toward regulating access to understanding.

5 Paratexts as Concealment Systems

The Shakespearean paratexts were produced within a culture that routinely embedded symbolic meaning in title pages, epigraphs, and dedicatory material. Dedications, epistles, and prefatory poems stood between reader and text as a site where symbolic density and interpretive play were cultivated.

Recent work on early modern paratexts further reinforces this point. As Thomas Berger and

Sonia Massai (2014) have shown, paratexts in English printed drama were not ancillary or ornamental; they were active sites of negotiation among printers, patrons, authors, and readers. Dedications, epigrams, and title pages could encode political allegiances, intellectual lineages, and claims to authority. From this perspective, Shakespearean paratexts are no longer anomalies inviting conspiratorial speculation, but typical paratextual spaces where layered meaning was expected.

Within this broader scholarly reorientation, Shakespearean concealment research is only now gaining limited representation in peer-reviewed literature. Recent work has begun to reexamine the Shakespearean paratexts using constrained formal and typographic analysis. Roger Stritmatter's work on "To the Reader" (2024) discusses a gematria-derived numerical pattern that encodes a secondary message dispersed within Jonson's poem.

A pivotal methodological shift occurred when Alexander Waugh proposed interpreting grid structures in Shakespearean paratexts within a Hermetic framework. In a lecture at Brunel University London, Waugh (2018) argued that these texts should be approached not as algorithmic encipherments but as symbolic constructions drawing on early modern Hermetic and alchemical traditions, including the use of emblematic verification markers such as the Triple Tau. He further demonstrated how gematria could be used to determine grid dimensions and how transposition of the texts into those grids allows secondary symbolic forms to emerge, carrying meanings not accessible through linear reading.

Much subsequent work engaging Waugh's reframing has emerged in authorship-attribution contexts that favor Edward de Vere, 17th Earl of Oxford. While such attributions have shaped the questions posed and the interpretations advanced, they do not exhaust the methodological implications of the reframing itself. Abstracted from questions of authorial identity, the central issue raised by the present study is whether the paratexts in question exhibit structured, non-random organization consistent with historically attested practices of symbolic or concealment-based secrecy. This work isolates this question from its polemical context and treats it as an independent problem in the history of cryptography.

Parallel research efforts, including computational analyses that test for non-random patterning, have further shifted the conversation. These studies do not claim that statistics can prove encryption, but they do indicate that certain features occur with frequencies inconsistent with chance, inviting further cryptographic, historical, and literary interpretation (Colombo and Chambers, under review).

Recent scholarship on the Shakespearean paratexts can be characterized by methodological camps, each defined by different assumptions about what constitutes cryptographic evidence.

One position, aligned with the Friedman tradition, requires encryption claims to satisfy modern cryptanalytic criteria such as explicit rules, demonstrable keys, and statistical resistance to chance. Shakespearean encryption studies are therefore viewed as compromised by symbolic interpretive flexibility.

A second, contextual-hermeneutic position treats Shakespearean paratexts as products of early modern cultures of secrecy, emphasizing spatial, emblematic, and redundant encoding and prioritizing historical intelligibility over algorithmic precision.

A third, emerging position seeks to integrate these perspectives through constraint-based hybrid methods. Here, quantitative analysis is used to establish boundaries within which interpretation may proceed. Statistical tests, control texts, and replication across multiple documents serve to identify non-random structure, while historical and literary context guides interpretation. This approach acknowledges the limitations of both rigid formal criteria and unconstrained hermeneutics.

This evaluation of the current state of research does not turn on the acceptance of any particular decipherment. Instead, it shows the need to situate recent work within a broader methodological and historical framework.

6 Figurative Acrostic Poetry as Precursor

This study proposes a historical framework in which ancient and medieval figurative acrostic poetry serves as a legitimate antecedent to the early modern concealment systems discussed here. Within this framework, the embedding of messages within grids and shapes is not

understood as novel, but as part of a long-standing learned practice.

The figurative acrostic poems of Publilius Optatianus (fourth century), Venantius Fortunatus (sixth century), and Rabanus Maurus (ninth century) were composed in grids over which symbolic figures were superimposed. These symbols highlight specific letters that, when read in sequence, form secondary poems or phrases, while simultaneously remaining part of the background poem (Brennan, 2019).

For example, in Rabanus Maurus's *De laudibus sanctae crucis*, the phrase *CRUX SALUS* is superimposed in the form of a cross onto the underlying poem (Figure 3).

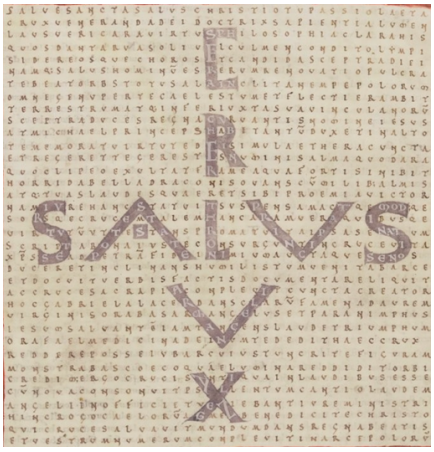


Figure 3. Rabanus Maurus, *De laudibus sanctae crucis*. Source: Bibliothèque nationale de France - gallica.bnf.fr, MS Latin 11685, f. 10v.

Each of these letters contains the name of an angelic order. The letters forming the names *Seraphin, Cherubin, Archangeli, and Angeli* are not anagrams, but occur in sequence (Figure 4). These letters contribute at the same time to the background poem.

Figure 4. Angelic orders spelled inside "Crux."

Below are two fourth-century figurative acrostic poems by Optatianus. One has highlighted letters that form a Chi Rho Christogram and the Latin name for "Iesus" (Figure 5a). The letters inside these colored shapes form a secondary poem, while remaining part of the primary poem.

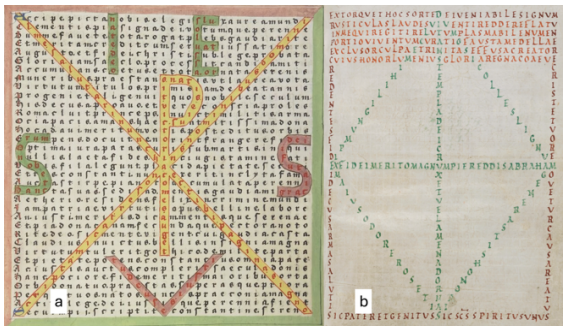


Figure 5. a) Optatianus, *Carmina*. 1468 printed edition. Source: Bibliothèque nationale de France - gallica.bnf.fr, MS Latin 8916, f. 68r; b) Fortunatus, *Carmen* 2.5. Source: St. Gallen, Stiftsbibliothek - e-codices.ch, Cod. Sang. 196.

Figure 5b shows an uncompleted figurative acrostic poem of Fortunatus, which reveals his process of construction.

These poems are not encrypted in the strict sense, because the shapes containing secondary meanings are visually indicated. They do, however, establish a conceptual grammar according to which meaning is distributed spatially across a grid, and symbolic forms function as reading keys. Trithemius, who studied and wrote about Rabanus Maurus, transmitted these ideas into the early modern period, and John Dee demonstrably read Trithemius (Ellison, 2016; Clucas, 2017).

7 John Dee and the Triple Tau

Dee's own manuscripts extend this tradition into Hermetic territory. In his notebooks he arranges letters into grids and superimposes geometric figures upon them, often crosses (Dee, 1584–85, 1585) (Figure 6a).

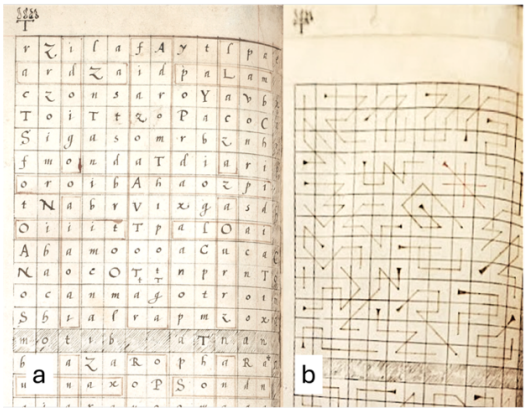


Figure 6. John Dee, Sloane. Source: British Library, MS 3193. a) f. 53v. b) f. 55v.

These shapes select specific letters for symbolic reading. The grids themselves are working structures in which position, column, and alignment carry meaning independent of the linear text, as illustrated in Figure 6b.

Dee filled notebook after notebook with such constructions, and the consistency of the practice across his manuscripts indicates a systematic preoccupation with superimposing symbolic forms onto textual grids as a method of revealing hidden order. These grid-based constructions also have a clear cryptographic analogue in the Cardano grille, an early modern form of encryption that selects meaningful letters from a gridded text through spatial masking.

Against this background, one emblem in particular has emerged as a focal point in recent Shakespeare encryption research: the Triple Tau. This symbol consists of three Tau crosses joined at their bases to form what looks like a capital *T* centered over the crossbar of an *H*. This construction conceals a fourth, upside-down Tau (Figure 7).

Figure 7. The Triple Tau and its hidden fourth *T*.

It has been proposed that, for Christian Hermetists such as Dee, the Triple Tau symbolized the fourfold Hermetic Trinity, which included the natural world as a hidden, fourth element (Waugh, 2018; cf. Dee, 1564, Theorem XX). This inclusion of nature in the divine is consistent with Dee's broader metaphysics, in which number and geometry mediate between the divine and material realms. The letter *T,* shaped like a cross, represents the elemental world; the shape of the circle (represented by the letter *O*) signifies the eternal, heavenly realm. Dee graphically enacts the joining of the eternal, solar circle with the elemental cross in *Monas Hieroglyphica* (1564).

In the context of Shakespearean paratexts, the relevance of the Triple Tau and its associated letters *T* and *O* lies in its function as a redundant marker of structure. In the *Sonnets'* dedication and "To the Reader," features interpretable as Triple Tau forms appear in the texts as typeset, through grouped *Ts* and *Os*. The Triple Tau appears again when the texts are arranged into specific, gematria-derived grids (Sec. 8). This recurrence provides a form of internal verification. For an audience familiar with Hermetic symbolism, such marks could function as signals of concealed structure, analogous to how alchemical emblems announce esoteric content. Importantly, the use of such a symbol implies that the concealed content was not intended for general readership.

8 Current Approaches

To distinguish recent work on Shakespearean concealment systems from early Baconian efforts, this section outlines how grid structures proposed for the *Sonnets'* dedication (1609) and the First Folio poem "To the Reader" (1623) are identified and independently verified.

Rather than rehearsing full decipherments, the focus here is on the procedural logic by which grid dimensions are constrained in advance and subsequently confirmed through symbolic redundancy, most notably by the appearance of the Triple Tau. The aim is to demonstrate that these analyses operate under explicit, testable constraints, rather than retrospective pattern selection.

The initial analytical task is to determine whether the text signals the presence of a grid-based mode of organization. In each case, this signal takes the form of a prominently placed representation of the Triple Tau. In the center of the *Sonnets'* dedication, the symbol appears as three *Ts* arranged in a triangular configuration, with a fourth *T* completing a diagonal of three *Ts* (Figure 1). In "To the Reader," the Triple Tau appears as four *Os* arranged at the end in a square (Figure 2). Within the Cabalistic framework employed by Dee and other Christian Hermetists, this substitution is not arbitrary. Four *Os* can be read as "4*O*," which visually signifies the number "40," which is a homophone for "four T." Similar phonetic substitutions are well attested in early modern Cabalistic practice and are explicitly discussed by Dee (1564; Whitby, 1988; Forshaw, 2017).

The presence of the Triple Tau serves as an indicator that the text may operate according to a structured, spatial logic susceptible to formal constraint. Once such a signal has been identified, the analytical task shifts to determining whether a grid can be constructed in

a non-arbitrary way and, if so, what parameters govern its dimensionality.

To this end, recent analyses employ a simple Cabalistic constraint documented in early modern sources: basic gematria, in which letters are assigned numerical values corresponding to their position in the alphabet (A = 1, B = 2, etc.). Gematria was believed to uncover divine truths put into place at the time of Creation. Dee explicitly endorses gematria, writing in *Monas Hieroglyphica* that "there are specific reasons for the shapes of letters, their positions or places within the order of the alphabet, their numerical value and many other things that must be considered" (Dee, 1564). The following gematric procedure is used for both texts: The grid's number of columns is determined by finding the letter that occurs in the text the same number of times as its gematric value.

For example, in the *Sonnets'* dedication, the letter *T* occurs nineteen times. Nineteen is also the gematric value of the letter *T,* because it is the nineteenth letter of the Latin alphabet (Figure 8).

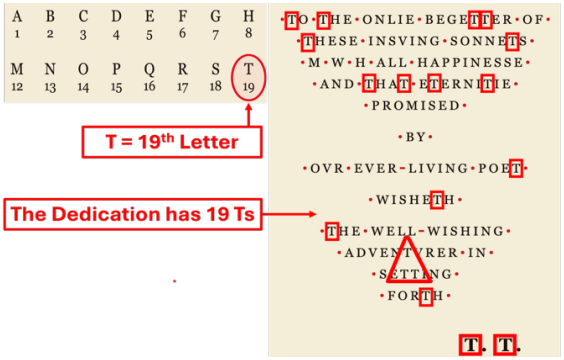


Figure 8. Gematria unlocks the *Sonnets'* grid.

No other letter in the dedication shares this property. When the text is transposed into a grid of nineteen columns across, further structural features emerge that are not present in adjacent grids (e.g., in grids of fewer or more columns).

The same procedure is used for "To the Reader." In that poem, the letter *N* appears thirteen times, corresponding to its position as the thirteenth letter of the Latin alphabet. Again, no other letter satisfies this condition. Transposition into a thirteen-column grid yields additional structural features absent from neighboring grid widths. The use of the same method in two distinct texts suggests a shared underlying practice.

What distinguishes these grid constructions from arbitrary pattern-seeking is the presence of an independent verification mechanism. In both texts, once the grid has been established using gematric constraint, a second Triple Tau appears within the grid itself, functioning as a symbolic confirmation that the correct dimension has been achieved. In both grids the Triple Tau appears in column 12, as three *Ts* aligned vertically, with a fourth *T* in column 13, to form the characteristic four-*T* structure of the symbol (Figure 9).

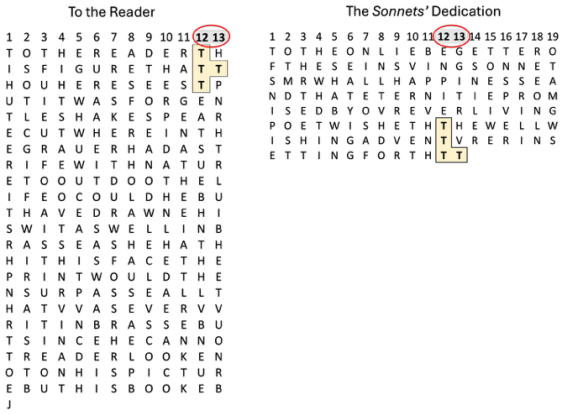


Figure 9. The Triple Tau in columns 12 and 13.

The number twelve represents the Triple Tau in that the symbol contains four *Ts,* each of which is made up of three lines (4 x 3 = 12). The relationship of four, three, and twelve is highly significant for Hermetic thinkers; Dee discusses it in his letter to Camden cited in section 4.

The recurrence of the same symbolic verification in the same column position across two separate texts reduces the likelihood that these patterns arose from chance. This is supported by quantitative analysis reported in Colombo and Chambers (under review). The Kruskal–Wallis test shows significant divergence of the proposed structural features from those found in adjacent grids ($p = 5.9e\text{-}05$ for the *Sonnets'* dedication; $p = 0.002$ for 'To the Reader'). Comparison against a corpus of contemporaneous paratexts yielded a p-value of 0.005 when both texts were analyzed together. Perplexity analysis of the *Sonnets'* dedication further suggests that its hidden message is more grammatically coherent than the original text, consistent with deliberate manipulation. Statistical analysis alone does not constitute proof of encryption; however, taken together, these results indicate that the structural features identified here are distinguishable from

chance at a level that warrants literary-historical interpretation.

The significance of the Triple Tau's appearance here is not merely structural. For readers trained in Dee's Hermetic framework, the symbol would have carried a specific meaning. It embodied the Hermetic claim that Creation constitutes a hidden fourth element of the Christian Trinity. Its appearance would have signaled that the text contains hidden knowledge deliberately withheld from general readership and reserved for initiated readers capable of recognizing the symbol. In this respect, the Triple Tau functions simultaneously as a verification mechanism and an access-control device.

A further point of methodological importance is that the Triple Tau does not function as the sole indicator of correctness. In both grids, it appears alongside additional features not presented here: repeated name signatures, directional cues, and symbolic Christograms that align with the same columnar and numeric logic. When the text is arranged into alternative grid widths, these features fragment or disappear. While individual features occasionally appear in a control group of contemporaneous paratexts, no control text replicates the combination or concentration of features.

The repetition of the Triple Tau is central to the interpretation of these structures as concealment systems. Grid selection is constrained in advance rather than determined by interpretive preference. The grid must satisfy multiple, independent constraints such as gematric frequency, symbolic verification, and structural coherence.

The significance of these examples lies not in the semantic content of any particular reconstructed phrase, but in the demonstration of procedural discipline. Grid dimensions are constrained before interpretation begins, and symbolic confirmation is embedded within the grid itself rather than supplied externally. The Triple Tau functions here not as an interpretive flourish but as a verification marker, analogous to a checksum in modern cryptographic systems.

In this respect, the Shakespearean paratexts illustrate how early modern concealment systems could combine numerical constraint with symbolic redundancy. They also exemplify why such systems fall outside the prevailing definition of cryptography, while remaining historically intelligible and analytically testable.

9 Methodological Implications

The study of non-algorithmic or concealment systems presents methodological challenges distinct from those posed by formal encryption systems, and these challenges help explain both the persistence of controversy and the reluctance of cryptographers to engage with such material.

First, concealment systems rarely provide a single decisive solution. They rely instead on redundancy, symbolic convergence, and contextual cues, none of which yields an unambiguous stopping point. This makes them resistant to the kind of closure that modern cryptanalysts expect.

Second, these systems often lack explicit keys. The "key" may consist of cultural knowledge, such as numerology, emblematic conventions, or theological symbolism, rather than a discrete object or rule. As a result, successful interpretation depends on historically informed constraint more than procedural decryption.

Third, concealment systems are vulnerable to accusations of arbitrariness. Without careful methodological discipline, symbolic readings can proliferate unchecked. This risk necessitates an unusually high standard of internal validation: repetition across texts, independent markers confirming structure, and the elimination of alternative grid arrangements.

Finally, for Hermetic thinkers, these encryptions enacted a real connection to the divine, formed by the very act of manipulating letters and numbers. Their purpose was not necessarily to convey information, but to preserve it and regulate access to it by presupposing particular forms of knowledge. This quality places these encryptions closer to steganography than to cipher in the strict modern sense. Such intentional ambiguity complicates both detection and evaluation.

One of the most significant methodological advances distinguishing contemporary efforts is the role assigned to quantitative analysis. Techniques such as frequency analysis, permutation testing, and nonparametric statistics (e.g., the Kruskal–Wallis test) have been used to evaluate whether proposed structural features,

such as symbolic clusters, repeated signatures, or grid-dependent alignments, occur at rates exceeding those found in adjacent grids or control texts. Similarly, language-model perplexity has been explored as a way to assess whether reconstructed phrases exhibit greater grammatical coherence than their surrounding text. If the configuration under examination is distinguishable from chance, then interpretation becomes historically interesting.

This approach aligns with recent work in alchemical cipher studies, where scholars such as Piorko, Lang, and Bean (2023) emphasize redundancy, internal validation, and contextual plausibility rather than single decisive readings. In this framework, statistics do not replace hermeneutics but discipline it.

Recognizing these methodological challenges does not invalidate the study of concealment systems; rather, it clarifies why they demand hybrid methods and why they were largely overlooked as the discipline of cryptography became professionalized.

10 Conclusion

The reemergence of research on the "Shakespearean ciphers" reflects a shift in disciplinary perspective rather than a reversal of earlier conclusions. The Baconian movement gave the topic cultural prominence without defensible methods. By contrast, the Friedmans' corrective intervention established modern cryptographic standards that tended to marginalize non-algorithmic secrecy practices. That the NSA traces its methodological lineage in part to the refutation of Shakespearean cipher claims is a reminder that disciplinary boundaries are historical artifacts, shaped by the problems a field was organized to solve.

The recent turn toward alchemical, Hermetic, and symbolic ciphers recognizes that cryptography, like any discipline, reflects the problems that historically defined its scope. Intelligence cryptography required algorithmic clarity; Renaissance secrecy often did not. Encryptions of this kind were not designed for transmission as much as preservation: concealing a hidden truth from the unworthy while enshrining it for those who could receive it. Accordingly, Shakespearean paratexts are understood as boundary objects whose study requires methods calibrated to a world in which cipher, steganography, and symbolic access control had not yet hardened into separate categories.

From a historiographical perspective, this reframing is significant. Shakespeare encryption research is separated from the realm of authorship advocacy and situated within the broader study of how secrecy functioned in Renaissance print culture. Whether particular interpretations of proposed concealment structures persuade future scholars is ultimately secondary to recognizing the question itself as methodologically legitimate.

A key contribution of this framework is the identification of ancient and medieval figurative acrostic poetry, namely the gridded, shape-bearing compositions of Optatianus, Fortunatus, and Rabanus Maurus, as a legitimate historical antecedent to early modern concealment systems. This lineage situates Shakespearean paratext research within an established textual tradition.

Ultimately, this study contends that early modern cultures of secrecy encompassed symbolic, diagrammatic, and concealment-based practices not captured by modern cipher definitions, and that Shakespearean paratexts exhibit features historically associated with such practices. Such features merit analysis through a combination of literary-historical contextualization and quantitative testing. The methodological legitimacy of such analyses rests on the use of explicit constraints, internal redundancy, and independent verification. Further research is needed to determine whether the statistical results reported here prove robust against more extensive control testing.

A reevaluation of Shakespearean paratexts exposes the limits of algorithm-centered definitions while advancing a historically grounded methodological pluralism. The value of this work lies largely in restoring a legitimate topic to the realm of scholarly inquiry. Doing so not only refines our notion of how secrecy operated in early modern print culture but also offers a principled framework for expanding the discipline of cryptography to include non-algorithmic models. It demonstrates the need for an approach that recognizes concealment systems as historically real, culturally meaningful, and methodologically intelligible when the analytical framework is designed to match the object of study.

References

Thomas L. Berger and Sonia Massai, editors. 2014. *Paratexts in English Printed Drama to 1642*. Cambridge University Press, Cambridge.

Ann Blair. 2021. *L'entour du texte: la publication du livre savant à la Renaissance*. Bibliothèque nationale de France, Paris.

Brian Brennan. 2019. Weaving with Words: Venantius Fortunatus's Figurative Acrostics on the Holy Cross. *Traditio*, 74:27–53. https://doi.org/10.1017/tdo.2019.13.

William Camden. 1586. *Britannia*. London.

Stephen Clucas. 2017. Introduction: John Dee, Alchemy, and Print Culture. *Ambix*, 64(2):107–114. https://www.tandfonline.com/doi/full/10.1080/00026980.2017.1356632.

Lyle Colombo and Paul Chambers. Under review. Hermetic Secrecy and Figurative Acrostics in Shakespearean Paratexts.

John Dee. 1564. *Monas Hieroglyphica*. Antwerp.

John Dee. 1584–85. *Claves Angelicae.* British Library, Sloane MS 3191, f. 55v.

John Dee. 1585. *Liber Scientiae Auxilii et Victoriae Terrestris*. British Library, Sloane MS 3191, f. 53v.

Ignatius Donnelly. 1888. *The Great Cryptogram: Francis Bacon's Cipher in the So-Called Shakespeare Plays*. R. S. Peale, Chicago.

Katherine Ellison. 2016. *A Cultural History of Early Modern English Cryptography Manuals*. Routledge, London.

Peter J. Forshaw. 2017. The Hermetic Frontispiece: Contextualising John Dee's Hieroglyphic Monad. *Ambix*, 64(2):115–139. https://doi.org/10.1080/00026980.2017.1353247.

Venantius Fortunatus. N.d. *Carmen* 2.5. St. Gallen, Stiftsbibliothek, Cod. Sang. 196, 39.

William Frederick Friedman and Elizebeth Smith Friedman. 1957. *The Shakespearean Ciphers Examined*. Cambridge University Press, Cambridge.

Elizabeth Wells Gallup. 1899. *The Bi-literal Cypher of Sir Francis Bacon Discovered in His Works and Deciphered*. Gay and Bird, London.

Ben Jonson. 1623. To the Reader. In *Mr. VVilliam Shakespeares Comedies, Histories, & Tragedies*. Isaac Iaggard and Ed. Blount, London. [First Folio.] Bodleian Library, Oxford, Arch. G c.7.

Ben Jonson. 1632. To the Reader. In *Mr. William Shakespeares Comedies, Histories, & Tragedies.* Thomas Cotes for Robert Allot, London. [Second Folio. Lord Leigh copy, with bookplates of Lord Leigh and Frederick S. Peck.] Walter Havighurst Special Collections Library, Miami University, Oxford, Ohio.

Ben Jonson. 1641. *Timber: Or Discoveries Made upon Men and Matter*. London.

David Kahn. 1967. *The Codebreakers: The Comprehensive History of Secret Communication*. Scribner, New York.

Benedek Láng. 2018. *Real Life Cryptology: Ciphers and Secrets in Early Modern Hungary*. Amsterdam University Press, Amsterdam.

Sarah Lang. 2023. Situating Ciphers Among Alchemical Techniques of Secrecy. In *Proceedings of the 6th International Conference on Historical Cryptology, HistoCrypt 2023*, pages 93-104. Linköping Electronic Conference Proceedings.

Rabanus Maurus. Ca. 1100–1115. *De laudibus sanctae crucis*. Bibliothèque nationale de France, MS Latin 11685, f. 10v.

Publilius Optatianus. 1468. *Carmina*. Bibliothèque nationale de France, MS Latin 8916, f. 68r.

Graham Parry. 1995. *The Trophies of Time: English Antiquarians of the Seventeenth Century*. Oxford University Press, Oxford.

Megan Piorko, Sarah Lang, and Richard Bean. 2023. Deciphering the *Hermeticae Philosophiae Medulla*: Textual Cultures of Alchemical Secrecy. *Ambix*, 70:150–183.

https://www.tandfonline.com/doi/full/10.1080/00026980.2023.2201744.

Rachel Reid. 2017. An Interpretation of John Dee's 'delta' from His Letter to William Camden. *Notes and Queries*, 64(2):247–250.

Roger Stritmatter. 2024. 'Ver had His Wit.' *South Atlantic Review*, 89(4):118–154. https://southatlanticmla.org/wp-content/uploads/2025/03/TOCSAR_89.4.pdf.

Johannes Trithemius. 1499/1606. *Steganographia*. Frankfurt.

Blaise de Vigenère. 1586. *Traicté des chiffres, ou secrètes manières d'escrire*. Abel L'Angelier, Paris.

Foster Watson. 1908. *The English Grammar Schools to 1660: Their Curriculum and Practice*. Cambridge University Press, Cambridge.

Alexander Waugh. 2018. Lecture on Hermetic Symbolism in Shakespearean Paratexts. Brunel University London. https://www.youtube.com/watch?v=XGn6eJkQlig.

Christopher L. Whitby. 1988. *John Dee's Actions with Spirits: 22 December 1581 to 23 May 1583*. Routledge, London.

Daniel R. Woolf. 2003. *The Social Circulation of the Past: English Historical Culture, 1500–1730*. Oxford University Press, Oxford.

"Was Early Modern Shorthand Cipher? Some Examples from Late Stuart England"

Andrea McKenzie
Department of History
University of Victoria, Canada
mckenzie@uvic.ca

Abstract

This paper addresses the question of whether early modern shorthand, a scribal technology first widely used in seventeenth-century England, qualifies as "cipher". In addition to the famous shorthand diary of Samuel Pepys (1633-1703), it will examine the previously undeciphered shorthand of several other late Stuart figures, with a particular focus on the lawyer and Member of Parliament Sir George Treby (c.1643-1700). Just as the authors of stenographic manuals touted shorthand as "secret writing", writers like Pepys and Treby clearly employed strategies to make their shorthand (or parts of it) difficult either to decipher or detect. Early modern shorthand can pose significant challenges for scholars, especially in cases where the system used is unknown; as with contemporary ciphertext, cracking these sources often requires painstaking contextual analysis currently beyond the powers of artificial intelligence.

1 Introduction

In July 1682 the Cambridge fellow Dr Nathaniel Vincent wrote the naval official Samuel Pepys to offer him and his royal patron, the duke of York, the right of first refusal of his new invention, a kind of secret writing he called "Cryptocovianicon" or "Monocrypticon". This was "a way of writing which can never be deciphered" by anyone but the person to whom the writer had furnished special "directions", which would however "not discover its way of writing"; after being read, its "characters vanish…so that no letter written by it can ever be a witness against its author". While Pepys responded with some commonsense objections (if the characters faded so quickly, how could copies be taken? And if the addressee were able to read the message, would he not become "master" of the secret and hence "impart it to more"?), he was nonetheless sufficiently intrigued both to correspond further about the subject with Vincent and meet him in person. Pepys's interest fizzled as his suspicions were presumably confirmed: the invention was too good to be true (Smith, 1841).

As Vincent perhaps knew, Pepys had a special interest in cryptography. The diary for which Pepys would become famous, written from 1660 to 1669, was almost entirely in shorthand – a scribal technology that had first originated in practical form in England in the early seventeenth century and which would remain an exclusively Anglophone practice until the eighteenth century (Henderson, 2008; Gardey, 2010). As recent scholarship has demonstrated, Pepys carefully curated his archive, destroying many of his own papers (and almost all his wife's) that would reflect badly upon him. He took careful measures to keep his shorthand diary from prying eyes, increasing the safeguards as time went on by writing sensitive passages – especially those relating to his sexual exploitation of and violence against female servants and other women and girls – in a polyglot of French and other languages and the occasional insertion of "dummy letters". At the time of his correspondence with Vincent, Pepys was likely debating whether he should destroy the diary, which he seems initially to have undertaken for his own private use. He ultimately decided to leave it with the other contents of his library for posterity, after taking elaborate precautions to ensure that the shorthand would eventually be deciphered, but only under certain conditions and by likeminded – and, hence, sympathetic – readers (Loveman, 2022 and 2025).

2 Scholarly Perspectives

There has been a persistent tendency on the part of both popular and academic writers to characterize Pepys' shorthand as cipher, a secret or "mysterious code" (Loveman, 2025; Akkerman and Langman, 2025) rather than a scribal technology conforming to basic rules that could be accessed – provided the system were identified and a manual could be located. It was long believed that the Cambridge scholar John Smith, who first deciphered Pepys's shorthand, did so solely by comparing Pepys's shorthand notes of Charles II's escape after the battle of Worcester (1651) with the printed account, thinking it was a cipher of the diarist's own invention and belatedly realizing, after his massive work of transcription was nearly complete, that the "key" had been in Pepys's library all along: Thomas Shelton's shorthand manual *Tachygraphy* (Matthews, 1934). However, as Kate Loveman has established, Smith's familiarity with certain arbitrary symbols leaves little doubt that he had identified the system as Shelton's but promoted the impression that Pepys's diary was written in "code" both to enhance his own prestige and to protect his intellectual labour (Loveman, 2025). This misconception has also persisted because of the tendency to underrate shorthand as a skill: after the late nineteenth-century feminization of the secretarial sector, the once lively antiquarian and scholarly interest in the history of stenography dried up rapidly (Gardey, 2001 and 2024). In other words, interest in Pepys's diary has reflected the ways in which "cipher" has been coded as male and shorthand as female, less worthy of study.

While in the last few years there has been a welcome resurgence of scholarly interest in shorthand, there is still a lot of work to be done. As one important recent work has emphasized: "Most texts written in, or with, shorthand are as inscrutable to us today as Egyptian hieroglyphs were in early modern Europe" (McCay, 2021). Since William Matthews, the shorthand expert and co-editor of the standard modern transcription of Pepys' diary, relatively few modern scholars have engaged in this tedious and difficult work: important exceptions include Frances Henderson and Timothy Underhill for seventeenth- and eighteenth-century England, and Linford Fisher for colonial America (Latham and Williams, 1971; Henderson, 2001 and 2005; Goldie, 2007; Underhill, 2013 and 2018; Fisher et al., 2014). Rather than focus on transcription, many scholars prefer to study shorthand as a genre or practice, or even as what James Dougal Fleming has described as an early form of the "infosphere". Fleming has characterized Pepys's shorthand as "legible, in principle, by anyone who cared to learn Shelton", concluding that "shorthand looks like cipher only until you have learned it" (Fleming, 2024). Yet this underestimates the difficulty not only of decoding Pepys's diary (a vast collaborative effort taking many years) but even other early modern shorthand, which while seldom as elaborately "encrypted" as Pepys's was, nonetheless tends to be ambiguous and personalized, and often explicitly aimed at thwarting detection. Certainly, the early modern purveyors of shorthand touted not only its brevity and speed, but also its "secrecy" (Rich, 1646). Pepys and other seventeenth-century writers, like the dissenter and court critic Roger Morrice, were evidently confident that their shorthand would have been impenetrable to contemporaries (Matthews, 1934; Goldie, 2007).

The expert on early modern English cryptology Katherine Ellison has defined "code" as the straightforward substitution of words or names by other words, symbols or numbers; "cipher", however, is more complex, with a "cooperative" and "relational grammar", necessitating "problem solving" and "contextual analysis" that computer analysis or even (failing significant human input) AI still cannot provide (Ellison, 2017 and 2022;

Ellison and Kim, 2017). Recent cryptographic breakthroughs, such as the decryption of Mary Queen of Scots' letters, written with homophonic substitution cipher, have been facilitated both by new technologies in "computerized cryptanalysis" and more traditional manual codebreaking techniques but also by intensive "linguistic and contextual analysis" and recourse to more conventional historical studies (Lasry, Biermann and Tomokiyo, 2023; Bossy, 1991 and 2001). Thus, deep and specialised scholarship remains indispensable in terms of making inferences about gaps or allusions that would have been much easier for contemporary insiders than for modern readers to understand (Desenclos and Lasry, 2025). Contextual analysis is no less critical to deciphering shorthand, not to mention identifying the system with which it was writing.

While Ellison has stopped short of defining shorthand as cipher, she and other scholars of early modern scribal culture, such as James Daybell, have seen it as, potentially, a species of "secret communication" (Ellison, 2022; Daybell, 2012). If shorthand fails to qualify as cipher because its "key" (the shorthand manual) is in the public domain, it should be noted that scholars of early modern secret writing have established that most "real life cryptography" was a far cry from the sophisticated "polyalphabetic" systems described by the pioneering scholar of codebreaking David Kahn (Láng, 2018; Kahn, 1980). Most routine diplomatic cipher consisted of simple homophonic systems in which letters, syllables and words corresponded to letters in ascending alphabetical order, making it intuitive and convenient to use, but also "very easy to break even on a trial-and-error basis", even apart from the risk of the key being intercepted (Láng, 2014). It was common for the same key to be used for months or even years on end and, even when changed, to maintain the same rules with minor variations; safeguards like adding "nullities" were often dispensed with altogether (Akkerman, 2011 and 2016; Marshall, 1994; Láng, 2018). In practice, then, "most ciphers were "rudimentary, intended merely to delay decryption" (Daybell, 2012). And while most shorthand could not have resisted a truly determined assault, it could similarly delay or deter closer examination, especially if the content appeared to be pedestrian – as in fact much of it was.

Indeed, one major bar to the decryption of early modern shorthand documents is not just that the work is tedious and time-consuming, but that the text, once transcribed, is often frustratingly prosaic (Boeddeker and McCay, 2024; Underhill, 2024).[1] The transcriber of the letter book of the seventeenth-century merchant Thomas Hill acknowledged herself to be "disappointed" that the letters had "failed to produce more interesting information" (Palmer, 2008). Many shorthand writers used this scribal technology primarily to save time and space to copy letters, record sermons, transcribe speeches or proceedings. Even those who wrote more personal communications – journals, notes, draft letters or even shorthand messages to others – tended to be circumspect, confining themselves to vague and cryptic references and evidently reserving the most sensitive information and confidences for face-to-face interactions. However, as the examples that follow suggest, the determined (and patient) researcher can, after long sifting through muck, sometimes retrieve a few gold nuggets.

3 Shorthand in Practice

The remainder of this paper will briefly discuss some examples of the shorthand I have encountered and deciphered in my own research, focussing primarily on the

[1] Interestingly, Benedek Láng (2018) has made similar observations about early modern cipher.

lawyer and MP Sir George Treby (c.1643-1700), but also referencing the previously undeciphered shorthand of two other contemporaries: the Anglican bishop William Lloyd (1627-1717) and the Anglo-Irish statemen Arthur Annesley, earl of Anglesey (1614-86), who had, like Treby, trained as a barrister. Recent scholarship has established that early modern shorthand, like cipher more broadly, was not the monopoly of elite, or even male writers (Láng, 2014; Daybell, 2012; McCay, 2021; Underhill, 2024). Nonetheless it was doubtless more often practiced by, and certainly most frequently preserved in the papers of the hyperliterate: those drawn from professions which required extensive writing and record keeping – government officials, the clergy and lawyers.

One reason early modern shorthand is difficult for researchers to crack is that there were many different systems, and variations between, and even within, different editions of the same systems.

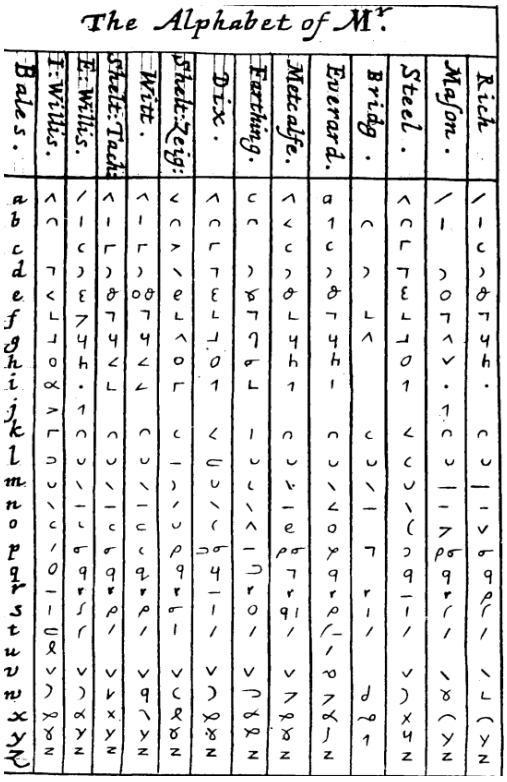


Figure 1: from Elisha Coles, *The Newest, Plainest, and the Shortest Short-hand* (1674)

Elisha Coles' 1674 pamphlet (Figure 1) identifies the "alphabets" of ten of the principal shorthand masters of his day; estimates of the total number of stenographic systems ranges to as high as over forty in the late seventeenth century (Matthews, 1934). The fact that all used similar or identical symbols to denote different letters, words or sounds made them very difficult to distinguish (Goldie, 2007; Boeddeker and McCay, 2024).

Early modern shorthand, consisting of different alphabetic or phonetic characters for letters, combinations of double and triple consonants and arbitrary characters for prefixes ("prepositions"), suffixes ("terminations"), and individual words, with many combinations and variants thereof, was clearly not a "code" involving simple substitution. Shorthand manuals often demonstrated how the same word could be written differently either by what Jeremiah Rich and Thomas Shelton called either "Alphabetical Rule" or "Marks for long Words". Users routinely personalized their shorthand by adapting or simplifying rules or even inventing new characters (McKenzie, 2021).

It is difficult to recognize that both Pepys and William Lloyd were using the same shorthand system, Shelton's *Tachygraphy*, both because of their writing choices and idiosyncrasies (see Figures 2 and 3). For instance, in both Shelton and Rich's systems, the plural "s" was signified by adding a dot or diacritic mark (a "tittle") to the left of the word. Lloyd ignored this rule entirely, manually inserting the shorthand character for "s" at the end of words. Pepys however followed it, as did Anglesey, who (like Pepys) used Shelton's system, although the latter was erratic – sometimes putting the plural tittle not on the left, but on the right, where it could be confused with words ending in vowels, where a diacritic mark was placed in different

positions according to which vowel (“a” to “u”) was being indicated.[2]

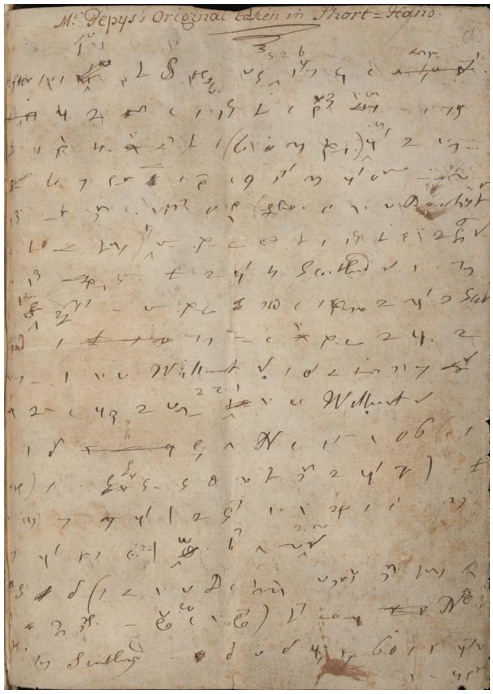


Figure 2: page from Pepys’ 1680 “Account of the Preservation of King Charles II”(Pepys Library, Magdalen College, Cambridge, 2141)

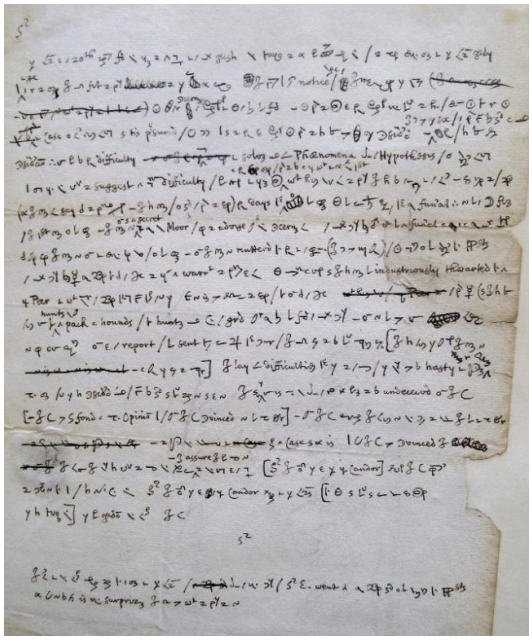


Figure 3: draft letter from Bishop William Lloyd to Sir Roger L’Estrange, 20 April 1686 (Gloucestershire Archives D3549/2/2/1)

Treby, who used Jeremiah Rich’s shorthand, generally chose not to indicate the plural form, leaving the grammatical number to be inferred from context. In the rare cases in which Treby indicated the plural he did so manually, like Lloyd, with the character for “s” instead of a tittle. The context of one of these cases suggests that he chose to add a plural “s” so as not to stumble over his notes when he read them aloud in committee (McKenzie, 2021).

In contrast to Pepys, Treby and Anglesey, whose shorthand seemed intended only for their private use – Anglesey, like Pepys, was apparently concerned with concealing sensitive passages in his journals – the Anglican bishop William Lloyd used his shorthand not only for notes and draft letters but in his correspondence with his son, several secretaries and current and former chaplains. It is likely that when writing shorthand letters intended to be understood by others, Lloyd and his correspondents ignored Shelton’s rule of using diacritical marks to avoid confusion with accidental marks resembling dots. (Lloyd also consistently rendered the pronoun “I”, a dot or tittle in Shelton, in longhand, possibly for the same reason).

But it is also possible some adaptations were chosen to make detection of the shorthand system more difficult: Lloyd and his circle, who consistently wrote the common word “is” in longhand, may have done so because the symbol in Shelton (1630) was distinctive and could thus have identified the system they were using.[3]

The fact that, as authors of shorthand manuals acknowledged, shorthand was easier to write than to read (Rich, 1654), and it was difficult to read even one’s own shorthand, meant that even proficient shorthand writers such as Pepys, Lloyd and Treby often used longhand for words that

[2] Medial vowels were typically not written out in early modern shorthand but indicated by the placement of the following consonant.

[3] The key in Lloyd’s papers in the Gloucestershire Archives makes it clear that Shelton’s symbol was known to Lloyd and his circle, and that their rendering it in longhand was a conscious choice.

could not be easily inferred from context, such as proper nouns. As Kate Loveman has demonstrated, Pepys also relied on occasional longhand words – typically names – as a "navigational" or "finding aid" (Loveman, 2025).

4 Shorthand and Secrecy

Conversely, some contemporary authors of journals or notes written largely in longhand wrote certain words in shorthand for the purposes of concealment. In his manuscript diaries, Arthur Annelsey, earl of Anglesey, discreetly inserted both short passages and individual words in shorthand which could be easily missed by someone not looking for such marks. Much of the shorthand consists of names or titles and seems to have been aimed at disguising the extent of Anglesey's contact with dissenting ministers, Catholic clergy and members of the political opposition, as well as his frequent visits to women of his acquaintance, to whom the earl officiously offered comfort or marital counselling. Anglesey also periodically added the shorthand for "wife", usually between the entries for two days (suggesting the early morning hours), followed by a number, presumably representing some sort of marital sex count – which restarted every year at on the Earl's anniversary of his wedding with Elizabeth Altham, 24 April.

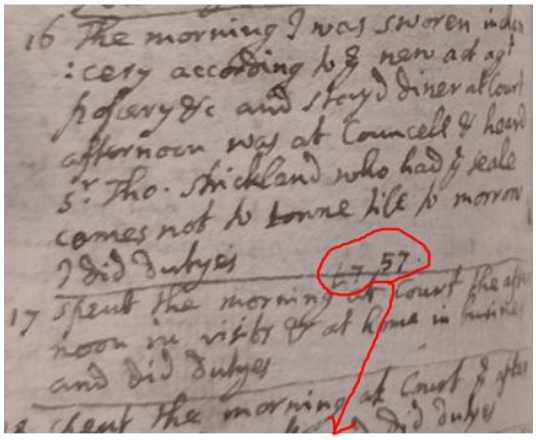
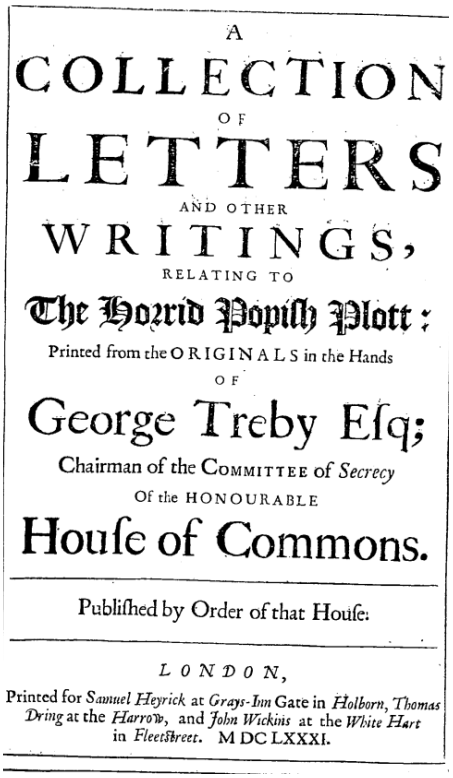


Figure 4: The shorthand for "wife", followed by "57", between the entries for 16 and 17 April 1673 (BL Add MS 40860, Diary of the Earl of Anglesey, 1671-1675, f. 45v).

Similarly, William Lloyd and his correspondents would sometimes depart from their common practice of writing proper nouns in longhand when they wanted to conceal sensitive material, using shorthand to make titles, names and other identifying terms opaque to prying eyes.

George Treby, a lawyer who would later become chief justice of the Common Pleas and Solicitor and Attorney General under William III, seems to have used shorthand in an exclusively solitary capacity, for recording trial proceedings, depositions, marginal comments, notes and draft addresses and letters. His shorthand was very difficult to crack because he left no key, no clue as to the system used, and no transcribed material. In the end I was able to establish that he was using Jeremiah Rich's system after detecting a pattern in the notes Treby had written, while Chair of the Committee of Secrecy investigating the Popish Plot (1678-81), on the back of the duke of York's former secretary Edward Coleman's confiscated papers.



A
COLLECTION
OF
LETTERS
AND OTHER
WRITINGS,
RELATING TO
The Horrid Popish Plott:
Printed from the ORIGINALS in the Hands
OF
George Treby Esq;
Chairman of the COMMITTEE of *Secrecy*
Of the HONOURABLE
House of Commons.
Published by Order of that House.
LONDON,
Printed for *Samuel Heyrick* at *Grays-Inn* Gate in *Holborn*, *Thomas Dring* at the *Harrow*, and *John Wickins* at the *White Hart* in *Fleetstreet*. M DC LXXXI.

Figure 5. Titlepage of the House of Commons' publication of a selection of Coleman's letters

On the backs of those letters not selected to be published by the House of Commons (see Figure 5) – i.e., because they were not sufficiently incriminating – Treby often wrote a symbol that only in Jeremiah Rich's system could signify "nothing", followed by other characters that meant "in it"; i.e., "nothing in it" or "nothing to [the] purpose" (Figure 6). While Treby had adapted Rich's shorthand and invented several new characters, my identification of the system allowed me to decipher enough of his notes to be able, over time, to infer modifications and innovations from context.

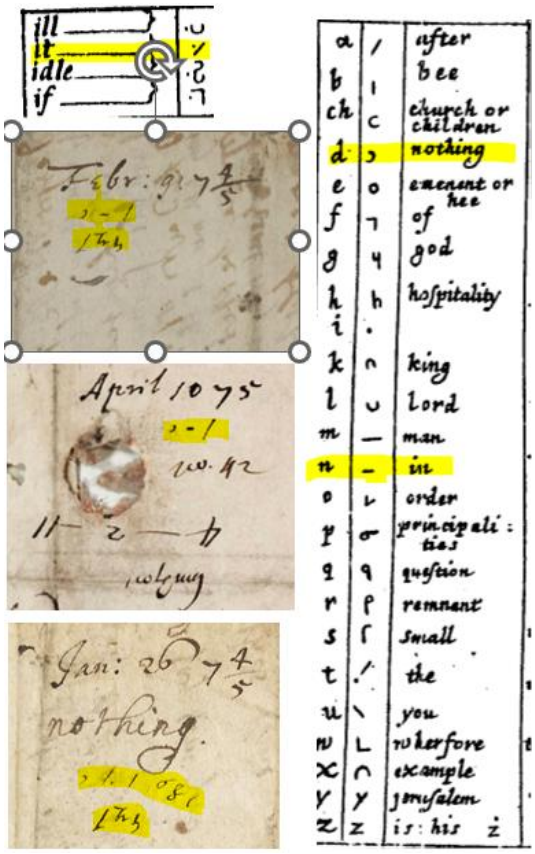


Figure 6. Jeremiah Rich's shorthand alphabet, from Samuel Botley's *Maximum in Minimo* (1674). Treby's shorthand notes on Coleman's letters, from top to bottom: "nothing in it[;] transcribed"; "nothing in it"; "nothing to the purpose[;] transcribed" (DRO D239/M/O)

In his case notes taken as a law student and newly minted lawyer, Treby sometimes made pointed comments in shorthand about the errors, inconsistencies and even corruption of various courtroom actors, including the judges (McKenzie, 2026). While he would become more cautious with age, he still let slip the occasional snarky remark. For instance, in the margin of a deposition of a Popish Plot suspect and would-be informant, Treby wrote in shorthand – next to a passage referring to Charles II's 1672 Declaration of Indulgence, seen by most English Protestants as a sinister plot to promote Catholicism – "as the king has promised".

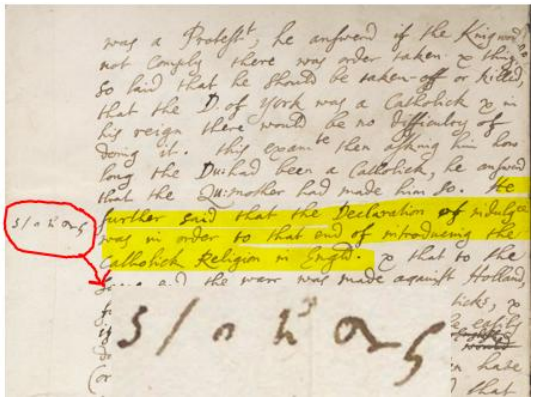
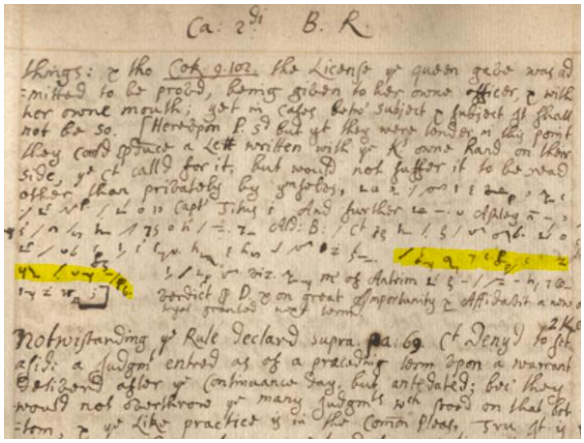
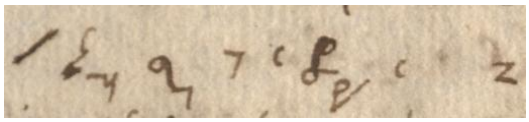


Figure 7. DRO D239/M/O/1623, Examination of Edward Fitzharris, March 1681. Highlighted text: "He further said that the Declaration of Indulgence was in order to that and of introducing the Catholick Religion in England"

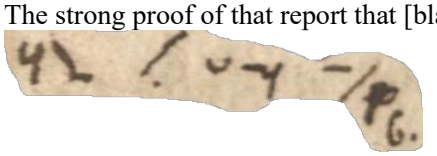
Like Lloyd and Pepys, Treby often wrote proper nouns difficult to infer from context in longhand. Also like them (as well as Anglesey), he sometimes rendered names in shorthand, presumably to add an extra layer of concealment by making it hard for any casual observer glancing over the page to identify sensitive passages. Treby also occasionally inserted blank spaces as though to underscore what was being omitted. In his reporting of one civil suit, in which both parties invoked the support of Charles II, the plaintiff brought a letter supposedly written by the king himself on his behalf to the judges. The latter refused to read it aloud, but conferred privately, obviously shocked, as Treby explained in shorthand, "that the letter was contradictory to [defence] testimony", even though the defendant claimed to have acted on the king's instructions. Treby noted in shorthand that the judges then gave the letter to a defence witnesses, Lord Ashley –

better known by his later title, the Earl of Shaftesbury. Ashley, seen by many as the leader of the first modern political party, the oppositional Whigs, read it and then said "softly however this should not alter his testimony. The strong proof of that report that" – and here there is a long blank space – " is given to lying notoriously".





The strong proof of that report that [blank] is

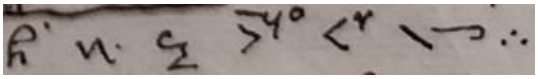


given to lying notoriously

Figure 8. Middle Temple MS2 C, Cases in the King's Bench, 1667-1672, 2:117, with detail of highlighted text

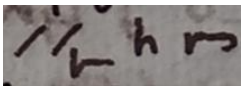
The clear implication was that the liar in question was Charles II himself, although Treby judiciously omitted the shorthand character for "king" which was readily identifiable, as it was the same in most major seventeenth-century stenographic systems (as well as the letter "k" in both Rich and Shelton).

Similarly, during the reign of Anne (1704-14), Bishop Lloyd and his correspondents often referred to the monarch indirectly as "she" or with various coded terms ("the town") rather than by her title, as the shorthand for "queen" in Shelton, as with most systems, was simply the letter "q".

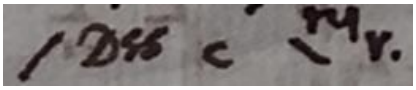


She very often changeth her mind

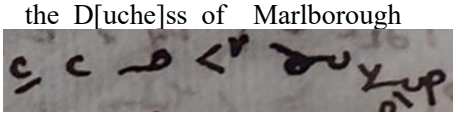
Figure 9. Gloucestershire Archives, D3549/2/2/7, Box 74, 267, William Lloyd junior to Bishop Lloyd, 24 June 1710



The town has turned



the D[uche]ss of Marlborough



out of all her employments

Figure 10. Gloucestershire Archives, D3549/2/2/7, Box 74, 267, William Lloyd junior to Bishop Lloyd, 4 January 1711

Not unlike Anglesey's insertions of small shorthand characters between diary entries, Treby occasionally wrote tiny shorthand words on the bottom or the edges of a page, sometimes upside-down, almost as a kind of steganography. On the back of an order of the Privy Council in 1691, when he was Attorney General, to investigate complaints that the Admiralty commissioners were being obstructed by Bristol magistrates, Treby wrote these cryptic lines (shorthand in italics): "Bristol *mayor* &c *spoke to lord* Pe[m]brook *who was far sent by* A[aro]n Smith &c *and could have nothing the* Sept. 9 91. (*After their affront[s] to the judge[*s])".

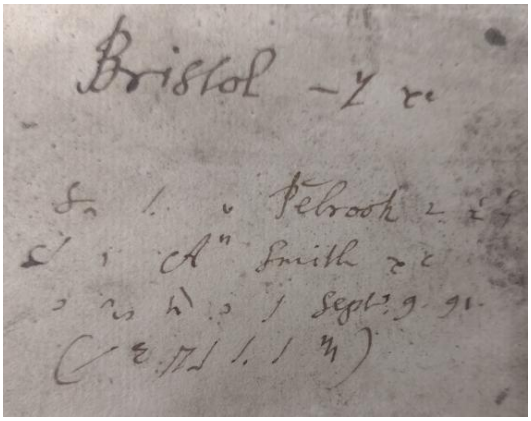


Figure 11. Derbyshire Record Office D239/M/O 1327 (23 April 1691)

Lord Pembroke was the lord of the Admiralty and Aaron Smith a Treasury employee; the mayor of Bristol was a militant Tory who had shortly before incited a mob to attack the circuit judge William Gregory in retaliation for the fraud conviction of a Tory customs surveyor. At the bottom of the Privy Council order, Treby has written what appears to be three names, in tiny shorthand characters: "Dudlestone Southern [?] Blathwayt[?]".

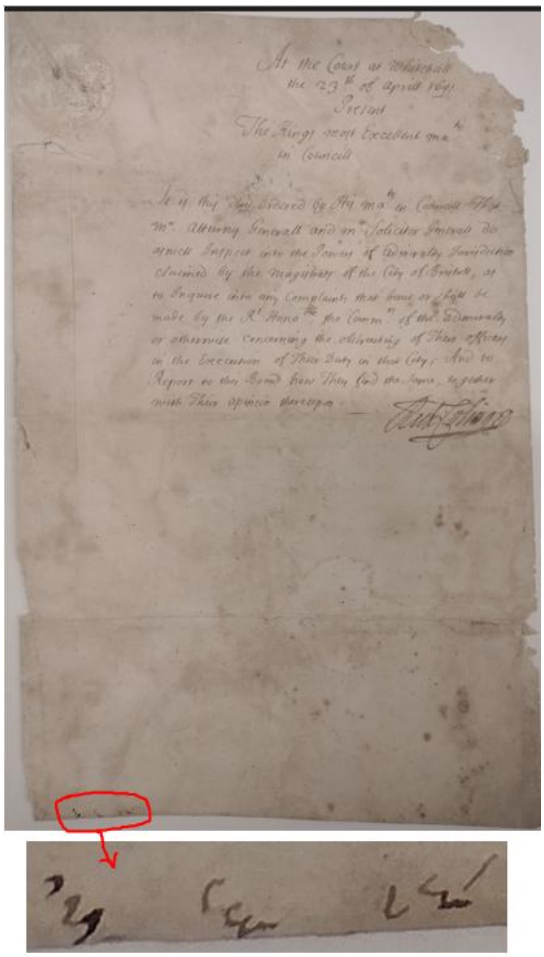


Figure 12. Front of previous, with detail of circled area

Sir John Dudlestone was a Bristol merchant and political ally of Knight's, but I have not been able to identify the other two individuals: who they were and why Treby wrote down their names remains a mystery.

6 Conclusion

Most of Treby's draft letters, like those of William Lloyd and his circle, were frustratingly vague, containing allusions to people and things that only his correspondent could understand. Many of Treby's shorthand notes, like some of Anglesey's, were banal, consisting of lawyerly queries, glosses, cross-references and reminders to himself. Arguably, the fact that much early modern shorthand is pedestrian worked to the advantage of such writers, for all that it frustrates the modern researcher: it was relatively safe to bury the occasional indiscreet remark in a larger body of apparently routine text. As I have suggested, painstaking palaeographical and historical sleuthing can yield some interesting material. However, in seeking to fill in the blanks we are often reminded that the ultimate key to early modern shorthand was not in the pages of the stenography manuals, but in the minds of the shorthand writers themselves – and they, like the characters in Vincent's mythical Monocrypticon, have vanished forever.

References

Nadine Akkerman and Pete Langman. 2025. *Spycraft: Tricks and Tools of the Dangerous Trade from Elizabeth I to the Restoration*. Yale University Press, New Haven.

Nadine Akkerman, ed. 2011. *The Correspondence of Elizabeth Stuart, Queen of Bohemia*, 3 vols. Oxford University Press, Oxford.

Nadine Akkerman. 2016. "Enigmatic Cultures of Cryptology" in James Dabyell and Andrew Gordon, ed., *Cultures of Correspondence in Early Modern Britain*. University of Pennsylvania Press, Philadelphia.

British Library (BL) Add MS 40860, Manuscript Diary of the Earl of Anglesey, 1671-1675.

Hannah Boeddeker and Kelly Minot McCay, ed. 2024. *New Approaches to Shorthand: Studies of a Writing Technology*. De Gruyter, Berlin and Boston.

John Bossy. 1991. *Giordano Bruno and the Embassy Affair*. Yale University Pess, New Haven.

John Bossy. 2001. *Under the Molehill: An Elizabethan Spy Story*. Yale University Press, New Haven.

Cambridge, Magdalen College, Pepys Library, 2141.

Elisha Coles. 1674. *The Newest, Plainest, and the Shortest Shorthand*. Peter Parker, London.

James Daybell. 2012. *The Material Letter in Early Modern England: Manuscript Letters and the Culture and Practices of Letter-Writing, 1512-1635*. Palgrave Macmillan, Basingstoke.

Derbyshire Record Office (DRO), George Treby Popish Plot Papers, D239/M/O.

Camille Desenclos and George Lasry. 2025. Cryptanalytic and historical challenges with unidentified encrypted documents from the early modern era. In *Proceedings of the 8th International Conference on Historical Cryptology, HistoCrypt 2025*, pages 41-51. Linköping University Electronic Press.

Katherine Ellison and Susan Kim. 2017. *A Material History of Medieval and Earl Modern Ciphers: Cryptography and the History of Literacy*. Routledge, New York.

Katherine Ellison. 2017. *A Cultural History of Early Modern English Cryptography Manuals.* Routledge, London.

Katherine Ellison. 2022. *Secret Writing in the Long Eighteenth Century*. Cambridge University Press, Cambridge.

Linford D. Fisher et al. 2014. *Decoding Roger Williams: The Lost Essay of Rhode Island's Founding Father*. Baylor University Press, Waco.

James Dougal Fleming. 2024. *Timothie Bright and the Origins of Early Modern Shorthand: Melancholy, Medicines, and the Information of the Soul*. Routledge, London.

Delphine Gardey. 2011. Une culture singulière ? Short-hand systems et abréviation de l'écriture en Angleterre à l'époque moderne. In Patrice Bret, Irina Gouzévitch, and Lilian Pérez, éd., *Les techniques et la technologie entre la France et la Grande-Bretagne XVII^{e}-XIX^{e} siècles*, pages 73-84. Centre de documentation d'histoire des techniques, Paris.

Delphine Gardey. 2001. *La dactylographe et l'expéditionnaire : Histoire des employés de bureau (1890-1930)*, Belin, Paris.

Gloucestershire Archives D3549, William Lloyd Papers.

Mark Goldie. 2007. *The Entring Book of Roger Morrice 1677–1691*, vol. 1: *Roger Morrice and the Puritan Whigs.* Boydell, Woodbridge.

Frances Henderson. 2008. Swifte and Secrete Writing in Seventeenth-Century England, and Samuel Shelton's *Brachygraphy*. *British Library Journal*, 4.

Frances Henderson. 2001. Reading, and Writing, the Text of the Putney Debates. In Michael Mendle, ed., *The Putney Debates of 1647*, pages 36-50. Cambridge University Press, Cambridge.

Frances Henderson. 2005. *The Clarke Papers,* volume V: *Further Selections from the Papers of William Clarke,* Cambridge University Press, Cambridge.

David Kahn. 1980. On the Origin of Polyalphabetic Substitution. *Isis*, 71(1): 122-27.

Benedek Láng. 2014. People's Secrets: Towards a Social History of Early Modern Cryptography. *Sixteenth Century Journal,* 45(2): 291-308.

Benedek Láng. 2018. *Real Life Cryptology: Ciphers and Secrets in Early Modern Hungary.* Routledge, London.

George Lasry, Norbert Biermann, and Satoshi Tomokiyo. 2023. Deciphering Mary Stuart's lost letters from 1578-1584. *Cryptologia*, 47(2): 101-202.

R. C. Latham and W. Matthews. 1971. *The Diary of Samuel Pepys*. G. Bell & Sons, London.

Kate Loveman. 2022. Women and the History of Samuel Pepys's Diary. *Historical Journal,* 65: 1221-1243.

Kate Loveman. 2025. *The Strange History of Samuel Pepys's Diary*. Cambridge University Press, Cambridge.

Alan Marshall. 1994. *Intelligence and Espionage in the Reign of Charles II, 1660-1685*. Cambridge University Press, Cambridge.

W. Matthews. 1934. Samuel Pepys, Tachygraphist. *Modern Language Review,* 29(4): 397-440.

Kelly Minto McCay. 2021. “All the World Writes Short Hand”: The Phenomenon of Shorthand in Seventeenth-Century England. *Book History*, 24(1): 1-36.

Andrea McKenzie. 2021. Secret Writing and the Popish Plot: Deciphering the Shorthand of Sir George Treby. *Huntington Library Quarterly*, 84(4): 783-824.

Andrea McKenzie. 2026. Off the Record: Legal Shorthand and George Treby’s Middle Temple Case Notes, 1667-72. *Journal of Legal History* (in press).

Middle Temple MS2 C, George Treby’s Cases in the King's Bench, 1667-1672.

June Palmer, ed. 2008. *The Letter Book of Thomas Hill 1660-1661: Westcountry Mercantile Affairs and the Wider World*, Volume 52. Devon and Cornwall Record Society, Exeter.

Jeremiah Rich. 1646. *Charactery*. Peter Cole, London.

Jeremiah Rich. 1654. *Semigraphy*. Nathaniel Brook, London.

Thomas Shelton. 1630. *Short-Writing: the Most Exact Method*, 2nd ed., J. Dawson, London.

Rev. John Smith. 1841. *The Life, Journals, and Correspondence of Samuel* Pepys. Volume 1. Richard Bentley, London.

George Treby. 1681.. *A Collection of Letters & other Writings Relating to the Horrid Popish Plot.* Samuel Heyrick, Thomas Dring, John Wickins, London.

Timothy Underhill and Timothy Peters. 2018. The Shorthand of Robert Willis, Physician-in-Extraordinary to King George III. *Electronic British Library Journal,* 2 (3):132.

Timothy Underhill. 2013. John Byrom and Shorthand in Early Eighteenth-Century Cambridge. *Transactions of the Cambridge Bibliographical Society*, 15(2).

Timothy Underhill. 2015. John Byrom and the Contexts of Charles Wesley’s Shorthand. *Wesley and Methodist Studies*, 7(1): 27-53.

Timothy Underhill. 2024. The Use of Shorthand by Women and Girls in Early Modern England. In Hannah Boeddeker and Kelly Minot McCay, *New Approaches to Shorthand. Studies of a Writing Technology,* pages 101-132. De Gruyter, Berlin.

Early Mechanical Cryptography and Binary Keying or The Possible Impact of the Damm Brothers on Leibniz's Machina Deciphratoria

Carola Dahlke
Deutsches Museum
c.dahlke@deutsches-museum.de

Magnus Ekhall
Independent Researcher
magnus.ekhall@gmail.com

Abstract

Mechanical cipher machines employing binary keying elements became widespread in the first half of the twentieth century. This paper examines the origins of binary keying in mechanical cryptography, motivated by Gottfried Wilhelm Leibniz's early work on binary arithmetic, his calculating machine (Machina Arithmetica), and his documented interest in cryptology.

By analysing Leibniz's surviving descriptions of a proposed cipher machine (Machina Deciphratoria), a reconstruction of this machine from 2012, and early cipher machines developed by Arvid Damm, we found no evidence that binary keying was intended in Leibniz's cryptographic ideas. Instead, binary keying appeared as a distinctly twentieth-century development.

The article highlights the possibility that reconstructions — when based on sparse sources — may unintentionally incorporate later design concepts into the reconstructed device, so that a historical device could be retroactively influenced by a modern idea.

1 Introduction

Mechanical cipher machines are often thought of as emerging in the early twentieth century, particularly those using a configurable and binary cipher key as well as some kind of stepping mechanism. In this context, a binary cipher key refers to a keying element composed of multiple components, each of which can assume one of two discrete states (e.g. active/inactive, tooth or no tooth, punched hole or no hole etc), thereby influencing the transformation applied during encryption. Examples of earlier mechanical devices used for cryptography certainly exist, but of simpler construction: cipher disks, slides, grilles or alphabet strips of various kinds.

In this paper, we distinguish between cryptographic devices and cryptographic machines. By "device" we mean an apparatus that assists the user in performing cryptographic operations without automating state changes between successive characters. By "machine" we refer to a mechanism that performs at least part of the cryptographic transformation automatically, typically through stepping, state progression, or key-controlled mechanical motion. In recent research there have been several cases where earlier devices have been interpreted as precursors of the modern mechanical cipher machine.

Our prime aim was to track down the point in time in the history of cryptography, where mechanical encryption started to become binary, i.e. when the cipher key is formed by a mechanism where parts or switches need to be set on or off. This paper reflects the detours and insights we made in this small study and the current state of our research.

2 Research Motivation and Initial Hypothesis

The initial hypothesis of this paper was to investigate the earliest use of a binary component used as a key in a cipher machine. It is well known that machines designed by Arvid Damm for the company AB Cryptograph in the early 1900s had this feature (Damm, 1917).

Basically, binary number systems were introduced to the European world by Gottfried Wilhelm Leibniz. It therefore made sense to start the search directly with Leibniz. Interestingly, his written estate contains references to a cipher machine, which was replicated a few years ago. Indeed, this recent reconstruction of Gottfried Wilhelm Leibniz' cipher device "Machina Deciphra-

toria", supposedly then more than 200 years prior to Damm's work, appeared to share several characteristics with the early machines of AB Cryptograph. A thorough comparison of the two devices was made. However, as the investigation progressed, it became clear that the evidential foundations for this comparison were uneven: the reconstructed "Machina Deciphratoria" was found to rely on a very limited set of primary, and only written sources. The path to Leibniz also led to other historical figures and machines (listed in the following section), which, although not helpful for the current question, nevertheless provided some interesting insights.

In this paper, the term *reconstruction* is used to describe the modern conceptual model of Leibniz's Machina Deciphratoria presented by Rescher et al. (2014). Given the very limited and vague nature of the surviving primary sources, this model should be understood as an interpretative construction rather than a historically correct mechanical design. Accordingly, references to a "reconstructed machine" in the following sections denote this conceptual model and do not imply that such a machine existed or was mechanically specified by Leibniz.

A research question was to investigate whether Damm might have been inspired by Leibniz. This was quickly found to be unlikely.[1] A closer look at an early Damm machine which has a binary type key can be found in section 6. Nevertheless, the two machines, i.e. the replica "Machina Deciphratoria" and the model A-1 machine from AB Cryptograph remained very similar, prompting the authors to investigate whether the replica of Leibniz's machine might even have been influenced by the work of the Damm brothers.

3 Sources and Methodology

This paper revisits the primary sources of Leibniz in order to collect the few texts that survive from Leibniz' own descriptions of his cipher machine idea. We also look at another early cipher device which has been attributed to being the world's first cipher machine: the "Chiffre-Machine" designed by Fredrik Gripenstierna. Primary sources in the form of letters sent to Arvid Damm by his older brother Ivar were investigated, as well as a manuscript on the subject of cryptology written by the two brothers together (National Security Agency, 2011), (National Security Agency, 2016). This material provided an insight on what the Damm brothers thought of contemporary cipher machines and that they strived to have a theoretical foundation for their work.

We briefly touch on other machines or inventors, such as the textile engineer Zschweigert, and the work of Vernam.

4 Leibniz's "Machina Deciphratoria": the Primary Sources

Starting in 1923 and still ongoing, Leibniz's very comprehensive and carefully sorted estate was and is still made fully accessible to the public by the Akademie-Ausgabe of the editing project Gottfried Wilhelm Leibniz. The publications have been edited using a historical-critical approach. This makes it convenient to trace his quotations on cryptography, and on a planned cipher machine.

The first time he mentions a Machina Deciphratoria is in a letter to Duke Johann Friedrich in February 1679. Previously, Leibniz describes his plan to construct a Machina Arithmetica. This is followed by this French[2] quote (see Figure 2 in appendix A) (Leibniz, 1679a):

> "This arithmetic machine makes me think of another beautiful machine that would be used to convert letters into encrypted letters and to decipher them: and this with great speed and in a way that is indecipherable to others. For I notice that most of the ciphers commonly used are easy to decipher; and those that are difficult to decipher are usually difficult to write, which causes busy people to abandon them. But with this machine, an entire letter could be converted into ciphers and deciphered almost as easily by the person who has the machine as it could be copied."

In the same year, Leibniz wrote a list of 16 issues in a memorial for Duke Johann Friedrich. Among them, he mentioned his plans to get work done on his Machina Arithmetica, and in this lines (see Figures 3 and 4 in appendix A) he added a

[1] Leibniz's ideas for a cipher machine can be found in his estate in letters and notes for a presentation. It is highly unlikely that Arvid Damm or his brother Ivar had access to the estate before it was gradually made public in 1923. However, it is certain that the Damm brothers knew the operating principles of the stepped drum.

[2] All translation were done by the authors

footnote about a machine for deciphering (Leibniz, 1679b):

> "Meanwhile, I wish to work diligently on the Machina Arithmetica (1), for which purpose I await a skilled craftsman, and I also wish to execute other matters. I have no doubt that Your Serene Highness will graciously assist me in this endeavour, as you have kindly offered to do."

And the footnote

> "(1) as well the machine for deciphering."

It is not until 1688 that the Machina Deciphratoria reappeared in his records. Leibniz was planning his important audience with Emperor Leopold I and listed all the scientific points he wanted to raise and to probably promote financial support. Needless to say, he prepared his presentation very thoroughly – there are a total of five versions for his audience in the documents, three of which mention the Machina Deciphratoria.

The first mention can be found in a version that is dated to August or September 1688 (Leibniz, 1688a) (see Figure 5 in appendix A):

> "What I have invented for Arcana in Mathesi Theoretica as well as Circa Leges Naturae et Causas Rerum is known to many, but the machines and useful practices that I have devised (except for the Arithmetic Machine and the improvements of clockworks) have mostly been kept secret and mentioned to almost no one, that I have them, waiting for an opportunity to present them in reality, so that they would not be published at an inopportune moment and exploited.
> Likewise, with my Machina Deciphratoria, a powerful man can correspond with many ministers in different ciphers, and without any effort, one can either write a cipher or understand what is sent to him in cipher, as if playing a musical instrument or clavichord, so that it is immediately available at the touch of a key and can simply be copied."

Attached to these lines written by Leibniz, the editors of Volume A IV,4 noted below (see Figure 6 in appendix A):

> "On the idea of the cipher machine, see also the remarks to Duke Johann Friedrich: our edition I,2 p.125 Z12-18; p. 223, line 30. It is not yet known whether Leibniz, who showed a sustained interest in the art of cryptography (cf. e.g. VI 4, note 239; I 13, p. 551, lines 12-16), continued to pursue his intention to construct such a machine or even put it into practice." [3]

The second mention can be taken from a small entry in August/September 1688 (Leibniz, 1688b) (see Figure 7 in appendix A):

> "Machines and useful practices: as my Machina Deciphratoria, the powerful man (potentate) has everything under control at once, as on the clavichord"

The third mention of the Machina Deciphratoria can be found in the second half of September 1688 (Leibniz, 1688c), and this is the most extensive description that Leibniz delivers of his idea, as far as the authors know (see Figure 8 in appendix A):

> "One of the most subtle inventions ever seen by humans is my Machina Arithmetica, which is admired in both the Royal Societies of London and Paris, even though only the effect has been seen in the poor model; but once I have the opportunity to employ craftsmen, I want to have several of them made to perfection for the chambers and observatories of great lords. A child can multiply and divide the most difficult examples on it, and everything happens in an instant, as it were, without any effort of the mind. And large numbers will be completed just as quickly as small ones. It is excellent for calculating entire tables, but it serves especially as a specimen of human mental power, in that a

[3] Following the suggestion of the editors, the authors as well checked the other mentioned entries on Leibniz' interest in cryptology. Just as mentioned by the editors, Leibniz is speaking about the art of cryptography. However, there is no other mention of a Machina Deciphratoria.

> machine can calculate what was otherwise considered proprium hominis.
>
> Based on the same principle, albeit much simpler, I have invented[4] a Machina Deciphratoria for persons of high rank. It is a small machine that is easy to carry. With it, a great lord can have many almost indissoluble ciphers at once and correspond with many ministers. However, since both the positioning in ciphers and the deciphering are laborious, the facility consists in the fact that one only has to grasp the given ciphers or letters as if one were playing on a clavicord or instrument, and the desired ones come out immediately and stand there; they only would have to be copied."

Leibniz does not seem to have elaborated on his ideas. No sketches or further descriptions have been found about how Leibniz imagined his machine would work. Although Leibniz reflects on the subject of cryptography and cryptanalysis in some of his correspondences and notes, his Machina Deciphratoria is not mentioned again, as far as the present volumes of the Academy edition reveal.

5 The Rescher Reconstruction of the Machina Deciphratoria

In 2012, Nicholas Rescher published a conceptual reconstruction of Leibniz's Machina Deciphratoria. The following description summarises the reconstructed machine as presented by Rescher, and does not imply that this design reflects a historically attested machine.[5]

The reconstructed machine is based around a horizontal hexagonal prism. On each of the six faces of the prism a slat with a scrambled alphabet is inserted. At any time, only the one side of the prism that is facing the operator is "active". The machine has a keyboard (as from a clavichord), one key per letter of the alphabet. When a key is pressed the corresponding letter on the scrambled alphabet slat is indicated. Additionally, the machine can change the active face of the prism by rotating it one step forward. Whether this is done or not is controlled by a Leibniz wheel with six pins.

A Leibniz wheel ("Staffelwalze, stepped drum") is a mechanical component invented by Leibniz and famously used in his Machina Arithmetica. It consists of a cylinder with a set of teeth (as on a cogwheel), but the teeth have varying length. Depending on how the Leibniz wheel is set up, different numbers of teeth can engage with other components.

In the case of the Rescher reconstruction, the wheel can be configured to rotate the prism in six different ways. This gives the machine a kind of irregular stepping between the alphabets (Rescher, 2012).

Rescher writes (2012) that "Leibniz's apparatus was in some regards akin to Arvid G. Damm's machine A-21 with its sliding revolving drum with 26 alphabetic faces". A better comparison is probably with Damm's A-1 from 1917.

It is worth pointing out again that there is no evidence that Leibniz's cipher machine was actually built during his lifetime. A comparison between Leibniz's surviving descriptions and the reconstructed machine shows that substantial extrapolation was required in order to derive a functioning mechanical design from the limited primary source material.

6 Fredrik Gripenstierna's "Chiffre-machine"

While the Leibniz reconstruction remains a modern conceptual model, the 18th-century work of Fredrik Gripenstierna (1728–1804) provides a concrete example of an early mechanical cryptographic device that was actually manufactured and demonstrated.

Gripenstierna was a Swedish military officer and nobleman who is credited with creating one of the first cipher machines (Beckman, 1999). His invention, which he calls a "Chiffre-Machine", consists of 57 wheels mounted on a single shaft. Each wheel is divided into two halves around its rim. One half bears the letters of the alphabet along with a few punctuation symbols, while the other half displays the numbers 00 to 99 arranged in a

[4] The original text uses the German word "ausgefunden" which is uncommon in modern German. It could mean either "discovered" or "invented" in this context. Please also compare the examples given in the Grimm DWB "Deutsches Wörterbuch by Jakob and Wilhelm Grimm", https://www.dwds.de/wb/dwb2/ausfinden

[5] More information on the exhibition in 2013, and pictures of the reconstructed machine, can be found e.g. on the website of the University of Pittsburgh: `https://pitt.libguides.com/c.php?g=12552&p=66419`

random order that is unique to each wheel.

The machine is operated by two people seated opposite each other. One person sees only the lettered halves of the wheels, while the other sees only the numbered halves. To encipher a message, the first person sets up a line of up to 57 letters, adjusting the wheels one by one. The second person then records the corresponding numbers visible on their side of the machine, producing the enciphered message (Beckman, 1999). This physical separation resembles the later 'red/black separation' principle in secure systems, in the limited sense that plaintext and ciphertext are handled in distinct operator domains.

The machine was accompanied by instructions describing how it should be used in such a way that the first wheel employed for enciphering the first letter was not always the same. The machine does not appear to have had a cipher key in the conventional sense, apart from these somewhat convoluted instructions determining where encryption should begin.

It is known that Gripenstierna's device was manufactured and demonstrated for the Swedish king Gustav III in 1786. Only one device is known to have been manufactured, no further mention of this device is known. The fate of the manufactured prototype is also unknown.

Gripenstierna himself wrote that he based his invention on principles he had received from his grandfather, Christopher Polhem (1661–1751) (Beckman, 1999). There are no known records on what part of the device that originates from Polhem and what Gripenstierna himself has contributed.

While it is evident that this device was actually constructed it is also worth pointing out that this apparatus is closer to a mechanical aid than an automated machine: there is no automatic stepping, no real cipher key (binary or otherwise). It is a manually operated polyalphabetic cipher device where the main feature being the separation of the cleartext and ciphertext domains.

7 Binary Keying in Early Damm Machines

The transition from such manual, mechanical aids to machines with true binary keying began in the early twentieth century with the collaborative work of Ivar and Arvid Damm.

Ivar Damm (1862–1917) was a Swedish mathematician and teacher. He studied at Uppsala university where he in 1896 presented his doctoral thesis "Bidrag till läran om kongruenser med primtalsmodyl" (Contributions to the Theory of Congruences with a Prime Modulus) (Thyselius, 1918). In 1899 he started work as a mathematics and physics teacher in Gävle, Sweden. His younger brother Arvid Gerhard Damm (1869–1927) trained and worked as an engineer working in the textile industry. Amongst other things he worked with Jacquard machines[6] at Vävskolan in Borås (Widman and Wik, 2017).

Both Arvid and Ivar had an interest in cryptology with Ivar perhaps having a more theoretical point of view (National Security Agency, 2011). In 1912 Arvid met George Lorimer Craig, a Scottish textile engineer, in Berlin and they developed several mechanical cipher machines together. There was not a lot of interest for these machines, but Arvid managed to gather enough interest for a patent consortium to be constructed in 1915 which later evolved into a company, AB Cryptograph, in 1916 (Widman and Wik, 2017).

During this time the Damm brothers exchanged correspondence in which they discussed various cipher methods and the theory behind them (National Security Agency, 2011). In 1917 they completed a manuscript titled "Kryptografiens grunddrag i systematisk framställning" (The fundamentals of cryptography in a systematic presentation). In this work, they define the terminology and theoretical foundations of cryptology, while also examining the practical use of ciphers in the form of machines and apparatus. Notably, none of Arvid Damm's machines are mentioned but at the same time almost all machines that are mentioned have their limitations and flaws highlighted (National Security Agency, 2016). The Damm brothers carefully studied the patents and other descriptions of contemporary cipher machines and devices. The highlighted limitations and flaws are not only of a cryptographic nature. In some cases it is instead related to a machine being difficult to use, or prone to having mechanical issues. Both the mathematical and engineering mindset has been used here (National Security Agency, 2016).

It is plausible that the experience with Jacquard machines and similar mechanical-digital equipment had an impact on the designs used for the

[6]The weaving patterns of Jacquard looms are fed with punch cards, an early form of programme control.

cipher machines designed by Arvid Damm. Ivar Damm's doctoral thesis deals with congruences with a prime modulus, an area of mathematics closely related to cryptology (Damm, 1896).

The early Damm machine models (starting with Cryptotyper in 1914) had a horizontal prism or cylinder that would step one step forward or backwards for each letter that was enciphered. The direction is determined by a chain which consists of a number of links. There are two types of links: high and low. The link type determines whether the stepping is forwards or backwards (Widman and Wik, 2017). The chain was configurable by the user and was a clear binary part of the cipher key.

Compared to Rescher's reconstruction of the Machina Deciphratoria, there are notable structural similarities. Like the reconstructed Leibniz machine, Damm's A-1 employs a prism carrying multiple cipher alphabets, with alphabet strips mounted on each face. The A-1 prism has 29 sides rather than six, but in both cases a single active alphabet is selected mechanically. Both designs incorporate a mechanism intended to produce irregular stepping of the prism: in the reconstructed Leibniz machine this role is assigned to a Leibniz wheel, while in Damm's A-1 the stepping behaviour is controlled by a configurable chain composed of high and low links. Although the historical evidence for such a mechanism in Leibniz's original conception is limited, the reconstructed machine closely resembles that of Damm's early machines. A photo of the A-1 with a covering hood removed can be seen in Figure 1.

Arvid Damm is not the only example of a cipher machine inventor coming from the textile industry of the early twentieth century. Rudolf Zschweigert, a German textile engineer, applied for a patent in 1919 for a mechanical cipher machine based on a permutation cipher (Schmeh, 2020).

Yet another example of early use of a binary cipher key is with the "Cipher Printing Telegraph System" described by Gilbert Vernam as being invented during the Great War. The system described used two punched paper tapes, one for the message and one for the key, which were binary added. A "machine perforator" was used to produce a third punched paper tape with the resulting enciphered message (Vernam, 1926). Used correctly, this system can implement what is now known as a one-time-pad: the only theoretically provable secure cipher. Vernam's system therefore represents one of the earliest clearly documented implementations of a binary keying mechanism in cryptographic machinery.

Figure 1: A photograph of the A-1 with the protective cover removed, exposing the prism with 29 alphabet strips (Häll, 2016).

8 Discussion: Leibniz and Binary Cryptography

Leibniz's well-known work on binary arithmetic makes it understandable that later interpretations have sought to associate his cryptographic ideas with binary mechanisms. However, while binary representation was of clear mathematical importance to Leibniz, there is no explicit evidence in the surviving sources that binary principles were intended to play a role in the Machina Deciphratoria.

On page 37, and in the footnote 229 on page 88, Rescher (2012) explains that Leibniz's memoranda give "a wealth of information" about his Machina Deciphratoria. "Given this detail, and considering what is known about Leibniz' calculating machine — and also about his ideas regarding cryptography — a conjectural reconstruction of his cryptographic machine is readily possible. [footnote 229]" Following this footnote 229 leads to Rescher's explanation on page 88:

> "However, this possibility only came to light in 2001 with the publication of A IV 4. In its wake, I have been able to devise a conceptual reconstruction of the apparatus with the assistance of my en-

Machine / Device	Year	Type	Main Properties	Binary Keying?
Machina Deciphratoria (Leibniz)	1679, 1688	Conceptual	Described as keyboard-operated (like a clavichord) to automate letter conversion.	No
Chiffre-Machine (Gripenstierna)	1786	Device	57 wheels on a single shaft. Featured physical domain separation ("red/black") between plaintext and ciphertext.	No
Model A-1 (A. Damm)	1917	Machine	29-sided prism with automatic, irregular stepping controlled by a configurable chain.	Yes
Vernam's System	1917	Machine	Utilized binary addition (XOR principle) of message and key via punched paper tapes.	Yes
Zschweigert's Machine	1919	Machine	Mechanical permutation cipher designed by a German textile engineer.	Yes
Rescher's Reconstruction	2012	Machine	Conceptual model using a 6-sided prism and a Leibniz wheel with pins to control stepping.	Yes (Interpretive)

Table 1: Summary of historical cryptographic devices and machines and the presence of binary keying.

> gineer friend Richard K. Arrangements are under way for a physical model of the machine to be fashioned by Messers Klaus B. and Wolfgang R. of Hannover who have expertise with the construction and operation of Leibniz's calculating machine."

When reading these quotations without researching the original passages in the Leibniz edition, it was easy to believe that much more than just the passages quoted above formed the basis for the Machina Deciphratoria.

As well, in Rescher's translation of the limited primary sources, subtle wording choices occasionally strengthen the impression that Leibniz's cipher machine was a concrete, realised artefact. For example, the French passage beginning "Mais par cette machine…" is rendered as "But with this machine of mine…" (Rescher, 2012). The possessive construction is not explicit in the original text and may suggest a degree of realisation or ownership that is not directly supported by the surviving sources. While minor in isolation, such translation choices contribute to an overall interpretation in which the Machina Deciphratoria appears better specified than the primary evidence suggests.

The connection to the Damm brothers is made in a footnote that Rescher links to on page 41 (Rescher, 2012). There he writes: "Various comparisons are suggestive. [Footnote 237]". Following this footnote to page 89, Rescher thanks two anonymous reviewers of his article in Cryptologia (Rescher, 2014): Both the Gripenstierna Chiffre-machine and the early Damm machines are referred to as "later analogues of Leibniz's apparatus". Even though Rescher himself was probably unfamiliar with the historical Swedish exhibits, the question remains whether the designers who worked with Rescher on the replica were subsequently influenced by their knowledge of existing machines. The fact is that neither Gripenstierna nor the Damm brothers could have copied the Leibniz machine, because it never existed - not even as a description, sketch or construction drawing.

In a review of Rescher's publication "Leibniz and Cryptography", Philip Beeley points out the weak link between the primary sources and the reconstructed machine. The fact that Rescher compares the reconstructed machine to the twentieth century Enigma machine is likened to "throwing all caution to the wind" (Beeley, 2014). As well, a critical essay from Fabian Dombrowski (2022) can be found on the matter. Dombrowski also addresses the scarcity of sources and raises the speculative question of whether the Machina Deciphratoria was perhaps never actually intended to be built, but was instead designed to secure funding from the rulers.

Rescher designed a machine based on the few surviving lines in which Leibniz refers to such a device. The construction was carried out with the assistance of experts experienced in the Machina Arithmetica, as well as collaborators who could have access to present-day knowledge of historical cipher machines. In this light, Rescher's later

description of the machine as an early Enigma is notable, as it invokes a comparison originating in much later cryptographic practice.

It seems to happen more often in the history of cryptology that big names can lead to a kind of glorification. For example, a similar uncertainty applies to Fredrik Gripenstierna's Chiffre-Machine. Although Gripenstierna explicitly attributes the underlying idea to his famous grandfather, Christopher Polhem, the surviving sources do not allow a clear distinction between what was originally Polhem's idea and what was developed or implemented by Gripenstierna himself. While the device was demonstrably constructed, the origin of its design principles therefore remains somewhat uncertain.

Concluding, Rescher's machine could be seen as an interesting mind game: Was it possible after all that Leibniz had been able to have such a machine constructed in theory? Leibniz invented the stepped drum, and by his extensive research and correspondence, he drew on a wealth of knowledge and tirelessly refined great ideas, as can be seen in his extensive estate. According to Dombrowski (2022) who refers to Leibniz (1682–1688), it is fairly certain that Leibniz was well acquainted with contemporary works on cryptography such as the Cryptomenytices from Selenus (Selenus, 1624). However, it seems to be historical fiction that Leibniz was the first to place permuted alphabets on a cylindrical form. We attribute this step to Gripenstierna (or possibly Polhem), and thereafter to Thomas Jefferson, as long as there is no earlier historical evidence.

Table 1 contains an overview of the majority of devices discussed in this paper.

9 Conclusion

This study finds no evidence that binary keying was part of Leibniz's ideas about cryptography. Instead, binary keying appears to be a development of the early twentieth century. The reconstructed Machina Deciphratoria, together with its exhibition and catalogue, shows how reconstructions based on limited sources can unintentionally create convincing but misleading interpretations.

This highlights the importance of returning to original sources and clearly separating historical evidence from later reconstructions, as failure to do so may contribute to the formation of myths in the history of cryptology.

Acknowledgments

The authors would like to thank the library of the Deutsches Museum, in particular Florian Preiss, for his help in obtaining historical sources. As well, we would like to thank three anonymous reviewers for their valuable comments and suggestions which we gratefully included to enhance our submission.

References

Bengt Beckman. 1999. *Världens första kryptomaskin*. FRA.

Philip Beeley. 2014. Leibniz and Cryptography: An account on the occasion of the initial exhibition of the reconstruction of Leibniz's cipher by Nicholas Rescher. *Leibniz Review*, 24:111–122.

Ivar Damm. 1896. *Bidrag till läran om kongruenser med primtalsmodyl*. Phd thesis, Uppsala Universitet.

Arvid Gerhard Damm. 1917. Apparatus for Producing Series of Signs. U.S. Patent US1233035A. Filed July 21, 1915; granted July 10, 1917.

Fabian Dombrowski. 2022. Kein Geheimnis. Leibniz und die Kryptologie des 17. Jahrhunderts. *Die junge Mommsen*, 4.

Peter Häll. 2016. Digitalt museum, eks0029825. Creative Commons Attribution 4.0 International (CC BY 4.0).

Gottfried Wilhelm Leibniz. 1679a. Nr. 110: Leibniz an Herzog Johann Friedrich, Februar (?) 1679, Eigenh. Konzept A (Hannover). In Leibniz-Archiv / Leibniz-Forschungsstelle Hannover, editor, *A I 2. Allgemeiner, politischer und historischer Briefwechsel*, page 125. Akademie Verlag 1927, Berlin. Akademie-Ausgabe.

Gottfried Wilhelm Leibniz. 1679b. Nr. 184: Leibniz für Herzog Johann Friedrich, Memorial. Oktober 1679, Eigenh. Konzept B (Hannover). In Leibniz-Archiv / Leibniz-Forschungsstelle Hannover, editor, *A I 2. Allgemeiner, politischer und historischer Briefwechsel*, page 223. Akademie Verlag 1927, Berlin. Akademie-Ausgabe.

Gottfried Wilhelm Leibniz. 1688a. Nr. 6: Aufzeichnung für die Audienz bei Kaiser Leopold i. August/September 1688. In Leibniz-Edition Potsdam der Berlin-Brandenburgischen Akademie der Wissenschaften, editor, *A IV 4. Politische Schriften*, page 27. Akademie Verlag 2001, Berlin. Akademie-Ausgabe.

Gottfried Wilhelm Leibniz. 1688b. Nr. 7: Kurzfassung einiger Ausführungen vor Kaiser Leopold i.

August/September 1688. In Leibniz-Edition Potsdam der Berlin-Brandenburgischen Akademie der Wissenschaften, editor, *A IV 4. Politische Schriften*, page 45. Akademie Verlag 2001, Berlin. Akademie-Ausgabe.

Gottfried Wilhelm Leibniz. 1688c. Nr. 8: Ausführliche Aufzeichnung für den Vortrag bei Kaiser Leopold i. Second half of September 1688. In Leibniz-Edition Potsdam der Berlin-Brandenburgischen Akademie der Wissenschaften, editor, *A IV 4. Politische Schriften*, page 68. Akademie Verlag 2001, Berlin. Akademie-Ausgabe.

National Security Agency. 2011. Papers on cryptography: Nemochiffer, linjalchiffer. Z104.D12. NSA Historical Archive, ID 2011.0101.0515.

National Security Agency. 2016. Kryptografins grundrag (handwritten manuscript). Z104.D13M 1917. NSA Historical Archive, ID 2016.0101.2213.

Nicholas Rescher. 2012. *Leibniz and Cryptography: An Account on the Occasion of the Initial Exhibition of the Reconstruction of Leibniz's Cipher Machine*. Office of Scholarly Communication and Publishing, University of Pittsburgh Library System, University of Pittsburgh.

Nicholas Rescher. 2014. Leibniz's Machina Deciphratoria. A Seventeenth-Century Proto-Enigma. *Cryptologia*, 28(2):103–115.

Klaus Schmeh. 2020. The Zschweigert cryptograph – a remarkable early encryption machine. In *Proceedings of the 3rd International Conference on Historical Cryptology HistoCrypt 2020*, pages 126–134. Linköping University Electronic Press.

Gustavus Selenus. 1624. *Cryptomenytices et cryptographiae libri IX*. Lunaeburgi: Exscriptum typis & impensis Johannis & Henrici fratrum, der Sternen, bibliopolarum Lunaeburgensium. Pseudonym of August von Braunschweig-Lüneburg.

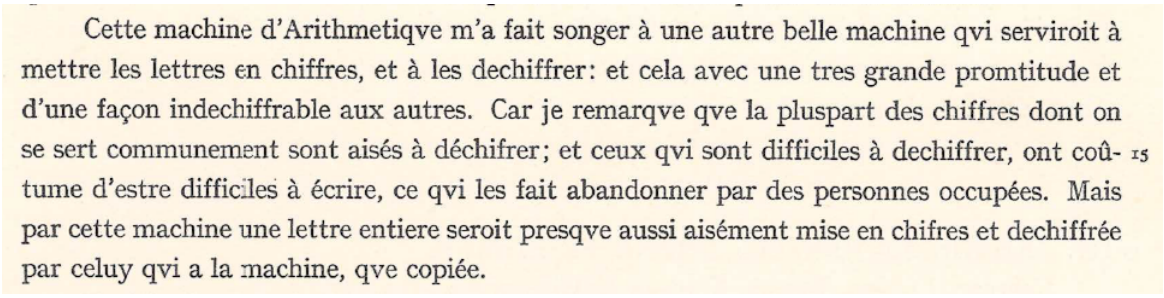
Erik Thyselius. 1918. *Vem är det?* P. A. Norstedt & söners förlag.

G. S. Vernam. 1926. Cipher printing telegraph systems. In *American Institute of Electrical Engineers proceedings*. Declassified copy hosted by the U.S. National Security Agency (NSA), Friedman Documents, Patent and Equipment Records, Folder 545, Document ID A4148856.

Kjell-Owe Widman and Anders Wik. 2017. *Damm och AB Cryptograph*. FRA.

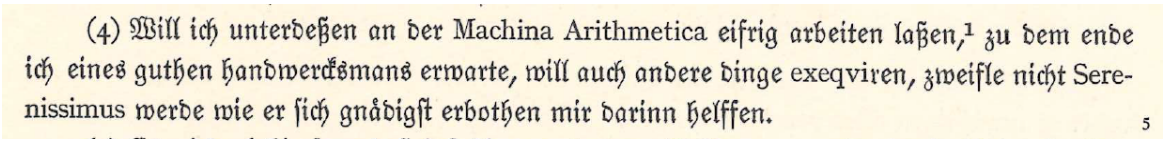
A Primary Sources by Leibniz on the Machina Deciphratoria

This appendix reproduces excerpts of the primary source material by Leibniz as discussed in the paper.



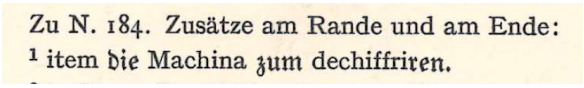
Cette machine d'Arithmetiqve m'a fait songer à une autre belle machine qvi serviroit à mettre les lettres en chiffres, et à les dechiffrer: et cela avec une tres grande promtitude et d'une façon indechiffrable aux autres. Car je remarqve qve la pluspart des chiffres dont on se sert communement sont aisés à déchifrer; et ceux qvi sont difficiles à dechiffrer, ont coûtume d'estre difficiles à écrire, ce qvi les fait abandonner par des personnes occupées. Mais par cette machine une lettre entiere seroit presqve aussi aisément mise en chifres et dechiffrée par celuy qvi a la machine, qve copiée.

Figure 2: Excerpt from A I,2 1697, p.125; please note that the expression "m'a fait songer à" marks his cipher machine as a product that exists only in his mind and not in reality - at least at this point in time. The same holds true for the use of the conditional to describe the machine ("par cette machine une lettre entière seroit ...")



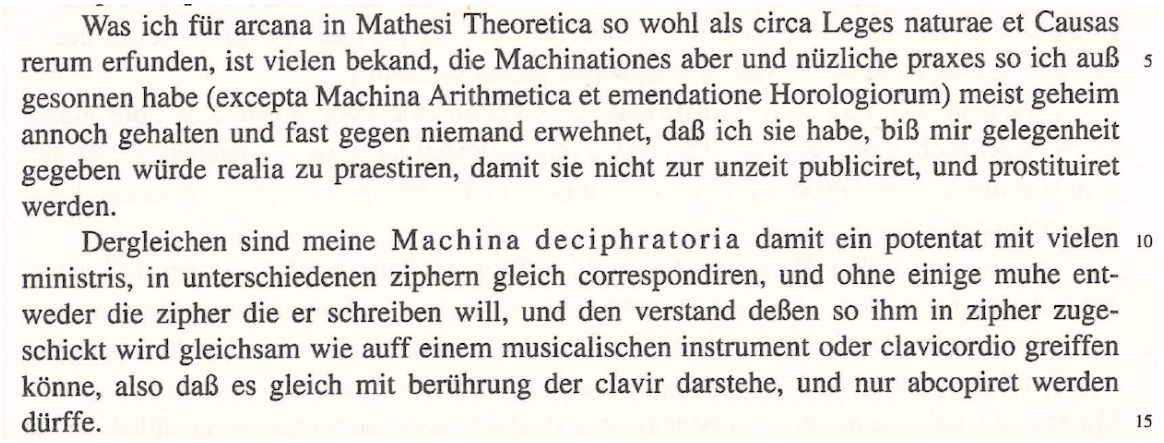
(4) Will ich unterdeßen an der Machina Arithmetica eifrig arbeiten laßen,[1] zu dem ende ich eines guthen handwercksmans erwarte, will auch andere dinge exeqviren, zweifle nicht Serenissimus werde wie er sich gnädigst erbothen mir darinn helffen.

Figure 3: Excerpt from A I,2 1679, p.223



Zu N. 184. Zusätze am Rande und am Ende:
[1] item die Machina zum dechiffriren.

Figure 4: Excerpt from A I,2 1679, p.223



Was ich für arcana in Mathesi Theoretica so wohl als circa Leges naturae et Causas rerum erfunden, ist vielen bekand, die Machinationes aber und nüzliche praxes so ich auß gesonnen habe (excepta Machina Arithmetica et emendatione Horologiorum) meist geheim annoch gehalten und fast gegen niemand erwehnet, daß ich sie habe, biß mir gelegenheit gegeben würde realia zu praestiren, damit sie nicht zur unzeit publiciret, und prostituiret werden.

Dergleichen sind meine Machina deciphratoria damit ein potentat mit vielen ministris, in unterschiedenen ziphern gleich correspondiren, und ohne einige muhe entweder die zipher die er schreiben will, und den verstand deßen so ihm in zipher zugeschickt wird gleichsam wie auff einem musicalischen instrument oder clavicordio greiffen könne, also daß es gleich mit berührung der clavir darstehe, und nur abcopiret werden dürffe.

Figure 5: Excerpt from A IV,4 1688, p.27

10 deciphratoria: Über die Idee der Dechiffriermaschine vgl. auch die Bemerkungen gegenüber Herzog Johann Friedrich: unsere Ausgabe I,2 S. 125, Z. 12–18; S. 223, Z. 30. Ob Leibniz, der an der Dechiffrierkunst anhaltendes Interesse zeigte (vgl. z. B. VI,4 N. 239; I,13 S. 551, Z. 12–16), die Absicht, eine solche Maschine zu konstruieren, weiter verfolgt oder sogar in die Tat umgesetzt hat, ist noch nicht bekannt. 23 lasten: vgl.

Figure 6: Excerpt from A IV,4 1688, p.27

Machinae und Nuzliche praxes: als Machina mea deciphratoria[,] potentat hat darinn viel ziphern zugleich alles im griff, wie aufm Clavicordio

Figure 7: Excerpt from A IV,4 1688, p.45

Eine der Subtilsten Inventionen so von Menschen gesehen worden, ist meine Machina Arithmetica so man in den beyden koniglichen Societäten zu London und Paris admiriret, da man doch nur die würckung in dem Schlechten Modell gesehen; wenn ich aber einmahl gelegenheit habe Handwercks leute zu halten, will ich deren etliche in Vollkommenheit vor großer Herren kammern und observatoria machen laßen, ein kind kan darauff die schwehrsten Exempel multipliciren und dividiren, und geschicht alles gleichsam in einem augenblick ohne arbeit des gemüths. Und große Zahlen werden eben sobald fertig als kleine. Ist treflich ganze tafeln auszurechnen, dienet aber sonderlich als ein Specimen der Menschlichen gemüthskraffte dadurch zu wege zu bringen daß eine Machina rechnen kan, welches sonsten proprium hominis gehalten worden.

Aus gleichen principio wiewohl viel leichter, habe eine Machinam deciphratoriam vor hohe Personen ausgefunden. Ist eine kleine Machinula die leicht bey sich zu fuhren. Darauff kan ein großer herr viele fast unauflößliche Ciphern zugleich haben, und mit vielen Ministris correspondiren; weilen aber sowohl die stellung in Ziphern als das deciphriren mühsam, so bestehet die facilitat darinn, daß man die gegebene Ziphern oder buchstaben nur greiffen darff als wenn man auff einem clavicordio oder Instrument spielte, so kommen die begehrten augenblicklich herauß und stehen da; durffen denn nur abgeschrieben werden[.]

Figure 8: Excerpt from A IV,4 1688, p.68

Establishing a Document Layout Analysis Baseline for Historical Cipher Keys

Raphaela Heil
Stockholm University
Sweden
raphaela.heil@ling.su.se

Alicia Fornés
Universitat Autònoma de Barcelona
Spain
afornes@cvc.uab.es

Benedek Láng
Eötvös Loránd University
Hungary
lang.benedek@gtk.elte.hu

Beáta Megyesi
Stockholm University
Sweden
beata.megyesi@ling.su.se

Abstract

Historical cipher keys encode mappings between plaintext elements and cipher symbols and are characterized by complex, heterogeneous handwritten layouts. This paper establishes a baseline for document layout analysis (DLA) of historical cipher keys using a newly annotated dataset of 350 images from European archives dating from ca. 1300 to 1850 CE. We evaluate four YOLO-based architectures under three conditions: training from scratch, cross-domain transfer from models pre-trained on DocLayNet and CATMuS in a class-agnostic setting, and fine-tuning of these pre-trained models on cipher key data. Results show that training from scratch is limited by data scarcity and unstable convergence, while direct transfer across DLA domains performs poorly. In contrast, fine-tuning consistently improves performance across all architectures, demonstrating the feasibility of adapting existing DLA models to cipher keys and supporting downstream tasks such as key extraction and comparative cryptographic analysis.

1 Introduction

A historical cipher key is a document that specifies the correspondence between plaintext elements and their encoded representations used in a cipher system. Such keys were widely used in diplomatic, military, and administrative contexts to enable the encryption and decryption of sensitive correspondence. Unlike ciphertexts, which consist mostly of encoded symbols with or without cleartext, cipher keys typically combine explanatory text, structured mappings, and graphical elements, resulting in complex and highly variable document layouts. Figure 1 shows two representative examples of historical cipher keys from Europe.

At a structural level, cipher keys most commonly contain one or more *alphabets*, which define the mapping between individual letters (or letter combinations) and their corresponding cipher symbols. In addition, many keys include *nomenclature elements*, which extend the alphabet by assigning codes to higher-level linguistic units such as names, places, titles, syllables, or frequently used words and phrases. These mappings are often arranged in structured forms such as tables, lists, or aligned rows and columns, but their exact visual realization varies across documents and time periods (Megyesi et al., 2024).

Cipher keys frequently exhibit heterogeneous layouts that combine multiple organizational principles within a single page. Alphabet and nomenclature sections may coexist with running text, marginal annotations, figures, decorative elements, stamps, or watermarks. Furthermore, mappings can be arranged horizontally or vertically, may involve one-to-one, one-to-many, or many-to-one relationships, and are often written by hand with limited visual consistency. This structural and visual diversity distinguishes cipher keys from both standard textual documents and more regular tabular material.

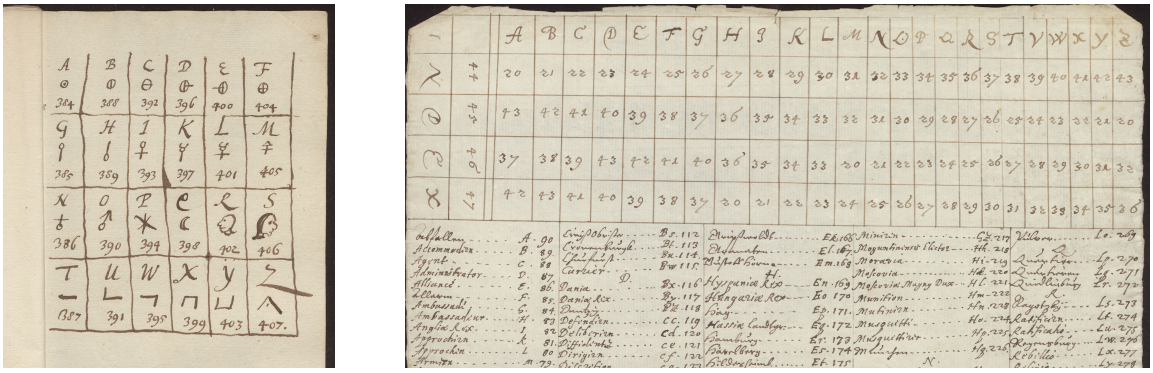


Figure 1: Examples of historical cipher keys from the Swedish National Archives. Left: DECODE record 4323, right: DECODE record 4329.

The automatic analysis of cipher keys therefore poses challenges that go beyond traditional handwritten text recognition. Before individual symbols or mappings can be transcribed or interpreted, semantically relevant regions of the page must first be identified and localized. Document layout analysis (DLA) addresses this need by segmenting cipher key pages into meaningful regions, making it a crucial preprocessing step for downstream tasks such as symbol recognition, key extraction, comparative cryptographic analysis, and assisted decryption.

On a general level, DLA identifies regions of interest, such as text blocks or figures, in document images, which can then be processed further, for example by applying handwritten text recognition (HTR) to text lines, or by performing structured information extraction from tables. In the specific case of cipher keys, the regions detected through DLA are directly tied to cryptographic interpretation. For example, identifying alphabet and nomenclature regions enables the extraction of plaintext–cipher symbol pairs, which can be used for decryption attempts or for comparing symbol inventories across different keys and potentially related ciphertexts. Beyond individual documents, automatic region identification also supports large-scale comparative studies, such as analyses of nomenclatures across entire collections (see e.g. Megyesi et al., 2024).

Given the relevance of layout analysis for such downstream tasks, this work aims to establish a baseline for document layout analysis of historical cipher keys. To this end, we compare the performance of four YOLO-based architectures (Redmon et al., 2016) under different training and fine-tuning conditions, using a newly established dataset of historical cipher keys. Concretely, we investigate the following questions:

1. How well do YOLO-based models perform on cipher key DLA, when trained *from scratch* on a comparatively small dataset?
2. How well do YOLO-based models, pretrained on other DLA domains, namely PDF documents and medieval manuscripts, perform in identifying regions of interest in cipher key images, in a class-agnostic setting?
3. Can the performance of models trained from scratch be improved by fine-tuning pretrained models with the cipher key dataset?
4. Can the performance for cipher key-specific classes, such as alphabet and nomenclature key regions, be further improved by limiting the training data?

2 Related Work

Historical cipher keys and their internal structure have been studied primarily from a cryptological and historical perspective. Large-scale analyses of European cipher keys have shown that most keys combine alphabet mappings with nomenclature sections that encode higher-level linguistic units such as names, places, titles, words, or phrases, often supplemented with nulls and special symbols (Megyesi et al., 2024). These mappings are typically organized in tables, lists, or aligned rows and columns, but exhibit substantial variation in layout, orientation, and complexity across time periods and regions. Studies of key usage and instructional material further highlight that cipher keys frequently include explanatory text and operational rules, which contributes to heterogeneous

page layouts (Láng et al., 2025). From a computational perspective, prior work has discussed the challenges of automatically extracting structured mappings from cipher keys and emphasized the need to first identify key components such as alphabet and nomenclature regions (Tudor et al., 2020). In addition, corpus-building efforts such as the DECODE project have provided standardized terminology and large collections of cipher keys and ciphertexts, enabling systematic comparative analysis (Megyesi et al., 2019). However, despite this growing body of work, the automatic layout analysis of cipher keys has received little attention, motivating the present study.

2.1 Document Layout Analysis

A significant body of literature examines DLA for various types of documents, ranging from printed corporate forms, to ancient handwritten documents. Binmakhashen and Mahmoud (2019) present a summary of DLA works until ca. 2019.

Recent works explore various kinds of deep learning approaches, such as the cross-attention-based HookNet, proposed by Wu et al. (2025).

Additionally, YOLO-based approaches have for example been examined in the context of printed documents (Li et al., 2025) and historical, Greek dictionaries (Ioakeimidou et al., 2024).

Besides this, several HTR frameworks, such as Loghi (van Koert et al., 2024; Klut et al., 2023) and Kraken (Kiessling, 2026), incorporate DLA approaches into their pre-processing pipeline, to identify textual regions which are then segmented into individual lines.

We are not aware of any prior works that study DLA for historical cipher keys.

2.2 DLA Datasets

Various DLA datasets have been presented in the literature, both as stand-alone projects and in the context of competitions. These datasets are often focused on a single domain or are limited to images of a specific document type (file format, content, layout).

The two datasets that form the basis for some of the pre-trained models used in this work are DocLayNet (Pfitzmann et al., 2022), consisting of PDF pages, i.e. contemporary digital material, and CATMuS (Clérice et al., 2024), focusing on medieval handwritten pages.

Other DLA datasets, which focus on a similar time frame, but different domain compared to our work, are for example the HORAE dataset (Boillet et al., 2019), containing images of prayer books from the late Middle Ages, the DIVA-HisDB (Simistira et al., 2016), a collection of challenging medieval documents, and SAM (Zottin et al., 2024), the ICDAR 2024 Competition on Few-Shot and Many-Shot Layout Segmentation of Ancient Manuscripts.

To the best of our knowledge, no prior dataset for DLA of historical cipher keys exists.

3 Study Design

The following sections briefly outline the different components of our study design, concluding with a description of the conducted experiments.

3.1 Data

The experiments in this work are based on a collection of 350 images of, predominantly handwritten, cipher keys from archives in Europe, sourced from the DECODE database (Héder and Megyesi, 2022). The original documents are estimated to have been created between ca. 1300 CE and ca. 1850 CE, with the bulk of the data originating from ca. 1500 CE to 1700 CE. The keys are constructed using various combinations of letters, numbers and graphical symbols.

During the digitisation process, the cipher key images were arranged as records, grouping documents that stem from the same book or collection. Each record has been annotated with several metadata fields, e.g. pertaining to age and provenance, as far as it could be established.

3.1.1 Data Annotation

All images were annotated manually by one annotator, with the support of several experts, when uncertainties arose. The annotation scheme was specifically designed for the use case of cipher keys, and their application in downstream recognition tasks and analyses. We follow the terminology established by (Mikhalev et al., 2023), resulting in the following cipher key-related regions, as well as a number of more general DLA classes. An annotation example is shown in Figure 2.

Alphabet Key Describes which alphabet element, including double letters, belongs to which alphabet code element(s). Note that this does not have to be a one-to-one mapping, and some (or all) alphabet elements may be represented by several alphabet code elements.

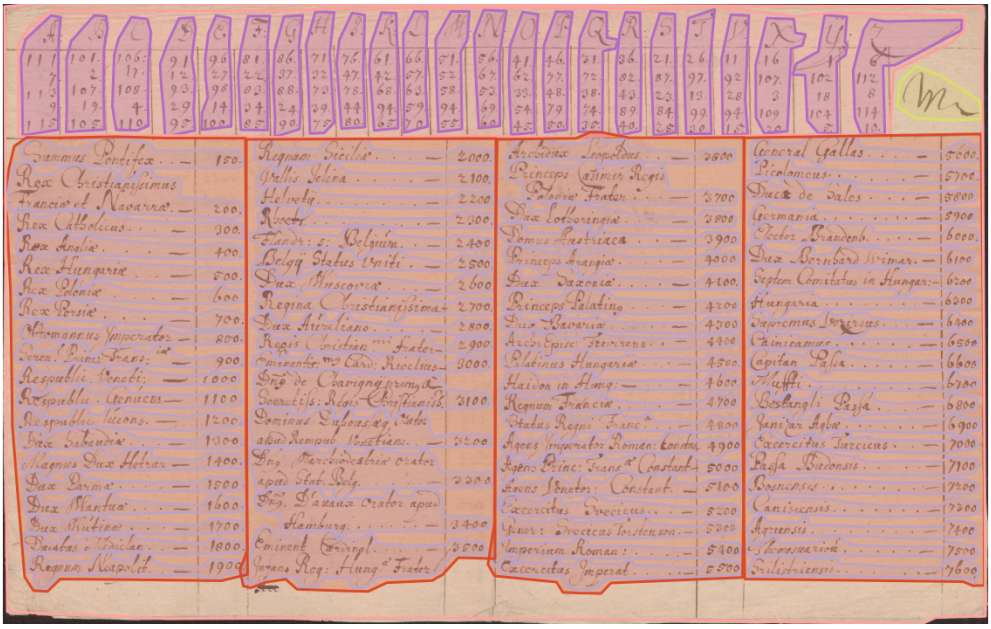


Figure 2: Sample of a cipher key, annotated with an alphabet key region, containing key columns, at the top, followed by four nomenclature key columns, containing key rows. Original image source: Swedish National Archives, DECODE record 4180.

Nomenclature Key Describes which nomenclature element belongs to which nomenclature code element(s). A nomenclature element is anything "larger" than an alphabet element, i.e. it can be a syllable, a name, a function, a content word, as well as a phrase.

Key Row Indicates a horizontal entry in a cipher key, i.e. a combination of a plaintext element and its corresponding code element, placed horizontally next to each other. Any order of plaintext and code is possible and mappings may appear in any form (i.e. one-to-one, one-to-many, many-to-one, many-to-many).

Key Column The vertical counterpart to key rows, i.e. a combination of a plaintext element and its corresponding code element, placed vertically above/below each other. Apart from the orientation, key rows and key columns do not differ in their structure.

Operational Element Groups various operational elements, such as nullities, nullifiers, duplication signs, and punctuation.

Text Any kind of textual area that does not fall into one of the other categories. This can for example be running text (paragraphs), headlines, page numbers, or annotations (marginalia).

Figure Any kind of drawing, illustration or figure that is not explicitly of textual nature. This may also include decorations and "doodles".

Table Any kind of table that is **not** a form of cipher key.

Stamp A short text, sometimes with a logo, stamped onto the page, for example indicating which archive indexed a given document.

Watermark Any form of watermark, either physically applied during the paper production process, or digitally added after the digitisation.

Page Indicates the borders of a page. This is primarily intended to delineate the actual page content from noise, stemming from the digitisation surface (e.g. table) and pages (or other artefacts) in the background. Images, showing a page spread, are annotated with two separate bounding boxes, unless the content runs seamlessly across the binding.

Fragment Marks the rare case in which a given image contains a page fragment, i.e. a piece of paper, torn off from the current, or another, page.

Other A catch-all label for any content that cannot be definitively assigned to any of the aforementioned categories, primarily designed to ac-

count for unforeseen annotation cases and to mark areas that should be reviewed by an expert.

3.1.2 Data Splitting

The 350 images were split into 50 pages for testing and 300 pages for 5-fold cross-validation, i.e. 60 images per validation fold. To reduce the risk of data leakage and biases, the original record-level grouping was maintained during the data splitting. Images from any given record always belong to the same subset and are never split across several. Additionally, a number of individual images were grouped into *virtual* records, as their metadata indicated that they were created and/or used by the same person(s). These virtual records were treated the same as all other records, i.e. never split across several subsets.

Besides this, an attempt was made to balance the cross-validation and test sets with respect to age and combination of symbol sets. However, due to the constraint of not splitting records, as well as some uncertainty in the recorded metadata, a perfectly balanced split cannot be guaranteed. Figure 3 summarises the age and symbol set distributions across the cross-validation and test sets. Note that individual folds within the cross-validation set were primarily balanced by the number of images, and a balanced distribution regarding other metadata fields therefore cannot be guaranteed.

3.2 Model Selection

Four different versions of the deep neural network architecture "YOLO" (stemming from the phrase "You only look once") (Redmon et al., 2016) were selected for evaluation, primarily motivated by the public availability of pre-trained DLA models.

Concretely, three different sizes (medium, large, extra large) of the YOLO *detection* architecture, version 11, and the extra large implementation of the YOLO *segmentation* architecture, version 8, were chosen. Both the medium-sized detection (Brunello, 2025), and the segmentation model (Yoann Schneider, 2024) were pre-trained on DocLayNet (Pfitzmann et al., 2022), adhering to the dataset's standard labels. The large and extra large versions of YOLO v11 (Mattingly, 2025) were pre-trained on CATMuS (Clérice et al., 2024), employing a subset of the SegmOnto (Gabay et al., 2024) vocabulary. Table 1 summarises the classes, covered by the two datasets, in comparison to ours. Detailed descriptions of each class can be obtained from the respective publications. However, even without these, the names should provide sufficient indications that these datasets differ in domain and only marginally overlap with the tasks investigated in this paper. We chose these pre-trained models despite these discrepancies because they are closer to the domain of cipher keys than for example YOLO models, pre-trained on the widely used COCO dataset (Lin et al., 2014), which consists of images of everyday scenes, containing for example animals and kitchen utensils. Besides this, smaller mismatches between domains and vocabularies reflect the reality of developing models for specialised collections, for which no perfectly fitting pre-trained model exists.

In addition to the pre-trained versions of the aforementioned models, we also evaluate the same architectures when trained from scratch. Table 2 summarises the different models, pre-training conditions and parameter sizes, and specifies the name by which each of them will be referred to for the remainder of this paper. Generally, names consisting only of a version and size refer to the models trained from scratch, while those containing a dataset name refer to their pre-trained, respectively fine-tuned, counterparts.

3.3 Evaluation

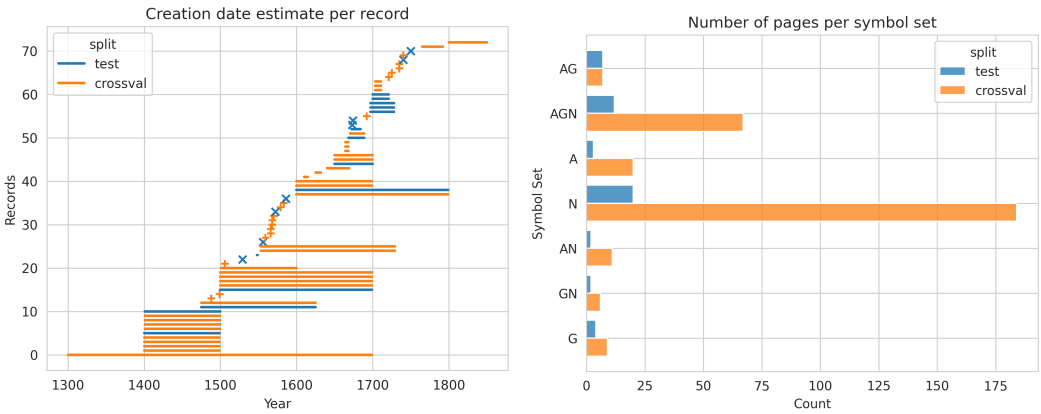
Unless otherwise stated, all models are trained and evaluated on all previously introduced classes, except for "other", which is excluded, due to its volatile nature. Prediction performance is reported as mean average precision, averaged over intersection over union thresholds, ranging from 50% to 95%, with a step-size of 5%. We use the standard implementation provided by the Ultralytics framework (Jocher et al., 2023b). Where applicable, test performances are summarised across the five fold-based models and standard deviations are reported in parentheses.

3.4 Experiments

Overall, we have constructed three major and two minor experiments to investigate the aforementioned research questions. Following the description of the general experiment protocol, each experiment is briefly introduced below.

3.4.1 General Experiment Protocol

All experiments are based on the Ultralytics YOLO implementations (Jocher et al., 2023a; Jocher and Qiu, 2024) and follow their default



(a) Lines indicate ranges of creation date estimates, e.g. the top-most record is estimated to have been created between 1800 and 1850 CE. Symbols (+ and x) mark records for which concrete years of creation have been determined.

(b) A = alphabet, G = graphic signs, N = numerical; combinations of letters indicate combinations of symbols sets, e.g. AGN = alphabet + graphic signs + numerical symbols

Figure 3: Creation date estimate per record (a) and distribution of symbol sets (b)

Table 1: Annotation classes used by the respective dataset, sorted by their dataset-specific IDs.

Dataset	Classes
Cipher Keys	Alphabet Key, Nomenclature Key, Figure, Table, Text, Stamp, Watermark, Other, Key Row, Key Column, Page, Fragment, Operational Element
DocLayNet (Pfitzmann et al., 2022)	Caption, Footnote, Formula, List-Item, Page-footer, Page-header, Picture, Section-header, Table, Text, Title
CATMuS (Clérice et al., 2024)	MarginTextZone, DefaultLine, MainZone, RunningTitleZone, NumberingZone, QuireMarksZone, HeadingLine, DropCapitalZone, StampZone, GraphicZone, InterlinearLine, DigitizationArtefactZone, DropCapitalLine, DamageZone, MusicLine, TitlePageZone, SealZone, MusicZone

parameters for learning rate, optimiser selection, etc., including the use of RandAugment (Cubuk et al., 2020) for augmentations. Modifications to the standard protocol were made regarding the batch size (8), to adapt to available GPUs (NVIDIA Tesla T4, NVIDIA Tesla A40). Training was terminated with a patience of 50 epochs, or when the wall time exceeded four hours, whichever condition was reached first. We follow the pre-trained models' defaults regarding input image sizes, i.e. 640px for 11l and 11x-based models, 1024px for 8x, and 1280px for 11m. Due to memory constraints, evaluations are limited to 200 detections per image.

3.4.2 Experiment 1: Training Models from Scratch

In a first instance, each of the selected models is trained from scratch, using each of the five cross-validation folds, i.e. yielding five distinct checkpoints. Each model is evaluated on the test set and the averaged performance is reported.

3.4.3 Experiment 2: Class-agnostic Evaluation of Pre-trained Models

Each of the four pre-trained models is evaluated on the test set, while disregarding class labels, i.e. assuming all detected and ground truth regions are of the same class. Class labels are discarded in this step, as the pre-trained models employ vocabularies that are distinctly different from that in our data and no meaningful mapping across all relevant categories could be established. As reference, we also evaluate the models from the previous experiment, i.e. those trained from scratch, under the same class-agnostic conditions.

3.4.4 Experiment 3: Fine-tuning of Pre-Trained Models

The four pre-trained models from experiment 1 are fine-tuned on the cross-validation data, following the aforementioned protocol. The obtained mod-

Table 2: Summary of models, considered in this work.

Name	Data Foundation	Architecture	Param Count
YOLO11m	Cipher Keys	YOLO11m	20M
doclaynet-11m	DocLayNet		
cipher-11m	DocLayNet → Cipher Keys		
YOLO11l	Cipher Keys	YOLO11l	25M
catmus-11l	CATMuS		
cipher-11l	CATMuS → Cipher Keys		
YOLO11x	Cipher Keys	YOLO11x	56M
catmus-11x	CATMuS		
cipher-11x	CATMuS → Cipher Keys		
YOLO8x-seg	Cipher Keys	YOLO8x-seg	71M
doclaynet-8x-seg	DocLayNet		
cipher-8x-seg	DocLayNet → Cipher Keys		

els are evaluated both in the class-based setting, for comparison with the results from the initial experiment, and in a class-agnostic fashion, in relation to experiment 2.

3.4.5 Experiment 4: Cipher Key-specific Modifications

The impact of the following modifications is examined, to determine whether a closer focus on selected classes can improve the prediction performance for these regions of interest:

1. limiting the fine-tuning to alphabet and nomenclature regions;

2. limiting the fine-tuning to classes of immediate relevance to cipher keys analysis, i.e. alphabet and nomenclature keys, cipher key rows and columns, operational elements, as well as the general page extent;

The outlined modifications are applied both to the cross-validation sets, and during evaluation, to the test set.

Table 3: Prediction performance across the four models, trained from scratch.

Model	mAP50-95
YOLO11m	0.1921 (± 0.11)
YOLO11l	0.1050 (± 0.14)
YOLO11x	0.1388 (± 0.13)
YOLO8x-seg	0.2046 (± 0.02)

4 Results and Discussion

4.1 Experiment 1: Training Models From Scratch

Table 3 summarises the performance of the four architectures, trained from scratch. All models exhibit low prediction performances, with mAPs below 0.21. As indicated by the comparably high standard deviations, all three YOLO11 architectures displayed stability issues during training, with about one third of the configurations failing to converge. Despite outperforming all other models in this experiment, and exhibiting much more stable training behaviour, the performances of the YOLO8 models remain low. Both observations, i.e. unstable training and low overall performance, can be explained by the limited amount of training data, which is insufficient to train the examined DLA models from scratch.

4.2 Experiment 2: Class-agnostic Evaluation of Pre-trained Models

The results for the pre-trained models, shown in Table 4, indicate that neither of the two DLA domains are directly transferable to cipher key regions. However, the two CATMuS-based models slightly outperform the ones pre-trained on DocLayNet, which may stem from some overlap in appearance between the medieval documents and the cipher key images, some of which also originate from the Middle Ages.

The models, trained from scratch, consistently outperform the pre-trained models, demonstrating some adaptation to the cipher key domain, despite the overall low class-based performance, as dis-

cussed in the previous section.

Table 4: Class-agnostic prediction performance of pre-trained models and their counterparts, trained from scratch.

Model	mAP50-95
doclaynet-11m	0.0672
catmus-11l	0.0705
catmus-11x	0.0822
doclaynet-8x-seg	0.0499
YOLO11m	0.3881 (± 0.21)
YOLO11l	0.1654 (± 0.22)
YOLO11x	0.2147 (± 0.19)
YOLO8x-seg	0.3987 (± 0.01)

4.3 Experiment 3: Fine-tuning of Pre-trained Models

Table 5 and Table 6 summarise the prediction performances of the fine-tuned models. As can be seen when comparing the results with those from the two previous experiments, all fine-tuned models consistently outperform their counterparts by considerable margins, across both evaluation modalities. These results highlight the efficacy of fine-tuning and the resulting successful domain transfer, both from PDFs and medieval manuscripts. In contrast to the models, trained from scratch, the fine-tuning process yields stable results across all five folds, with no further convergence issues being observed.

Considering the class-level performances, it can be summarised that nomenclature keys and key rows consistently achieve higher prediction performances (mAP50-95 of 0.53-0.66, respectively 0.57-0.65) than their counterparts, alphabet keys and key columns (mAP50-95 of 0.36-0.41, respectively 0.42-0.55). A straightforward explanation for this can be found in the imbalanced distribution within the two pairs, with the former appearing four to six times more frequently than the respective latter classes.

Regarding classes with low mAP scores, figures and operational elements stand out, with performances below 0.02. Overall, these results can be explained by the low number of occurrences. However, for the latter class, its high level of visual variation may be a contributing factor. These variations stem from the underlying annotation scheme. *Operational elements* summarise several sub-categories, such as nullities and duplication signs, which may appear in several different forms, for example in the shape of tables, i.e. similar to alphabet/nomenclature keys, or as running text. In order to properly handle these diversities, a different approach will have to be found, such as explicitly defining and using the respective subcategories. However, a closer investigation of this is beyond the scope of this work.

Table 5: Prediction performance of fine-tuned models, summarised across all classes.

Model	mAP50-95
cipher-11m	0.4114 (± 0.03)
cipher-11l	0.4031 (± 0.05)
cipher-11x	0.4102 (± 0.04)
cipher-8x-seg	0.3757 (± 0.01)

Table 6: Class-agnostic prediction performance of fine-tuned models.

Model	mAP50-95
cipher-11m	0.5884 (± 0.02)
cipher-11l	0.5157 (± 0.03)
cipher-11x	0.5440 (± 0.02)
cipher-8x-seg	0.5500 (± 0.02)

Table 7: Prediction performance (mAP50-95) for the models, fine-tuned only on alphabet and nomenclature key regions (*limited*), and their counterparts, trained on all regions, with the evaluation limited to alphabet and nomenclature key regions (*original*).

Model	Limited	Original
cipher-11m	0.4851 (± 0.02)	0.4849 (± 0.02)
cipher-11l	0.4095 (± 0.06)	0.4704 (± 0.05)
cipher-11x	0.4095 (± 0.04)	0.4611 (± 0.03)
cipher-8x-seg	0.5525 (± 0.03)	0.5385 (± 0.02)

4.4 Experiment 4: Cipher Key-specific Modifications

The following subsections briefly summarise the results and analyses for several smaller experiments, all pertaining to cipher key-specific modifications, i.e. disregarding classes that are not of relevance to (parts of) the cipher key analysis, such as watermarks and stamps.

4.4.1 Limiting Fine-tuning to Alphabet and Nomenclature Key Regions

Table 7 presents the performance differences between models, trained only on alphabet and nomenclature key regions, and those trained on all available labels. Both original cipher-11l and cipher-11x architectures considerably outperform their limited counterparts. For the two models, based on DocLayNet (cipher-11m and cipher-8x-seg), limiting the training data did improve the overall performance, albeit by a very modest margin. Given that other classes in the dataset are also of relevance for downstream analyses, the performance gain is not large enough to justify the limited training approach. It may, however, be of interest as part of a pipeline that approaches the segmentation hierarchically, i.e. identifying alphabet/nomenclature regions before applying a secondary model for the subsegmentation of rows/columns.

4.4.2 Limiting Fine-tuning to Cipher Key-relevant Classes

Table 8 summarises the performance differences between models, trained only on the classes "alphabet key" and "nomenclature key", "key row" and "key column", "operational element" and "page", and their counterparts trained on all labels but evaluated only on the selected class-subset. The prediction performances differ only marginally, which can likely be explained by the fact that the selected classes, with the exception of operational elements, are well-represented in this focused cipher key dataset, and are therefore expected to perform well, regardless. While the excluded classes may not have immediately obvious downstream use cases, their presence clearly does not hinder the training. It is therefore not necessary to exclude them, and they can be maintained to allow for a more complete view of the data in subsequent processing steps.

5 Qualitative Analysis

In order to complement the extensive quantitative analyses of the previous sections, we present a brief qualitative analysis of the DLA results. All presented annotations were obtained from the best-performing checkpoint of experiment 3.

Figures 4 and 5 exemplify two typical failure cases, in which the DLA models identify non-textual artefacts as regions of interest. This generally pertains to blank areas, page edges, or parts of the digitisation surface (e.g. table surface) that are included in the document image. While the latter two can be easily removed by cropping the image, the former requires adaptations during model training, e.g. by including more diverse samples of watermarks.

Table 8: Prediction performance (mAP50-95) for the models, fine-tuned only on the classes alphabet and nomenclature keys, cipher key rows and columns, operational elements and page (*limited*) and their counterparts, trained on all regions, with the evaluation limited to the same classes (*original*).

Model	Limited	Original
cipher-11m	0.4492 ($\pm$ 0.02)	0.4537 ($\pm$ 0.01)
cipher-11l	0.4589 ($\pm$ 0.03)	0.4497 ($\pm$ 0.03)
cipher-11x	0.4540 ($\pm$ 0.01)	0.4617 ($\pm$ 0.02)
cipher-8x-seg	0.4857 ($\pm$ 0.01)	0.4834 ($\pm$ 0.01)

Finally, Figure 6 shows a successful example of DLA. While minor artefacts remain, alphabet and nomenclature regions, as well as the contained key columns, respectively rows, are generally correctly identified and delineated.

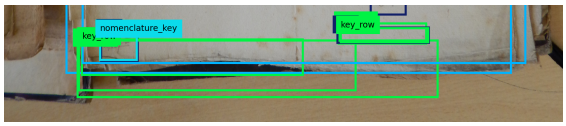


Figure 4: Example for edge artefacts mistakenly being identified as regions of interest.

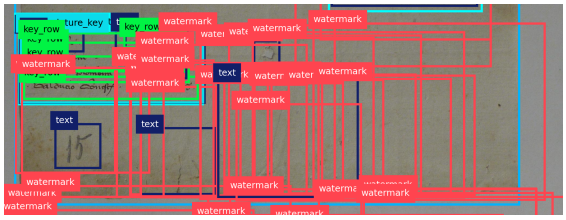


Figure 5: Example of overexpression of watermark regions (red) in empty areas.

6 Conclusion

In this work, we have studied four YOLO-based models in order to establish a baseline for DLA of cipher keys. We have demonstrated that:

1. a small dataset of 240 training images (per

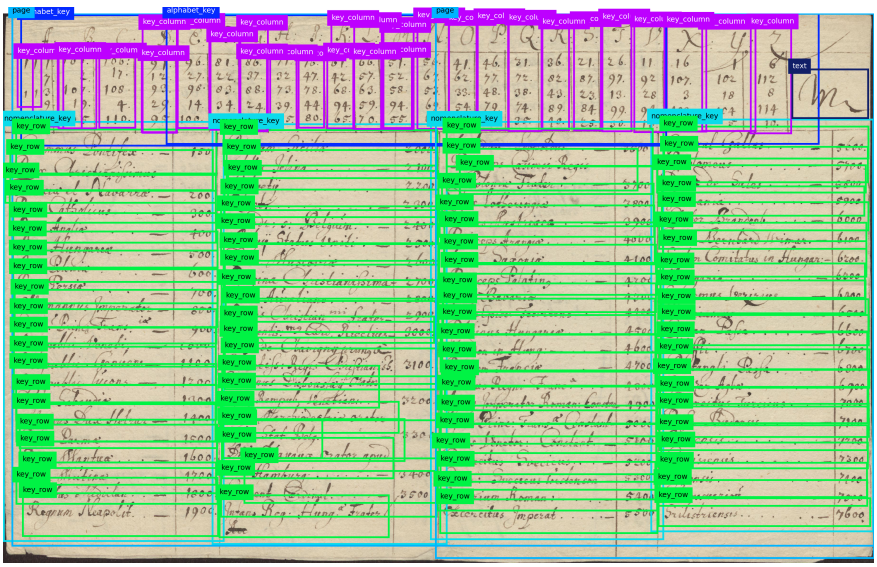


Figure 6: Sample output of the best-performing model from experiment 3. Ground truth shown in Figure 2.

fold) is not sufficient to train the selected architectures from scratch.

2. models, pre-trained on other DLA domains, are not directly applicable to our dataset, even when used in a class-agnostic fashion.

3. considerable improvements can be obtained by fine-tuning pre-trained DLA models, adapting them to the cipher key domain. Improvements are obtained consistently, across all pre-trained configurations, despite a small training set, and a considerable domain gap, between the pre-training and fine-tuning datasets.

4. the two examined training modifications did not yield sufficiently large or consistent improvements. Overall, neither of them can therefore be recommended for implementation.

Overall, the evaluated models only achieved a moderate mean average precision on our newly introduced cipher key dataset. Besides the small amount of data, the difference in performance, compared to state-of-the-art DLA models, can also be explained by the kinds of information that we are looking to extract from cipher key images. In contrast to conventional DLA, which focuses on physical aspects of the layout, our cipher key annotation scheme requires a certain level of semantic analysis.

Even though the presented work is limited to YOLO-based models, it presents a first baseline for DLA of cipher keys. Future work should expand these investigations to other model architectures. In addition to this, an extension of the annotations, to more images, either genuine or through the creation of synthetic samples, may be of interest. As an alternative to increasing the dataset size, few-shot approaches could be explored.

Data Availability

Due to image copyright constraints, the dataset cannot be shared publicly at the time of writing. All trained models are available in the following repository: 10.5281/zenodo.19911174.

Acknowledgements

This work has been supported by Riksbankens Jubileumsfond, grant M24-0028: Echoes of History: Analysis and Decipherment of Historical Writings (DESCRYPT). The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725. Alicia Fornés acknowledges financial support for her general research activities from ICREA under the ICREA Academia (Departament de Recerca i Universitats de la Generalitat de Catalunya), and also, she has partial support from the Spanish project PID2024-157778OB-I00

(SUKIDI) from the Ministerio de Ciencia e Innovación, the Departament de Cultura of the Generalitat de Catalunya, and the CERCA Program / Generalitat de Catalunya. We gratefully acknowledge Adelaida López for her help in annotating the data.

References

Galal M. Binmakhashen and Sabri A. Mahmoud. 2019. Document Layout Analysis: A Comprehensive Survey. *ACM Comput. Surv.*, 52(6), October.

Mélodie Boillet, Marie-Laurence Bonhomme, Dominique Stutzmann, and Christopher Kermorvant. 2019. HORAE: an annotated dataset of books of hours. In *Proceedings of the 5th International Workshop on Historical Document Imaging and Processing*, HIP '19, pages 7–12. Association for Computing Machinery.

Alessandro Brunello. 2025. Armaggheddon/yolo11-document-layout (Hugging Face). `https://huggingface.co/Armaggheddon/yolo11-document-layout`, version 0a03ab4.

Thibault Clérice, Ariane Pinche, Malamatenia Vlachou-Efstathiou, Alix Chagué, Jean-Baptiste Camps, Matthias Gille Levenson, Olivier Brisville-Fertin, Federico Boschetti, Franz Fischer, Michael Gervers, Agnès Boutreux, Avery Manton, Simon Gabay, Patricia O'Connor, Wouter Haverals, Mike Kestemont, Caroline Vandyck, and Benjamin Kiessling. 2024. CATMuS Medieval: A Multilingual Large-Scale Cross-Century Dataset in Latin Script for Handwritten Text Recognition and Beyond. In *Document Analysis and Recognition - ICDAR 2024*, pages 174–194. Springer Nature Switzerland.

Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. 2020. Randaugment: Practical Automated Data Augmentation With a Reduced Search Space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June.

Simon Gabay, Ariane Pinche, Kelly Christensen, and Jean-Baptiste Camps. 2024. SegmOnto: A Controlled Vocabulary to Describe and Process Digital Facsimiles. *Journal of Data Mining & Digital Humanities*, Dec.

Mihály Héder and Beáta Megyesi. 2022. The DECODE Database of Historical Ciphers and Keys: Version 2. In *Proceedings of the 5th International Conference on Historical Cryptology, HistoCrypt 2022*, pages 111–114, Linköping University Electronic Press.

Despoina Ioakeimidou, Stavros N. Moutsis, Konstantinos Evangelidis, Konstantinos A. Tsintotas, Panagiotis E. Nastou, Elpida Perdiki, Emmanouil Gkinidis, Nikos Tsoukatos, Antonis Tsolomitis, Maria Konstantinidou, and Stamatios Busses. 2024. Cyril's Lexicon Layout Analysis Through Deep Learning. In *2024 IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 1–6.

Glenn Jocher and Jing Qiu. 2024. Ultralytics YOLO11. `https://github.com/ultralytics/ultralytics`, version 11.0.0.

Glenn Jocher, Ayush Chaurasia, and Jing Qiu. 2023a. Ultralytics YOLOv8. `https://github.com/ultralytics/ultralytics`, version 8.0.0.

Glenn Jocher, Jing Qiu, and Ayush Chaurasia. 2023b. Ultralytics YOLO, January. `https://github.com/ultralytics/ultralytics`.

Benjamin Kiessling. 2026. Version 5 of the Kraken ATR Engine for the Humanities. In *Document Analysis and Recognition – ICDAR 2025*, pages 443–458. Springer Nature Switzerland.

Stefan Klut, Rutger van Koert, and Ronald Sluijter. 2023. Laypa: A Novel Framework for Applying Segmentation Networks to Historical Documents. In *Proceedings of the 7th International Workshop on Historical Document Imaging and Processing*, HIP '23, pages 67–72. Association for Computing Machinery.

Dong-Lin Li, Shih-Kai Lee, and Yin-Ting Liu. 2025. Printed document layout analysis and optical character recognition system based on deep learning. *Scientific Reports*, 15(1):23761, Jul.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *Computer Vision – ECCV 2014*, pages 740–755. Springer International Publishing.

Benedek Láng, Beáta Megyesi, Nils Kopal, Vasily Mikhalev, Crina Tudor, and Michelle Waldispühl. 2025. Cipher key instructions in early modern Europe: analysis and text edition. *Cryptologia*, 49(5):416–442.

William Mattingly. 2025. biglam/medieval-manuscript-yolov11 (Hugging Face). `https://huggingface.co/biglam/medieval-manuscript-yolov11`, version 8f5eddd.

Beáta Megyesi, Nils Blomqvist, and Eva Pettersson. 2019. The DECODE Database: Collection of Historical Ciphers and Keys. In *Proceedings of the 2nd International Conference on Historical Cryptology, HistoCrypt 2019*, pages 69–78, Linköping University Electronic Press.

Beáta Megyesi, Crina Tudor, Benedek Láng, Anna Lehofer, Nils Kopal, Karl de Leeuw, and Michelle Waldispühl. 2024. Keys with nomenclatures in the early modern europe. *Cryptologia*, 48(2):97–139.

Vasily Mikhalev, Nils Kopal, Bernhard Esslinger, Michelle Waldispühl, Benedek Láng, and Beáta Megyesi. 2023. What is the Code for the Code? Historical Cryptology Terminology. In *Proceedings of the 6th International Conference on Historical Cryptology, HistoCrypt 2023*, pages 130–138, Linköping University Electronic Press.

Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S. Nassar, and Peter Staar. 2022. DocLayNet: A Large Human-Annotated Dataset for Document-Layout Segmentation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, pages 3743–3751. Association for Computing Machinery.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June.

Foteini Simistira, Mathias Seuret, Nicole Eichenberger, Angelika Garz, Marcus Liwicki, and Rolf Ingold. 2016. DIVA-HisDB: A Precisely Annotated Large Dataset of Challenging Medieval Manuscripts. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 471–476.

Crina Tudor, Beáta Megyesi, and Benedek Láng. 2020. Automatic Key Structure Extraction. In *Proceedings of the 3rd International Conference on Historical Cryptology, HistoCrypt20*, pages 146–152, Linköping University Electronic Press.

Rutger van Koert, Stefan Klut, Tim Koornstra, Martijn Maas, and Luke Peters. 2024. Loghi: An End-to-End Framework for Making Historical Documents Machine-Readable. In *Document Analysis and Recognition – ICDAR 2024 Workshops*, pages 73–88. Springer Nature Switzerland.

Fei Wu, Mathias Seuret, Martin Mayr, Florian Kordon, Jochen Zöllner, Sebastian Wind, Andreas Maier, and Vincent Christlein. 2025. Lightweight cross-attention-based HookNet for historical handwritten document layout analysis. *International Journal on Document Analysis and Recognition (IJDAR)*, 28(3):409–427, Sep.

TEKLIA Yoann Schneider. 2024. yolo-v8-segmenter/DocLayNet. `https://gitlab.teklia.com/dla/models/-/tree/master/yolo-v8-segmenter/DocLayNet`, version 5d3d0478.

Silvia Zottin, Axel De Nardin, Gian Luca Foresti, Emanuela Colombi, and Claudio Piciarelli. 2024. ICDAR 2024 Competition on Few-Shot and Many-Shot Layout Segmentation of Ancient Manuscripts (SAM). In *Document Analysis and Recognition - ICDAR 2024 Proceedings, Part VI*, pages 315–331. Springer-Verlag.

Unsupervised Feature Learning via Convolutional Autoencoders for Cross-Manuscript Comparison in Historical Cryptanalysis

Alejandra Reinares,
Giuseppe De Gregorio,
Alicia Fornés
Computer Vision Center
Department of Computer Science
Universitat Autònoma de Barcelona, Spain
alejandrareigue2004@gmail.com
{gdegregorio, afornes}@cvc.uab.cat

Abstract

The study of historical enciphered manuscripts is fundamental to understanding our cultural heritage, yet a vast corpus of these archives remains inaccessible due to the complexity of ancient cryptographic systems. Traditional analysis relies heavily on manual expertise, a process that is labor-intensive and difficult to scale across the immense volume of unstudied documents. This paper proposes a novel, fully unsupervised framework for the automated comparative analysis of images of historical ciphers. Our approach leverages Convolutional Autoencoders (CAE) to learn intrinsic morphological features directly from manuscript images, bypassing the need for labeled datasets or prior knowledge of the cipher keys. By projecting symbols into a high-dimensional latent space, the system generates a "similarity fingerprint" for each manuscript, enabling a quantitative comparison of diverse documents. Experimental results demonstrate that this method effectively identifies relationships between ciphers, grouping them by cryptographic tradition. This framework provides historians with a powerful computational tool to detect shared lineages and map the evolution of secret communication across history.

1 Introduction

Preservation and analysis of historical handwritten manuscripts are essential for the study of cultural heritage. However, a vast amount of these archives remain unreachable due to the use of sophisticated encryption systems. For historians and linguists, the enciphered nature of these documents creates a significant barrier, making primary sources inaccessible for traditional analysis.

The challenge of deciphering encrypted documents is highly intriguing for historians and linguists, but before a document can be decrypted, it must identify its underlying cipher system and its potential relationship with other known cryptographic traditions.

Traditionally, this preliminary identification has relied on the manual expertise of paleographers and cryptologists. While effective, this manual process is labor-intensive, time-consuming, and difficult to scale against the immense volume of unstudied material. Furthermore, many historical ciphers utilize unique, non-standard symbolic alphabets, which limit the applicability of standard Optical Character Recognition (OCR) tools.

To address these limitations, this paper proposes a fully unsupervised framework for automated comparison of historical ciphers. Unlike supervised deep learning models, which require large-scale labeled datasets, our approach leverages Convolutional Autoencoders (CAE) for neural feature extraction (Masci et al., 2011; Wickramasinghe et al., 2021). By training an autoencoder to reconstruct cipher symbols, we extract high-dimensional morphological features from the latent space of the network. These features allow for an objective comparison between different manuscripts, regardless of their language or the specific key used.

The primary contribution of this work is to propose a framework for producing a cross-manuscript relationship mapping. By computing similarity metrics between the feature sets of diverse documents, our system can identify similar

cryptographic systems. This provides historians with a powerful computational tool to detect "families" of ciphers and trace the evolution of cryptographic practices across different regions and periods. This approach enhances historians' computational toolkit for analyzing encrypted archives at scale, overcoming previous barriers imposed by manual analysis constraints and the idiosyncrasy of symbol sets (Chen et al., 2015; Wickramasinghe et al., 2021).

2 Related Work

Historical ciphered manuscripts represent a significant blind spot in our understanding of history. These documents, used for diplomatic, military, and secret society communications, were protected by invented symbolic alphabets that mixed digits, Latin or Greek letters, graphical signs like alchemical or zodiac symbols, and even invented symbols. The primary bottleneck in the study of encrypted documents is the identification of the underlying cipher alphabet, a process that has traditionally been manual, time-consuming, and requires specialized expertise.

The emergence of large-scale digital initiatives has provided the necessary foundation for computational analysis. The DECRYPT project (Megyesi et al., 2020) established a cross-disciplinary infrastructure to centralize scattered archival materials and provide a standardized set of state-of-the-art tools for their analysis. As these digital efforts have expanded, the field has increasingly turned to computer vision to overcome the problem. However, the unique nature of the documents, each with its own custom alphabet and a lack of large amounts of training data, makes supervised Deep Learning methods unsuitable. This has driven research towards unsupervised methods that can analyze manuscript images directly without prior knowledge. Recent advances in unsupervised representation learning, including Variational Autoencoders (Kingma and Welling, 2014), contrastive learning methods such as SimCLR (Chen et al., 2020), and self-supervised vision transformers such as DINO (Caron et al., 2021), have demonstrated strong feature extraction capabilities across a wide range of visual domains.

More related to ciphers, early work, such as Baro et al. (2019), focused on the challenge of transcribing a single cipher. Their pipeline involved segmenting symbols, clustering them based on visual features like SIFT descriptors, and then using label propagation to assign a consistent label to all instances of the same symbol. This proved that unsupervised transcription was feasible.

The field has recently transitioned from traditional computer vision to Deep Learning to handle the high variability of handwritten scripts. Yin et al. (2019) developed a framework that segments ciphered manuscripts into isolated symbols, utilizes a pre-trained Siamese Neural Network (SNN) for feature extraction, and groups them using a Gaussian Mixture Model (GMM). While this approach represents a significant step toward automated conversion of images to plaintext, its performance was often hindered by the sensitivity of the feature extractor to segmentation quality. A notable performance gap was observed between manually and automatically segmented symbols, suggesting that traditional SNNs may struggle to maintain robust representations in the presence of the noise and fragmentation typical of historical manuscripts.

Souibgui et al. (2021) introduced few-shot learning architectures, which require only a small handful of training examples, even on unseen data from unfamiliar manuscripts. The method demonstrated that models could generalize to unseen manuscripts using only a minimal support set of labeled examples. While effective, these methods still depend on a degree of supervision or pre-existing labeled data from similar domains.

The most relevant area to this study is the automated comparison of cipher writing systems. Méndez et al. (2024) proposed a Cipher Similarity Index (CSI), which utilizes a mutual-kNN graph and entropy-based partitioning. The core of the CSI is the analysis of how the entropy of the clusters changes during this process: documents with similar alphabets resist separation, maintaining high entropy for longer, which results in a higher similarity score. Their method provides a more stable measure of similarity by analyzing how clusters resist separation during iterative partitioning.

Chen et al. (2021) introduced an unsupervised method for alphabet matching by jointly clustering symbols from two different ciphers. They proposed a similarity metric based on the ratio of "mixed clusters" containing symbols from both ci-

phers. While groundbreaking, this approach was highly sensitive to handwriting styles and clustering parameters, often resulting in flattened similarity scores that struggled to distinguish between subtle stylistic and structural differences.

3 Methodology

The proposed framework follows a multi-stage pipeline divided into four main phases: (i) Image Preprocessing and Segmentation, (ii) Neural Feature Extraction via Convolutional Autoencoders, (iii) Unsupervised Symbol Clustering, and (iv) Cross-Manuscript Similarity Analysis.

3.1 Preprocessing and Symbol Segmentation

The transition from a raw manuscript page to individual glyphs is a critical phase, as any error in segmentation directly propagates to the feature extraction stage. Historical manuscripts often suffer from physical degradation and irregular writing alignment. To standardize the input, the following steps are performed:

- **Binarization**: Following a qualitative evaluation of several local and global thresholding methods — including Otsu, Gaussian, Adaptive, Niblack, and Sauvola — Sauvola binarization (Sauvola and Pietikäinen, 2000) was selected as the most effective method across all cipher collections. It was applied with a window size of $w = 125$ pixels and $k = 0.2$ uniformly across all cipher collections.
- **Skew Correction**: We estimate and rectify the global rotation of the document to ensure that text lines are horizontally aligned, which is essential for accurate vertical projection.
- **Morphological Cleaning**: Binarized images are processed using morphological smoothing to reduce salt-and-pepper noise and bridge fragmented strokes within individual characters.
- **Line Segmentation**: Connected Component Analysis was employed to identify blobs corresponding to individual characters or symbols. The resulting bounding boxes were subsequently grouped according to their vertical proximity and alignment, enabling the reconstruction of text lines.
- **Line Noise Filtering**: Unlike standard approaches, we employ a modified filtering technique inspired by Jindal et al. (2023). We estimate the middle zone of each text line through linear regression. This central axis serves as a reference to filter out artifacts:
  1. Adjacent-line Noise: Components touching the image borders that do not intersect the middle zone are discarded.
  2. Granular Noise: Components below a cipher-specific area threshold are eliminated.

  This line noise filtering ensures that the final line image contains only the core symbols, as illustrated by the comparison between the raw and cleaned versions in Figures 1 and 2.

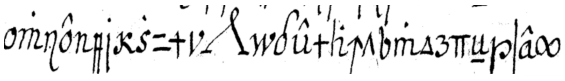


Figure 1: Original binarized line from the Copiale cipher.

Figure 2: Result of the filtering process using linear regression to isolate the middle zone and remove adjacent-line artifacts

3.1.1 Character Segmentation and Diacritic Merging

The final stage of the preprocessing involves isolating individual symbols using Connected Component Analysis. Given the complexity of handwritten ciphers, a simple bounding-box approach is insufficient.

- Contextual Merging: Historical scripts often feature composite glyphs or detached elements (e.g., accents or diacritics). Our algorithm classifies components into "main symbols" and "small components."
- Vertical Re-association: Small components are contextually re-merged with their nearest vertically-aligned main symbol. This prevents the system from treating a single composite character as multiple distinct entities, ensuring that the input to the Convolutional Autoencoder is a complete, semantically meaningful glyph.

Figure 3: Example of Character Segmentation of an image line of the Copiale cipher

3.2 Feature Extraction via Convolutional Autoencoder

The core of our methodology lies in the unsupervised learning of morphological features. We implement a Convolutional Autoencoder (CAE) to learn a representation specific to historical ciphered scripts. The choice of a CAE over a standard pre-trained CNN is motivated by the need to learn features that are strictly intrinsic to historical scripts, avoiding the semantic bias of models trained on natural images (Yosinski et al., 2014). We acknowledge that more recent unsupervised representation learning methods, such as contrastive learning and self-supervised vision transformers, have demonstrated strong performance in general visual domains. However, the CAE was selected for this task for three concrete reasons. First, the **domain shift** between natural images and historical cipher symbols is substantial: models such as DINO and SimCLR are pretrained on large corpora of natural photographs, and their learned representations encode texture, color, and semantic content that are entirely absent in manuscript images. While fine-tuning could partially mitigate this, it would require labeled data that is precisely what this unsupervised framework is designed to avoid. Second, the **extremely small dataset size** is poorly suited to the large-batch contrastive objectives that methods like SimCLR rely on for stable training; a lightweight CAE trained from scratch converges reliably under these constraints. Third, the **interpretability** of the reconstruction objective provides a direct and verifiable signal that the learned features encode symbol morphology: if the CAE reconstructs the symbol faithfully, the latent vector necessarily encodes its shape. The exploration of more powerful self-supervised alternatives, and a systematic comparison of their performance under the data constraints of historical cryptanalysis, is an important direction for future work.

Figure 4 shows the CAE architecture. The encoder compresses a grayscale input image of size 1×100×100 pixels into a compact latent representation through three strided convolutional stages. The first stage applies a Conv2d layer with 16 filters, kernel size 3×3, stride 2, and padding 1, reducing the spatial dimensions to 50×50. The second stage applies a Conv2d layer with 32 filters and the same parameters, yielding a 25×25 feature map. The third stage applies a Conv2d layer with 64 filters, producing a 13×13 feature map. Each convolutional layer is followed by a ReLU activation. The use of strided convolutions instead of pooling operations preserves spatial information during downsampling (Springenberg et al., 2014). The resulting 64×13×13 tensor is flattened and passed through a fully connected linear layer followed by ReLU, producing a 64-dimensional latent vector. This compact representation serves as the morphological signature of the symbol. The decoder mirrors the encoder symmetrically. A fully connected layer projects the 64-dimensional vector back to a 64×13×13 tensor, which is then progressively upsampled through three transposed convolutional layers with 64, 32, and 16 filters respectively, each followed by ReLU activations, until the original 1×100×100 dimensions are recovered. A Sigmoid activation is applied at the final layer to constrain the output pixel values to the range [0, 1], consistent with the normalized binary input images. The network is trained end-to-end by minimizing the Mean Squared Error (MSE) between the original input image x and its reconstruction $\hat{x}$:

$$\mathscr{L}_{rec}(x,\hat{x}) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2$$

Training is performed jointly on the symbols of the two manuscripts being compared, using the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 1×10^{-3}, a batch size of 32, and for 100 epochs. To ensure balanced representation, when the two symbol sets differ in size, the larger set is randomly subsampled to match the size of the smaller one prior to training. By forcing the model to reconstruct each symbol faithfully, the bottleneck is encouraged to encode all information relevant to shape, stroke thickness, and topology, while discarding noise. Crucially, because the CAE is trained from scratch on each manuscript pair, it learns features that are specific to the visual characteristics of the scripts under analysis, rather than inheriting biases from large natural-image datasets.

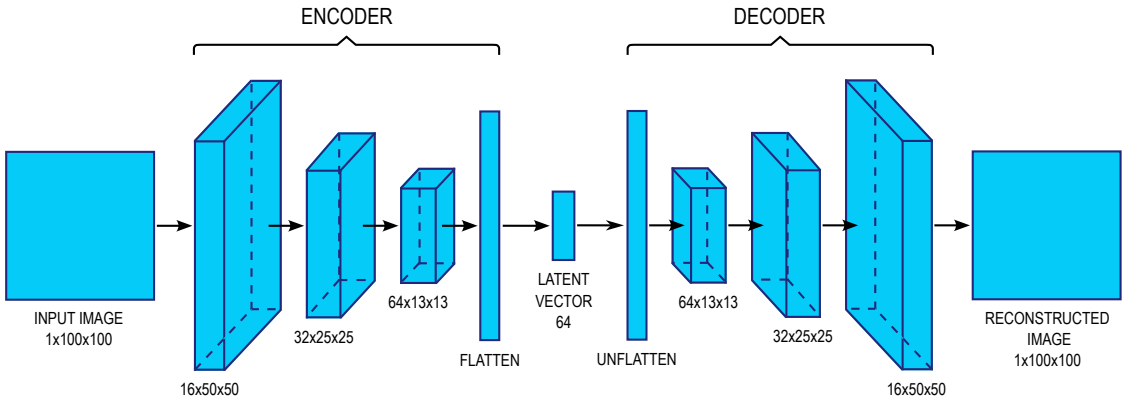


Figure 4: Architecture of the Convolutional Autoencoder used for unsupervised feature extraction. The encoder applies three strided convolutional layers (kernel size 3×3, stride 2, padding 1) with 16, 32, and 64 filters respectively, each followed by ReLU activation, progressively reducing the spatial dimensions from 1×100×100 to 64×13×13. The resulting tensor is flattened and projected via a fully connected layer into a 64-dimensional latent vector. The decoder mirrors this process using transposed convolutions to reconstruct the original 1×100×100 image, with a Sigmoid activation at the output constraining pixel values to [0, 1]. The network is trained end-to-end by minimizing the Mean Squared Error between input and reconstruction.

3.3 Unsupervised Clustering and Symbol Discovery

Once the CAE is trained, the 64-dimensional latent vectors are extracted for every segmented symbol. To organize these features into meaningful character classes without prior knowledge of the alphabet size, we employ Agglomerative Hierarchical Clustering (AHC) (Murtagh and Contreras, 2012).

3.3.1 Hierarchical Grouping and Threshold Optimization

Unlike partitioned methods such as K-means, AHC adopts a bottom-up approach that does not require a predefined number of clusters (k). We utilize Ward's linkage criterion (Ward, 1963), which minimizes the total within-cluster variance (Within-Cluster Sum of Squares). At each iteration, the algorithm merges the two clusters that result in the smallest increase in variance, effectively building a dendrogram that represents the morphological relationships between symbols at varying levels of granularity.

To objectively determine the optimal cutting point for the dendrogram, our framework evaluates a range of distance thresholds using the Silhouette Score (S). This metric quantifies how similar an object is to its own cluster (cohesion) compared to other clusters (separation). For a data point i in cluster C_k, the average intra-cluster distance is defined as:

$$a(i) = \begin{cases} \dfrac{1}{|C_k|-1} \displaystyle\sum_{\substack{j \in C_k \\ j \neq i}} d(i,j), & \text{if } |C_k| > 1 \\ 0, & \text{if } |C_k| = 1 \end{cases}$$

where $d(i,j)$ represents the Euclidean distance. The average nearest-cluster distance is given by:

$$b(i) = \min_{C_j \neq C_k} \left[\frac{1}{|C_j|} \sum_{j \in C_j} d(i,j) \right]$$

The Silhouette coefficient for a single point is then:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

The global Silhouette Score S is the mean of $s(i)$ across all N points in the dataset. By maximizing S, the system automatically identifies the threshold that yields the most naturally defined character classes, avoiding both over-segmentation (fragmenting a single character type) and under-segmentation (merging distinct symbols).

3.3.2 Post-Clustering Quality Control

To ensure the integrity of the discovered alphabet, a dual-criteria filtering mechanism was adopted to categorize the resulting clusters based on their reliability. First, cluster cardinality was used as a reliability indicator: clusters containing fewer

than five elements were considered statistically insufficient and were therefore flagged as noise or rare outliers, as they do not provide adequate evidence to represent a stable character class. In addition, cluster compactness was evaluated by measuring the internal consistency of each cluster through the average distance of its members from the corresponding centroid. Clusters exhibiting high variance were identified as potentially impure, as they are likely to group together visually dissimilar symbols that the convolutional autoencoder was unable to clearly discriminate.

Applying a percentile-based threshold (20%), the system automatically filters the output into three categories:

- Rejected because of insufficient size
- Rejected because of high internal variation
- Accepted (compact, well-defined clusters)

This refinement process ensures that the symbols displayed in Figure 5 represent a high-confidence mapping of the cipher's alphabet, allowing for a reliable comparison between different manuscripts sharing similar visual characteristics.

The clusters displayed in Figure 5 demonstrate how symbols from two different ciphers are grouped when they share similar visual characteristics.

3.4 Cross-Manuscript Similarity Metric

To quantify the relationship between two different manuscripts, hereafter referred to as C_A and C_B, we perform a joint clustering of their extracted symbols. The core intuition is that if two manuscripts share a similar cryptographic system, their symbols will be morphologically indistinguishable to the CAE and will consequently be grouped into the same clusters.

Following the methodology proposed by Chen et al. (2021), we define the Alphabet Similarity Index based on the composition of the resulting clusters. After the hierarchical clustering process, each cluster is analyzed to determine the provenance of its members. We distinguish between two types of groupings:

1. **Homogeneous Clusters**: Groups containing symbols exclusively from either C_A or C_B, suggesting unique character classes.
2. **Heterogeneous (Mixed) Clusters**: Groups where symbols from both manuscripts coexist. These clusters indicate a direct morphological overlap between the character classes of the two sources.

Let C_{mix} be the number of heterogeneous clusters containing at least one symbol from both manuscripts, and C_{total} be the total number of clusters identified. The similarity metric is calculated as follows:

$$\text{Similarity}(\text{C}_A, \text{C}_B) = \frac{C_{\text{mix}} \times 100}{C_{\text{total}}}$$

4 Results

In this section, we evaluate the performance of our unsupervised framework in identifying relationships between different cipher manuscripts. We compare our results with the baseline established by Chen et al. (2021) and analyze the stability of our system under varying confidence thresholds.

4.1 Dataset

The dataset follows the same experimental setup as Chen et al. (2021), allowing for a direct comparison with the baseline. It comprises a diverse collection of historical enciphered manuscripts sourced from the DECODE database (Megyesi et al., 2019): ASV-France (AF), Borg (B), Chiffrenschlüssel (CS), Copiale (C), Ramanacoil (R), Zodiac (Z), and five variants of the Vatican cipher (V1, V2, V3, V6, V7). Each document presents unique challenges in terms of symbology and handwriting style. Table 1 summarizes the entire dataset.

4.2 Comparative Similarity Analysis

To evaluate the framework's ability to detect relationships between different manuscripts, we computed a pairwise similarity matrix across the entire dataset. The results of our Convolutional Autoencoder approach are presented in Table 2, while Table 3 reports the baseline results obtained by Chen et al. (2021).

Our method demonstrates a high discriminative power in identifying manuscripts that belong to the same cryptographic tradition. The highest similarity scores are consistently found among the Vatican ciphers ($V1$ through $V7$). Specifically, our framework achieves a peak similarity of 84.4% between Vatican 3 and Vatican 6, significantly outperforming the baseline's peak of 62.9%.

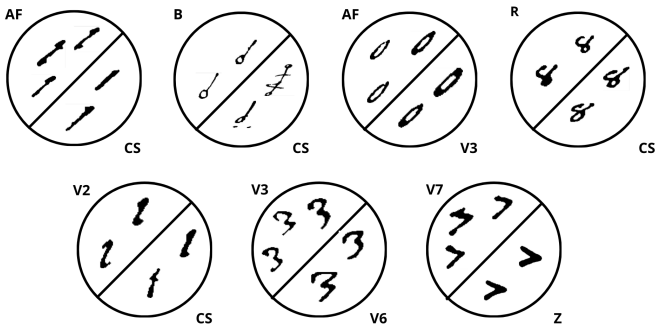


Figure 5: The image illustrates the system's ability to group morphologically similar glyphs extracted from diverse historical sources. Each cluster contains symbols that, despite originating from different manuscripts (identified by capital letters), share a high degree of visual and structural affinity in the 64-dimensional latent space.

Table 1: Dataset overview. DECODE IDs follow the same experimental setup as Chen et al. (2021).

Collection	Record ID	Image IDs
AF	19	110, 111, 112, 114, 115
B	210	1571, 1609, 1625, 1679, 1697
CS	1407	6585, 6586, 6587
C	2	10, 15, 26, 42, 51
R	1564	6975, 6978, 6980, 6985, 6986
V1	5	66, 67, 68
V2	11	76, 77, 78, 79, 80
V3	16	690
	17	100
V6	63	494
	65	506
	67	519
	68	523
V7	183	1176
	186	1187
	187	1193, 1194, 1195
Z	—	z13, z32, z340, z408-1,2,3

A critical observation arises when comparing manuscripts that share the same abstract alphabet but differ in handwriting style. For instance, Vatican 2 and Vatican 7 utilize the same cipher system, yet our method yields a similarity score of only 13.7% (compared to 24.7% in the baseline). A similar phenomenon is observed when comparing ASV-France with the Vatican series despite their alphabetical similarity, the scores remain low.

The results also show that the Zodiac cipher is specially different to the others. In fact, the Zodiac cipher represents a unique case in our dataset, as its symbology is visually distinct from any other historical manuscript analyzed. Our method effectively identifies this outlier: while the baseline provides similarity scores for the Zodiac cipher ranging between 5% and 12%, our system yields significantly lower scores, mostly between 1% and 3%.

4.3 Probabilistic Classification and Affinity Analysis

While pairwise similarity provides a direct measure of morphological affinity, a useful cryptanalytic tool should also allow a researcher to gauge how strongly a manuscript is associated with a particular cipher family. To this end, we transform the raw pairwise similarity scores into a normalized distribution, enabling a clearer ranking of candidate families for each manuscript. To achieve this, we apply a temperature-scaled Softmax function to the similarity scores of each manuscript against all others:

$$\sigma(\mathbf{z}, T)_i = \frac{e^{z_i/T}}{\sum_{j=1}^{K} e^{z_j/T}} \tag{1}$$

where z_i represents the input similarity score for a specific cipher and K is the total number of manuscripts in the comparison set. A temperature of T=0.15 was used, which produces a sharply peaked distribution that amplifies the contrast between the highest-scoring candidate and the remaining ones. The Vatican cipher variants (V1–V7) are treated as a single family in this analysis, all sharing the same script and alphabet, with the maximum pairwise similarity score across all Vatican variants used as the representative score for that group.

The resulting values represent a relative affinity ranking. They indicate how strongly each manuscript is attracted to a given cipher family

Table 2: Percentage of similarity between different pairs of ciphers achieved by the proposed method.

%	B	CS	C	R	V1	V2	V3	V6	V7	Z
AF	2.6	10.1	9.2	29.4	11.5	6.8	28.0	33.8	33.3	2.1
B	—	22.3	11.0	24.2	4.0	8.1	11.7	18.4	48.1	3.3
CS	—	—	4.2	50.0	25.5	3.6	9.9	6.9	4.8	4.5
C	—	—	—	27.2	44.6	18.4	24.7	29.0	21.2	6.7
R	—	—	—	—	20.4	37.1	43.5	35.3	22.9	6.1
V1	—	—	—	—	—	**61.2**	**75.5**	**71.7**	**55.3**	7.4
V2	—	—	—	—	—	—	22.9	**72.6**	13.7	1.0
V3	—	—	—	—	—	—	—	**84.4**	15.7	3.4
V6	—	—	—	—	—	—	—	—	**64.4**	11.0
V7	—	—	—	—	—	—	—	—	—	17.2

Table 3: Percentage of similarity between different pairs of ciphers achieved by Chen et al. (2021).

%	B	CS	C	R	V1	V2	V3	V6	V7	Z
AF	11.0	20.9	05.9	07.7	07.4	11.0	18.6	09.9	07.3	04.5
B	—	21.5	19.1	13.3	14.1	20.2	25.9	23.8	08.7	05.2
CS	—	—	14.7	18.1	17.5	37.0	43.8	35.2	14.9	12.2
C	—	—	—	10.3	21.1	14.6	21.1	20.4	09.4	07.1
R	—	—	—	—	08.9	05.4	08.1	07.6	03.7	08.8
V1	—	—	—	—	—	32.2	39.6	39.0	20.9	06.1
V2	—	—	—	—	—	—	54.8	46.2	24.7	07.8
V3	—	—	—	—	—	—	—	62.9	25.0	07.7
V6	—	—	—	—	—	—	—	—	24.0	05.0
V7	—	—	—	—	—	—	—	—	—	02.8

Table 4: Summary of normalized affinity scores achieved by the proposed method.

%	AF	B	CS	C	R	V	Z
AF	—	05.23	08.63	08.11	31.23	41.75	05.05
B	03.07	—	11.44	05.37	12.97	63.95	03.21
CS	04.61	10.41	—	03.10	65.86	12.85	03.16
C	05.68	06.40	04.06	—	18.84	60.22	04.80
R	10.76	07.61	42.46	09.28	—	27.62	02.28
V1	01.15	00.70	02.92	10.46	02.08	81.82	00.88
V2	01.07	01.16	00.86	02.32	08.04	85.83	00.72
V3	02.07	00.70	00.62	01.66	05.83	88.72	00.40
V6	03.05	01.10	00.51	02.23	03.38	89.07	00.67
V7	07.66	20.58	01.15	03.42	03.82	60.74	02.62
Z	11.54	12.48	13.52	15.66	15.12	31.69	—

Table 5: Summary of normalized affinity scores achieved by Chen et al. (2021).

%	AF	B	CS	C	R	V	Z
AF	—	14.78	28.67	10.55	11.88	24.51	09.61
B	10.79	—	21.66	18.52	12.55	29.15	07.33
CS	11.52	11.93	—	07.62	09.55	52.93	06.43
C	09.65	23.20	17.34	—	12.92	26.45	10.44
R	12.83	18.57	25.67	15.26	—	13.86	13.81
V1	05.69	08.91	11.13	14.13	06.28	48.65	05.22
V2	03.36	06.19	19.04	04.27	02.31	62.12	02.72
V3	03.41	05.55	18.30	04.02	01.69	65.39	01.64
V6	02.14	05.40	11.55	04.30	01.83	73.23	01.54
V7	10.34	11.30	17.13	11.87	08.13	33.60	07.63
Z	13.38	13.97	22.28	15.90	17.80	16.67	—

compared to the rest of the dataset. Importantly, these are not probabilistic confidence scores in a statistical sense, nor do they measure model uncertainty; the sharpness of the distribution is partly a function of the chosen temperature, and results should be interpreted accordingly.

The resulting affinity values for our method are presented in Table 4, alongside the baseline results from Chen et al. (2021) in Table 5. Our method shows a clear separation between the primary assignment and the remaining candidates, particularly within the Vatican series, where normalized scores exceed 80–89% compared to the baseline peak of 73%. The margin between the primary classification and the second-best candidate is much wider in our model. This suggests that the latent features extracted by the Autoencoder are not only more precise but also better at suppressing cross-cluster noise.

4.4 Stability and Threshold Evaluation

A key requirement for a computational tool in the Humanities is reliability. To test the stability of our classifications, we analyzed how the system's output changes as we increase the confidence threshold. This experiment simulates a real-world scenario where a historian might only be interested in "high-certainty" leads.

The results, summarized in Table 6, highlight several important aspects of the proposed approach. For the Vatican (V) and AVS-France (AF) ciphers, the classifications remain remarkably stable even when the threshold is increased up to 0.8, indicating that the clusters labeled as Accepted in the methodology are highly cohesive and well separated in the latent space. In parallel, entries marked with a dash (–) correspond to cases in which the system deliberately refrains from assigning a classification, reflecting a conservative decision-making strategy whereby higher thresholds favor the absence of a result over

low-confidence matches. When compared with the baseline approach, which fails to provide stable classifications for the AF and V7 ciphers at higher thresholds, the proposed framework maintains consistent identifications.

5 Discussion

The results presented in the previous section provide a comprehensive overview of the capabilities and limitations of using Convolutional Autoencoders for the unsupervised analysis of historical ciphers. The following discussion synthesizes these findings, focusing on morphological representation, the impact of stylistic variance, and the practical utility of the system for archival research.

The system's ability to identify manuscripts belonging to the same cryptographic tradition, like the case of the Vatican group, suggests that CAE-learned features are highly effective at capturing shared morphological structures. The strong correlation observed within the Vatican series confirms that the latent space can represent the shape of symbols. From a historical perspective, this capability allows the automated grouping of documents by "families", providing researchers with a reliable starting point to trace common archival origins or shared cryptographic protocols even when the specific keys remain unknown.

A significant finding of this study is the model's sensitivity to handwriting style (the "scribal hand"). Our analysis revealed that ciphers sharing the same underlying alphabet but written by different scribes were often perceived as distinct by the network. This indicates that the CAE perceives the "texture" of the stroke and individual calligraphic nuances as primary features, which can occasionally overshadow the abstract geometric shape of the character.

While this sensitivity poses a challenge for identifying abstract alphabetical identity across different manuscripts, it opens a promising new avenue for computational stylometry. The same features that "limit" alphabet identification can be repurposed to identify individual scribes or specialized workshops. In this sense, the latent space acts as a dual-purpose tool, capable of analyzing both the message (the cipher system) and the messenger (the scribe).

The distinct treatment of the Zodiac cipher highlights the superior rejection capabilities of the CAE-based approach. By mapping unrelated or visually disparate symbols to distant regions of the latent space, the system generates a robust "negative" result. This is particularly evident when comparing our scores to the baseline; whereas traditional features might find superficial similarities, our model yields significantly lower similarity indices for outliers. This reduction in "false-positive" connections is crucial for historical research, as it prevents the suggestion of misleading links between unrelated cryptographic traditions.

The higher affinity scores achieved by our framework suggest that features learned through unsupervised deep learning are more robust against the inherent "noise" of historical handwriting than traditional handcrafted features. By training the network to reconstruct the symbols, the CAE is forced to prioritize structural strokes over background artifacts or degradation. This enables the system to bridge the gap between different hands more effectively than the baseline, particularly when the scribal styles are reasonably consistent.

A fundamental challenge in this domain remains the scarcity of exhaustive ground truth. In many cases, the absence of decrypted versions makes it impossible to definitively validate the similarity scores. However, by using the Vatican ciphers as a partial ground truth, we have demonstrated that the system's confidence aligns with known historical facts.

It is important to emphasize that this framework is not intended to replace the paleographer or the historian. Instead, it serves as a "computational compass". In the vast and often overwhelming landscape of archives, our tool points researchers toward the most morphologically and stylistically reliable connections. By providing a computational hypothesis based on objective visual evidence, the system guides scholars toward unexplored archival relationships while significantly reducing the labor-intensive nature of manual manuscript comparison.

6 Conclusion

In this paper, we presented a novel, fully unsupervised framework for the comparative analysis of historical enciphered manuscripts. By leveraging Convolutional Autoencoders for neural feature extraction and hierarchical clustering for symbol discovery, our approach allows for the objective mapping of relationships between diverse

Table 6: Stability analysis of classification outputs across incremental confidence thresholds.

Cipher		AF		B		CS		C		R	
		Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline
Threshold	0.4	**V**	–	V	–	R	V	V	–	CS	–
	0.6	–	–	V	–	R	–	V	–	–	–
	0.8	–	–	–	–	–	–	–	–	–	–

Cipher		V1		V2		V3		V6		V7		Z	
		Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline
Threshold	0.4	**V**	**V**	**V**	**V**	**V**	**V**	**V**	**V**	**V**	–	–	–
	0.6	**V**	–	**V**	**V**	**V**	**V**	**V**	**V**	**V**	–	–	–
	0.8	**V**	–	**V**	–	**V**	–	**V**	–	–	–	–	–

manuscripts without the need for manual labeling or pre-defined alphabets.

The experimental results demonstrate that the learned latent features are highly effective at identifying shared cryptographic traditions, particularly within the Vatican cipher families, where our system achieved a peak similarity of 84.4%. Furthermore, the implementation of a probabilistic confidence metric via a Softmax function showed that our model possesses superior discriminative power compared to traditional baseline methods, providing more stable and certain classifications even under high-threshold constraints.

A key finding of our research is the system's sensitivity to individual handwriting styles. While this poses a challenge for identifying abstract alphabetical identity across different scribes, it reveals a significant opportunity for computational stylometry and author identification within the context of secret archives. This duality suggests that the latent space captures a rich representation of the manuscript that encompasses both its cryptographic structure and its physical execution.

In conclusion, our framework offers historians and paleographers a "computational compass" to navigate vast archives. By automating the preliminary stages of manuscript comparison, the tool significantly reduces manual labor and points toward previously unexplored connections between documents.

Future work will pursue two main directions. First, we will investigate style-invariant feature extraction methods to further isolate alphabetical content from scribal nuances, potentially leveraging domain-adapted contrastive learning or self-supervised vision transformers trained specifically on historical document images. Second, we will conduct a systematic comparison between the CAE-based approach and more recent representation learning methods (including Variational Autoencoders, SimCLR, and DINO) under the specific data constraints of historical cryptanalysis, to establish whether the performance gains of modern methods outweigh the challenges posed by small dataset sizes and extreme domain shift. Finally, we plan to involve a historical paleographer to conduct a thorough evaluation of the method's strengths and weaknesses, with the goal of further improving it. Specifically, a comparison between an expert's judgment of cipher similarity and the similarity scores generated by our method would provide important insights.

Acknowledgments

This work has been partially supported by Riksbankens Jubileumsfond, grant M24-0028 (Echoes of History: Analysis and Decipherment of Historical Writings, DESCRYPT), the Spanish project PID2024-157778OB-I00 (SUKIDI) from the Ministerio de Ciencia e Innovación, the Departament de Cultura of the Generalitat de Catalunya, and the CERCA Program / Generalitat de Catalunya. Alicia Fornés acknowledges financial support for her general research activities from ICREA under the ICREA Academia (Departament de Recerca i Universitats de la Generalitat de Catalunya).

References

Arnau Baró, Jialuo Chen, Alicia Fornés, and Beáta Megyesi. 2019. Towards a generic unsupervised method for transcription of encoded manuscripts. In *Proceedings of the 3rd International Conference on Digital Access to Textual Cultural Heritage*, pages 73–78.

Mathilde Caron, Hugo Touvron, Ishan Misra, Herv'e J'egou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging properties in self-supervised vision transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9630–9640.

Kai Chen, Mathias Seuret, Marcus Liwicki, Jean Hennebert, and Rolf Ingold. 2015. Page segmentation of historical document images with convolutional autoencoders. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1011–1015.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*.

Jialuo Chen, Mohamed Ali Souibgui, Alicia Fornés, and Beáta Megyesi. 2021. Unsupervised alphabet matching in historical encrypted manuscript images. In *Proceedings of the 4th International Conference on Historical Cryptology, HistoCrypt 2021*, pages 34–37. Linköping University Electronic Press.

Amar Jindal and Rajib Ghosh. 2023. Word and character segmentation in ancient handwritten documents in devanagari and maithili scripts using horizontal zoning. *Expert Systems with Applications*, 225:120127.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*.

Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. 2011. Stacked convolutional auto-encoders for hierarchical feature extraction. In *Artificial Neural Networks and Machine Learning–ICANN 2011*, pages 52–59. Springer.

Beáta Megyesi, Nils Blomqvist, and Eva Pettersson. 2019. The DECODE database: Collection of historical ciphers and keys. In *Proceedings of the 2nd International Conference on Historical Cryptology, HistoCrypt 2019*, pages 69–78. Linköping University Electronic Press.

Beáta Megyesi, Bernhard Esslinger, Alicia Fornés, Nils Kopal, Benedek Láng, George Lasry, Karl de Leeuw, Eva Pettersson, Arno Wacker, and Michelle Waldispühl. 2020. Decryption of historical manuscripts: the DECRYPT project. *Cryptologia*, 44(6):545–559.

Martín Méndez, Pau Torras, Adrià Molina, Jialuo Chen, Oriol Ramos-Terrades, and Alicia Fornés. 2024. Structured analysis and comparison of alphabets in historical handwritten ciphers. In *European Conference on Computer Vision*, pages 330–344. Springer.

Fionn Murtagh and Pedro Contreras. 2012. Algorithms for hierarchical clustering: An overview. *Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery*, 2:86–97, 01.

Jaakko Sauvola and Matti Pietikäinen. 2000. Adaptive document image binarization. *Pattern Recognition*, 33(2):225–236.

Mohamed Ali Souibgui, Alicia Fornés, Yousri Kessentini, and Crina Tudor. 2021. A few-shot learning approach for historical ciphered manuscript recognition. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5413–5420. IEEE.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. 2014. Striving for simplicity: The all convolutional net. *CoRR*, abs/1412.6806.

Joe H. Ward. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58:236–244.

Chathurika S. Wickramasinghe, Daniel L. Marino, and Milos Manic. 2021. Resnet autoencoders for unsupervised feature learning from high-dimensional data: Deep models resistant to performance degradation. *IEEE Access*, 9:40511–40520.

Xusen Yin, Nada Aldarrab, Beáta Megyesi, and Kevin Knight. 2019. Decipherment of historical manuscript images. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 78–85.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328.

Modeling ENIGMA with Integer Linear Programming

Kevin Knight
Threeven Labs
kevin.crawford.knight@gmail.com

Abstract

We model the German ENIGMA cipher machine as a large integer linear program (ILP), allowing us to solve certain encipherment and decipherment problems with a generic ILP solver.

1 Introduction

ENIGMA is an electromechanical device that was used by the German military to encrypt secret communications before and during World War II. It was the subject of intense code-breaking efforts in Poland (Rejewski, 1984) and the United Kingdom (Welchman, 1982). Researchers since then have conducted a number of ENIGMA analyses using specially-designed software (e.g., Gillogly, 1995; Bagnall et al., 1997; Sullivan and Weierud, 2005; Ostwald and Weierud, 2017; Lasry et al., 2019; Sommervoll and Nilsen, 2021; Kopal and Esslinger, 2022).

In this paper, we investigate the use of a general-purpose problem solver to model ENIGMA. We first describe the workings of the machine as a set of variables and constraints in an integer linear program (ILP).

Disclaimer: The use of the name Adolf Hitler in this paper is solely to illustrate a technical question concerning Enigma settings in a historical cryptographic context. No endorsement, glorification, or political meaning is intended.

We then use a generic ILP solver to answer questions such as "Is there an initial ENIGMA setting that encrypts the string KEVINKNIGHT as ADOLFHITLER? If so, what setting uses the fewest number of plugboard wires?"

Advantages of this approach include:

- Questions are put to the solver in declarative form, as values for specific variables or constraints on those values, and no cryptanalytic software or special search algorithms need to be written. Instead, the ILP solver employs general-purpose search strategies, such as branch-and-bound and cutting planes.
- The ILP solver returns optimal answers.

The main disadvantage is slow speed, due to the generic problem solver's optimality and its lack of ENIGMA-specific strategies. In this paper, we investigate what can and cannot be done by modeling ENIGMA this way.

2 Integer Linear Programming

Any ILP consists of variables, constraints, and an objective function. For example:

Integer variables: x, y

Maximize: $2x + y$

Subject to:

$x + y \leq 6.9$
$y - x \leq 2.5$
$y > 1.1$

The goal of an ILP solver is to find integer assignments[1] to the variables that satisfy the constraints and maximize the objective function. Readers unfamiliar with ILPs may want to solve the above puzzle by hand. Dantzig (1949) popularized linear programming after World War 2 and introduced the first generic solver, the simplex algorithm.

3 ILP for Simple Substitution

Ravi and Knight (2009) show how to model a simple substitution cipher as an ILP. Since this cipher is much simpler than ENIGMA, we briefly review their method.

[1] In this paper, we further constrain each variable to take on the value 0 or 1.

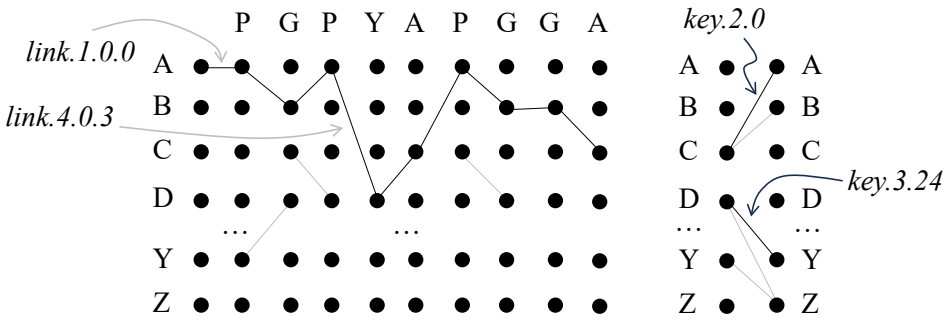


Figure 1: Decrypting a simple substitution cipher with integer linear programming (ILP). The ciphertext runs along the top, and there is a row for each plaintext letter A-Z. The ILP contains two types of binary variables, *link.k.i.j* and *key.i.j*. Their values are constrained so that *link* variables with value 1 make up a single, connected path that assigns a plaintext letter type to each ciphertext token. Here, the *link* variables assign the plaintext ABADCABBY to the ciphertext PGPYAPGGA.

Figure 1 visualizes decipherment using a rectangular lattice. Ciphertext letter tokens (n of them) run along the top, and plaintext letter types (26 of them) run down the left-hand side. Any particular decipherment is a linked sequence of nodes through the lattice, where each node assigns a plaintext type to a ciphertext token. Separately, there is a key connecting plaintext and ciphertext types.[2]

The ILP variables are binary, valued at 0 or 1.

- *link.k.i.j* = 1 iff the decipherment contains a link between plaintext types i and j underneath ciphertext token k. Here, k ranges from 1 to $n-1$, where n is the length of the ciphertext, while i and j range from 0 (A) to 25 (Z).

- *key.i.j* = 1 iff the key maps plaintext letter type i onto ciphertext letter type j.

For a ciphertext of length 50, the ILP will contain $49 \cdot 26 \cdot 26$ link variables and $26 \cdot 26$ key variables, for a total of 33,800 distinct variables.

To restrict the values these variables are allowed to take, we need a number of "Subject to" constraints. First, we ensure that link variables with value 1 form a connected path through the lattice (rather than a scattering of unconnected links), by requiring that the sum of "on" links entering a node equals the sum of "on" links exiting that same node.

$$\forall_{k=1}^{n-2} \; \forall_{i=0}^{25} \; : \; \sum_{j=0}^{25} link.k.j.i = \sum_{j=0}^{25} link.(k+1).i.j$$

Next, we require that the chosen links are consistent with the key:

$$\forall_{k=1}^{n-1} \; \forall_{i=0}^{25} \; : \; \sum_{j=0}^{25} link.k.i.j = key.i.c_k$$

where c_k is the type of the kth cipher token. Of course, this is only useful if the key is restricted to be one-to-one, so we also add:

$$\forall_{i=0}^{25} \; : \; \sum_{j=0}^{25} key.i.j = 1 \quad \Big| \quad \forall_{i=0}^{25} \; : \; \sum_{j=0}^{25} key.j.i = 1$$

By this time, we have also ruled out link settings that represent zero or multiple connected paths. The total number of constraints for a 50-letter cipher comes to 2728.

Lastly, we need a way to prefer one link sequence over another. A reasonable objective is to maximize the probability of the plaintext $p_1 \ldots p_n$, or equivalently, the negative log probability, which we approximate with letter bigram probabilities:

$$P(p_1 \ldots p_n) \sim \sum_{i=1}^{n-1} -logP(p_{i+1}|p_i)$$

$$P(p_{i+1}|p_i) = \frac{\text{count}(p_i p_{i+1})}{\text{count}(p_i)}$$

[2]Throughout, *token* refers to a letter found in running text, such as the second A in KOALABEAR, while *type* refers to a member of some fixed vocabulary, such as a letter in the range A-Z.

Counts are taken over a large plaintext corpus unrelated to the cipher at hand. Fortunately, we can write this objective as a weighted linear combination of the ILP variables:

$$\text{minimize: } \sum_{k=1}^{n-1} \sum_{i=0}^{25} \sum_{j=0}^{25} link.k.i.j \cdot logP(j|i)$$

Given a particular ciphertext, we can now write out the relevant constraints and invoke the solver, which sets the *link* and *key* variables to maximize the objective. Inspecting the variables with value 1 lets us read off the plaintext and key. After that, we can be sure there is no better-scoring plaintext than the one returned.

Finally, if we have additional information about the cipher, we can add it in the form of constraints before solving. For example, we may encode that no letter enciphers to itself:

$key.0.0 = 0$ where 0 stands for A
$key.1.1 = 0$ where 1 stands for B
...

or that space always enciphers as space:

$key.26.26 = 1$ where 26 stands for space

Ravi and Knight (2009) build ILPs for a large number of synthetically-generated ciphertexts and report decipherment accuracy rates under the ILP solver's exact search.

4 ENIGMA

ENIGMA is an electromechanical device that implements a complex substitution cipher. As illustrated in Figure 2, when a plaintext key is pressed on the keyboard, a ciphertext letter lights up in the display area. The wiring of ENIGMA dictates the mapping between plaintext and ciphertext letters. Electricity passes through a *plugboard* (or *steckerboard*), then through a series of three rotors, then through a *reflector*, and finally back through the rotors and plugboard to the cipher-letter display.

This would seem to implement a simple substitution cipher, but each time a keyboard key is pressed, rotors may turn, changing their wiring patterns.

Configuring ENIGMA with a secret key consists of these steps:

1. Select a reflector (inventory: B or C). Each of the two reflectors has a fixed wiring pattern.

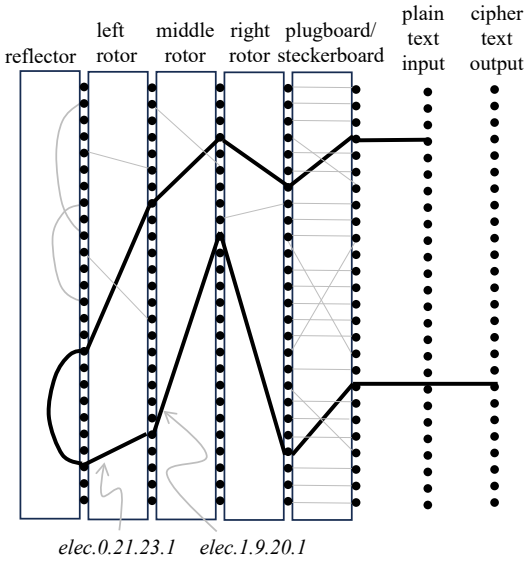


Figure 2: Electricity passing through wires inside ENIGMA. Electrical contact points are represented by black circles. Once a reflector is selected ("B-type" or "C-type"), the wiring inside the reflector is fixed. Likewise, once plugboard wires are inserted, that wiring is also fixed. Each time a plaintext letter is entered on the keyboard, one or more of the rotors will turn, changing the wiring pattern.

2. Select a rotor (inventory: I, II, or III) for the left-hand slot. Each of the three rotor-types has a fixed wiring pattern.
3. Select a ring setting for the rotor. This rotates the internal wiring relative to a physical notch on the rotor.
4. Place the rotor into the left-hand slot and turn it to an initial position, called the indicator setting.[3]
5. Repeat this process for the middle rotor and the right rotor. Each rotor should be of type I, II, or III, and each type should be used exactly once.

By manipulating the ring and indicator settings, we independently decide on the initial wiring pattern and the initial notch position for a rotor. As we type plaintext letters, the right rotor's notch advances. When it reaches the top, it forces the middle rotor to advance, changing *its* wiring pattern and advancing its own notch. When the middle rotor's notch reaches the top, it similarly advances

[3]We will use "position," "indicator," and "indicator setting" interchangeably.

the left rotor. The left rotor has a notch, but since it never engages another piece of ENIGMA, its position is irrelevant. Finally, due to a curious feature called *double-stepping*, the notch on the middle rotor never remains at the top. Once it gets there, it automatically advances again with the next keypress.

When two ENIGMAs are configured in the same way, they can send encrypted messages to each other. Sending and receiving are symmetric—to decrypt, we simply type the ciphertext and read the plaintext off the display.

5 ILP for ENIGMA

This section overviews our ILP for ENIGMA, with Figures 3-8 providing the complete specification. Readers uninterested in the details of the ILP model can skip this section and associated figures, and proceed to Section 6, where we describe tests.

5.1 Variables

Figure 3 shows the variables in the ILP model. All variables are binary, with values 0 or 1. In naming variables and writing constraints, these conventions apply:

- w ranges over the rotor inventory (0 ... 2)
- f ranges over the reflector inventory (0 ... 1)
- p ranges over rotor positions (0 ... 2)
- t ranges over time steps (0 ...)
- i, j range over letter types (0 ... 25).

Variables *elec.p.i.j.t* represent the flow of electricity through the rotors. Figure 2 previously illustrated this for two such variables. Similarly, *elec.u.i.j.t* and *elec.s.i.j.t* model electrical flow through the reflector and plugboard. Taken together, these variables are analogous to the *link* variables of Section 3.

To be precise, *elec.p.i.j.t = 1* if electricity passes through rotor p (0 = left, 1 = middle, 2 = right) between contact point i on the rotor's right side and contact point j on its left side, at time t. Note that i represents the rotor's right side regardless of the direction electricity is flowing.

Electricity between two positions at time t is only possible if there is a physical wire connecting those positions at time t. The presence (or absence) of a wire is modeled by *wire.p.i.j.t*.

The initial setting of rotors is modeled by *init.w.p.i.j* = 1 if a rotor of type w (0 = I, 1 = II, 2 = III) is slotted into position p (0 = left, 1 = middle, 2 = right) with indicator setting i and ring setting j. The reflector selection is captured by *reflector.0* (B-type) and *reflector.1* (C-type). The *init* and *reflector* are connected to *wire.p.i.j.0* and *wire.u.i.j* variables, which capture the state of the machine at time $t = 0$. The initial plugboard wiring is given directly through the values of *wire.s.i.j*.

If the plaintext letter at time t is i, then *plaintext.i.t* = 1, and likewise for *ciphertext.i.t*. The plaintext starts at time $t = 0$, while the ciphertext starts at time $t = 1$. This is because the first electrical circuit is closed only after the first plaintext letter is typed.

The rest of the variables track notch positions and rotor turning, the latter of which results in *wire.p.i.j.t* having a potentially different value from *wire.p.i.j.(t-1)*.

5.2 Constraints

Figure 4 shows constraints controlling electrical flow. First, *elec.p.i.j.t* $\leq$ *wire.p.i.j.t*, so that electricity can flow only if a wire is present. Second, for each contact point, the total electricity coming into the contact point equals the total electricity going out, the same idea as the constraints on *link* variables in Section 3.

Figure 5 gives constraints on the initial configuration of rotors. For example, each rotor from the inventory (I, II, and III) can be selected no more than once, and each rotor slot position (0 = left, 1 = middle, 2 = right) must be filled exactly once. Figures 6, 7, and 8 provide constraints that ensure consistency between the initial ENIGMA configuration, plaintext, ciphertext, and electrical flow between contact positions at each point in time.

In setting up an ILP for a cipher of length 50, we generate a total of 517,266 variables and 417,894 constraints (after de-duplication).

6 Fidelity of the ILP Model

We first test the model in "open mode" by setting the time steps to $n = 4$ and invoking the ILP solver with an empty objective function. The solver easily finds a value for each variable such that the constraints are all satisfied. Note that we have not supplied any plaintext or ciphertext, so finding values for *plaintext.i.t* and *ciphertext.i.t* is part of the solver's job.

Here is a subset of the variables assigned value 1:

- *elec.0.i.j.t* = 1 iff electricity flows from contact i (right side) of left rotor to contact j (left side).
- *elec.1.i.j.t* = 1 (same for middle rotor).
- *elec.2.i.j.t* = 1 (same for right rotor).
- *elec.u.i.j.t* = 1 iff electricity flows between contacts i and j on reflector.
- *elec.s.i.j.t* = 1 iff electricity flows from contact i to contact j on steckerboard.
- *init.w.p.i.j* = 1 iff rotor type w slots into position p with indicator setting i and ring setting j.
- *wire.s.i.j* = 1 iff the plugboard connects position i (on keyboard side) with position j
- *reflector.f* = 1 indicates choice of reflector type B ($f = 0$) or C ($f = 1$)
- *wire.u.i.j* = 1 iff a reflector wire connects contact position i with j
- *plaintext.i.t* = 1 iff tth plaintext letter (starting at $t = 0$) is ith letter of alphabet (a=0 ... z=25)
- *ciphertext.i.t* = 1 iff tth ciphertext letter (starting at $t = 1$) is ith letter of alphabet
- *notch.p.i* = 1 iff the notch on the pth rotor (0 ... 2) is initially at position i
- *notches.i.j.t* = 1 iff if the notch on the middle and right rotors are at position i and j at time t
- *wire.p.i.j.t* = 1 iff the pth rotor has a wire connecting contact i (right side) to j at time t
- *rotate.p.t* = 1 iff the pth rotor rotates at time t
- *wire.p.i.j.t.rotate* = 1 iff the wire connecting i and j is inside rotor p that rotates at time t
- *wire.p.i.j.t.norotate* = 1 iff the wire between i and j is inside non-rotating rotor p at time t

Figure 3: Variables in the ILP model for ENIGMA.

$$\forall_{t,p,i,j} : elec.p.i.j.t \leq wire.p.i.j.t$$

$$\forall_{t,p,i,j} : elec.s.i.j.t \leq wire.s.i.j \quad \Big| \quad \forall_{t,p,i,j} : elec.u.i.j.t \leq wire.u.i.j$$

$$\forall_{t,i} : plaintext.i.t + ciphertext.i.t = \sum_j elec.s.i.j.t$$

$$\forall_{t,i} : \sum_j elec.s.j.i.t = \sum_j elec.2.i.j.t \quad \Big| \quad \forall_{t,i} : \sum_j elec.2.j.i.t = \sum_j elec.1.i.j.t$$

$$\forall_{t,i} : \sum_j elec.1.j.i.t = \sum_j elec.0.i.j.t \quad \Big| \quad \forall_{t,i} : \sum_j elec.0.j.i.t = \sum_j elec.u.i.j.t$$

$$\forall_{t,i} : \sum_j elec.u.j.i.t = \sum_j elec.0.i.j.t \quad \Big| \quad \forall_{t,i} : \sum_j elec.0.i.j.t = \sum_j elec.1.j.i.t$$

$$\forall_{t,i} : \sum_j elec.1.i.j.t = \sum_j elec.2.j.i.t \quad \Big| \quad \forall_{t,i} : \sum_j elec.2.i.j.t = \sum_j elec.s.j.i.t$$

$$\forall_{t} : \sum_i plaintext.i.t = 1 \quad \Big| \quad \forall_{t} : \sum_i ciphertext.i.t = 1$$

Figure 4: Constraints on electrical flow, ensuring a single connected path such as that shown in Figure 2. Note that this formulation includes redundant constraints to emphasize the bidirectional electrical flow; redundancies are removed by the ILP solver in pre-processing.

$$\forall_w : \sum_{p,m,r} init.w.p.m.r \leq 1 \quad \Big| \quad \forall_p : \sum_{w,m,r} init.w.p.m.r = 1$$

Figure 5: Constraints on initial rotor selection and placement, ensuring no rotor type is selected more than once, and that no rotor position is left empty or filled with multiple rotors.

$$\sum_i wire.s.i.i \geq 26 - 2 \cdot \text{steckermax} \quad \Big| \quad \sum_i wire.s.i.i \leq 26 - 2 \cdot \text{steckermin}$$

$$\forall_i : \sum_j wire.s.i.j = 1 \quad \Big| \quad \forall_{i<j} : wire.s.i.j = wire.s.j.i$$

$$\sum_f reflector.f = 1 \quad \Big| \quad \forall_{f,i} : wire.u.i.R[f]_i \leq reflector.f \quad \Big| \quad \forall_i : \sum_j wire.u.i.j = 1$$

$$\forall_{p,i,j} : wire.p.i.j.0 = \sum_{w,m,r} init.w.p.m.r \quad \text{(add to sum only when } P[w]_{i+m-r} = j+m-r\text{)}$$

```
Note:
    P[0] = rotorIwiring     = EKMFLGDQVZNTOWYHXUSPAIBRCJ
    P[1] = rotorIIwiring    = AJDKSIRUXBLHWTMCQGZNPYFVOE
    P[2] = rotorIIIwiring   = BDFHJLCPRTXVZNYEIWGAKMUSQO
    R[0] = reflectorBwiring = YRUHQSLDPXNGOKMIEBFZCWVJAT
    R[1] = reflectorCwiring = FVPJIAOYEDRZXWGCTKUQSBNMHL
```

Figure 6: Constraints on initial wiring of plugboard, reflector, and rotors, bounding the amount of steckering, and ensuring wiring is consistent with the initial configuration of the rotors and reflector.

$$\forall_{p,i} : notch.p.i = \sum_{w,j} init.w.p.Q[w] - i \pmod{26}.j$$

$$\forall_i : notch.1.i = \sum_j notches.i.j.0 \quad \Big| \quad \forall_i : notch.2.i = \sum_j notches.j.i.0$$

$$\forall_{t,i,j} : notches.i.j.t+1 = \begin{cases} 0 & \text{if } i = 0 \text{ and } j < 25 \\ notches.i+1.0.t & \text{else if } j = 25 \\ notches.25.j+1.t + notches.0.j+1.t & \text{else if } i = 25 \\ notches.i.j+1.t & \text{otherwise} \end{cases}$$

```
Q[w] is notch position of rotor w, namely:
    Q[0] = Q     Q[1] = E     Q[2] = V
```

Figure 7: Constraints on notch positions, ensuring that notches move when rotors turn. When a notch reaches top position, it engages the rotor to the left, forcing it to turn. In addition, the middle rotor "double steps," as its notch cannot remain in top position. The update is complicated by the fact that while each configuration of rotors has a deterministic successor, some have two possible predecessors.

$$\forall_t \; : \; rotate.2.t = 1$$

$$\forall_t \; : \; rotate.1.t = \sum_{i,j} notches.i.j.t \text{ such that } i = 0 \text{ or } j = 0$$

$$\forall_t \; : \; rotate.0.t = \sum_{i} notches.0.i.t$$

$$\forall_{t,p} \; : \; \sum_{i,j} wire.p.i.j.t.rotate = 26 \cdot rotate.p.t$$

$$\forall_{t,p,i,j} \; : \; wire.p.i.j.t = wire.p.i.j.t.rotate + wire.p.i.j.t.norotate$$

$$\forall_{t,p,i,j} \; : \; wire.p.i.j.t = wire.p.i+1.j+1.t-1.rotate + wire.p.i.j.t-1.norotate$$

Figure 8: Constraints on rotor turning, ensuring that physical locations of rotor wires are updated appropriately.

init.2.0.21.0	*init.0.1.24.17*	*init.1.2.4.23*
reflector.0	*wire.s.2.3*	*wire.s.3.2*
plaintext.24.0	*plaintext.12.1*	*elec.0.24.5.1*
ciphertext.12.1	*ciphertext.8.2*	...

We can interpret this as follows:

Left rotor III, indicator V, ring setting A
Middle rotor I, indicator Y, ring setting R
Right rotor II, indicator E, ring setting X
AB CD EF MO NP QR ST UV WX YZ
Reflector B
Plaintext: (A) Y M D W
Ciphertext: M I A V (A)

Or, more compactly:

III I II (Reflector B) VYE ARX
AB CD EF MO NP QR ST UV WX YZ
YMDW → MIAV

Using an independent ENIGMA simulator GUI, we confirm that plaintext YMDW is encrypted as MIAV under these settings.

We also want to test edge cases like *double stepping*. A quick test is to add these two constraints:

notch.2.0 = 1	right rotor
notch.1.1 = 1	middle rotor

The first constraint requires the right rotor to be in top notch position 0, ready to engage the middle rotor on the first keypress. The middle rotor starts in notch position 1. The solver now yields:

I III II (Reflector B) RUE ACT
AB CD EF MO NP QR ST UV WX YZ
WQYV → HLGG

An independent ENIGMA simulator confirms the double-stepping action of the machine as it encrypts WQYV as HLGG.

7 Experiments

In this section, we use our ILP to answer questions about ENIGMA.

7.1 A Constant Sequence Question

> *Question 1:* What is the largest n for which an unsteckered ENIGMA can be configured to encipher A^n as Z^n?

To prohibit plugboard wires, we add 26 constraints of the form *wire.s.i.i* = 1, one for each i from 0 to 25 (A-Z). We constrain *plaintext* and *ciphertext* variables to hold various amounts of As and Zs.

This configuration encrypts AAAA as ZZZZ:

II I III (Reflector C) VQS KVJ
AAAA → ZZZZ

However, there is no configuration that encrypts AAAAA as ZZZZZ. When we ask for a solution, the ILP solver simply returns Infeasible.[4]

If we allow steckering, we *can* find solutions for $n > 4$. For example:

[4] All runs use v13.0.1 Gurobi Optimization (2024) on an Intel Core i9-13950HX CPU laptop with 128 GB RAM, running Ubuntu 22.04.5 LTS. The open-source HiGHS 1.14.0 is slower, failing to solve AAAA → ZZZZ within 3 hours.

II I III (Reflector B) QQT KTY
AM CJ EP FV GQ HR LZ OT SW UY
AAAAA → ZZZZZ

II III I (Reflector B) RUQ OEC
BP CD FJ GX HY KS MQ NW OU RZ
AAAAAA → ZZZZZZ

7.2 An Eleven-Letter Question

Question 2: Is there an ENIGMA configuration that encrypts KEVINKNIGHT as ADOLFHITLER, using the standard German Army practice of 10 plugboard wires?

We cast this question as an equivalent maximization problem.

Question 3: Given plaintext KEVINKNIGHT, what ENIGMA setting generates a ciphertext that maximally resembles ADOLFHITLER?

We supply the plaintext KEVINKNIGHT and add this objective function:

ciphertext.0.1 +	(first letter = A)
ciphertext.3.2 +	(second letter = D)
ciphertext.14.3 +	(third letter = O)
...	
ciphertext.17.11	(eleventh letter = R)

As the ILP solver strives to maximize this objective, it emits a series of increasingly better solutions, ultimately answering *Question 2* in the affirmative:

III I II (Reflector B) URF ZKY
AB CD EF MO NP QR ST UV WX YZ
KEVINKNIGHT → **LDQBMADZNEA**
Objective value: 1

I II III (Reflector B) QDZ HXF
AS BW CE DR FL GV IX JT KO NZ
KEVINKNIGHT → **SAOLFXITVER**
Objective value: 7

III II I (Reflector B) AYR DEA
AS CL DO ER FG HZ IY KX NQ TU
KEVINKNIGHT → **ASOLFHHTLLR**
Objective value: 8

I II III (Reflector B) ZGL QAR
AV BC DG EY FL HM IX JT NZ OS
KEVINKNIGHT → **ADOLFHITQER**
Objective value: 10

III II I (Reflector B) ADP FXK
AU BL CI EV FR GW HS JN OT QZ
KEVINKNIGHT → **ADOLFHITLER**
Objective value: 11

Question 4: Is the solution unique? Are there other configurations that also encrypt KEVINKNIGHT as ADOLFHITLER with 10 plugboard wires?

There are several ways to answer this question. One is to add the following constraint and re-run the solver, preventing it from choosing the same rotor settings:

$$init.0.2.15.10 + init.1.1.3.23 + init.2.0.0.5 < 3$$

Another is to ask the solver for multiple optimal solutions, a strategy that tends to yield only minor changes in plugboard wiring:

III II I (Reflector B) ADP FXK
AU BL CI EV FR GW HS JN **MZ** OT
KEVINKNIGHT → **ADOLFHITLER**

Or, we can insist on using Reflector C instead of B (by adding constraint *reflector.1* = 1), in which case we get yet another solution:

I II III (Reflector C) QER TYQ
AG BT CV DO FX HQ IS KW LN RU
KEVINKNIGHT → ADOLFHITLER

Question 5: What ENIGMA configuration encrypts KEVINKNIGHT as ADOLFHITLER with the fewest number of plugboard wires?

Here, we simply maximize the sum of *wire.s.i.i* variables, i.e., one for each letter that the steckerboard maps to itself. Although this objective is easy to state, it is difficult for the ILP solver to optimize. We interrupt the solver after this 6-wire solution:

II I III (Reflector C) VRR AKO
AL DS HP JK TX VW
KEVINKNIGHT → ADOLFHITLER
38.4hrs

Because of the interruption, we cannot rule out a solution with fewer plugwires. However, a similar query for the shorter pair KEVIN → ADOLF terminates with one plugwire, so we at least conclude that neither name pair can be mapped by an unsteckered ENIGMA.

7.3 Lazy Plaintext

During World War II, British codebreaker Mavis Lever noticed a long Italian Navy ENIGMA ciphertext message without any L's. ENIGMA cannot encode a letter as itself, so she assumed the Italian operator had sent a test message by lazily choosing a plaintext of repeated L's (Woo, 2013).

> *Question 6:* How long must such a "lazy" ENIGMA intercept be to permit key recovery?

First, we randomly construct an unsteckered key and use it to encrypt L* = LLLLL ...

III I II (Reflector C) KFS ISC
L* → JUJOFGZQGVPAGPGGRGI
MDRYKRGOABAMAWTRETW
JZPBVUCTKKQYDCZGDNY
TNRWPQEZHZONANSJCUW
AJSIEZOWUOCJDGMKAIB
USYJYMDPZMNWRIDFDFP
QAOKEGDVUIKIAKVUOUH ...

Given an intercepted plaintext/ciphertext pair of length 3 *only* (namely, LLL → JUJ), the ILP solver guesses:

I II III (Reflector C) IDV PFU
LLL → JUJ M...

The proposed solution is consistent with LLL/JUJ, but it fails to match the original key and incorrectly encrypts a fourth L as M instead of O.

Below are keys recovered from intercepts of lengths 5, 8, and 14, respectively. The third key's encryption of the next 100 Ls matches the key exactly.

III I II (Reflector C) TQW SEG
LLLLL → JUJOF GZQ R...
5.2hrs

III I II (Reflector C) SBF QOP
LLLLLLLL → JUJOFGZQ GVPAR...

III I II (Reflector C) TQS SEC
LLLLLLLLLLLLL → JUJOFGZQGVPAGP
14.8hrs

Now we make a different key, this time using ten stecker wires:

III I II (Reflector C) KFS ISC
AF CN DY EB GH IR JQ KL OT PS
L* → HSTSTTQGBSXJZGAGVZST ...

With an intercept of length 10, the solver's solution achieves the goal, although it is totally unrelated to the original key:

I III II (Reflector B) IUD ANW
AE BM GY HO IQ JT LV NP RX SW
LLLLLLLLLL → HSTSTTQGBS PWR

As discussed by Ostwald and Weierud (2017), accurate key recovery for such a short plaintext/ciphertext pair is impossible in the presence of steckering.

7.4 Crib Attack Largely Unsuccessful

In a practical situation, we may intercept a long ciphertext with a guess as to how some small part of it decodes. For example, consider a 32-letter ciphertext generated by an unsteckered ENIGMA, plus a guessed decipherment (or *crib*) of the first seven letters:

```
CT: QYPUQNWNBBVHRMANZCWQSLBYSSSDQPZN
PT: FARBEIT????????????????????????
```

In this case, we want to recover a key that:

- is consistent with the crib
- yields sensible plaintext when applied to the entire cipher

We can enforce the former with ILP constraints (as before) and encourage the latter with an ILP objective. Perhaps the simplest approximation to "sensible plaintext" is a unigram objective function:

$$\text{minimize: } \sum_{t=1}^{n} \sum_{i=0}^{25} plaintext.i.t \cdot logP(i)$$

where P(0) is the probability of A, P(1) is the probability of B, etc. Because log probabilities are negative, we minimize instead of maximize.

Question 7: Which key setting yields the most probable plaintext for the 32-letter cipher above?

The ILP solver is unable to find a solution after four days of runtime. However, if we fix the middle rotor to match the key (rotor I, position F, ring setting S), the solver decides to optimize the "English-ness" of the plaintext by finding a very good ENIGMA setting:

```
CT: QYPUQNWNBBVHRMANZCWQSLBYSSSDQPZN
PT: FARBEITFROMMETOPAINTAROSYPICTURE
 III I II (Reflector C) AFT YSD
 Objective: 141.54
```

Allowing for the possibility of steckered configurations, however, confounds the crib attack. The ILP solver finds a setting that produces a plaintext completion with a better letter-unigram score, but which is gibberish:

```
CT: QYPUQNWNBBVHRMANZCWQSLBYSSSDQPZN
PT: FARBEITDUGNEISDHSEAMRWOHITDTNMAK
 II III I (Reflector B) UVK WOY
 AR BL CI DZ EP FN GM OT QU SX
 Objective: 114.13
```

Question 8: Can a higher-order n-gram model distinguish the correct ENIGMA key from others?

Unknown. The answer boils down to whether two different ENIGMA settings could produce two distinct yet equally-sensible plaintexts. In theory, the ILP solver could answer this question by returning the top N solutions under a strong English objective function, but it is not fast enough in practice. However, longer ciphers are known to succumb to simple letter n-gram scoring.

8 Ciphertext-Only Attack

In the previous section, the crib serves to narrow down the possible ENIGMA keys, while the objective function prefers keys that generate sensible plaintexts. Of course, nothing stops us from simply deleting the crib constraints.

Question 9: Can ILP solve a long ENIGMA cipher without a crib?

It seems not, at least for the current model and search algorithm; the runtimes for ILP ciphertext-only attacks with n-gram English models are prohibitive.

9 Conclusion

By modeling ENIGMA as an integer linear program (ILP), we are able to ask questions about the machine very simply, by adding constraints on the plaintext, ciphertext, and/or settings of the machine. These questions can then be answered by a generic ILP solver without any special-purpose programming. The flexibility to constrain any part of the machine's operation makes it easy to explore.

At the same time, we find that for more difficult questions, the ILP solving speed decreases and its genericity/optimality can become liabilities rather than strengths. We would like to experiment in future work on whether significant speedups can be obtained by changes to the ILP model or the search parameters of a generic solver. Just for example, an ILP designer can suggest which variables might be given higher priority in early branching.

All software used in this paper appears at `github.com/kevincrawfordknight/enigma-ilp`.

10 Acknowledgments

We thank the anonymous reviewers for their valuable comments.

References

Anthony J. Bagnall, Geoff P. McKeown, and Victor J. Rayward-Smith. 1997. The cryptanalysis of a three rotor machine using a genetic algorithm. In *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 712–718. Morgan Kaufmann.

George B. Dantzig. 1949. Programming of interdependent activities: II mathematical model. *Econometrica*, 17(3):200–211.

James J. Gillogly. 1995. Ciphertext-only cryptanalysis of Enigma. *Cryptologia*, 19(4):405–413.

Gurobi Optimization. 2024. Gurobi Optimizer Reference Manual.

Nils Kopal and Bernhard Esslinger. 2022. New Ciphers and Cryptanalysis Components in CrypTool 2. In *Proceedings of the 5th International Conference on Historical Cryptology, HistoCrypt 2022*, pages 127–136. Linköping University Electronic Press.

George Lasry, Nils Kopal, and Arno Wacker. 2019. Cryptanalysis of Enigma double indicators with hill climbing. *Cryptologia*, 43(4):267–292.

Olaf Ostwald and Frode Weierud. 2017. Modern breaking of Enigma ciphertexts. *Cryptologia*, 41(5):395–421.

Sujith Ravi and Kevin Knight. 2009. Attacking decipherment problems optimally with low-order n-gram models. *Cryptologia*, 33(4).

Marian Rejewski. 1984. How Polish mathematicians deciphered the Enigma. In Władysław Kozaczuk, editor, *Enigma: How the Poles Broke the Nazi Code*, pages 1–33. University Publications of America, Frederick, MD.

Åvald Åslagson Sommervoll and Leif Nilsen. 2021. A genetic algorithm attack on Enigma's plugboard. *Cryptologia*, 45(3):194–226.

Geoff Sullivan and Frode Weierud. 2005. Breaking German Army Ciphers. *Cryptologia*, 29(3):193–232.

Gordon Welchman. 1982. *The Hut Six Story: Breaking the Enigma Codes*. McGraw–Hill, New York.

Elaine Woo. 2013. Mavis Batey dies at 92; renowned code-breaker for Britain in WWII. *Los Angeles Times* obituary, November 23, 2013.

A Brief Guide to the Authentication of Cryptanalytic Claims

Richard B. Shapiro
Independent scholar
Massachusetts, USA
Rick@Rickshapiro.com

Abstract

Information theory provides powerful tools for the analysis of cryptographic systems. These tools may be used to discredit pseudo-cryptographic claims, validate legitimate claims, and assess whether a ciphertext is likely to be decipherable. Although information theory can be found in most cryptology textbooks, what is often lacking is practical guidance that addresses different types of classical cipher systems. This article presents information theory with a minimum of technical details, focusing on a few powerful concepts and several basic formulae. It demonstrates how to test various types of cryptographic systems. Of special concern is the difficulty of authenticating short ciphertexts. It proposes further work that would aid in the authentication of historical ciphers.

1 Introduction

How does one know that a cryptanalytic solution is valid? Might another key have produced a different message? This uncertainty most frequently arises when the ciphertext is relatively short. Fortunately, quantitative tests are available that can authenticate cryptanalytic solutions. They also provide an effective means to discredit pseudo-cryptographic claims.

This article has two principal arguments. The first is that pseudo-cryptography can be most persuasively discredited using quantitative methods rather than qualitative arguments. Typically, pseudo-cryptographic claims are repudiated by pointing to the overly free rules used in the deciphering process. While this may suffice, qualitative arguments may strike claimants as subjective and unfair. A more effective attack against such claims is achieved by demanding that claimants meet an objective standard based on quantitative measurement. Claims that do not meet this standard could then be treated as lacking authentication. However, as explained in Section 10, there are some mitigating factors. But for those claims that employ overly free methods (pseudo-cryptography), failed authentication should be treated as invalidating the claim.

Is pseudo-cryptography really a serious problem? Unfortunately, pseudo-cryptographic claims frequently arise in the popular press (Schmeh, 2012), and on some occasions have appeared in peer-reviewed publications (see Section 2). The danger posed by unchallenged pseudo-cryptography is that non-cryptologists might come to believe that the field lacks rigor, leading to legitimate cryptographic claims being questioned.

My second argument is that the quantitative analysis of legitimate cryptanalytic claims on a routine basis would be beneficial. True, the validity of many cryptanalytic solutions is abundantly clear and therefore formal validation may be unnecessary. But the validity of the decryption of shorter ciphertexts is often uncertain. As James Reeds (1977) explains, there may be "several completely different meaningful plaintexts which, when enciphered by completely different keys, result in the same cipher text." Examples of short ciphertexts include encrypted marginalia, alchemical formulae, and shorter messages such as the Zodiac Killer's Z340 ciphertext. A further benefit of quantitative analysis is that it can be used to estimate whether an unbroken ciphertext can be deciphered.

This article builds upon Benedek Láng's (2025) typology of pseudo-cryptology. He observes that some cryptographic methods are more susceptible

to pseudo-cryptographic claims, and he therefore examines claims based on the type of cryptographic system. This article extends his conceptual framework by demonstrating how quantitative authentication methods can be applied to several types of cryptographic systems. Further work is required to extend this conceptual framework to other types of cryptographic systems (for example, transposition ciphers).

Section 2 examines the most common form of pseudo-cryptography, para-steganography, which typically employs excessively free methods. Section 3 provides a brief introduction to information theory. Section 4 describes how cryptographic solutions are authenticated, and how that process can often be reduced to relatively simple arithmetic formulae. Sections 5, 6, 7, and 8 discuss the authentication of several types of cryptographic systems: "ad hoc steganography," simple monoalphabetic ciphers, polyalphabetic ciphers, and homophonic ciphers, respectively. Section 9 argues that the authentication process appropriately follows Bayesian probability principles, eschewing the frequentist probability approach. Section 10 identifies opportunities for further work and argues for a wider use of information theory.

2 Para-steganography

Cryptologists have on many occasions disputed pseudo-cryptographic claims. William and Elizebeth Friedman (1958) demonstrated that many so-called "Shakespearean ciphers" are without merit. More recently, Klaus Schmeh (2012) produced a broad survey of dubious cryptographic claims. Benedek Láng (2025) characterizes various types of pseudo-cryptographic claims. In their critiques, these cryptologists most frequently encounter a loosely regulated form of steganography. Schmeh coined the term "para-steganography" to describe the unsystematic, or barely systematic, use of steganography. For example, letters may be arbitrarily extracted from a text and rearranged to produce a "deciphered" message. The practitioners often misleadingly refer to this as "anagramming." Anagrams are more properly defined as rearrangements of short sequences of characters. In para-steganography, far greater liberties are often taken.

In a book on Shakespeare's *Sonnets*, Harvard professor Elaine Scarry (2016) asserts that Shakespeare and another sonneteer, Henry Constable, were lovers and that they embedded each other's name in their sonnet sequences. Her three-step procedure begins with her (mostly) arbitrary selection of one line of poetry from among many. In one case, she selects the line, "MY L**OVE** M**Y** T**R**UET**H** A**N**D **BL**A**C**K DISDAI**N**D **ESTA**TE." She then arbitrarily chooses certain letters in the line, shown in bold. Finally, she "anagrams" (i.e., rearranges) the letters to spell "HENRY CONSTABLE."

Of course, this practice allows far too many degrees of freedom; indeed, almost any plaintext can be extracted. Although deeply flawed, such practices have nonetheless appeared in the work of some fine scholars. A good example is a claim made by R. L. Winnick (2009), published by one of the world's leading academic presses. He argues that the name WRIOTHESLEY (Henry Wriothesley, the third Earl of Southampton) is steganographically embedded in Shakespeare's *Sonnets*. Wriothesley was the dedicatee of two of Shakespeare's poetry books and some scholars believe that he may be the young man to whom Shakespeare dedicated his *Sonnets*.

Winnick claims that it is significant that each of the letters needed to form WRIOTHESLEY appears twice in three of work's lines: a "double WRIOTHESLEY." However, the *Sonnets* has over 2100 lines, and if one calculates the probability of this occurring by chance, the calculated incidence is close to the actual incidence of three lines.[1] An alternative quantitative test is to determine the incidence of a double WRIOTHESLEY in an independent sample set of sonnet poetry not written by Shakespeare. Indeed, Winnick performed this test and found that the incidence of double WRIOTHESLEY lines was about the same as that found in Shakespeare's *Sonnets*. Thus, ironically, Winnick offers conclusive evidence against his own thesis. Indeed, this raises an essential point: when evaluating a para-steganographic claim, if one cannot show statistical significance, it is very likely inauthentic. Indeed, another plaintext may be extracted with an equal claim to validity.

Nevertheless, Winnick, undeterred, persists in his claim. He admits that it "cannot be proven by statistical means" but then offers two defenses.

[1]This can be calculated by determining the probability that any given poetry line contains two sets of WRIOTHESLEY letters out of the approximately 45 letters in each line.

First, he asserts that cryptographic claims should be understood as similar to qualitative literary judgments, such as recognizing symbols or discerning the meaning of polysemous words. This reveals the problem at the core of pseudo-cryptography: the failure to recognize cryptology as a scientific discipline.

In his second defense against the negative results of his own statistical analysis, Winnick offers what I would characterize as "special pleadings." He argues that in those lines in which WRIOTHESLEY appears, the letters are bunched closely together within the poetic lines. He also argues that the three lines that contain a double WRIOTHESLEY are of special poetic significance. In my view, these arguments, made ex post facto of his "discovery," must be seen as rationalizations. Such explanations are almost always available because some detail or artifact can always be judged to be significant.

Winnick is not the only literary critic to engage in anagrammatic speculations. Martin Dodsworth (2017) notes that eminent scholars Alastair Fowler, Helen Vendler, and Christopher Ricks have published similar claims with respect to Shakespeare's *Sonnets*. What motivates these specious claims? The *Sonnets* promises that its subject, an unnamed young man, will enjoy renown in the future, and this suggests to some that the text might by some means hide his identity. Of course, this is no excuse to practice pseudo-cryptography. Indeed, early modern English poets were wary of pseudo-cryptographic practices because they understood the limitations of anagrams. According to Dodsworth, the English only used anagrams as a parlor game.

I have considered Winnick's claim at length because its methods and logic are representative of pseudo-cryptographic claims. Indeed, his example can guide us in formulating a set of rules for evaluating and discrediting such claims. First, all claims must be subjected to quantitative testing. This may be accomplished by either (1) comparison to incidence rates in other texts, (2) a probability calculation, or (3) information theory tests as described in Section 4. Second, failed quantitative analysis cannot be justified by special pleadings because it is almost always possible to find some rationale for a desired result.

3 Information as entropy

This section provides a brief overview of some essential concepts in information theory, and a few basic arithmetic formulae needed for authenticating cryptographic solutions. Claude Shannon developed a revolutionary theory of information in the 1940s and applied it to cryptology. His theory is frequently employed by computer scientists, especially in the field of data compression. Yet it is less frequently employed by practitioners of historical cryptography.

Shannon's (1948) revolutionary insight was to use entropy to measure the amount of information in a text. Entropy is a measure of the order (or disorder) of a system. Shannon's measurement of information uses the same statistical mathematics as the measurement of thermodynamic entropy in physical systems (as described by Boltzmann). As is practiced in thermodynamics, Shannon measures information logarithmically.

Shannon also recognized that natural languages contain redundant information. For example, if every other letter of a text is missing, one can often guess the value of the missing letters. Redundancy is also apparent when we make use of the type-ahead feature while texting. Of course, this is fundamental to cryptanalysis, for without redundancy, cryptanalysis is impossible. Also, redundancy is what makes natural languages compressible. If a language is reduced to its most theoretically compressed state (a practical impossibility), we have what might be called "pure information," but is properly referred to as entropy. The difference between this pure information and the language's normal appearance is called "redundancy."

One of Shannon's (1951) contributions was to measure the redundancy of the English language. Figure 1 provides a schematic view of Shannon's conception of the order (or entropy) found in the English language. He envisioned a series of steps by which a sequence of letters that exhibits no apparent order progressively approaches a grammatically correct English sentence. The labels inside the trapezoid figure describe the level to which an English text is approximated; the messages to the right of the trapezoid are sample texts for each level of approximation.

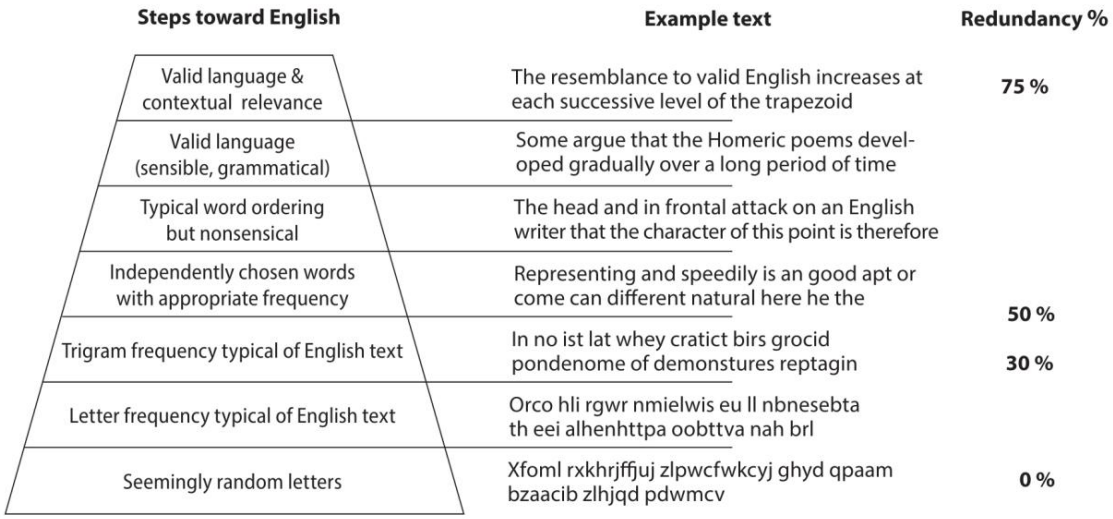


Fig. 1 Shannon Information: Successive approximations to English

At the lowest level of the trapezoid, the sequence of letters shows no apparent pattern. Such a sequence, seemingly random, cannot be significantly compressed—it approaches perfect concision and thus consists of 100% information with no redundancy. At the next level up, Shannon's first step toward English, the letters have no meaningful order but duplicate the individual letter frequencies of English.

At the third level, Shannon increased his approximation to English by duplicating trigram frequencies. The redundancy for trigrams in English is about 30%. Shannon then extrapolated from trigrams to 8-grams and reports a redundancy of about 50%.[2] This is marked at just above the trapezoid's third level.

At the fourth level, only sequences of letters that are valid English words are included, but the words lack any meaningful order. The fifth level improves the resemblance to English by mimicking English word order, but the text is still nonsensical. Not until the sixth level do we have a meaningful and grammatical English sentence. In climbing up each level of the trapezoid, the number of qualifying texts is diminished exponentially. At the sixth level, only an extremely small number of the sequences of unordered letters from the lowest level qualify as valid English.

The seventh and top level of trapezoid adds a further important qualification: contextual relevance. The sample text at the sixth level is grammatical and sensible English but it concerns the origin of the Homeric poems, a matter irrelevant to cryptology. In contrast, the sample text at the seventh level is descriptive of Figure 1 itself. This is an important distinction because when a cryptogram is deciphered, one expects the deciphered text to have some contextual relevance. Indeed, the cryptanalytic process often involves a crib which is guessed based on context.

To quantify information, we first ask how many bits are required to specify each English character. Five bits can specify a 32-letter alphabet ($2^5 = 32$); for a 26-letter alphabet we calculate $\log_2 26 \approx 4.7$ bits. This is known as "the absolute rate of language."

Shannon conducted a set of experiments in which people guessed at the letters of a text. He estimated the redundancy of English to be between 1.3 and 0.6 bits per character, out of the 4.7 bits, the absolute rate of language. This corresponds to a redundancy of 72% and 87%, respectively.[3] Others have measured it between 77% and 86% (Levitin, 721). Throughout this article, a value of 75% is applied, which falls at the conservative end of the range. Most other European languages have similar redundancy levels.

4 The authentication of cryptanalysis

We now have an estimate that English is 75% redundant and that the remaining 25% is a theoretical compression of English to its ultimate level of concision. This allows us to answer the following useful question: for an English text of n characters, how many valid and contextually

[2]See Shannon (1949, page 700).

[3]Shannon (1951, page 64).

relevant texts are there? This will allow us to calculate the probability that a plaintext arises by chance when attempting to guess at a key.

If English is 75% redundant, then for each character in English, the redundant amount is 75% of 4.7 or 3.53 bits per character. The remaining part, the information in its most concise form, is 1.17 bits per character. We now have an estimate of the number of valid and relevant English texts for a plaintext that is n characters long: $2^{1.17n}$ (exponentiation inverts the log function).[4] For a 25-letter plaintext, this is equal to approximately 760 million possible texts.

We now ask the following question: for what length message are we most likely to find that a single spurious solution arises by chance? This is known as the "unicity distance." Calculating the unicity distance provides a means to authenticate a cryptanalytic claim. If the length of the cryptogram is significantly greater than the unicity distance, then the chance of a spurious solution is slight. Shannon (1949) was the first to calculate unicity distances. The unicity distance (U) is defined by the following formula:[5]

$$U = H(K) / R \qquad \text{Formula 1}$$

"H" denotes entropy and H(K) is the entropy of the key, known as "key equivocation" or "key space." A spurious decryption could arise from any key value, and key equivocation is a measure of all possible keys that could be applied. "R" is the plaintext redundancy measured in bits per character. For English, this is the estimate of 3.53 bits per character given above. The result, U, is the number of characters in the unicity distance.

Formula 1, in effect, finds the entropy balance point between the redundant component of the message ($R \cdot U$) and the key equivocation, H(K). Unless the message redundancy is larger than the key equivocation, a spurious key may decipher to what appears to be a valid plaintext.

To demonstrate the use of Formula 1, we take as an example a simple substitution cipher. In this cipher, the key is ideally a random mapping of each of the 26 letters in the English alphabet to another one of those 26 letters. We begin by calculating the entropy of the key space or key equivocation. Assuming the key is a random sequence of the 26 letters of the alphabet, with each letter appearing just once, the number of permutations is 26 factorial.[6] The information content or entropy of the number of the keys is calculated using base 2 logarithms:

$$H(K) = \log_2 26! \approx 88.4$$

Using Formula 1, we now calculate the unicity distance:

$$U = H(K) / R \approx 88.4 / 3.53 \approx 25$$

A simple substitution cipher of 25 characters is most likely to have a single spurious solution. To validate a cryptanalytic solution, the number of characters must be greater than 25 characters. How much greater? Reeds (1977) provides a formula which promises that there will be no spurious solution, at a 99.8% confidence rate.[7] Formula 2, which Reeds derived, is used along with the values discussed above, to calculate what I call the "authentication distance" (AD) for a simple substitution cipher:

$$AD = H(K) / R + 20/R \approx 25 + 20 / 3.53 \approx 30.7 \qquad \text{Formula 2}$$

The approximately 31 characters represent an increase of about 25% above the unicity distance. Yet is Formula 2 conservative enough—one that would be appropriate to use as a standard for authentication among cryptologists? One might argue for a higher threshold, say, 50% above the unicity distance, rather than 25%. This additional length is intended to account for inaccuracies in estimating the redundancy of language and some biases found in language that our model cannot fully capture.[8] An appropriate threshold to use for authentication purposes is worthy of further investigation.

The concept of unicity distance was extended by both Reeds and Deavours to aid in the determination of whether a cipher is likely to be breakable. Suppose a cryptanalyst intends to crack a cipher using bigram or trigram frequencies, then

[4]See Deavours (1977, page 47), who presents a similar value using base 10 logarithms and exponentiation.

[5]For further explanation, see: Lasry (2018, page 35); Reeds (1977, page 235); or Deavours (1977, page 46).

[6]If the key is randomly selected, then all elements of the key are equiprobable. If the key is not randomly selected, then one must calculate key entropy on a weighted basis.

[7]That is, three standard deviations. Reeds (1977, page 238).

[8]Reeds models variations in language redundancy using a Poisson distribution. But actual redundancy, if measured empirically, might vary somewhat from his model.

one can use a different value for language redundancy based solely on bigram and trigram frequencies. Figure 1 shows that the redundancy is approximately 30% for trigrams. The redundancy per bit is then 30% of 4.7 or 1.41 bits per character. Using Formula 1 with this new value of R (instead of 3.53), we now calculate a "breakable distance" of 63 characters.[9]

The calculations of unicity and breakable distances are not hard limits—one may sometimes be able to crack a cipher that falls below the unicity distance. For example, suppose one has a strong reason to believe that a crib is present such as the name of an encrypted message's addressee. This might significantly reduce the effective key equivocation, allowing the cipher to be broken.

Unicity distance is not the only way to authenticate a cryptanalytic solution. Deavours (1977) asserts that Shannon's "unicity point formula is couched in information theoretic terms but a simpler approach is possible." He then employs standard probability calculations in conjunction with his estimate for the redundancy of English. The basic principle remains the same whether one uses Shannon's formula or probability calculations. The basic parameters are key equivocation, the number of potentially valid plaintexts, and the number of all possible plaintexts, valid or not. These three values determine the probability that a spurious decryption will arise. In the examples presented below, the authentication method chosen is the one most appropriate to the circumstances.

5 Ad hoc steganography

Section 2 discussed para-steganography, the loosely regulated selection of letters to form a plaintext. An effective way to discredit these bogus claims is to demand that the claimant provide a calculation of the key equivocation. In the case of Winnick's claim, equivocation occurs at three levels: selection of lines, selection of letters within lines, and the "anagrammatic" rearrangement of those letters. The key equivocation is so large that authentication convincingly fails.[10]

Nevertheless, some unorthodox steganographic practices may be valid—what might be called "ad hoc steganography." Rather than a steganographic system such as a Cardan grille, a short message may be opportunistically embedded. For example, poets have occasionally placed their names in the acrostic of a poem. An interesting case is that of a late sixteenth-century dialogue on love, *Contramours*, published under the pseudonym Battista Fregoso. The acrostic in a fourteen-line prefatory poem spells out THOMAS SEBILLET. It is only through this use of steganography that scholars have been able to identify the author. We can authenticate this by calculating what is essentially the key equivocation of the steganographic claim. If the volume has five prefatory poems (the likely place for a poet's hidden name), and either an acrostic or telestich is acceptable, then the key equivocation is 5 times 2 or 10, an extremely low value. Obviously, the probability of a spurious appearance of the name of a potential poet in one of these 10 locations is very remote.

Suppose we were to expand our steganographic search to include a fixed character count inward, say, from 1 to 10 characters from the poem's acrostic on each poetic line. In that case, the key space would still be relatively small. But if we were to allow the inward character count to be arbitrary and independent for each line, the key space would expand to 10^{14} (10 possible positions for each of 14 lines). This high value for key equivocation would discredit any claim that a putative plaintext is valid.

My goal here is to avoid the categorical rejection of every ad hoc or para-steganographic claim. This would likely strike claimants as arbitrary and unfair. Instead, the cryptologist critiquing such claims should demand that the claimant provide a quantitative analysis. The claimant's most essential task is to calculate what is effectively the key equivocation. This calculation must carefully consider every arbitrary path taken to produce the putative plaintext, which in the vast majority of cases will lead to an authentication failure.

Under this standard, might it sometimes be possible to validate an anagram that hides an author's name? François Rabelais published *Pantagruel* under the pseudonym Alcofribas Nasier, which is an anagram of his name (ignoring the cedilla). Rabelais' authorship is certain from historical evidence, but what if it were not?

We will attempt to validate the anagram using a probability calculation. We must calculate key equivocation, the number of all possible plaintexts, valid or not, and the number of

[9]This is close to Deavours' (1977) estimate of 55. The difference is due to different redundancy estimates.

[10]This lengthy calculation cannot be included here.

potentially valid plaintexts. First, we calculate the key equivocation of an anagram, which is the number of permutations of its letters, that is, all possible transformations. We must be careful to account for duplicate letters in the anagram (A, I, R, and S). The number of permutations is 16! / (3! · 2 · 2 · 2) ≈ $4.4 \cdot 10^9$.

Next, we calculate the number of meaningless (or not) plaintext messages. This is similar to the absolute rate of language; however, we must account for the fact that an anagram's letters are not random but adhere to the frequencies of letters in the 23-letter Middle French alphabet. This consideration is often missed by pseudo-cryptographers when they provide probability calculations. This reduces the effective alphabet size to perhaps 15 from 23 letters (i.e., the alphabet could theoretically be encoded using 15 equiprobable letters). Then the number of possible plaintexts, sensible or not, is 15^{16}, equal to approximately $6.6 \cdot 10^{18}$.

Next, we calculate the number of meaningful plaintexts. Suppose that scholars judged that only 10 writers in France potentially had the skill to write this brilliant novel. Then 10 is arguably the number of valid plaintext messages.

What is the probability of hitting upon one of these 10 valid plaintexts for any given key? It is 10 out of $6.6 \cdot 10^{18}$, or 1 out of $6.6 \cdot 10^{17}$. But we have $4.4 \cdot 10^9$ chances (the number of keys) to hit on one of these 10 valid plaintexts. The probability of a spurious anagrammatic deciphering is then the division of these two numbers, which is 1 out of $1.5 \cdot 10^8$. As a result, we can be relatively certain that the anagram is intended to be deciphered to François Rabelais.

However, there is unobvious error in my logic. Perhaps Alcofribas Nasier is merely a pseudonym and not an anagram of anyone's name. This possibility should be factored in. But even if we estimate that there is merely a 1 out of 100 chance that the author decided to anagram his true name, the authentication probability is still strong: 1 in $1.5 \cdot 10^6$. It is essential when calculating probabilities to carefully factor in all assumptions (see Section 9 on Bayesian priors).

This example illustrates that in rare cases ad hoc steganography can be valid. But this is only true if one places a restriction—a special case assumption—on what constitutes a valid plaintext. Alternatively, a severely restricted key space might allow authentication. Any claimant must carefully state all assumptions and then perform a valid probability calculation.

Dodsworth (2017) finds some evidence of the practice of hiding information in anagrams among the French (in contrast to the English).

6 Simple monoalphabetic ciphers

Is it possible to authenticate a monoalphabetic cipher of only 8 characters? If the key is short enough, then it may be, though with some uncertainty.

An enciphered marginal note appears in an early edition of Edmund Spenser's *Faerie Queene*. The note, likely written in 1597, is of interest because scholars would like to understand how a poet's contemporaries read and interpreted the poet's work (Hough, 1964). Some modern scholars believe that various characters in Spenser's fictional poem are intended to allegorically represent historical individuals. They wish to know whether Spenser's contemporaries also read in an allegorical manner.

The ciphertext of the marginal note is given below (it employs the 24-letter Elizabethan alphabet). If a Caesar shift of 12 is applied to the ciphertext (either up or down), the following plaintext is produced:

Ciphertext: YB: YRFGRE
Plaintext: LO: LESTER (Lord: Lester)

LO is an abbreviation for "Lord" and LESTER is an alternative spelling of Leicester (spelling was not standardized). Lord Leicester, Robert Dudley, was an intimate friend of Queen Elizabeth and an influential statesman. The marginal note appears next to the third line of Spenser's epic poem, which introduces the Redcross knight. Modern scholars do not necessarily identify the Redcross knight with Leicester, but apparently this early modern reader did.

Should we trust this decryption? If this were a substitution cipher, the key equivocation would be too large to allow any cryptanalysis. But as the key equivocation of a Caesar shift is extremely small (24 possibilities), a good case can be made for this decryption's authenticity. We can test that none of the other 23 Caesar shifts produce anything intelligible, but can the plaintext be quantitively authenticated? Formula 1 gives a unicity distance of about 1.3, but our measurement of the redundancy of English is not necessarily reliable for such an extremely short text.

First, we calculate key equivocation, which is simply 24, the number of Caesar shifts. Next, we calculate the absolute rate of language, which is

the number of all possible sequences of 8 letters in a 24-letter alphabet, which is equal to $24^8 \approx 110$ billion. Now we must estimate the number of valid and relevant plaintexts. Due to the extreme shortness of the message, we cannot rely on language redundancy calculations. We might choose a conservative number based on the number of 8-letter words in English, which is perhaps 10,000. This is conservative because the plaintext is a gloss on a poetic line, which sharply restricts the range of potential plaintexts.

Our probability calculation is straightforward. The chance that any given key produces a plaintext is 10,000 out of 110 billion or 1 out of 11 million. For any of the 24 keys, the chance of a spurious decryption is 24 times greater, which is 1 out of 458,000. We can be almost certain that our decryption is not spurious, even though the plaintext is only 8 characters in length.

7 Polyalphabetic ciphers

In polyalphabetic ciphers, a key is chosen and then applied successively to each letter in the plaintext to encipher it. The key is a fixed length and then used repetitively until the message is fully enciphered. Suppose the key length is 16 and each element of the key is a number from 1 to 26, chosen randomly and applied as an arithmetic shift. Suppose that we apply it to a short message of only 16 characters. This is called perfect secrecy (or a one-time pad), and no cryptanalysis is possible because the redundancy of language is entirely hidden.

We now consider a message of 21 characters, which makes a second use of the first 5 characters of the key. Can a cryptanalytic solution be authenticated?

We first calculate the entropy of the key, each element of which is a number from 1 to 26:

$$H(K) = \log_2 26^{16} \approx 75.2.$$

We next use Formula 1 to calculate the unicity distance:

$$U = H(K) / R \approx 75.2 / 3.53 \approx 21.3$$

The ciphertext length (21) is almost the same as the unicity distance (21.3) and thus authentication fails.

We next consider the case of a Vigenère polyalphabetic cipher, which employs an easy-to-remember key phrase instead of a key with randomly selected numbers. Suppose our plaintext message is HE DISCOVERED LOGARITHMS (length: 22) and our key is NAPIER WAS CUNNING (length: 16). Now we must recalculate the entropy of the key differently because it is no longer a random sequence of numbers. Further, it is contextually relevant to the plaintext: John Napier discovered logarithms.

How do we estimate the entropy of a memorable phrase key? That is, how many possible memorable phrases might be specified by the 16-character key? In Section 4, we posed the same question for plaintext possibilities, and our answer was given by calculating the pure information content of the plaintext: $2^{1.17n}$, where n is the number of characters. For n = 16, the value is $2^{18.7}$ (approximately 426,000). The entropy of the key, H(K), is $\log_2 2^{18.7} = 18.7$. Now that we have the key entropy, we can calculate the unicity distance:

$$U = H(K) / R \approx 18.7 / 3.53 \approx 5.3$$

In the case of the memorable phrase, the unicity distance drops from 21.3 to 5.3. Our ciphertext length of 22 is about four times larger than the unicity distance and thus the cryptanalytic result is easily past the authentication threshold. This demonstrates that polyalphabetic ciphers have a weakness. If a crib is successfully guessed, and crib sliding is attempted, the memorable phrase key may appear. If instead the key had consisted of randomly selected numbers, the cipher would have been far better protected.

It should be noted that the calculation of the entropy of the 16-character key is somewhat uncertain as the redundancy of English may be overstated at short lengths. According to Deavours (1977), the redundancy level of 75% for English is only reached at 20 characters or more. However, as the key includes NAPIER, it is strongly linked to the plaintext, and this compensates for the key being slightly shorter than 20 characters.

8 Homophonic ciphers

Homophonic ciphers improve security by assigning multiple cipher symbols to the same plaintext letter to defend against frequency counting attacks. We consider a system with n symbols and a 26-letter plaintext alphabet. If each plaintext letter is assigned to at least one ciphertext symbol (key space = 26!) and the

remaining symbols (n minus 26) are assigned to any plaintext letter (key space = 26^{n-26}), then the key equivocation is given by:[11]

$$H(K) = \log_2 (26! \cdot 26^{n-26}) \qquad \text{Formula 3}$$

However, this greatly overstates key equivocation because it includes many extremely unlikely key possibilities. For example, if each of 26 symbols is assigned to a single plaintext letter, it is extremely unlikely that every remaining symbol is assigned to a letter that seldom appears such "Z." This obviously makes no sense because symbols are assigned for the purpose of smoothing the frequency counts of cipher symbols. Thus Formula 3 overestimates the key space. An open question is whether it is possible to estimate, in a straightforward manner, the entropy of a homophonic cipher more precisely. To obtain a more realistic estimate of the key space, one must assume that the additional ciphertext symbols are primarily assigned to higher frequency letters.

I was unable to find any survey of homophonic ciphers that estimates the likely range of ciphertext symbol allocation. It appears that there is considerable variation in how symbols are allocated: in some homophonic ciphers, only vowels have additional symbols; in others, a greater range of plaintext letters is assigned to multiple symbols. A survey that examines the characteristics of the key spaces of solved historical homophonic ciphers might prove valuable. It would allow one to estimate the key space of unsolved homophonic ciphers more accurately. This would provide a better estimation of the unicity distance, and by extension, the chances that a cipher might be broken.

When authenticating a solved homophonic cipher, the key equivocation could be estimated based on the distribution of cipher symbols. Yet is it valid to assume a specific distribution given that that is the result that one is attempting to evaluate? In any case, although the unicity distance of homophonic ciphers has been addressed in some articles, additional work is required to more accurately assess key equivocation. The present alternative is to use Formula 3, even though it overestimates key equivocation.

The Z-340 (Zodiac Killer) cryptogram is a short homophonic cipher that demonstrates the potential difficulties in calculating unicity distance. It has 63 ciphertext symbols and is 340 characters long. Competing solutions to this cipher raised concerns about authentication, and particularly the challenges encountered in authenticating homophonic ciphers. Von zur Gathen (2023) begins by calculating an even higher estimate of unicity distance than that provided in Formula 3.[12] Significantly, he recognizes another set of problems in calculating unicity distance: the Z340 solution contains various anomalies, including spelling errors and poor grammar. He puts considerable effort into accounting for these anomalies, which have the effect of increasing the unicity distance. Von zur Gathen's analysis of the Z340 cipher serves as a valuable reminder that redundancy estimates assume (possibly inaccurately) that a clean plaintext is under test.

9 A Bayesian perspective

In reviewing pseudo-cryptanalytic claims, I sometimes found that the claimants offered probability calculations as part of their argument. Without exception, these probability calculations only considered the probability of the putative plaintext message (i.e., the message they hoped to find) spuriously appearing. This follows what is known as a "frequentist" probability model, rather than a Bayesian model. In Bayesian probability calculations, one considers "the priors," that is, all potential hypotheses. Rather than calculate the probability of any single hypothesis H, one must calculate the probability of all possible hypotheses, $H_1 \ldots H_n$. In the context of crypt-analytic validation, $H_1 \ldots H_n$ are all potential cryptanalytic solutions.

In authenticating a cryptanalytic solution, one should consider not only the plaintext message being tested but all valid plaintext messages. Shannon's authentication process is inherently Bayesian because his calculations factor in all potential plaintexts (see the second paragraph of Section 4). These potentially valid plaintexts are essentially Bayesian priors, hypotheses $H_1 \ldots H_n$.

Pseudo-cryptographers invariably calculate based on a frequentist model of probability, which poisons the calculation with circular reasoning. Instead, any authentication method must consider a full range of potential plaintext solutions. This is illustrated by the claims that discover that WRIOTHESLEY is enciphered somewhere in Shakespeare's *Sonnets* (see Section 2). These claims adopt a single hypothesis to the exclusion

[11] See Dhavare et. al. (2013, page 254).

[12] His formula is $H(K) = \log_2 26^{63}$.

of all other possible encrypted names. Shakespeare could have enciphered any one of a hundred other names into his *Sonnets* (or no name at all). When performing quantitative authentication tests, one must either use Shannon's methods, which are inherently Bayesian, or if calculating probabilities, factor in all valid potential plaintexts in one's calculation.

10 Conclusions and future work

Shannon information is a powerful tool that could be applied more widely in the study of historical cryptography. Many fields set a numerical standard for the validity of results. For example, medical research sets "$p < .05$" as the threshold for statistical significance. Might cryptologists set a standard threshold for authentication distance at, say, 50% above the unicity distance? Probability calculations could also be assigned an authentication threshold.

These standards might serve to deter or repudiate pseudo-cryptographic claims. Even if some pseudo-cryptographers are not deterred, their claims could then be challenged for failing to authenticate against the standard.

Standards would also aid in the appraisal of legitimate claims in which the ciphertext is short. If a cryptographic claim meets the standard, it may then be presented as "authenticated." True, some cryptanalysts will not feel comfortable calculating unicity distance, but they might be encouraged to seek the help of an applied mathematician or computational linguist.

Claims that fail authentication are uncertain rather than necessarily false. This is especially true when historical or literary context is present, as this cannot be captured in a mathematical model.

The calculation of the unicity distance of homophonic ciphers would benefit from further research. Knowing the most common mapping patterns of ciphertext symbols to plaintext letters might prove valuable for cryptanalysts.

A generally available tool for the calculation of unicity, authentication, and breakable distances might aid in the evaluation of unsolved ciphers. It would provide some indication of the cipher's difficulty. Perhaps these functions could be integrated into CrypTool (a widely used deciphering tool), or an independent tool could be built reasonably quickly using AI.

Shannon's extraordinary contribution to the field continues to pay dividends eighty years on.

References

Cipher A. Deavours. 1977. Unicity points in cryptanalysis. *Cryptologia*, 1(1):46–68.

A. Dhavare, R. M. Low, and M. Stamp. 2013. Efficient cryptanalysis of homophonic substitution ciphers. *Cryptologia*, 37(3):250–281.

Martin Dodsworth. 2017. The Elizabethan anagram and Shakespeare's sonnets. *The Review of English Studies*, 68(286):666–688.

William F. Friedman & Elizebeth Friedman. 1958. *The Shakespearean ciphers examined: An analysis of cryptographic systems used as evidence that some author other than William Shakespeare wrote the plays commonly attributed to him*. Cambridge University Press, New York.

Graham Hough. 1964. *The First Commentary on the Faerie Queene.* Privately published. Pages 1–2.

Benedek Láng. 2025. A Typology of Pseudo-Cryptology. In *Proceedings of the 8th International Conference on Historical Cryptology, HistoCrypt 2025*, pages 90–100. Linköping University Electronic Press.

George Lasry. 2018. *A methodology for the cryptanalysis of classical ciphers with search metaheuristics*. Kassel University Press, Kassel, Germany. Pages 35, 37–38.

L. B. Levitin and Z. Reingold. 1994. Entropy of natural languages: Theory and experiment. *Chaos, Solitons & Fractals*, 4(5):709–743. Page 721.

James Reeds. 1977. Entropy Calculations and Particular Methods of Cryptanalysis, *Cryptologia*, 1(3):235–254. Page 235.

Elaine Scarry. 2016. *Naming thy name: Cross talk in Shakespeare's sonnets*. Farrar, Straus, and Giroux, New York. Page 20.

Klaus Schmeh. 2012. The Pathology of Cryptology — A Current Survey. *Cryptologia*, 36(1):14–45.

Claude Shannon. 1948. A Mathematical Theory of Communication. *Bell system technical journal* 27(3):379–423.

Claude Shannon. 1949. Communication Theory of Secrecy Systems. *Bell system technical journal*, 28(4):656–715.

Claude Shannon. 1951. Prediction and Entropy of Printed English. *Bell system technical journal*, 30(1):50–64.

J. von zur Gathen. 2023. Unicity distance of the Zodiac-340 cipher. *Cryptologia*, 47(5):474-488.

R. H. Winnick. 2009. "Loe, here in one line is his name twice writ": Anagrams, Shakespeare's Sonnets, and

the Identity of the Fair Friend. *Literary Imagination*, 11(3):254–277. Page 257.

Only Gentlemen Read Each Other's Mail: Over 50 Years of Sittler Codebooks in the Dutch Diplomatic Service

Jip Boer
iHub, Radboud university
Nijmegen - The Netherlands
jip.boer@ru.nl

Abstract

The Dutch diplomatic service's prolonged reliance on commercially available Sittler codebooks, despite repeated indications the books had been compromised, presents an interesting case study spanning the late nineteenth century to the early 1930s. With the use of the Sittler code as its focal point, this article explores how cryptographic practices were shaped as much by social norms, administrative capacities and financial constraints as by awareness of strategic security risks.

Drawing primarily on archival material from the Dutch Ministry of Foreign Affairs, it demonstrates that continued use of the Sittler code reflected neither ignorance nor technological backwardness. Instead, it exposes a class-based, "gentlemanly", conception of secrecy in which codes fulfilled a primary function of shielding information from subordinate staff and telegraph personnel, rather than from senior officials of foreign governments. Only under the external pressures of the First World War did concerns of foreign cryptanalytic threats lead to incremental security measures.

1 Introduction

As the famous quote by U.S. Secretary of State Henry Stimson goes, 'Gentlemen do not read each other's mail', this article will demonstrate, Dutch gentlemen of the diplomatic class thought very differently of the matter. From at least 1878 to 1931, and possibly beyond, the Dutch diplomatic service relied on various editions of the commercially available 'Sittler codebooks' for confidential communications. These French-language dictionary codebooks were authored by the water company director and later postal and telegraph director F.J. Sittler (dates unknown). Within Dutch diplomatic circles, the code was commonly known as the 'Sittler code' after its author, or simply, the 'French code'. As one might expect, over the roughly fifty years in which these codebooks were in use, the Dutch Department of Foreign Affairs in the Hague had repeatedly received warnings that its codes had been compromised.

The historical considerations and attitudes towards diplomatic communication security in neutral countries has been a little studied subject. The Netherlands is an interesting case because from a modern perspective it is difficult to understand why the Dutch diplomatic service continued to rely on compromised codebooks for such an extended period. This article examines that continued use, taking the Sittler code as a case study to explore historic attitudes toward communication security among Dutch diplomats and policymakers, as well as the considerations that ultimately led to the institutionalisation of cryptologic capabilities within the Dutch state apparatus. In this article, I will argue that the answer to the question is three-fold.

In true Dutch fashion, the first and most straightforward factor was financial in nature. Producing codebooks for the entire diplomatic service was a laborious task that required a certain expertise and thus was costly. By contrast, the commercially available Sittler codebooks were relatively inexpensive. Their use also reduced the length of messages and thus the cost to transmit them telegraphically.

The second factor was the existence of a somewhat nonchalant attitude towards secrecy within Dutch diplomatic circles, especially prior to the First World War. Diplomats during this period, mostly hailing from the nobility and gentry, generally displayed little concern about their communications being known to foreign government

officials. Possible explanations for this attitude range from the idea that the neutral Netherlands was not a significant European power and had few things to hide, to the existence of a class-based conception of secrecy that meant some Dutch diplomats were primarily concerned with shielding their communications from the prying eyes of "commoners" rather than the gentlemen in senior positions of foreign governments. Evidence of this "gentlemanly" attitude will be provided, in the form of historical correspondence.

The third factor was the gradual introduction of additional security measures on top of the Sittler code, mainly during and after the First World War. Increasing geopolitical pressure threatened Dutch neutrality and positive results of cryptanalytic efforts seemingly galvanized the Dutch Department of Foreign Affairs to introduce these additional measures. Techniques such as superencipherment would come to be applied on top of the Sittler codebooks, mitigating some of their weaknesses and reducing the perceived urgency of replacing them altogether.

1.1 Sources

This article is mainly based on the archives of the Cabinet of the Minister of Foreign Affairs of the Netherlands.[1] This Cabinet was responsible for dealing with all confidential matters until the start of the Second World War. This included correspondence with Dutch diplomatic envoys and matters related to Dutch ciphers used for this correspondence. The several archival entries concerning these matters will be central to this paper. The sources consist of a somewhat eclectic, largely chronologically ordered collection of codebooks, correspondence with envoys related to the use of codes, correspondence from sellers trying to advertise new cryptographic systems, and correspondence with other Ministries' Departments discussing the establishment of an interdepartmental cipher bureau. Correspondence by diplomats was typically addressed to the Minister of Foreign Affairs, but letters were delivered to and handled by the Cabinet. Furthermore, this paper uses the shortened versions "Cabinet of Foreign Affairs" or simply "Cabinet" interchangeably.

First an important note regarding Dutch cryptologic terminology at the time. In most of the correspondence the terms code and cipher are used interchangeably. Basically, any message that was communicated in numbers could be considered to be in 'cipher'. This also led to the codebooks (*codeboek*) being alternately referred to as cipher book (*cijferboek*) or even encipher book (*vercijferboek*) by some diplomats.

This article continues with a brief description of the codebooks in question. In the sections following thereafter, it presents a chronological overview of how these codebooks were used and how thinking on their use evolved. This overview includes an examination of several of the security incidents that took place as well as the developments towards institutional cryptology within the Dutch government.

2 The Sittler codebooks

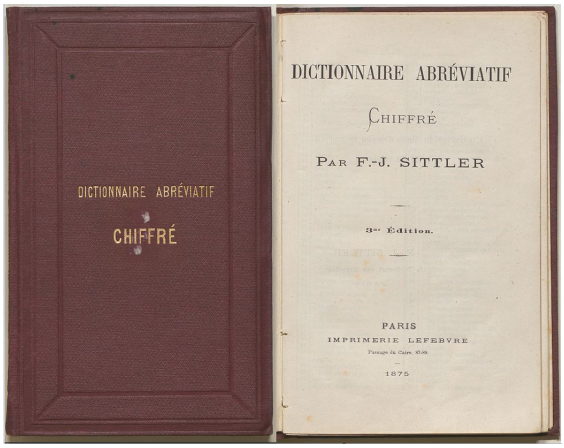


Figure 1: This 1875 third edition of the 'Sittler-booklet', used by the legation in London, is the earliest version present in the archive of the Cabinet of the Minister of Foreign Affairs; NL-HaNA-2.05.18, no. 149.

The Sittler codebooks, or *Dictionnaire Abrévatif Chiffré*, as seen in Figure 1 above, were commercially available hundred-page booklets. They were initially published by the Paris-based *Imprimerie Lefebvre* and later sold by the international book wholesalers Boyveau & Chevillet.[2] The earliest mention of a Sittler book being used for Dutch diplomatic communication can be

[1] Nationaal Archief, The Hague (NL-HaNA), Inventory of the Cabinet Archives of the Ministry of Foreign Affairs, 1871-1940, 2.05.18, nos. 149-153; Hereafter NL-HaNA-2.05.18.

[2] Abbreviated cipher dictionary; Lefebvre Printing Company. These and all following translations have been done by the author.

found in 1878.[3]

French words, or combinations of words, were encoded by taking the two assigned page numbers and following these with the two numbers assigned to each word on the page.[4] These four-digit numbers would then be written down in groups of five. Since most of the entries in the booklet exceeded four letters, encoding them in this way shortened messages, making them cheaper to send. Minor adjustments, such as changing rarely used words, to words not included in the book or changing page numbers, offered limited additional security.[5] This did have an adverse side effect, since the many different Dutch consulates and legations possessed codebooks with different paginations and thus could not communicate with one another. This resulted in larger consulates, like that of Constantinople for example, having to maintain a multitude of codebooks corresponding with those of smaller consulates in the region.

3 The long road to awareness

Before, during, and after the First World War, the Dutch Cabinet of Foreign Affairs received numerous indications, usually through correspondence with Dutch diplomats, that the codes used by the diplomatic service had been compromised or broken by foreign governments. The next two sections trace how the Cabinet and Department of Foreign Affairs, as well as the diplomats themselves, responded to these concerns, examining their views and considerations regarding communication security, the incidents that occurred, and the measures taken to secure communications further.

The period from the adoption of the Sittler codebooks up until the First World War passed without major developments in terms of improvements to procedure and security. Despite incidents regularly occurring, envoys made mistakes and at several points indications were received that foreign governments read Dutch communications. Nevertheless, improvements were sporadic and small-scale, some legations implemented security tricks like adding nulls, reversing the code when written down, using multiple sets of paginations or adding a certain number to specific groups of numbers based on a chosen lawbook or day of the year, but no major, universal, adaptations were made.[6]

3.1 Van Vredenburch's first memorandum

In 1908, the diplomat Jhr.[7] dr. C.G.W.F. van Vredenburch (1874-1927), sent a letter to the Cabinet stating that the Dutch use of the Sittler codebook was known to German, Austrian and possibly Russian officials, and that any "*bureau noir*" would make short work of them. In an appended memorandum, he proposed the implementation of additional superencipherment or self-produced codebooks for confidential telegrams unique to each legation. These propositions were promptly shot down by the Cabinet. An unknown official argued in their reply that no code was completely safe, and because the "key" (i.e. pagination), was not universal to all codebooks, only few had been compromised. The official goes on to conclude that creating distinct superencipherments or codebooks for each legation simply was too labour-intensive to be feasible.[8] This response illustrates that the Cabinet prioritised administrative convenience over absolute secrecy, accepting the risk of compromised communications as a normal part of diplomatic practice.

One reason for this lack of urgency possibly lay in a common sentiment regarding the purpose of encoding communications. As the following section will demonstrate, the perceived primary aim of using codes was not to prevent content from being known by foreign governments, but rather to shield information from subordinate officials and telegraph personnel, people who, because of their social background, were implicitly assumed to be less trustworthy and discreet.[9]

[3] NL-HaNA-2.05.18, no. 149, Cyfer voor het gezantschap te London, June 26, 1878.

[4] Meaning the 42nd word on page 55 would be encoded as 5542.

[5] Over the decades these paginations were divided in increasingly smaller segments, a codebook from 1879 was cut up in two even sections, pages 99-50 followed by pages 00-49, later codebooks, like the one used in 1921 by the Dutch legation in Warsaw, were divided up in many uneven sections, pages 36-44, 09-00, 87-99, 56-45, 10-21, 65-74, 22-35, 86-75 and 57-64.

[6] Correspondence concerning these mistakes and security tricks can be found in NL-HaNA-2.05.18, no. 149.

[7] Jhr., short for Jonkheer, is the title carried by Dutch lower nobility.

[8] NL-HaNA-2.05.18, no. 150, Letter Jhr. van Vredenburch to the Minister of Foreign Affairs (MinFA), Lisbon, March 4, 1908; Response from Cabinet March 24, 1908; Dutch institutional cryptographic expertise to perform these tasks had not been present for a time, possibly since the early nineteenth-century. See de Leeuw, 2000.

[9] For a foundational work on class distinctions and resulting social practices and judgements see Bourdieu, 1984.

3.2 De Stuers' letters

A letter by the long-time Dutch resident minister in France, A.L.E. ridder[10] De Stuers (1841-1919), presents a clear and frank example of this attitude regarding the use of codes.[11] When the First World War was still in its infancy, in September of 1914, De Stuers wrote to the Cabinet of Foreign Affairs in response to a French ban on communications in cipher by envoys of neutral nations. In this letter, De Stuers reports that he had assured Pierre de Margerie (1861-1942), the political director of the French Ministry of Foreign Affairs, the following:

> The cipher had no state secrets to conceal, it was only used to assure messages of a certain delicate nature would not be disclosed to subordinate officials and to the public of the postal offices.

In isolation, this statement might be read as an insincere reassurance, offered by a diplomat seeking to placate French officials. In context, however, it reveals De Stuers' underlying view of the main purpose of using codes. As he indicates in the same letter that he was far less troubled by French government officials reading his correspondence.

De Stuers further explains that he had gotten de Margerie to agree to a concession whereby the latter would arrange for encoded Dutch dispatches to be sent to the French ambassador in the Hague, who would have them delivered to the Dutch Department of Foreign Affairs. But, to appease the supreme commander of the French armed forces, General Joffre, who apparently was not a fan of diplomats, de Margerie requested to be informed of the nature and subject of the dispatches.[12] De Stuers found himself agreeable to the request since:

> The officials of the *Kabinet noir* (sic) - that in regular times deciphered all ciphered telegrams – have gone to war. Since the contents of our telegrams were previously known in their entirety, it matters little whether I now disclose the subject matter discussed.

He goes on to somewhat ironically add the following remark:

> Incidentally, I intend to replace our cipherbook with a new one in any case.

This letter perfectly demonstrates the aloof attitude of De Stuers at the outset of the war. He writes nonchalantly about French officials reading Dutch correspondence almost as a commonly known fact that warranted little concern. This unconcerned attitude, echoing that same untroubled point of view in the Cabinet reply to Van Vredenburch six years earlier, seems to have been typical of the Dutch diplomatic service in the decades leading up to WWI.

3.3 Growing concerns

De Stuers' letter, along with other archival material, demonstrates that this nonchalance was not a result of naïveté regarding the existence and success of foreign black chambers, either before or after WWI. Foreign efforts to decipher Dutch communications are referenced in correspondence between high government officials as well as in letters by Dutch diplomats, often in relation to a mistake having been made or in relation to having received indications their communications had been compromised.[13]

As WWI progressed, Dutch neutrality continued to be threatened and diplomatic controversies occurred (Abbenhuis, 2006; van Tuyll van Serooskerken, 2001). These external developments heightened awareness of the risks associated with insecure diplomatic communications. At the same time, Dutch General Staff cryptanalysts started booking the first results in breaking German codes in April of 1915 (Jacobs and van Kampen, 2024). The intelligence derived from these efforts likely reached the Minister of Foreign Affairs

[10] *Ridder* (knight) is the lowest noble rank after Jonkheer.

[11] NL-HaNA-2.05.18, no. 151, Letter A.L.E. ridder De Stuers to MinFA, Bordeaux, September 21, 1914.

[12] De Stuers remarks in his letter that: '[General Joffre,] like so many that know little of diplomats, must think they are nothing but official spies in an embroidered uniform'.

[13] To list a number of examples: Rijks Geschiedkundige Publicatieën (RPG) 116, pages 390-391, 570; NL-HaNA-2.05.18, no. 152, Letter Minister of War to Ministers of the Navy, the Colonies and MinFA, February 19, 1919; Letters of diplomats: NL-HaNA-2.05.18, no. 150, Letter Jhr. van Vredenburch to MinFA, Lisbon, March 4, 1908; NL-HaNA-2.05.18, no. 151, Letters ridder van Rappard to MinFA, Washington February 28, 1916 and October 18, 1916; Letter D. baron van Asbeck to MinFA, Tokio, March 9, 1915; Letter Jhr. van Vredenburch to MinFA, Stockholm, March 27, 1919; Letter Jhr. Van Weede van Berencamp to MinFA, July 2, 1919; NL-HaNA-2.05.18, no. 152, Letter P.B. Hubrecht to MinFA, Washington D.C., September 14, 1921; Letters W.A.F. baron Gevers to MinFA, Berlin April 12, 1923 and June 9, 1926; Letter J.C. Pabst to MinFA, Tokio, June 14, 1923; Letter consul in Jeddah to MinFA, Jeddah, July 4, 1923.

and the chief official of his Cabinet shortly thereafter,[14] leading them to further realise the risks associated with the evident shortcomings of existing Dutch communication security.

Many of the letters cited above include explicit questions about whether existing procedures should be revised or whether better alternatives to the Sittler codebooks were available.[15] Importantly, such concerns persisted even after additional measures were introduced, indicating a sustained shift in attitude toward communication security.

3.4 First additional measure

As far as we can surmise from the archive of the Cabinet of Foreign Affairs, in July 1915, the first additional encryption method was introduced to improve the security of diplomatic communications, initially called '*cijferstammen*' (cipherstems).[16] Because the Cabinet archive mainly contains correspondence about the cipherstems and their deliveries, it is difficult to reconstruct this first measure and its use fully, but we do know several things. The means to use this new method were mainly provided to the more important legations, usually those in capitals of warring powers. With the adoption of the cipherstems, a distinction started to be made between regular messages that required encryption with just the Sittler booklet, and "secret" messages that required an additional layer of encryption through superencipherment.

To provide this extra layer, the cipherstems were introduced. The single surviving example of a sheet with such cipherstems is shown in Figure 2. The sheet shows the same sequence of 34 numbers written by hand three times. The sequences are spaced out evenly over the page and a grid has been drawn in pencil in which the numbers appear. According to separately found instructions,

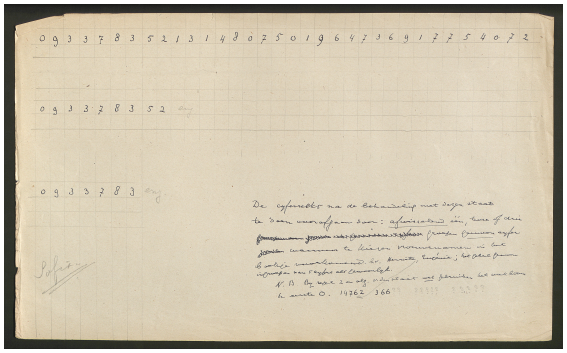


Figure 2: The only surviving example of the cipherstems (date unknown), used between 1915 and 1919. The text says: 'The series of numbers after treatment with these stems should be preceded by: alternately one, two or three groups of the regular cipher for which female names appearing in the booklet should be chosen.' In the bottom left corner, the name of the Bulgarian capital, Sofia, is pencilled in; NL-HaNA-2.05.18, no. 151.

this sequence of 34 numbers would be added in its entirety to a similarly long sequence of numbers that resulted from a regular encoding using the Sittler booklet.

The cipherstems were distributed in batches designated A, B, or C. These batches contained between fifteen and thirty-five sheets each. Unfortunately, no documentation within the Cabinet archive details the differences between the three designations. Correspondence does show a progression in their adoption, some legations first received undesignated cipherstems in 1915 (retroactively designated A). In the following years, the legations received batches of cipherstems marked B or C, always accompanied by an instruction to destroy the previously used stems. It is unclear how often a single stem was used. A 1919 letter by a diplomat suggests adopting a different sheet of cipherstems for each day of the month would further improve security, indicating each sheet differed from the others, and that they were used more frequently than once a month.[17]

As correspondence between the Cabinet and diplomats shows, the idea of using codes because of possible indiscretions by subordinate staff or telegraph personnel persisted later into the war and after it, but with one important difference. Now a clear distinction was being made between

[14] Intelligence officers made frequent visits to the chief official of the Cabinet during the war, dr. W.I. Doude van Troostwijk (1868-1957); See NL-HaNA-2.04.53.21, no. 17, Diary van Woelderen.

[15] See note 10; One such procedure was the requirement for consuls to send cleartext versions of their telegrams to the accounting department (staffed with unknown civil servants) in Netherlands through regular mail services every quarter, to account for telegraph costs made. A consul noted that these letters would be subject to censorship, giving host countries access to cleartext versions of coded communications.

[16] In the following years they would come to be alternately referred to as *cijferstaten* (cipher tables), the same name a later system would also carry. For clarity I will refer to this system only as the cipherstems and to the later system only as the *cijferstaten*.

[17] NL-HaNA-2.05.18, no. 151, Appendix letter Van Vredenburch to MinFA, March 27, 1919.

the use of the *geheimcijfer* (secret cipher) and the *gewoon cijfer* (regular cipher), the latter being employed 'exclusively to avoid possible indiscretions by telegraph personnel', and the former being employed only 'for telegrams of which the confidentiality is truly of importance'.[18] Interestingly, in contrast to other nations (Ferris, 2020; Larsen, 2017; Lasry, 2018), the Dutch diplomatic service seems to have continued using the same codebook for communications that required different levels of secrecy until 1931.[19]

3.5 Continued advocacy

The adoption of additional security measures did not stop the questioning of the use of the Sittler booklets. In 1919, the same Van Vredenburch who had criticised the use of the Sittler booklets before, wrote another memorandum to the Cabinet and the new Minister of Foreign Affairs in the Netherlands. Because Van Vredenburch had since studied contemporary cryptologic literature, he was now able to level much more substantive criticism at current Dutch practices, specifically the use of the Sittler booklets. His memorandum details the shortcomings of the Dutch encryption methods, some of the incidents that further compromised security, common cryptologic principles, the methods and techniques used by cryptanalysts, and suggestions on improvements in security. Van Vredenburch refers to the 'cipher of Sittler' as a 'one-part cipher of the simplest sort'. He goes on to sum up the reasons why 'Sittler's booklet is especially compromised':

> 1°. The booklet is commercially available, 2°. It has been in use for years, 3°. It has been in possession of representatives of foreign powers (among others in Tangiers, Caracas or Teheran), 4°. Honorary consular staff, non-Dutch persons, have been in possession of the books (among others in Bucharest, Belgrade, Lisbon, Kristiania etc.), 5°. Because almost none of the legations possesses a safe and the booklet is usually kept in a possibly unlocked drawer, 6°. Because the legations simply rip up draft papers and throw these in the refuse and do not burn them.[20]

As proof of the insecurity of the ciphers, Van Vredenburch follows with an anecdote of when he was the Dutch envoy to Vienna in 1906. There, he received a Sittler-encrypted telegram from the Netherlands containing a message for the Foreign Minister of Austria-Hungary. Before Van Vredenburch could relay this message, he had the displeasure of finding it had been decrypted and published in that day's evening edition of the *Neue Freie Presse*, a Viennese newspaper. He surmises that the Austro-Hungarian '*Chifferbureau*' had been able to promptly decrypt the message and had leaked it to the paper.[21] One could speculate that this earlier episode drove Van Vredenburch to share his earlier criticism and to educate himself on contemporary cryptology.

In the letter accompanying the memorandum, Van Vredenburch seems to plead for the establishment of more permanent cryptologic expertise, and urges the Cabinet to circulate his letter to the departments of War and the Colonies. He stressed that this had become more urgent than ever before, as Dutch communications not only had to be protected from foreign agents, but also against the agents of the '*Bolscheviki*, who are certainly no less to be feared'.[22] This concern leads to Van Vredenburch's closing remarks, in which he points to developments taking place in Sweden:

> As Sweden had unpleasant experiences with its unreliable cipher during the World War, the Commission responsible for reorganising this Department proposed that a separate office, headed by an official with the rank of chief archivist and assisted by a few clerks, should henceforth be responsible for ciphering and deciphering.[23]

In the surrounding context of his description of the development, Van Vredenburch seems to recommend a similar institutional arrangement for

[18] NL-HaNA-2.05.18, no. 151, Letter Cabinet to W.J. Oudendijk, Dutch envoy in Petrograd, May 16, 1918.

[19] After 1919, this was also true for the German diplomatic service, their codebook had the main purpose of shortening messages, in contrast to Dutch use, German communications were always superenciphered in different ways depending on their level of confidentiality (van der Meulen, 1998).

[20] NL-HaNA-2.05.18, no. 151, Appendix letter Van Vredenburch to MinFA, March 27, 1919.

[21] Ibidem.

[22] This letter was written just five months after a failed socialist revolution in the Netherlands, for additional information, see Linmans, 2024.

[23] NL-HaNA-2.05.18, no. 151, Letter Van Vredenburch to MinFA, March 27, 1919.

the Dutch Department of Foreign Affairs. Later that same year, a Dutch interdepartmental cipher bureau was indeed established, formally under the auspices of the Department of Foreign Affairs. Whether Van Vredenburch's letter played a decisive role in the bureau's establishment is unknown, but the letter demonstrates that members of the diplomatic service were actively advocating for improved communication security and the abandonment of the Sittler code.

4 The *Cijferbureau*

Following these developments with Dutch diplomatic codes as well as Dutch cryptanalytic successes during the First World War (Jacobs and van Kampen, 2024), a Dutch interdepartmental cryptologic agency practicing both cryptography and cryptanalysis was established in late 1919, becoming official in 1920 (de Leeuw, 2015).[24] This was a relatively unique arrangement for that time (Gentry, 2019). The '*Cijferbureau*' as it came to be called, was headed by H. Koot (1883-1959), a former officer that had shown a great level of proficiency in both cryptography and cryptanalysis during the First World War (Jacobs and van Kampen, 2024).[25] According to the personal journal of a high-ranking Dutch intelligence officer, the recently appointed Minister of Foreign Affairs, Jhr. H.A. van Karnebeek (1874-1942), had been the main proponent of the establishment of the *Cijferbureau*.[26] This marked a shift in thinking within the highest levels of the Dutch government regarding the necessity of establishing a more permanent cryptologic expertise.

4.1 Ramping up security

Shortly after the establishment of the *Cijferbureau*, the use of the Sittler booklets was updated to provide additional security. The earliest mention of this update can be found in September of 1920.[27] From now on, many of the booklets would have two different sets of paginations, meaning each page had two different page numbers. One was written in blue pencil, which was known as the regular cipher or blue pagination, and worked similarly as before.[28] The other pagination was written in black pencil, known as the secret cipher or black pagination. It was unique to each booklet and thus could mainly be used to communicate with the Cabinet, or with delegations to international conferences if identical codebooks were provided. In addition to this different pagination, the secret cipher would also be superenciphered using '*cijferstaten*'.[29] Like the cipherstems, this method for superenciphering seems to have initially only been made available to the larger and more important consulates, as well as attendees of international diplomatic conferences.

4.2 The *cijferstaten*

The *cijferstaten* were a similar, albeit updated and more sophisticated version of the cipherstems. They comprised two to four file folders with six couplets of rows cut out of the front page each. In between these cut-away-rows were a series of "randomised" typed numbers. Differing from the earlier system, all rows contained a differing chain of numbers, see Figure 3. They were to be used as follows: a sender would place an empty page inside the folder, in the cut-out row above the numbers, users were instructed to fill in the ciphers (single file) following an encoding with the black pagination of their Sittler booklet. Then, each number of the resulting code would be added to the corresponding number from the chain of randomised numbers below. The resulting sum would be written down in the cut-out row below the numbers. The numbers were added modulo 10, so, 5+9 would become 4. This new system was both easier to apply (by only adding single digit numbers together), and provided better security than the previous system because it was much less repetitive.

Perceptive readers might have observed that this method, if used correctly, could function as a one-time pad (OTP). This was, however, not how the system was used. Recipients of the *cijferstaten* periodically (with unknown frequency), or after a request, received two or four file folders with a total of 12 or 24 numbered lines of 32 randomized numbers.[30] Because the Sittler code ascribed

[24] The several army branches and Ministry of the Colonies had come to similar conclusions about the need to improve communications security by 1920.

[25] Rijks Geschiedkundige Publicatieën (RPG) Grote Serie 116, pages 391, 570.

[26] NL-HaNA-2.04.53.21, no. 17, Diary van Woelderen, Entry March 3, 1919.

[27] NL-HaNA-2.05.18, no. 152, Letter Cabinet to Jhr. Mr. dr. R. de Marees van Swinderen, envoy in London, September 2, 1920.

[28] Note that this pagination would be similar for some books, but was not universal

[29] Cipher tables; On occasion they were also referred to as '*optelstaten*' or addition tables

[30] The number of folders received varied, it is unclear why,

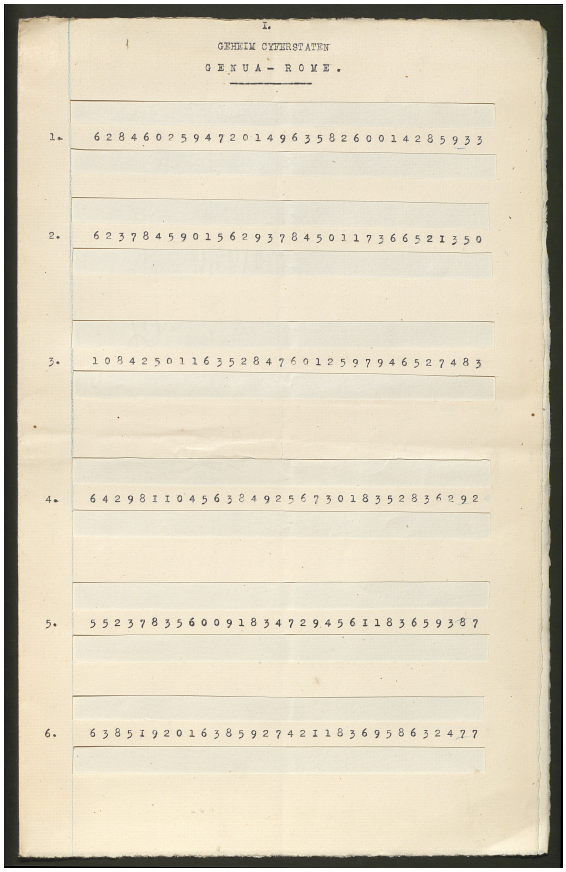


Figure 3: These *cijferstaten* were designed to superencipher communications between Dutch diplomats in Rome and attendees of the Genoa Economic and Financial Conference of 1922; NL-HaNA-2.05.18, no. 152.

four numbers to every word, this meant that without repetition, the *cijferstaten* could only superencipher a maximum of 96 or 192 words. So instead, its users were instructed to start the superenciphering with one of the 12 or 24 lines corresponding to the serial number of the telegram they were enciphering. If a serial number exceeded the total amount of lines, the starting line would be calculated modulo 12 or 24 depending on the number of folders. Meaning, if the telegram had the serial number of 55 and was superenciphered using two folders, the starting line for superenciphering would be the seventh (4x12+7=55). When a message exceeded 96 or 192 words, the numbered rows were to be reused in the same order. Correspondence between Dutch diplomats and the Cabinet suggests there possibly was a limited amount of allowed re-uses, or a limited timeframe in which they were to be used, but this amount is never specified in the letters nor in the available instruction materials.

4.3 Further incidents

The added security measures did not, however, prevent further incidents from occurring, especially not ones caused by human error. The earliest incident after the adoption of the *cijferstaten* known to us happened just a year later in September 1921. Following this incident, the temporary Dutch envoy to Washington, P.B. Hubrecht (dates unknown), wrote an apologetic, non-official, private letter to a relation within the Department of Foreign Affairs. In this letter, he describes an episode where, what he calls 'an unfortunate concurrence of events', led to a miscommunication between himself and a staff member. This miscommunication resulted in the same message being transmitted twice, once in the secret cipher, and once in cleartext.[31] He goes on to inquire whether it might be a good idea to replace the five rows of the *cijferstaten* that were used to superencipher the message, which got an affirmative reply from the Cabinet of Foreign Affairs.[32]

Although the proper adoption of the *cijferstaten* reduced the number of reports by diplomats that their communications were being read by foreign governments, they were not invulnerable. This became apparent during Dutch German trade negotiations taking place in 1926. Following these negotiations, W.A.F. baron Gevers (1856-1927), who had been critical of the continued use of the Sittler booklets before,[33] wrote that he had ample grounds to believe that the secret cipher was known to the German government. He reported receiving very clear indications that at least one ciphered telegram was known in its entirety to the German government. According to Gevers, this had negatively affected the outcome of agreements regarding seaport exemption rates.[34] This episode illustrated the very direct consequences of inse-

but it is likely that consulates that corresponded more, received more folders. When Queen Wilhelmina travelled abroad, she received just a single folder with six lines to communicate in secret.

[31] This was not an uncommon occurrence, this type of mistake was and would continue to be the most prevalent type of human error.

[32] NL-HaNA-2.05.18, no. 152 Letter P.B. Hubrecht, Washington D.C., September 14, 1921; Ibidem, Reply by Cabinet.

[33] Baron Gevers had even proposed alternative cipher systems twice before.

[34] NL-HaNA-2.05.18, no. 152, Letter W.A.F. baron Gevers to the MinFA, Berlin, June 9, 1926.

cure communications. Other than scribbled notes asking and confirming that the telegram Gevers mentioned was indeed sent in secret cipher, the response to Gevers' remarks from the Cabinet is sadly lost to us.

4.4 More partial solutions

At some point in the early 1920s, an additional, possibly universal code was adopted for consular communications.[35] The so-called 'Bibabo-code' or 'bigramcode'. Sadly, correspondence regarding the specifics of the Bibabo-code is scarce. From circumstantial mentions, it seems to have been a cipher system designed to reduce telegraph costs and allow for communications between all consulates. The Bibabo-code made use of a Sittler book without adjusted pagination to encode messages. It is likely the resulting numbers would then be transformed into letter bigrams, which could form fictive words. Interestingly, with the adoption of the Bibabo-code, the use of the regular cipher was not abolished, meaning there were now three different levels of communications making use of the same codebooks, with each individual diplomat being entrusted to weigh the costs and benefits of using a particular system.[36]

4.5 Cipher machines?

In 1923 and 1924, the Department of Foreign Affairs was approached by the director of the Amsterdam-based trading company Koopman & Co., who acted as an agent for '*Hamelin*'(sic) from the Stockholm-based A.B. Cryptograph.[37] In February 1924 Koopman proposed a meeting to be set up between Hagelin and Koot for them to discuss a cryptographic machine. Koot accepted the meeting and later that same week, it took place.[38] It is difficult to ascertain what the results of the meeting were, but we have no evidence of cipher machines being used by the Dutch diplomatic service until the Second World War. We do know however, that other parts of the Dutch government did adopt cipher machines. Karl de Leeuw has described how in 1915 the Dutch Navy adopted a cipher machine on an experimental basis for the duration of the First World War (de Leeuw, 2003). The earliest known purchase of commercial cipher machines was done by the Department of the Colonies. By the beginning of 1926, an order had been placed for an unknown number of machines produced by A.B. Cryptograph, called the 'electric cipher machine "DAMM"' by Dutch officials. This description likely refers to the two-rotor machine B13, developed by the Swedish inventor A.G. Damm (1869-1928). Even with the adoption of cipher machines by the Ministry of the Colonies in 1926, a priority of safeguarding information from subordinates was still present. In an official document to the desk of the Minister of the Colonies, Koot states:

> In my opinion, in addition to cipher machines, however excellent they may be from a cryptographic point of view, there will always continue to be a need for separate secret codes, not only for the highest government bodies, but also for the secret radio operations between the Netherlands and the Dutch East Indies.[39]

His point being that even though cipher machines might provide better security, for communications of the highest levels, it was preferable to make use of a different, easier to use, cryptographic system (a system that he himself would provide). Such a system would not require the intercession of a specialized cipher machine operator, who was presumably regarded as less discreet. Here Koot makes the (likely correct) assumption, that the gentlemen in the highest government circles could not be trusted to operate a cipher machine by themselves, requiring a more manageable system, even if that meant it was less secure.

5 The Dutch code

Starting in 1931, after over fifty years of continuous use, the Sittler booklets finally started to be phased out within the Dutch diplomatic service. The *Cijferbureau* had developed new Dutch-language codebooks. Unfortunately, none of these books have been found in this archive or other publicly available archives, and very little circumstantial information regarding their nature remains. The new books were also used to send communications in both a regular cipher and a

[35] The first mentions can be found from January 1925, but it had been established earlier; NL-HaNA-2.05.18, no. 152.

[36] Ibidem

[37] Ibidem, Letter Koopman February 4, 1924; A misspelling of Hagelin; For more information on Koopman & Co see `https://www.cryptomuseum.com/manuf/koopman/`

[38] Letter Cabinet to Koopman & Co, February 5, 1924.

[39] NL-HaNA-2.13.70, no. 1812.

secret cipher with *cijferstaten* for superencipherment. The books were issued together with a secret-cipher-condenser manual, which described how to transcribe messages into pronounceable ten-letter words, making them cheaper to send.[40] Basically, the way codebooks were used changed very little, but the books themselves were replaced. This did not yet mean the definitive end of the Sittler books however, as recipients of the new Dutch codebooks were instructed to keep their Sittler books, in case any messages necessitated the use of French.

6 Conclusions

The continuous use of the Sittler codebooks by the Dutch diplomatic service was not the result of simple ignorance or technological backwardness, but rather a collection of practical, cultural and institutional considerations that shaped policy for over fifty years. The continued use of the Sittler booklets, despite the recurring reports by diplomats that secret communications were compromised, can partially be explained through a practical lens, their low cost and the convenience of commercial availability, practical applicability by cryptographic novices, and reduced telegraph expenses. This shows that secure diplomatic communications were not seen as strictly essential for safeguarding national interests.

Equally significant was the prevailing diplomatic and class culture in which secrecy was understood as much in social as in strategic terms. Before WWI, the use of codes and ciphers primarily had the function of shielding sensitive communications from telegraph clerks and subordinate staff, not from foreign governments, whose senior officials would have likely been regarded as social peers. This attitude can help explain the tolerance for insecure communications and slow cryptographic reforms. Only under the geopolitical pressures of the First World War, when Dutch neutrality was increasingly threatened, the stakes of careful diplomacy rose, and the benefit from cryptanalysis of German codes became clear, did this perspective begin to adapt. The gradual implementation of additional security measures during and after WWI, like the different levels of secrecy, the different cipher tables and the establishment of the *Cijferbureau*, illustrates a more incremental approach to security improvements, rather than decisive breaks in practices.

This article has tried to show how cryptographic practices were shaped as much by social norms, notions of trust, administrative pragmatism and financial constraints as by awareness of strategic security risks. Practical, social and cultural considerations were continually balanced with more abstract concerns of secrecy and communication security in a changing diplomatic and geopolitical context. This case has shown it is valuable to study cryptology not merely as a technical domain, but also as a praxis of secrecy. In other words, by viewing cryptology as a way to 'do' secrecy, we find that how, when, why and where it is applied, is shaped by a myriad of factors. In turn, this can help us better understand shifting conceptions of state secrecy and the role of communications in international relations historically.

[40] This manual, as well as correspondence concerning the new codebooks can be found in: NL-HaNA-2.05.18, no. 153; The condenser is specifically described as being different from the Bibabo-code, because the condenser did not solely consist of bigrams.

Acknowledgements

The author would like to thank Bart Jacobs, Rowin Jansen and Florentijn van Kampen for their valuable feedback.

References

Maartje Abbenhuis. 2006. *The Art of Staying Neutral. The Netherlands in the First World War, 1914–1918*. Amsterdam Univ. Press, Amsterdam.

Pierre Bourdieu. 1984. *Distinction, A Social Critique of the Judgement of Taste*. Routledge & Kegan Paul, London.

Karl de Leeuw. 2000. *Cryptology and statecraft in the Dutch Republic*. Universiteit van Amsterdam, Amsterdam.

Karl de Leeuw. 2003. The Dutch Invention of the Rotor Machine, 1915-1923. *Cryptologia*, 27(1):73–94.

Karl de Leeuw. 2015. The Institution of Modern Cryptology in the Netherlands and in the Netherlands East Indies, 1914–1935. *Intelligence and National Security*, 30(1):26–46.

John Ferris. 2020. *Behind the Enigma, the Authorised History of the GCHQ, Britain's Secret Cyber-Intelligence Organisation*. Bloomsbury Publishing, London.

John A. Gentry. 2019. Selective SIGINT: Collecting Communications Intelligence While Protecting One's Own. *International Journal of Intelligence and Counterintelligence*, 32(4):647–676.

Bart Jacobs and Florentijn van Kampen. 2024. A new perspective on Dutch WWI codebreaking with its international ramifications. In *Proceedings of the 7th International Conference on Historical Cryptology, HistoCrypt 2024*, pages 135–145. Linköping University Electronic Press.

Daniel Larsen. 2017. British codebreaking and American diplomatic telegrams, 1914–1915. *Intelligence and National Security*, 32(2):256–263.

George Lasry. 2018. Deciphering German Diplomatic and Naval Attaché Messages from 1914-1915. In *Proceedings of the 1st International Conference on Historical Cryptology, HistoCrypt 2018*, pages 55–64. Linköping University Electronic Press.

Wouter Linmans. 2024. *Revolutiekoorts: Onrust en oproer in November 1918*. Athenaeum, Amsterdam.

C. Smit. 1964. *Bescheiden Betreffende de Buitenlandse Politiek van Nederland, 1848-1919, derde periode 1899-1919, Vijfde deel, 1917-1919, Eerste stuk*. Number 116 in Rijks Geschiedkundige Publicatieën, Grote Serie. Martinus Nijhoff, The Hague.

Michael van der Meulen. 1998. The Road to German Diplomatic Ciphers - 1919 to 1945. *Cryptologia*, 22(2):141–166.

Hubert P. van Tuyll van Serooskerken. 2001. *The Netherlands and World War I, Espionage, Diplomacy and Survival*. Brill Publishers, Leiden.

Carel Albert van Woelderen. 1919. *Dagboek van Van Woelderen, een medewerker van GSIII, met toegang, kopieën, 1916-1919*. Number 17 in NL-HaNA, 2.04.53.21 Inventaris van de collectie De Meijer: Binnenlandse Veiligheidsdienst (BVD). 1916-1940.

Encrypted official telegrams of the Vichy government

André Falut
French National Archives
andre.falut@culture.gouv.fr

Abstract

After the installation of Pétain's government in Vichy, telegrams sent to its administration were received by a central telegraph station, which retained copies of the messages. A large proportion of the surviving messages, now kept at the French National Archives, is encrypted. This paper provides initial observations on these messages and their context, focusing on the years preceding the full occupation of mainland France.

1 Introduction

In a note from 9 October 1915, the general staff of the French army alerted to the fact that copies of telegrams sent, received or in transit, whether clear or encrypted, were archived by telegraph stations, violating directives to destroy ciphertexts once decrypted.[1] A quarter of a century later, during another world war, the Vichy central telegraph station archived thousands of encrypted telegrams among the official messages it received or handled in transit.[2]

Following the German invasion of 1940 and the armistice on June 22, the government of Philippe Pétain established itself in Vichy, south of the line of demarcation between the occupied and unoccupied territories. To meet its communication needs, the Vichy telegraph station became a "central" station, modelled on the 103, rue de Grenelle central station in Paris, now in the occupied zone. A specialized telegraphic network was planned to link the government to regional prefectures through an interministerial station. However, in January 1943, only the head of state, the head of the government, and the ministries of Agriculture and of the Interior were connected to this station;[3] it is unclear whether the network was ever fully operational.

The decree of 28 April 1939 on wartime telegraphic correspondence had prohibited the use of ciphers or agreed-upon languages between private correspondents (*Journal officiel*, 1939: 10822). These restrictions remained in place, along with the surveillance of correspondence (telegraphic or otherwise) by the Technical Controls Service, which was instituted in August 1939 and maintained by the Vichy government for policing and intelligence purposes (Berlière, 2018: 1081-1095). The use of encryption was thus reserved to official messages.[4]

2 Encrypted transmissions under armistice terms: restrictions and margins of maneuver

It has sometimes been asserted[5] that the armistice convention required the French authorities to hand over the codes and ciphers they used to the German authorities. This assertion is not fully accurate: article 14 of the convention only stated that wireless telegraphy (T.S.F.) transmissions in the non-occupied zone would be subject to special regulations. But when, a week after the armistice, they were permitted to resume for official messages, it was indeed on the condition that any encryption of these messages would use methods that would allow the German authorities to decrypt them.[6]

Telegrams sent via cable by and to the Vichy administration did not fall under this obligation, unless they crossed the demarcation line. In the

[1] Service Historique de la Défense (SHD) [Archives of the French Ministry of Defense] GR 19 N 1418, "Notes de principe".

[2] Archives nationales de France (AN) [French national archives], F/90/14450-F/90/14519.

[3] AN, 72AJ/2235, "Note du Ministère de l'Intérieur, Service des Transmissions de l'Intérieur, aux préfets régionaux", 29 January 1943.

4 In practice, encrypted transmissions did occur outside of official telegrams sent to and by the central administration; while the cryptanalytic activities covertly maintained during that period in France are beyond the scope of this paper, communications between "PC Cadix" and posts in Clermont-Ferrand, Londres and Alger used such transmissions, as described in Bertrand (1973).

5 For example in Henri Ribadeau-Dumas (1976).

[6] AN, 19800127/3, "Comptes-rendus de séances de la Sous-Commission des Transmissions, Résultats acquis au 15 juillet 1940".

occupied half of French territory, German authorities monitored transmissions and imposed further restrictions. Transmissions across the demarcation line were particularly surveilled, limited to a list of authorized correspondents and sometimes suspended without explanation.[7] There, too, the Vichy authorities were only permitted to send and receive encrypted telegrams using methods they had submitted beforehand to the German telegram control authority (*Telegramprüfstelle*), now based at the Paris central station.[8]

Reports from the Subcommission for Transmissions of the French delegation to the Armistice commission in Wiesbaden show French representatives repeatedly requesting exceptions to this disclosure obligation for diplomatic and governmental telegrams. Thus, on 12 September 1940, lieutenant-colonel Sézerat, head of the Subcommission, declares to German colonel Negendanck that "currently, cipher telegrams for our diplomatic posts are sent through foreign wireless stations or the English cable network, because we are not allowed to transmit telegrams the Germans cannot decipher through French wireless stations".[9] Two weeks earlier, Negendanck had pointed out an incident he described as a double infraction: a telegram using an undisclosed code had been sent from Rabat to Dakar via Berne and the English station of Ongar. Sézerat, initially suggesting that the message might be a hoax, later explained that it should have been transmitted by the Casablanca-Dakar cable, but had to be sent wirelessly due to an interruption. As it used an undisclosed code, it could not be transmitted by French stations, hence the use of a foreign station.

The request to allow diplomatic transmissions to use undisclosed encryption methods seems to have been granted by the beginning of October 1940. However, while the need for Vichy government members to transmit genuinely secret messages during trips to Paris, previously expressed by Sézerat, was stated twice again by the French delegation's president, general Paul Doyen, that request seems to have been definitively rejected in April 1941.[10]

[7] AN, 72AJ/2235, "Délégation française pour les Transmissions auprès du MBF".

[8] *Ibid*, "Compte-rendu du 10 juillet 1941".

[9] AN, 19800127/3, "Comptes-rendus de séances de la Sous-Commission des Transmissions".

[10] *Ibid*, "Note pour la direction des services d'armistice à Vichy", 30 April 1941.

The Subcommission's reports also contain indications on the codes and encryption keys disclosed to the German authorities, as well as details of their demands. On 31 July 1940, for example, Sézerat handed a copy of the *Code des Colonies* to Negendanck, who asked for a list of the stations that would use this code as a prerequisite to its authorization. On 12 September, Sézerat provided the current encryption keys used with this code. Telegrams that the German authorities could not decipher using disclosed methods were often handed to the Subcommission, which sent them to French services for decryption and handed back the plaintext result, along with information on the encryption method. On 28 October, Negendanck requested further information on an encryption method, as the note by the French Cipher Service (*Service du Chiffre*) accompanying a recent "translation" did not fully clarify how it was performed. On occasion, the reports also provide details such as the use of "KESAKO" (an alternative spelling of a colloquial interjection meaning "what is it?") as a cipher key.

In these reports, discussions of codes and ciphers generally focus on systems used for military and diplomatic messages. This focus is likely related both to the interest of German authorities for such messages, and to the likelihood that they would be sent wirelessly outside of metropolitan France, providing opportunities for interception. But some Vichy administrations frequently sent and received encrypted telegrams as part of their communications with mainland stations via cable. This was the case of the Ministry of the Interior, which we will examine further below.

3 A variety of encryption methods

Very little remains of the Vichy central station telegrams prior to March 1943.[11] The reuse of earlier telegrams (including encrypted ones) for new messages in a context of paper shortage, evident in the summer of 1944, accounts for part of the lacunae, but other factors probably contributed to the loss of most of the 1940-1942 messages. Only 17 days of received telegrams have been preserved for the period between 9 May 1941 and 1 November 1942, with between one and a few hundred telegrams for each day. While sparse, these records attest to the use of

[11] AN, F/90/14450 to F/90/14453.

various encryption methods by different administrations.

Encrypted telegrams are primarily addressed to the Ministries of War, the Navy, the Air, Foreign Affairs and the Interior. For each recipient administration, over the course of a single day, messages from a single origin may display more than one encryption system. For example, on 13 July 1941, the Admiralty in Vichy received from the Navy in Casablanca encrypted messages in 5-letter groups with a seemingly random distribution of vowels and consonants, 6-letter groups with an equal proportion of vowels and consonants and 5-number groups. Cleartext indications prohibiting the wireless transmission of some messages, such as "A TRANSMETTRE UNIQUEMT PAR FIL" ("TO BE TRANSMITTED VIA WIRE ONLY") or "EN AUCUN CAS PAR TSF" ("IN NO CASE VIA TSF") are likely to indicate undisclosed encryption methods, if only through their correlation with higher requirements of secrecy.

We found plaintext versions of encrypted messages in two cases. In the first case, that of telegrams from the French military attachés in Ankara,[12] Bucarest[13] and Budapest,[14] the forms used for plaintext transcription indicate that the messages are "encrypted with a cipher" (*chiffré*) rather than with a "disclosed code" (*code déposé*). The corresponding encrypted messages consist of 5-number groups and are much shorter than their plaintext versions, possibly due to the use of a code as part of the encryption method; we have not identified this method so far.

4 Theory and practice of encryption at the Ministry of the Interior

The second case is that of the Ministry of the Interior. Here, in addition to plaintext versions of encrypted messages and to the codebook used for most of them, instructions regarding encryption can be compared to its actual practice.

Prior to November 1942, these transmissions emanated mainly from department prefects in the non-occupied zone. Starting in March 1941, registers from the Ministry's Cipher Service (*Bureau du Chiffre*) provide plaintext versions of most of these messages (a few are either absent or left untranscribed with the indication "Texte réservé" replacing the transcription).[15] These messages often relate to policing and repression, answering warrants for surveillance, arrest, and internment in camps, or requests for information on strangers and suspected political enemies sent by the central administration.

The transcription forms provide four options to indicate the encryption method: "simple", "secret", "very secret" and "personal".[16] Most of the messages are described as using the "simple" and, from 1942 onwards, the "secret" system. The "very secret" system is used much less frequently, and only a few messages, none of which seem to be preserved in the collection for this period, are described as using a "personal" system. In 1942, the "simple" system becomes "secret"; "simple" now describes the original version of Code 23, a method based on a codebook in which punctuation marks, words or syllables (sorted in alphabetical order) are each replaced by a 5-letter code (also in alphabetical order).[17] For example, "acceptable" is encrypted as "bamik", and the next word, "accepter/acceptation", as "bamil"; further in the codebook and in the alphabet, "instruction" is encrypted as "kaduz", "suspicion" as "remiz".

On 1 June 1940, circular letter n°1180 prescribed replacing this very simple system by one in which each word or syllable in the codebook was encrypted as a 5-number group, combining three arbitrary numbers attributed to each page and two numbers corresponding to a line in the page.[18] On 25 April 1941, a new circular letter instructed prefects to further encrypt messages by using an additive key. On 1 March 1941, an instruction from the Cipher service had already issued reminders on best practices for the composition, encryption and transmission of encrypted messages, stressing the importance of avoiding repetitions or formulaic phrasings.[19]

In practice, most of the preserved telegrams sent to the Ministry of the Interior during this period still use the circular letter 1180 system

[12] SHD, GR 7 NN 2 1160, telegrams from 21 April 1942.

[13] SHD, GR 7 NN 2 1161, telegram from 23 April 1942.

[14] SHD GR 7 NN 2 1162, telegrams from 21 and 23 April 1942.

[15] AN, 19790067.

[16] Until the beginning of April 1942, the transcription forms provide additional categories, left unused: "diplomatic", "diplo-secure", "commercial" and "armistice", the latter presumably for methods disclosed to the German authorities.

[17] A copy of the codebook can be found in AN, 19940278/1.

[18] AN, 19940278/1.

[19] *Ibid.*.

without an additive key, as in this message sent by the Vaucluse prefect on 10 May 1941:

> "17811 25037 25727 21473 31210 44157 65775 61888 19143 94139 23412 23439 62338 37252 65355 31852 65355 33962 65355 73165 34077 79520 17811 14789 31715 97060 53238 68558 68596 34037 97856 36789 49367 25039 36760 21479 47916 25043 73165 96819 79541 64738"

The first three numbers in each group refer to a page in the codebook. In the first group, the reference "178" directs to page 43, and the two last numbers "11" give the relevant line: line 11 of page 43, "*étranger*" ("stranger"). The next group begins with "250", one of several possible references (alternated to avoid repetitions) for a flyleaf of grammatical variants and punctuation codes, where line 37 is "plural": the two groups together mean "strangers". Fully decrypted, the message reads:

> "strangers appearing your telegram number 3472 from 10 May neither detained nor interned nor knowing (*sic*) stranger service my department. However searches have been ordered and will let you know result if positive"[20]

This message is fairly typical of the encryption practices observed in these transmissions. Despite the recommendations of the Ministry's Cipher Service, weaknesses such as repetitions (e.g. the three occurrences of "65355" for "*ni*", "neither"/"nor") and formulaic phrases ("21473 31210" for "your telegram" and "14789 31715" for "my department" are very frequent in the corpus) are common. By April 1942, the numbers designating each page of the codebook had changed, but still preceded the same line numbers. Due to this permanence, the aforementioned weaknesses and the continued absence of additive key use in most messages, new page references are easy to find.

Messages in the original, "letter" version of Code 23 were sent by the prefects of Bordeaux and Chalon-sur-Saône, in the occupied zone, on 22 and 25 April 1942. Additionally, from February 1942 onwards, some "very secret" messages bear a "machine" stamp;[21] no corresponding telegrams remain for this period, but later ones (for instance, from the prefect of Tulle on 25 March 1943[22]) appear as 5-letter groups with a low proportion of vowels; the nature of this new method could be the subject of further research.

5 Conclusion

Until November 1942, while the official telegraphic transmissions of the Vichy government could be restricted to encryption methods disclosed to the German authorities when material and geopolitical conditions made monitoring and interception possible,[23] these restrictions did not apply universally, nor to all potentially intercepted messages. This complex status of encrypted transmissions during the first years of the Vichy government is reflected in the remaining official telegrams of that period.

Following the invasion of the southern zone, the number of Vichy authorities receiving encrypted telegrams was significantly reduced: most of them were addressed either to Pierre Laval or to the Ministry of the Interior. While these messages seem less varied in their form, they form a much denser collection, and, together with earlier Vichy telegrams, are likely to be useful sources for historical cryptology.

Acknowledgements

The author would like to thank Adrien Silvestre for kindly providing tools to perform semi-automated frequency checks, David Kenyon for his insightful observations on the distribution of vowels in the messages, and his colleagues at the Service Historique de la Défense and the Archives nationales for their valuable help in accessing relevant archives. Thanks are also due to Camille Desenclos and to the anonymous reviewers for their helpful comments and suggestions on this paper, the final version of which was thus greatly improved.

[20] Literally and group by group: "étranger-pluriel-figurant-votre-télégramme-numéro-trois-mille-quatre cents-soixante-douze-du-dix-mai-ni-détenu-ni-interné-ni-connaître-participe présent-service-étranger-mon-département-point-toutefois-recherche-2e mot ligne-pluriel-ont-été-prescrire-féminin pluriel-et-vous-faire-futur-connaître-résultat-si-positif".

[21] AN, 19790067/1/4.

[22] AN, F/90/14455.

[23] That is to say, in the cases of wireless transmission and transmission through occupied territory.

References

Archives nationales de France, France, Pierrefitte-sur-Seine (AN). 19790067, 19800127/3, 19940278/1, 72AJ/2235, F/90/14450-F/90/14519.

Jean-Marc Berlière. 2018. *Polices des temps noirs: France 1939-1945*. Perrin, Paris.

Gustave Bertrand. 1973. *Enigma ou la plus grande énigme de la guerre 1939-1945*. Plon, Paris.

Journal officiel de la République française. Lois et décrets. Année 1939. Imprimerie des journaux officiels, Paris.

Henri Ribadeau-Dumas. 1976. Essai d'historique du chiffre de l'armée de Terre, 5ème partie. *Bulletin de l'A.R.C, nouvelle série*, 4:16-52.

Service Historique de la Défense, France, Vincennes (SHD). GR 19 N 1418, GR 7 NN 2 1160-1162.

Collaboration and Collation: Breaking German diplomatic ciphers in 1942

Dermot Turing
Kellogg College, University of Oxford, UK
dermotturing@btinternet.com

Abstract

German diplomatic signals enciphered in the Floradora double-additive system were broken at the end of 1942 as a result of Anglo-American collaboration. The American contribution was a novel and ingenious system of machine search for likely additive combinations, which allowed the choice of additives to be revealed. The paper suggests a reconstruction of the machine methodology, which used punched-card sorting and collation.

1. Introduction

Much has been written about codebreaking in World War II. The focus has predominantly been on military and machine ciphers, notably the Enigma machine; but little attention has been given to diplomatic signals of this era, or to cryptography using superenciphered codebooks which was the system of choice at the time of the conflict, for Allied, Axis and neutral powers alike. Notable exceptions to the gaps in the literature on non-machine systems are the work of Alvarez (1997; 2000) and Denniston (1995) on diplomatic signals, and of Donovan (2004) and Simpson (2011) on breaking superenciphered codes.

Encrypted diplomatic signals are at risk of being perceived to be of secondary importance to military ones in wartime. Yet major strategic understandings were built from decrypted diplomatic signals in World War II. The most famous examples come from the breaking of Japanese diplomatic traffic. Filby (1995) gives an example of the excitement caused in intelligence circles from one particular Floradora decrypt. But, as yet, no systematic analysis of the surviving German diplomatic decrypts appears to have been done.

Another area which has received scant attention in the cryptology literature is the main technique deployed by breakers of code-and-superencipherment systems, especially when attacked using punched card technology. The work of Christensen (2014), Christensen and Antrobus (2015) and Budiansky (2001) ventures into this terrain, and there is the first-hand account by Whelan (1995) of punched-card machinery used at Bletchley Park, but beyond acknowledging the importance of this type of equipment little is said about the specific algorithms, processing methods or machine-programming used for codebreaking. The problem of non-declassification of relevant documentation dogs the researcher; and the problem is compounded by the difficulty that present-day researchers are unlikely to have personal practical experience of punched card machinery which began to fall out of use in the 1960s.

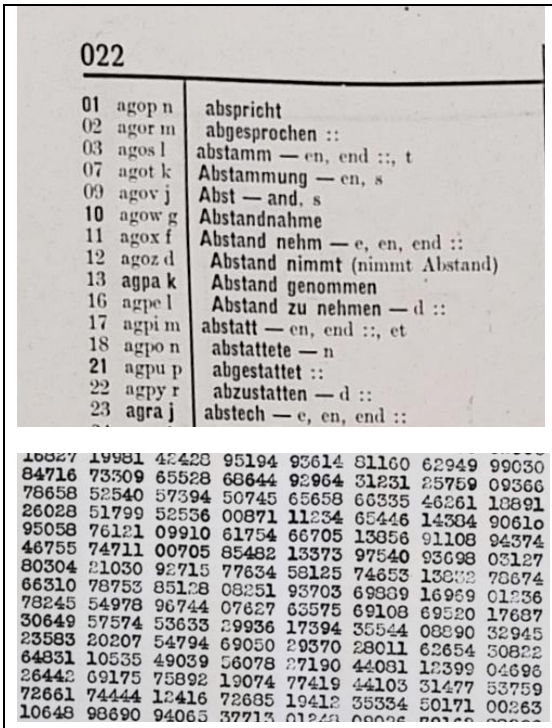
This paper aims to make a small inroad into this largely unexplored territory. German diplomatic signals presented a difficult challenge to the Anglo-American codebreakers because of the complexity of the system used. The encryption used by the Auswärtiges Amt (German Foreign Office) was based on a 'non-hatted' codebook,[1] the Deutsches Satzbuch (DeSaB). The code-groups found in the DeSaB were encrypted by superencipherment, that is to say by adding (by non-carrying or modular addition) numerical groups in a superencipherment table. Two principal superencipherment systems were used: a machine-generated one-time-pad, and a static book of 'additives' deployed in a novel and challenging way. This paper looks only at the

[1] In a 'hatted' book the numerical groups representing plain-text words and phrases are randomized. In a non-hatted book the groups are in numerical order in step with the plain-text. The word 'hatted' is a nod to placing raffle-tickets in a hat and picking them out at random.

second of these systems, called the 'Grundverfahren' by the Germans, and codenamed 'Floradora' by the British and 'GEC' or 'Keyword' by the Americans assigned to the task of decryption. By the spring of 1943, using punched card machinery, the Floradora system had been mastered. In his account (1995), P.W. Filby, who led the Floradora effort towards the end of the war, said that the system was 'the most important diplomatic cipher'; the breakthrough was, accordingly, of great significance. It was regarded by the Americans as 'the most important German problem in respect to volume of traffic and value of intelligence' (ASA, 1946, p. 83), at least until the one-time-pad system was solved in 1945 (as to which see Filby (1995) and Phillips (2000)).

In this paper, an attempt is made to reconstruct the technical-algorithmic approach used to solve the Floradora system with punched card machinery. The solution was made much easier through a collaborative approach between the U.S.-based Signal Intelligence Service (SIS) at Arlington Hall in Washington, D.C., whose machines and technical expertise were used, and a U.K.-based diplomatic codebreaking team in Berkeley Street, London: 'the cooperation with GCCS [the U.K.'s Government Code & Cypher School, of which the Berkeley Street operation was a branch] was of great importance in speeding the work,' according to the history of the SIS (ASA, 1946, p. 86). Technical collaboration became essential because of a breakdown in relations among the British codebreakers over access to punched card technology and prioritization of cryptanalytic tasks.

The paper is presented as follows. The Floradora system is described, together with a summary of the problem which it presented to the codebreakers. A very brief outline of known use of punched-card techniques for cryptanalysis during World War II is then given. The British difficulties in using punched cards are outlined, together with the rescue of the situation through cross-Atlantic collaboration and an American breakthrough in machining methodology which enabled the solution. Much of this has been described before (Filby, 1995; ASA, 1946). But the existing descriptions suggest that the algorithm 'ran a possible crib through 50,000,000 possibilities in two hours' (ASA, 1946, p. 87). If that statement is taken at face value, it implies a processing rate of almost 7,000 cards a second, far beyond the speeds of any punched-card technology available in 1942. This paper then attempts to unravel the mystery of supersonic card-processing and to suggest a reconstruction of the machine technique involved.



022

01	agop n	abspricht
02	agor m	abgesprochen ::
03	agos l	abstamm — en, end ::, t
07	agot k	Abstammung — en, s
09	agov j	Abst — and, s
10	agow g	Abstandnahme
11	agox f	Abstand nehm — e, en, end ::
12	agoz d	Abstand nimmt (nimmt Abstand)
13	agpa k	Abstand genommen
16	agpe l	Abstand zu nehmen — d ::
17	agpi m	abstatt — en, end ::, et
18	agpo n	abstattete — n
21	agpu p	abgestattet ::
22	agpy r	abzustatten — d ::
23	agra j	abstech — e, en, end ::

84716	73309	65528	68644	92964	31231	25759	09366
78658	52540	57394	50745	65658	66335	46261	18891
26028	51799	52536	00871	11234	65446	14384	90610
95058	76121	09910	61754	66705	13856	91108	94374
46755	74711	00705	85482	13373	97540	93698	03127
80304	21030	92715	77634	58125	74653	13802	78674
66310	78753	85128	08251	93703	69889	16969	01236
78245	54978	96744	07627	63575	69108	69520	17687
30649	57574	53633	29936	17394	35544	08890	32945
23583	20207	54794	69050	29370	28011	62654	30822
64831	10535	49039	56078	27190	44081	12399	04696
26442	69175	75892	19074	77419	44103	31477	53759
72661	74444	12416	72685	19412	35334	50171	00263
10648	98690	94065	37713				

Figure 1: Excerpt from the DeSaB and a set of typical Additives

Note: Additives shown for illustrative purposes only; not considered to be from the Tangenstafel.
Sources: National Archives and Records Administration, US (NARA) HCC Box 46 Nr 226; The National Archives, UK HW 53/55

2. The Floradora Problem

The Floradora system differed from the typical superencipherment scheme. Descriptions of the system are to be found in Erskine (2003), van der Meulen (1998), the first-hand accounts by Adolf Paschke (1957), P.W. Filby (1995) and his erstwhile section-head, Patricia Bartley (1942). The elements were as follows:

- The DeSaB itself was not a problem. As an unhatted book, it had been largely solved by the Americans before the war

(ASA, 1946, p. 82). That will have been no surprise to the Germans, who used the DeSaB without superencipherment for non-secret traffic, and themselves said that 'it was not regarded by the German Foreign Office as being secret' (Thoem, 1945). Both the SIS (American) and British codebreaking services acquired copies in 1940 (ASA, 1946, p. 82; Filby, 1995). DeSaB 3, in use at the beginning of the war, was replaced by DeSaB 4, but again this had been compromised. Thus, the cryptanalytic problem was in the superencipherment.

- Instead of adding a single five-digit numerical group to the five-digit code group, two groups were chosen from the same superencipherment table (the 'Tangenstafel') and both were added. Doubling-up the added cipher groups in theory squared the number of possible combined additives: the table was made up of 10,000 lines of additive with six groups per line, giving 100 million potentially available combined-additive lines – an enormous search space likely to defeat anything but the most sophisticated cryptanalysis. A page of additive lines was also acquired by the British in 1940, consisting of 50 lines.[2] A most significant feature of the additive tables was that lines 5001 to 10000 were repeats, in complementary form, of lines 1 to 5000: so, if line 3000 began with the group 67228, line 8000 began with 43882.

  Patricia Bartley discovered the complementary character of the additive tables (Filby, 1995) and explained its purpose herself (Bartley, 1942). Adolf Paschke, the German official responsible for diplomatic cryptography, later confirmed the reason for the complementarity (Paschke, 1957, p. 39): namely, to simplify decryption.

- Each of the two lines of additive selected by the message creator was notified to the message recipient by means of an 'indicator' system which encrypted this vital information. The system had two components: a 'Kenngruppe' and a bigram. Each component directed the code clerk to a four-digit number. The identification number of the line of additive to be used – the indicator – was the sum of these two numbers. The scheme of the encryption system, and the full details of the bigram component, were known to the Allies from a set of documents acquired in 1940 (ASA, 1946, p. 85). That left unknown only the four-digit number, known as the 'Schlüssel', referenced by the Kenngruppe.

The indicator system and the process for encryption used in Floradora are described in more detail in the Appendix, together with an explanation of the complementary nature of the Tangenstafel for decryption.

During the course of 1942, reconstruction of the missing additive lines took place through the fortuitous alignment of messages in depth, and by extrapolation from runs using strings of known additive. Such progress, however, did not allow for rapid and complete decryption of intercepted signals. Without knowledge of the 'Schlüssel', identifying which of the additive lines was being used for a particular signal remained obscure.

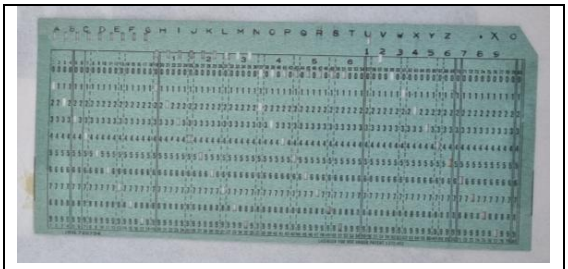


Figure 2: A punched card

Source: NARA RG457 HCC Box 1425 Nr 4692

3. Punched card machinery

Whelan (1995) gives the most comprehensive description of the uses of punched card machinery as an aid to cryptanalysis at Bletchley Park. Various machines were used: card punches, sorters, tabulators, collators and so on. Whelan explains how standard IBM data cards (measuring 180×82.5 mm and

[2] ASA (1946), p. 85, says it was 50 lines, which agrees with Bartley (undated 1943 report, UK National Archives file HW 53/22); Filby's (1995) account says 40 lines.

containing 80 columns each punchable in 12 places; see Figure 2) were used to record data such as the content of intercepted messages.

The paper also describes selected procedures used to solve specific code-breaking problems. These include 'code group and cypher group differencing', locating tetragraph repeats across different naval Enigma messages; finding 'depths' (cases where a cipher sequence was used to encipher more than one plain text signal) and so forth. These techniques deserve to be fully researched and explicated. Unfortunately the use of punched-card machinery for anything other than business accounting applications is weakly represented in the literature, so the challenge for cryptological research is twofold, requiring understanding both of the machinery, its capabilities and limitations, and of the cryptanalytic problems targeted by machine processes.

Application of pre-war punched card technology to cryptanalysis was not unique to Bletchley Park. American and German codebreakers made extensive use of punched card machinery, and the Germans themselves assessed the vulnerability (Frowein,1945) of their own machine cryptography in terms of the punched-card resources likely to be needed to break their systems.

Despite this ubiquity, however, little attention has been given by historians to the methodologies and algorithms used to tackle complex cryptanalytical problems of this era. The principal usefulness of the technology was to attack superenciphered codes, which were pervasively used by Allied and Axis powers alike for both military and civil purposes.

4. World War II: Berkeley Street v Bletchley Park

Bletchley Park's punched-card unit had its headquarters on site in Hut 7 (later, in Block C) and a local outstation at Drayton Parslow. Whelan (1995) describes the organization, with a powerful array of machinery, 500 staff and consuming 2 million cards a week. The unit was headed by Frederic Freeborn, who, according to witnesses and corroborated by archival material,[3] ruled his small empire like an autocrat (Simpson, 2011, p. 139; Greenberg, 2022, p. 234). Freeborn disliked being told which of the many demands on his section should be prioritized, or being told anything at all by another section head who was female. This state of affairs was bound to lead to a clash when a new card-processing proposal was submitted by Bartley for the solution of Floradora in May 1942.

> 'Because of its size, the problem without machinery is for all practical purposes, insoluble, and labour on it wasted. Unless some kind of priority can be given it in Mr. Freeborn's section, therefore it might as well be dropped altogether. Mr. Freeborn is of the opinion that he cannot allot enough labour or machinery to attack it thoroughly and that if he were to carry out our proposals as originally made it would take him 9 months.'[4]

The problem was large because (it seems) that Berkeley Street wanted to compile a catalogue of all 100 million combinations of additive, which could be tested against intercepts to see which combination was in use in any particular case. Freeborn might have been forgiven some scepticism as to how useful the catalogue would be, even if he could produce it. However, Bartley had a second source of expertise to call upon. This was the punched-card team working on the Floradora system at Arlington Hall, with whom she had been in liaison since before the attack on Pearl Harbor. Bartley expressed scepticism over Freeborn's time estimate, for 'the U.S.A. have already done a similar operation … and with far less machinery and several false starts took only 3 months to do so, it would perhaps be possible for Mr. Freeborn to cut down his estimate by a few months.' The Americans, apparently, had better algorithms than Bletchley Park.

The impasse between Berkeley Street and Bletchley Park continued. As Filby remarked as late as December 1943, 'Machining has always been a major trouble at B. St; the machines are at Bletchley and it is not easy to

[3] UK National Archives files HW 14/26, HW 14/33, HW 14/35, HW 64/67, HW50/72.
[4] Patricia Bartley, untitled report, 31 May 1942. UK National Archives file HW 53/44.

bridge the gap of fifty miles.'[5] A gap of 3,500 miles would prove easier to bridge. By December 1942, the missing-Schlüssel problem was on the threshold of solution:

> 'The Americans, however, have now produced a scheme by which they can test combined adder assumption by machine without making up the 100,000,000 index of combined lines. They sent us a description of this scheme, but after discussing the matter very fully with Mr. Freeborn… Mr. Freeborn calculated that a single assumption by this method would take up to eight hours to test… However, Captain Johnson now tells us that the U.S. are able to test one possibility in one and a half to two hours…'[6]

The American team had had a breakthrough. Their system was indeed able to carry out the test in two hours. By the following February even Freeborn reported that Floradora was proceeding 'at great pace far exceeding all expectations'.[7] Something remarkable, not explained in open British files, had happened.

5. Sorting and collation

The following discussion attempts to reconstruct the algorithm used to simplify the search for the Schlüssel, devised and implemented at Arlington Hall. Finding the Schlüssel was broken into the following stages. First, if the actual additive lines could be identified for a single message, the components used to build the indicator for the additives used in that message could be analysed to glean the element unknown to the codebreakers, namely the 'daily' Schlüssel. Once the Schlüssel was revealed, the indicators in all other messages would be readable for the applicable two-day period, allowing straightforward decoding of them all.

5.1 Finding candidate additives

Finding the additive lines for a given message exploited the concept of cribbing[8] or predictable code-groups. 'Stereotyped phraseology is the rule… testing is based on our assumption of what a given message will say in the first two groups or the last two groups' (Reynolds, 1945, p.36). The search is then for a likely pair of additive lines which would, when combined with the crib, yield the observed intercepted signal.

Expressed algebraically:

$$G = P+A_1+A_2 ,$$

where G is the intercepted group, P is the plain group (crib) taken from the DeSaB, and A_1 and A_2 are the additives taken from the two lines identified by the indicator. G and P are both known (in the case of P, assumed), so that

$$G\text{-}P = A_1+A_2 ,$$

and G-P is a value easily calculated, which we can call V, a constant for all possible additives under test.

Clearly $V\text{-}A_1 = A_2$, so a search through all values of $V\text{-}A_1$ to find a matching A_2 ought to yield a candidate[9] pair of additives which might be in use for the message. Hence, the British cryptanalysts imagined the search space to be huge: for each of the 10,000 values of $V\text{-}A_1$ all 10,000 values of A_2 would need to be compared, meaning 100 million comparisons (or 50 million, if the complementary nature of the second half of the additives table is taken into consideration).

5.2 Avoiding 50 million comparisons

The American process was designed by Lt Stephen Dunwell. The surviving description, which is very brief, is as follows (Reynolds, 1945, p.37):

> 'The cryptanalysts therefore assume a meaning for a pair of groups and subtract the appropriate code groups from the cipher, which results in combined additive. This is sent to the machine section for testing to determine the two additive lines which when added together match the assumption. The total possible number of combinations

[5] Filby report for Lt Col O'Connor, 23 December 1943. The National Archives UK file HW 53/44.
[6] Untitled report by Bartley, 3 December 1942. The National Archives UK file HW 53/44.
[7] Weekly report, 25 February 1943. The National Archives UK file HW 77/10.
[8] The term 'crib' was used by British cryptanalysts to denote a probable plain-text word or code-group – in other words a guess at the unenciphered raw material underlying the enciphered intercept.
[9] Only a candidate, since there are theoretically 10 ways (for each digit in a group) to combine two additives to construct V.

> of additive lines is 50,000,000. By extending the first eight digits of the uncombined additive lines in one deck of 10,000 cards, and sorting them so as to add the assumption to each line, and then collating them against another deck containing the 10,000 additive lines, it is possible to perform the equivalent of 100,000,000 additions of additive in a two hour period with one machine operator.'

Although far from exhaustive, this passage contains significant clues to the process which might in fact have been adopted. The central machining components are:

- two decks of cards punched with single-line additive
- one deck *sorted* 'so as to add the assumption to each line'
- *collating* the one deck against the other.

If a deck of cards punched with the values of A_2 is reordered, so that the smallest[10] values are first, and a second deck of cards with the values of $V\text{-}A_1$ is similarly reordered, a method of comparison is possible to circumvent a need for the brute-force testing of 100 million additions.

No archival material yet researched gives detail on the scheme used for punching the relevant data into cards for this specific problem. However, following the general methodologies set out by Whelan (1995), one may surmise the following. Whelan explains that of the twelve hole positions in each column, known as X, Y, 0, 1, 2, …, 9, for numerical values only the 0-9 hole positions were used. Four of the 80 columns of the standard IBM card would be punched with the number of the line of the additive table from which the card's additive data has been taken (a four-figure number): so a card containing additive from line 3354 would be identified with holes punched at positions 3, 3, 5 and 4 in adjacent columns in a 'reference' section of the card. Likewise, a further eight columns would be punched with the actual additive values found in the first eight digits[11] of that line of the additive table.

$V\text{-}A_1 < A_2$	$V\text{-}A_1 = A_2$	$V\text{-}A_1 > A_2$
Reject A_1 to A_1 bin Retain A_2 Test next $V\text{-}A_1$	Pass both cards to 'possible match' bin	Reject A_2 to A_2 bin Retain A_1 Test next A_2

Figure 3: Logic Table

5.3 Collator procedure

The comparison was done by a collating machine, described by Reynolds (1945, Appendix) as 'a machine for merging or selecting cards. The principal applications include merging operations where two files of cards are merged together, and selecting, where particular cards are selected from a file.' In a post-war document, the Army Security Agency (a successor organization to the SIS) described the collator as follows:

> 'Cards are introduced through two hoppers, selection is accomplished by means of the plugboard and the finished product, a combined file, emerges. If the Sorter is the dealer, then this is the arranger of the hand into suits.'[12]

Cards from the decks in the two hoppers of a collator can be read by pairs and compared. The values read in are handled by a selector unit, which directs the cards to output bins according to whether the values match, or pass or fail some test hardwired into the unit. It seems that selector units could be rigged via the plugboard to accommodate bespoke tasks (IBM,1961).

As reconstructed, the testing principle envisages that each stack of cards is ordered so that the values punched into the cards present to the collating machine in ascending order. Thus, as a card from one or other stack arrives

[10] Or largest; the point is for them to be in ascending or descending value order.

[11] Reynolds gives no explanation as to why eight digits were used. It may be that the choice related the the108 permutations existing in a randomly-chosen eight-digit decimal number to the 108 additive combinations. To avoid too many 'false matches' a minimum number of digits has to be taken, so the choice may have been a compromise between processing time and an excessive proportion of false hits (falling with every additional test digit).

[12] 'The Role of the IBM at the Army Security Agency' (undated but likely about 1946). National Archives and Records Administration US RG 457 Entry A1-9032 (Historic Cryptographic Collection or 'HCC') Box 1425 Nr 4692.

in the machine to be tested, the value of additive (or in the case of the A_1 cards, the value V-A_1) might be lower, equal to, or higher than that on the other card. The machine is set up to reject the lower-value card of the pair currently being tested, while the higher-value one of the pair is retained to test against the next card in the opposite deck (see Figure 3). A brute force approach, by contrast, would require each card to be tested against each other, because a matching value may be anywhere in the deck. The logical approach of value-sorted cards means that no match could exist for a rejected card.

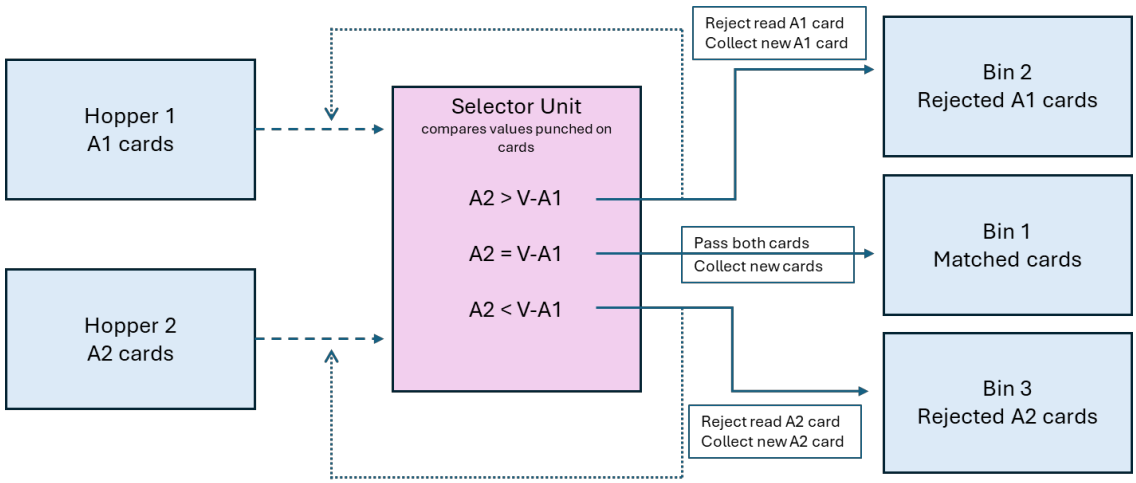


Figure 4: Schematic representation of Collator action

Source: Author

5.4 Preparation of card decks

The decks of cards could have been constructed as follows. One deck comprises simply the unaltered additive, with the cards rearranged into ascending order of punched value.[13] (.) The other deck needed to test the values of V-A_1. This could, in theory, have been done by creating a new deck of cards with the value of V-A_1 punched in place of A_1, but that would have been wasteful: an unmodified deck can be reused again tomorrow for another test, whereas a modified deck is useless after the test has been carried out.

An alternative, and less wasteful, scheme would be to hardwire value V into the selector unit of the collator which made the logical switches set out in the logic table above. The value V would be added automatically by the selector to the value punched on the A_1 card being read, giving a comparator of V+A_1 for the A_2 card being read. But, as discovered by Bartley, $+A_1$ for the additives in line 5001 is the same value as $-A_1$ for the additives in line 0001, so there is no need to create a subtraction procedure: all the values of V-A_1 will be tested simply by testing the whole deck if the machine can add V to the value punched into the card each time an A_1 card is to be read.

That then leaves the question of how to ensure that the A_1 deck cards are correctly ordered, with the cards yielding the lowest values of V+A_1 presented first. There is in fact a simple way to do this when the unmodified A_1 deck is sorted. With a 'normal' sort, cards bearing eight-digit numbers will be sorted according to the rightmost digit first, the sorter putting all cards having 0 in the eighth place into the 0 bin, those with 1 in the 1 bin, and so on. When all cards have been through this process, they are collected in order and the machine now sorts according to the seventh place, so that all cards with 0 in the seventh place go in the 0 bin, et cetera; but due to the first sort, the first

[13] Cards will have been labelled and separately punched so that their index location in the additive table will not be lost on reordering

cards to reach the bin will be those with 0 in the eighth place; the 0 bin will then have cards with the sequence ~00 at the bottom, then ~01, ~02 etc on top. Eight passes of the cards are necessary to complete the sort.

The machine operator could, however, carry out the reordering so that the lowest values of $V+A_1$, rather than the lowest absolute values of punched data, present first in the reordered deck. This can be done by changing the order of removal of cards from the bins after the first pass of cards through the sorter. Instead of taking the cards from the 0 bin to be handled first on the second pass, followed by those from the 1, 2, 3 etc bins, the order of stacking can be Caesar-shifted according to the value V so that the net result after adding-in V yields zero.

This can easily be seen with an example. Let us assume that the value V is 52098447. After the first pass through the sorter, all cards in the 3 bin will have a 3 punched in the eighth place. All these cards, when the 7 is added from the eighth place of V, will in practice be read by the collator as if 0 were in the eighth place on the card (because 7+3=0) – which is to say, the lowest value. All cards in the 4 bin will in practice read as 1, and so forth. So, before the second pass through the sorter, the cards will be ordered with the 3-bin cards to go first, then the 4-bin, and so on with the 2-bin cards going last. On the second pass through the sorter, it is the cards in the 6 bin which yield 0 when combined with the seventh digit of V, so these are picked to go first through the third sort. And so on, the Caesar-shifts involved being the complement of V (58012663), and the instructions given to the machine-operator to take first the 3-bin, 6-bin, 6-bin, 2-bin, etc, cards for successive passes). After the final pass, the cards will be in ascending order of value $V+A_1$.

5.5 Overview of collator process

So, to summarize, it is suggested that two decks of cards, one sorted according to the plain values of additive in ascending order, the other sorted according to the values of additive plus what Reynolds calls 'the assumption' also in ascending order, were placed into the two feeder hoppers of the collator. If the collator took about two hours to process the decks, that implies a processing rate of 83 cards per minute (cpm) for each deck, well within the 240 cpm capability of the IBM 077 machine pictured in Reynolds's paper—the nature of the Floradora test, requiring several tests on individual cards from one stack or the other, presumably reduced the processing rate below the 240 cpm theoretically achievable for simpler runs.[14]

6. Conclusion

The collation process separates from the deck pairs of cards representing additive lines believed to constitute the message's encipherment. Additional groups of intercept can be tested using further additives from the same line to check that meaningful plain-code groups emerge, and thereby allow the analyst to reject false pairs of cards. The cards identify the rows of additive chosen from the Tangenstafel, and by subtracting from these the known numerical equivalents of the bigrams included in the signal, the values of 'Schlüssel I' and 'Schlüssel II' are recovered. With these, the indicators of all messages for the period in which those Schlüssel values are in use can be interpreted.

It is not certain that the process described here is indeed the one that was used, nor is it guaranteed that machine programming could adapt the operation of a collator to achieve the add-in of the value V during the logical operation of the selector. However, it seems possible that the machine could either be wired to do this or that a peripheral unit could have been added, much in the way that IBM machinery was rigged by William Friedman's team to assist in the decryption of the Japanese 'Purple' cipher (Rowlett, 1998). It is possible that still-classified materials on punched card operations at Arlington Hall might confirm or condemn the suggestions set out in this paper.

The investigation carried out on Floradora machining does, however, indicate that reconstructions of the complex machine processes and algorithms in use during World War II are possible. Whelan's paper (1995) points toward some more of these algorithms,

[14] It is possible that the two hours mentioned in the source material refers to time for sorting. With four sorters working at 400 cpm under an efficient operator, sorting might have been done in under two hours. But it is difficult to fit both processes into a two-hour time frame.

as do other archival materials.[15] Perhaps, with further research, some clawback out of undeserved obscurity might be possible for these interesting yet poorly understood cryptanalytic techniques.

Acknowledgments

The author wishes to thank the President of Kellogg College, Oxford for a Visiting Fellowship, and the anonymous referees of this paper's draft for their constructive comments.

References

ASA [Army Security Agency]. 1946. *History of The Signal Security Agency. Vol 2: The General Cryptanalytic Problems.* https://www.nsa.gov/Helpful-Links/NSA-FOIA/Declassification-Transparency-Initiatives/Internal-Periodicals-Publications.

David Alvarez. 1997. No Immunity: Signals Intelligence and the European Neutrals, 1939-45. *Intelligence and National Security,* 12(2):22-43.

David Alvarez. 2000. *Secret Messages – Codebreaking and American Diplomacy, 1939-1945*. University Press of Kansas, KS, US.

Patricia Bartley. 1942. *The German Diplomatic Cyphers*. Unpublished. The National Archives UK, HW 53/5 and HW 53/38.

Stephen Budiansky. 2001. Codebreaking with IBM Machines in World War II. *Cryptologia,* 25(4):241-255.

Chris Christensen. 2014. The National Cash Register Company Additive Recovery Machine. *Cryptologia,* 38(2):152-177.

Chris Christensen and Jared Antrobus. 2015. The Story of Mamba: Aligning Messages Against Recovered Additives. *Cryptologia,* 39(3):210-243.

Robin Denniston. 1995. Diplomatic eavesdropping, 1922–44: A new source discovered. *Intelligence and National Security,* 10(3):423-448.

Peter Donovan. 2004. The Breaking of the JN25 Series of Ciphers 1939-1945. *Parabola*, 40(2):1-9.

Ralph Erskine. 2003. From the Archives: What the Sinkov Mission Brought to Bletchley Park. *Cryptologia,* 27(2):111-118.

[15] For example, 'Report on Cross-footing Pre-sensing Gangpunch Machine', HCC Box 1009 Nr 3174; 'General Breakdown of I.B.M. Unit' HCC Box 939 Nr 2729.

P.W. Filby. 1995. Floradora and a Unique Break into One-Time-Pad Ciphers. *Intelligence and National Security,* 10(3):408-422.

Joachim Frowein. 1945. *Report on Interrogation of Lt. Frowein of OKM/4 SKL III, on his Work on the Security of the German Naval Four-wheel Enigma.* TICOM Report No I-38. Unpublished in complete form. National Archives and Records Administration US, RG 457, Entry P-4, Box 1.

Joel Greenberg. 2022. *The Bletchley Park Codebreakers in their own words.* Greenhill Books, Barnsley, UK.

International Business Machines Corporation [IBM]. 1961. *Punched Card Data Processing Principles. Section 4: The IBM Collator.* https://ibm-1401.info/PunchedCard_Section-4.pdf.

Michael van der Meulen. 1998. The Road to German Diplomatic Ciphers -1919 to 1945. *Cryptologia,* 22(2):141-166.

Adolf Paschke. 1957. *Das Chiffrier- und Fernmeldewesen im Auswärtigen Amt: Seine Entwicklung und Organisation.* Unpublished. Politisches Archiv, Auswärtiges Amt, Berlin, Nr. VS-6025.

Cecil Phillips. 2000. The American Solution of a German One-time-pad Cryptographic System (G-OTP). *Cryptologia*, 24(4):324-332.

C.N. Reynolds. 1945. *SIS History Machine Branch.* Unpublished. National Archives and Records Administration US, RG 457, Entry A1-9032 (Historic Cryptographic Collection), Box 1019 Nr 3247.

Frank B. Rowlett. 1998. *The Story of Magic.* Aegean Park Press, Laguna Hills, CA, US.

Edward Simpson. 2011. Solving JN-25 at Bletchley Park: 1943-5. Chapter 9 and Appendix V in Ralph Erskine and Michael Smith, eds., *The Bletchley Park Codebreakers*. Biteback Publishing, London, UK.

Wilhelm Thoem. 1945. *On German Diplomatic Ciphers*. TICOM Report No. I-56. https://archive.org/details/ticom/TicomI-56/mode/2up.

Ronald Whelan. 1995 (undated but date inferred). *The Use of Hollerith Punched Card Equipment in Bletchley Park.* The National Archives UK, HW 25/22.

Appendix: Floradora in detail

Indicator Procedure

The codebreakers used the term 'indicator' to refer to the message key, i.e. information pointing to the two lines of additive used for encipherment. The identity of the lines of additive was concealed in the following way.

The message preamble contains a 'Kenngruppe' followed by eight letters. The Kenngruppe is used by the code clerk to find two four-figure numbers (called 'Schlüssel I' and 'Schlüssel II') from a secret table – the one component of the Floradora system not acquired by the Anglo-American codebreakers. Say these numbers are 3915 and 2280.

The eight letters comprise two bigrams, formed as a reverse-repeated quartet such as QGHA AHGQ. Each of the component bigrams QG and HA refers to another four-figure number in a second secret table. This table had been acquired by the Allies. Say these numbers are 7441 and 2690. (This part of the scheme was called the 'Spalierverfahren'.)

So now the message recipient has two pairs of numbers, which are added using non-carrying arithmetic (where 7+8=5, not 15):

Schlüssel I	3915	Schlüssel II	2280
Bigram QG	7441	Bigram HA	2690
Total (indicator)	0356	Total (indicator)	4870

The totals indicate the row numbers of additive, taken from the additive tables (called the 'Tangenstafel') to be used in enciphering the plain text.

Encryption Procedure

Imagine the secret signal is 'Spain propose attack [on] Gibraltar'. The code clerk looks up the associated code groups in the 'DeSaB' dictionary. The additive groups from the Tangenstafel are added, using non-carrying arithmetic, to arrive at the transmitted group:

Plain text	Spain	Propose	Attack	Gibraltar
Plain code groups	72731	29550	36139	88421
Additive from row 0356	80304	21030	92715	77634
Additive from row 4870	79246	91505	87297	01884
Trans-mitted group	21271	31085	05031	56839

Complementarity of the additive

Adding the 'complement' of a number has the same effect, in non-carrying arithmetic, as subtraction. A number's complement, when added to the first number, yields 00000. An attractive feature of complements is to simplify subtraction. For example:

91254 – 18846 = 83418 = 91254 + 92264.

Each half of the Tangenstafel comprises the complement of the other half. So, if the additive 18846 is located in line 2000, its complement 92264 will be found in line 7000. The decoding clerk can add 92264 to the group received instead of subtracting 18846.

Source: Bartley (1942).

The Encrypted Notes of Murderer Petras Dominas

Klaus Schmeh
Schmeh.org
klaus@schmeh.org

Abstract

Petras Dominas was a Lithuanian-born criminal active in Germany and the Netherlands during the 1960s. His violent series of robberies—including two murders—was documented extensively in contemporary media. Dominas left behind a corpus of encrypted writings consisting of roughly twenty volumes of ciphertext, created between 1952 and 1964. His manually constructed encryption system, comprising more than one thousand distinct symbols, employs homophones as well as glyphs representing entire words or syllables. Although visually reminiscent of shorthand, the cipher was designed not for speed but for secrecy, and its complexity posed substantial challenges to cryptanalists. After multiple failed attempts by various experts, the system was ultimately deciphered by Dieter Bäusch of the German Zentralstelle für das Chiffrierwesen (ZfCh). Based on exclusive materials provided by the German Federal Intelligence Service (Bundesnachrichtendienst, BND), this paper presents the first public examination of Dominas's encrypted texts. The recovered German plaintext reveals not a diary, as sometimes claimed, but detailed accounts of Dominas's criminal activities, notes on his legal representation, and a range of personal reflections and narrative fragments.

1 Introduction

In 2015, while conducting a Google search using the keywords "encryption" and "crime", I came across an article published in 1966 by the German weekly magazine *Der Spiegel* (Der Spiegel 1966). The article discussed the secret writing system of a criminal named Petras Dominas.

According to the article, during his career as a murderer and robber, Dominas kept extensive encrypted notes—amounting to roughly twenty volumes in total. He used a cipher that he apparently invented himself. The article stated that deciphering his writings proved to be a difficult task. An illustration accompanying the article shows an excerpt from Dominas's encrypted notes (Figure 1). However, the article provides no description of Dominas's encryption method, and I was unable to find further information online.

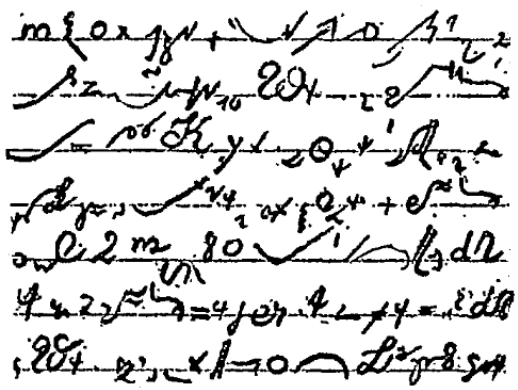


Figure 1. This excerpt from Petras Dominas's encrypted notes was published in *Der Spiegel* in 1966.

However, via the German Federal Intelligence Service (Bundesnachrichtendienst, BND), I received extensive documentation regarding Dominas's cipher and its eventual decipherment. The present work is based primarily on these materials obtained from the BND.

2 The Murderer Petras Dominas

Petras Dominas was a Lithuanian by birth and worked professionally as a typographer. He was born around 1929 and is believed to have died around 1972. Little is known about his life prior to 1959. In that year, Dominas was convicted in Germany and sentenced to four years in prison. Following his release, he quickly returned to criminal activity, embarking on a violent series of robberies during the 1960s that spanned both Germany and the Netherlands.

Between 1963 and 1964, Dominas and his accomplices carried out a string of armed robberies, primarily targeting jewelry stores. The operations were characterized by their precision, boldness, and frequent use of firearms. His criminal record includes the following incidents:

- *19 November 1963 – Bochum, Germany*: Robbery of a jeweler; loot valued at 30,000 Deutsche Marks. A female passerby was shot in the abdomen.
- *3 January 1964 – Witten/Ruhr, Germany*: Robbery of a jeweler; loot valued at 73,000 Marks. A pedestrian was shot in the thigh.
- *28 February 1964 – Dortmund, Germany*: Robbery of a jeweler; loot valued at 97,000 Marks.
- *1 April 1964 – Essen, Germany*: Robbery of a jeweler; loot valued at 30,000 Marks.
- *20 April 1964 – Gelsenkirchen, Germany*: Robbery of a jeweler; loot valued at 30,000 Marks.
- *2 May 1964 – Siegen, Germany*: Robbery of a jeweler; loot valued at 115,000 Marks.
- *7 June 1964 – Amsterdam, Netherlands*: Robbery of a jeweler; loot valued at 60,000 Marks.
- *31 July 1964 – Gelsenkirchen, Germany*: Robbery of a jeweler; loot valued at 115,000 Marks. An exchange of gunfire with police occurred during the escape.
- *25 August 1964 – Lünern near Soest, Germany*: Armed robbery of a truck stop; amount of stolen goods unknown. Two people were shot dead.
- *26 September 1964 – Amsterdam, Netherlands*: Robbery of a jeweler; loot valued at 380,000 Marks. Shots were fired at two passersby.

The June 1964 robbery in Amsterdam was witnessed by a woman and her daughter. In their statements, they reported that one of the three assailants bore a resemblance to a well-known Dutch television personality, quizmaster Theo Eerdmans. Although Eerdmans had no connection to the crime, his photograph was compared with images in criminal archives, which ultimately contributed to the identification of Petras Dominas.

Dominas was arrested one day after the second Amsterdam robbery, bringing an end to his series of violent crimes.

3 Dominas's Encrypted Notes

According to the said *Der Spiegel* report, the German criminal police (Kripo) first discovered encrypted notes belonging to Petras Dominas as early as 1959. Among the possessions of Dominas's attorney, investigators also found a notebook written in cipher. Over time, more than a dozen additional encrypted notebooks were uncovered in various hiding places connected to Dominas.

These notebooks—ordinary school exercise books and ledgers—were distinctively decorated. Dominas covered their outer pages with children's drawings and pressed flowers, lending them a deceptively innocent appearance. The series of encrypted manuscripts is believed to have been created between 1952 and 1964.

At first, the police were unable to decipher the writings. Even the Federal Criminal Police Office (Bundeskriminalamt, BKA) in Wiesbaden failed to make progress. Believing initially that the texts might be based on a modified form of

shorthand, investigators enlisted Gottlieb Jahn, a Krefeld specialist in stenographic systems from Eastern Europe. His efforts were unsuccessful.

Next, the police turned to Mansour Elias Mansour, a Jordanian merchant and sworn court interpreter in Düsseldorf, hoping his expertise in Arabic scripts might provide clues. Mansour, however, could not identify the writing. Members of the Jewish community in Düsseldorf were also consulted, but they confirmed no resemblance to Hebrew or related alphabets.

Next Dr. Ludwig Franzheim, former head of the Cologne Customs Criminal Institute and a specialist in invisible inks, was brought in. His analysis confirmed that no secret ink or chemical encoding methods had been used. Meanwhile, investigators scoured the libraries of every prison in which Dominas had ever been incarcerated, searching for books that might have inspired his system. They found nothing.

At one point, the public prosecutor in Dortmund presented Dominas with samples of his own secret writings and asked for a translation. Dominas reportedly replied, “Pay me 1,500 dollars, and I’ll read you a few pages. You’ll never find the code.”

Ultimately, according to *Der Spiegel*, the case was referred to the “Office for Deciphering Affairs” (Dienststelle für Dechiffrierwesen) in Bad Godesberg-Mehlem—most likely the Zentralstelle für das Chiffrierwesen (ZfCh), the Federal Republic’s central cryptographic authority at the time. There, an unnamed expert undertook the task of decryption. After four months, he achieved a breakthrough; the full decipherment took approximately eight months in total.

Dominas’s background likely contributed to the sophistication of his cryptographic system. Born in Lithuania, he was noted for his high intelligence, exceptional memory, and strong linguistic abilities—he spoke Lithuanian, German, English, Polish, and Dutch. After the war, he had trained in Brussels as a typographer, a profession demanding meticulous attention to symbols and structures.

Over nearly a decade, Dominas refined and expanded his system. By its final form, it contained more than a thousand unique symbols, all of which he memorized. No practice sheets or cipher keys were ever found. Expert assessments described his work as “a unique achievement in the field of cryptography.” *Der Spiegel* concluded unequivocally: “There is no doubt—Petras Dominas, aged 37, devised one of the most sophisticated secret scripts ever confronted by police experts.”

When the decoding was finally completed, the prosecutor read aloud several passages from the decrypted notebooks in Dominas’s presence. Dominas appeared surprised and reportedly remarked, “Those are just notes for a detective novel.”

The available sources do not specify what role, if any, the deciphered texts played in the legal proceedings against Dominas.

4 The BND Files

After reading about the Dominas case in *Der Spiegel*, my curiosity was piqued, and I wanted to learn more. A contact of mine suggested that the Federal Intelligence Service (Bundesnachrichtendienst, BND) might hold information related to Dominas. At the time of Dominas, the aforementioned ZfCh was under the authority of the BND.

In June 2023, I therefore contacted the BND via email, requesting access to any available materials concerning the secret writing of Petras Dominas. After nearly two years of waiting, in March 2025 I finally received two extensive PDF files from the agency.

The first PDF file contained a 223-page collection of documents titled “Tagebücher des Verbrechers Petras Dominas, gelöst von Dieter Bäusch” (“Diaries of the criminal Petras Dominas, deciphered by Dieter Bäusch”). It is

likely that Dieter Bäusch was the same expert from the ZfCh mentioned in the Spiegel article.

The first document in this collection (page 4) is a letter dated September 17, 1970, from Bäusch to Dr. Erich Hüttenhain, then head of the ZfCh. The purpose of the letter was to submit the following materials to Hüttenhain.

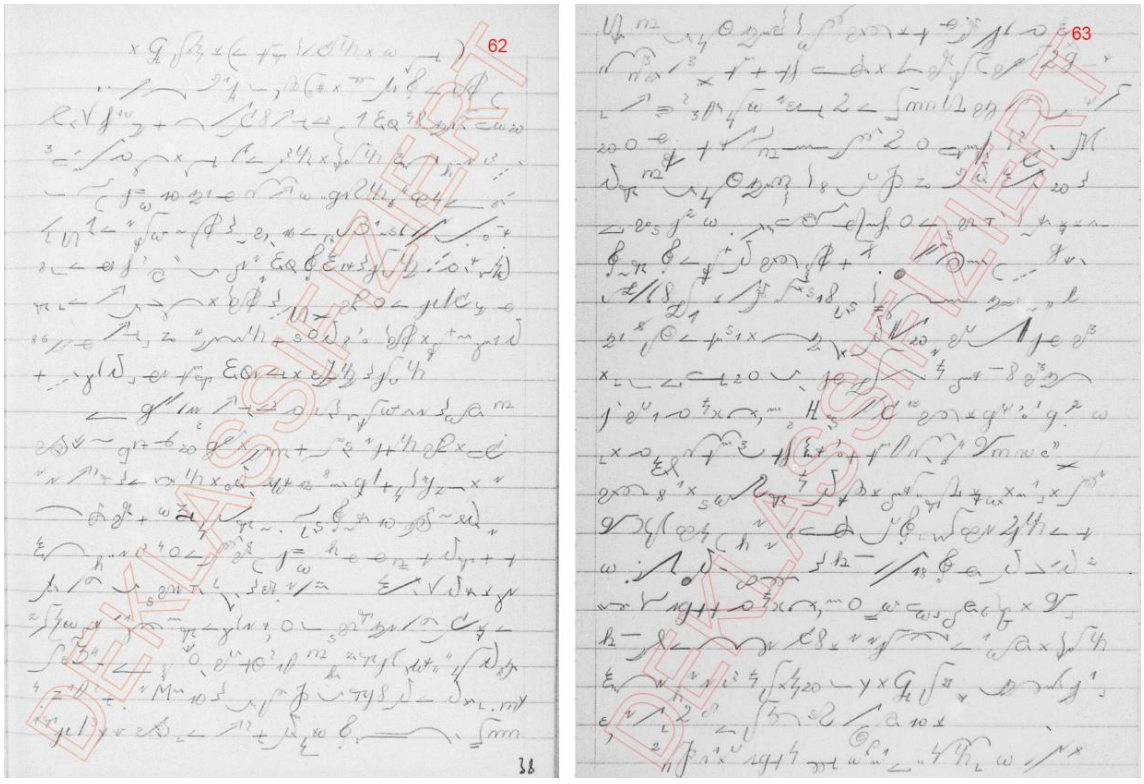


Figure 2. Two pages from the encrypted notes of Petras Dominas

Pages 5–111: A handwritten treatise titled “Die Entzifferung der Geheimschrift des Dominas” (“The Decipherment of Dominas’s Secret Writing”).

Pages 112–163: Another handwritten document titled “Die Geheimschrift des Petras Dominas (bearbeitet von Dieter Bäusch)” (“The Secret Writing of Petras Dominas, edited by Dieter Bäusch”).

Pages 164–189: A list of cipher symbols and their meanings—essentially, the decryption key.

Pages 190–203: Copies of nine press articles about Dominas:

1. General-Anzeiger Bonn, 12 December 1964 – Contains information about the loot, but none about the cipher (General-Anzeiger Bonn 1964).
2. Welt am Sonntag, 13 December 1964 – Mentions the secret writing, but adds no new details (Welt am Sonntag 1964).
3. Welt am Sonntag, 11 September 1966 – Again references the cipher, without new information (Damrow 1966).
4. Bonner Rundschau, 4 September 1966 – A brief article unrelated to Dominas, reporting instead on an alleged codebreaking machine (Bonner Rundschau 1966).
5. General-Anzeiger, 10 September 1966 – Mentions the cipher but provides no further insights (General-Anzeiger Bonn 1966).
6. Der Spiegel, Issue 41/1964 – Not identical to the Spiegel article that first drew my

attention; contains no details about the cipher (Der Spiegel 1964).

7. Bonner Rundschau, 13 November 1965 – Refers to twenty volumes of Dominas's diary written in cipher, noting that three volumes (a total of 750 pages) had already been deciphered (Hiller 1965).

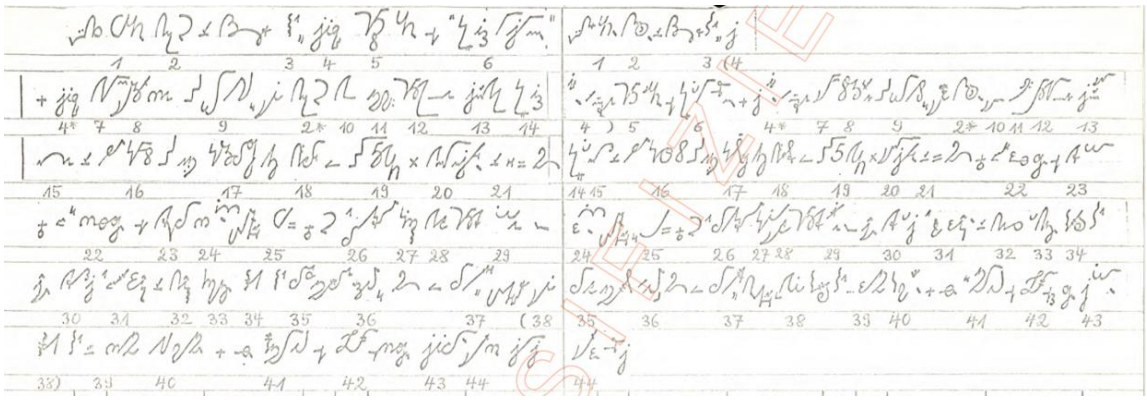


Figure 3. Two encrypted passages (left and right) with similar content. The cryptanalyst Dieter Bäusch employed comparisons like these to break the cipher.

8. Bonner Rundschau, 20 August 1965 – Mentions twenty notebooks of encrypted entries by Dominas (Bonner Rundschau 1965).

9. Quick, 1964 (issue not specified) – Contains no information on the encryption itself (Quick 1964).

Pages 204–223: Various notes by Bäusch, the purpose of which is often unclear. Some appear to be frequency analyses.

The second PDF file comprised 72 pages. After two cover sheets, it contained 70 pages of encrypted notes written by Dominas (Figure 2). These are likely only a fraction of the approximately twenty notebooks mentioned in the press articles.

5 The Breaking of the Cipher

The two handwritten treatises by Bäusch contained in the first PDF provide extensive insights into the cipher system used by Dominas. The encrypted manuscript itself presents a number of challenges to decipherment: it contains no illustrations, headings, typographical highlights, or plaintext insertions. Its structure is limited solely to paragraph divisions, offering minimal contextual guidance.

Bäusch assumed that the plaintext language of the cipher was German—an assumption that later proved correct. According to his notes, the alphabet of the cipher comprises more than one thousand distinct symbols. From this, Bäusch inferred that the system either employed homophones (indicating a homophonic cipher) or that some symbols represented entire words or syllables (suggesting the use of a nomenclator). It turned out that both was the case

Unfortunately, Bäusch does not describe in detail the exact methods he used to break the cipher. He provides no information on statistical analyses or cribs that might have aided his decryption. Instead, he merely states that he compared passages with a similar or identical plaintext in order to derive information from their differences—a method he refers to as the identification of compromising repetitions. His manuscripts include numerous examples of this comparative approach (see Figure 3).

Apparently, it was relatively easy to find such compromising repetitions, as many text segments occurred twice or even more within the encrypted material. It is likely that Dominas copied certain sections of his text, possibly as an

exercise to practice the use of newly added symbols in his cipher system. This repetitive structure may have inadvertently provided Bäusch with critical clues that enabled his partial or complete decipherment of the system.

6 The Cipher

Although Dominas's secret script bears some resemblance to shorthand, according to Dieter Bäusch it was not designed to accelerate writing. Instead, its sole purpose was to ensure confidentiality.

combination, a word, or a punctuation mark. No transposition takes place, and distinctions between upper and lower case are not observed. It frequently occurs that Dominas encodes the same word differently when it appears multiple times, by dividing it into varying syllables or letter units.

Symbols within a word are often connected or merged. However, these connections have no semantic significance—the manner of connection carries no meaning. Word boundaries, on the other hand, are consistently maintained.

Figure 4. An excerpt from Dominas's encrypted notes (left) alongside the corresponding plaintext

In general, the system constitutes a monoalphabetic substitution cipher. Each symbol corresponds to a letter, a syllable, a letter

Some symbols differ only by their size, position, or stroke thickness, yet convey distinct meanings. Conversely, certain symbols are

polyphonic, bearing multiple, usually related meanings. In other cases, different (often visually similar) symbols represent the same value—homophones are thus used. The number of representations assigned to a character does not correlate with its frequency in the German language. Instead, geometric or aesthetic properties appear to have been the decisive factors. For example, the relatively infrequent letter D can be represented by four symbols, each depicting a rotated variant (90°, 180°, or 270°) of the letter U. By contrast, the frequent E has only two equivalents.

Dominas expanded and modified his script over time. The earliest documented stage dates to around 1952. At this point, the position and size of symbols were irrelevant, and punctuation marks as well as numerals remained unencrypted.

Around 1959, Dominas revised the system once more. The use of dot markings was largely abandoned, while the position and size of symbols gained structural importance. Once again, several characters assumed new meanings.

After 1960, Dominas introduced a few additional symbols, though the system otherwise remained largely stable.

The cipher key reproduced in Appendix A (pages 164–189 in the first PDF) is unfortunately undated but likely represents the final stage identified by Bäusch.

In conclusion, Dominas's secret script lacks complete internal consistency. Many symbols changed in form or meaning over time, while others were added or discarded. It is evident that the system evolved organically rather than being based on a fixed logical framework. Since Dominas appears to have been the sole user of this cipher, there was no need for unambiguous or permanent rules. He could freely adapt the script to his personal requirements—an approach that would hardly be practical in a multi-user encryption system.

7 Content of the messages

The two PDF documents contain several passages of plaintext, but no complete decryption. Still, it becomes clear that Dominas's notes are not a diary, though this term has been used in various press reports. In fact, the records consist of notes of different types and purposes.

Some of the encrypted messages contain detailed descriptions of the robberies, while others provide information concerning Dominas's legal counsel. In addition, they include a variety of personal reflections, observations, and narrative fragments.

Figure 4 presents an excerpt from Dominas's encrypted notes alongside its corresponding plain-text passage for comparison. An English translation is provided in the following:

Explosive powder consists of saltpeter "69," sulfur "10," and coal "21." Gunpowder consists of "64," "10," and "16." Only potassium saltpeter can be used for production, and only powdered rod sulfur can be used as sulfur. Ammonium nitrate explosives are called safety explosives because they have a low explosion temperature. They consist of approximately seventy to ninety percent ammonium nitrate as an oxygen carrier and charcoal, naphthalene, or oil as additives. Mixtures of ammonium nitrate HN four with finely powdered aluminum metal are called ammonal and are characterized by an explosive effect similar to that of blasting gelatin. They are not very sensitive to heat, shock, or friction. Guncotton is produced by immersing cotton or cotton wool in nitric acid. After about an hour, the liquid is spun off and washed away, and the resulting guncotton is washed with plenty of water, boiled, and then dried. It is almost insensitive to simple impact and shock, especially when soaked in paraffin, and when ignited it deflagrates instantly without any explosion. However, it explodes very violently when an initial impulse is given. Explosion wave seven thousand meters. Nitroglycerin is formed when nitric acid is carefully dripped into glycerin. This process is very dangerous, as

nitroglycerin decomposes easily at temperatures above thirty degrees Celsius. By absorbing diatomaceous earth, it becomes significantly less sensitive to shock and pressure and loses very little of its explosive power. Absorptions of nitroglycerin in diatomaceous earth are called dynamite. Because moisture displaces the dangerous nitroglycerin from the dynamite, the cartridge is wrapped in paraffin or parchment paper. However, since it still attracts moisture, dynamite must not be stored for long periods. Explosive gelatin is produced by allowing nitroglycerin to act on guncotton, which then swells. It has a greater explosive power than dynamite and is insensitive to moisture. Trinitrophenol (called picric acid), which was originally used as a yellow dye for silk and wool, was later used as an explosive for filling grenades. Phenol is a colorless crystalline mass; its solution in water is antiseptic and is called carbolic acid. With nitric acid, it forms the explosive trinitrophenol or picric acid, which is toxic. Mercury salt or fulminate of mercury is produced by complicated reactions of alcohol on a solution of mercury in HNO, which serves as an explosive agent. Trinitroglycerin is an oily, colorless, odorless toxic liquid. Wet guncotton is more suitable for explosions and cannot be ignited by a normal flame; it explodes on initial ignition, which means that it explodes when it receives an extremely rapid physical shock, which is caused by the explosion of fulminate of mercury. So if a torpedo filled with wet or dry guncotton has a detonator made of this material and explodes, ...

8 Conclusion

The cipher created by Petras Dominas represents an exceptional example of the use of cryptography by a criminal actor. For a manually constructed encryption system, it is remarkably complex and presents significant challenges to cryptanalysis.

The materials evaluated for this study—archival records from the German Federal Intelligence Service (BND)—provide a valuable overview of the system and its context. Nevertheless, several important questions remain unanswered:

- How exactly did Bäusch succeed in breaking the cipher?
- What, in precise terms, do the encrypted notes contain?
- In addition to the pages made available by the BND, there must have been numerous others. Do these still exist somewhere?

It would also be of great interest to learn more about Dominas's later life after 1964. The available sources do not even mention the year of his death. Any reliable information or leads regarding these open questions would be highly appreciated.

Acknowledgments

The author would like to thank the Historisches Büro of the Bundesnachrichtendienst (BND).

References

Bonner Rundschau. 1965. Anwältin gab Darlehen für Raubüberfall. In *Bonner Rundschau*, 13 November 1965.

Bonner Rundschau. 1966. Ende des Codes. In *Bonner Rundschau*, 4 September 1966.

Der Spiegel. 1964. Blut und Brillanten. In *Der Spiegel* 41/1964.

Der Spiegel. 1966. Unter gepreßten Blumen. In *Der Spiegel* 9/1966.

Hildegard Damrow. 1966. Drei Monate Monsterprozeß gegen Bandenchef Dominas. In *Welt am Sonntag*, 10/11 September 1966.

General-Anzeiger Bonn. 1964. Dominas-Raub gefunden. In *General-Anzeiger Bonn*, 12 December 1964.

General-Anzeiger Bonn. 1966. Geheimschrift eines Verbrechers von Spionage-Experten entziffert. In *General-Anzeiger Bonn*, 10/11 September 1966.

Hans Hiller. 1965. Der große Schweiger Dominas. In *Bonner Rundschau*, 20 August 1965.

Quick. 1964. Die Anwältin und ihre Töchter. In *Quick* 40/1964.

Welt am Sonntag. 1964. Das Tagebuch soll Dominas überführen. In *Welt am Sonntag*, 13 December 1964.

Appendix A. The key

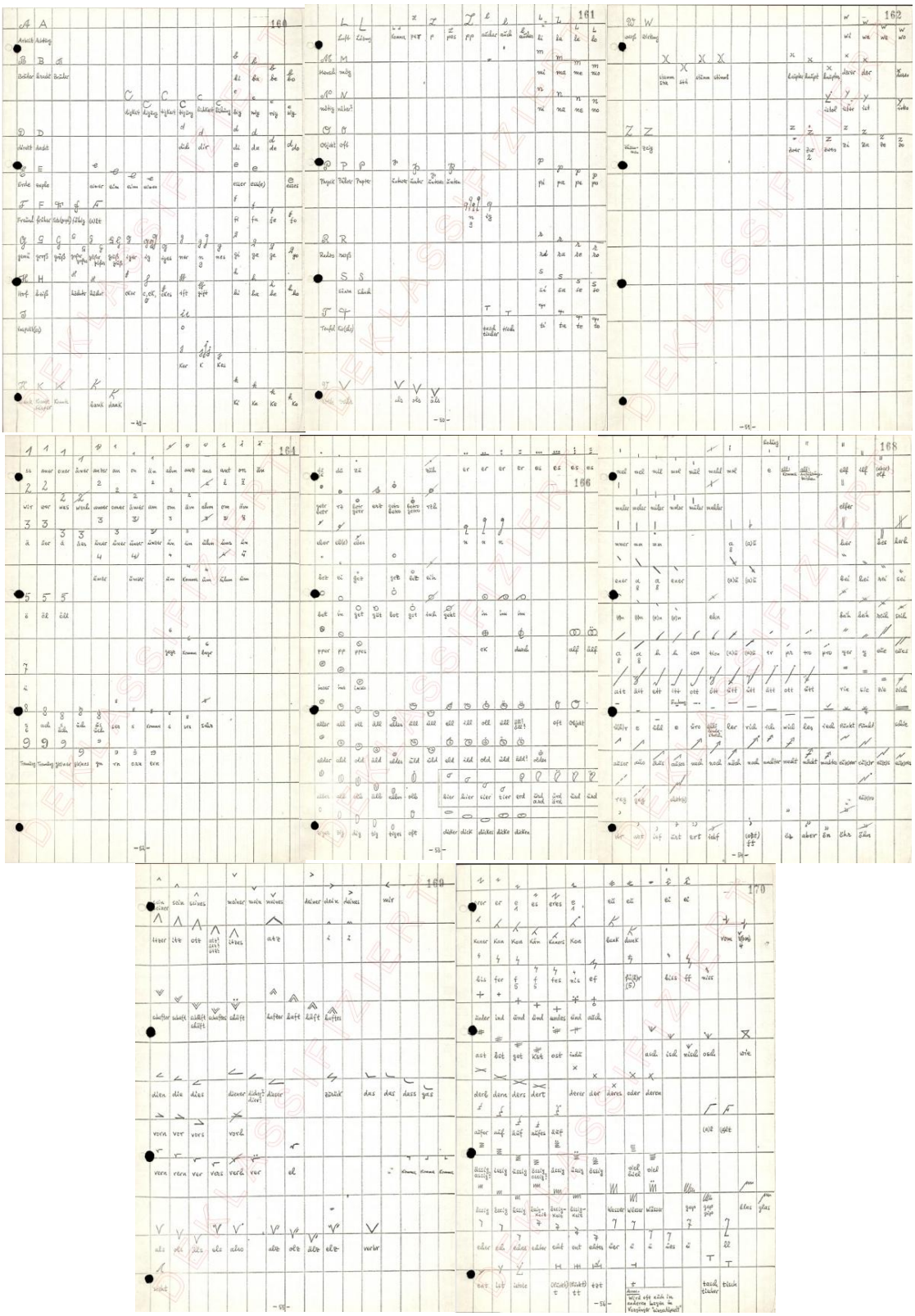


Figure 5. The key of Dominas's cipher (part 1), as derived by the cryptanalyst Dieter Bäusch

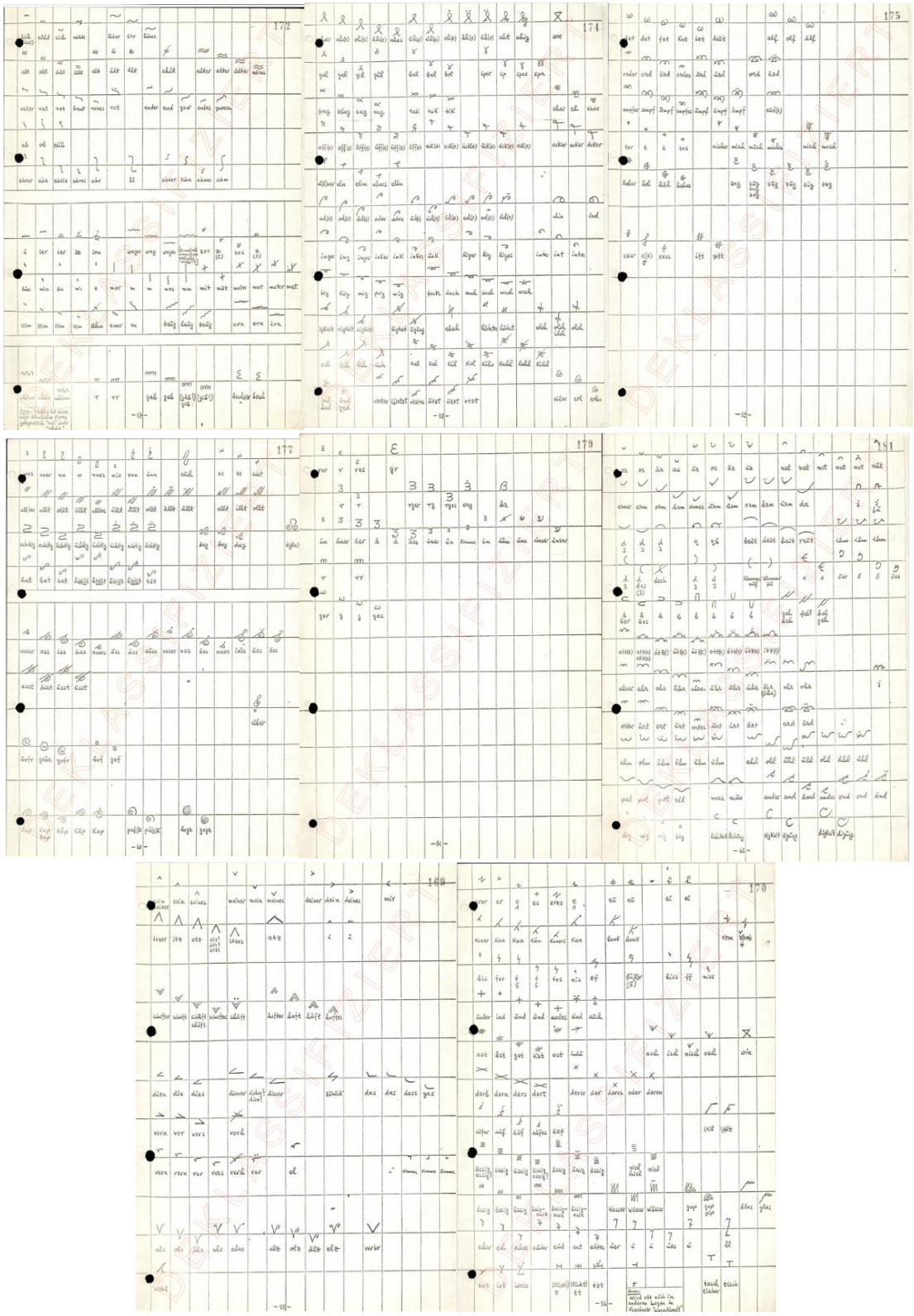


Figure 6. Part 2 of the key

HCPortal: Ten Years of Development

Eugen Antal
Slovak University of
Technology in Bratislava
Slovakia
eugen.antal@stuba.sk

Pavol Zajac
Slovak University of
Technology in Bratislava
Slovakia
pavol.zajac@stuba.sk

Abstract

HCPortal is an online portal focusing on historical cryptology. It has now been in active development for ten years. During the development, many new changes and additional modules were added. We present the current state of the portal, with a focus on an overview of the main modules and their user interface specifics.

1 Introduction

The Portal of Historical Ciphers (HCPortal) is an online portal consisting of several web pages and tools, each related to historical cryptology (Antal and Zajac, 2020). The first version of the portal was developed in 2016, focusing on two aspects at that time: creating a collection of historical ciphers, alongside tools for analysing these ciphers (Antal and Zajac, 2018). The content of the portal was gradually expanded, featuring a modern Angular application for browsing the database of cryptograms since 2017. The main goal was to publish accessible data for enthusiasts and researchers to support their work and to promote the research in the area of historical cryptography.

In 2019, a new education aspect was introduced, focusing on various cryptanalytic techniques (Antal and Zajac, 2021). The database of cryptograms was expanded with a new separated database in 2020, and gradually upgraded in the following years. In 2021, a virtual museum and a special online tool (which can be used to create custom nomenclator ciphers) were integrated into the portal.

The user interface of the HCPortal was redesigned in 2022 (see Figure 1). When designing new components, the contributors focused exclusively on online accessible applications and modern web technologies such as Angular, React, Laravel, etc. In the same year, a new version of the

Figure 1: HCPortal - main portal page.

portal was introduced with a special focus on the following three aspects:

- education,
- promotion of historical cryptology,
- source of historical data (databases and annotated datasets).

The education module was redesigned to better support education goals and interactive teaching methods. Old, separate databases were integrated, and access to data was updated to facilitate easier access.

In 2024, an additional database of cryptologists was added as well. In 2025, a new web application was added for storing and sharing datasets used for various machine learning tasks, related to historical encrypted manuscripts and cipher keys.

In the rest of the article, we describe the current state of the portal with respect to selected areas of the portal's use. We point out how the specific parts of the portal support researchers, educators, and enthusiasts in the area of historical cryptology.

2 Related Work

Presently, numerous websites and online tools are accessible on the internet that specialize in historical cryptology. Notably, over the past decade, a substantial number of digital resources have emerged, providing support for research in this field and facilitating data access.

Enthusiasts can find various information about codes and unsolved cryptograms on various blogs and websites (Cryptiana, 2026; Crypto Cellar Research, 2026; Cipherbrain, 2022).

Websites such as dCode (2026), Cryptii (2026), CrypTool (2026) offer implementations of different classical ciphers. These and similar sources provide students and enthusiasts with an opportunity to interact with the cipher algorithm and learn basic encryption and decryption steps.

Online museums (Crypto Museum, 2026) and simulators (Virtual Colossus, 2026) provide detailed information about historical ciphers and cipher machines, shedding light on their operation and significance.

Databases of historical cryptograms, cipher keys, datasets, and other materials are equally important. These databases are primarily used in research projects focused on analyzing and decrypting historical ciphers (Megyesi et al., 2020; Antal and Zajac, 2020). They also enable a comprehensive study of historical ciphers, contributing to a deeper understanding of their significance.

We aim to position HCPortal as a bridge between various cryptology resources. We integrate all the above-mentioned aspects under a single umbrella with a unified design, while providing links to additional specialized resources.

3 Using HCPortal for Education

One of the main aspects of the portal is to support education goals and interactive teaching methods[1] on the topic of historical cryptology. For this reason a special *Education*[2] module was created. At the current state, it is divided into two main parts: cryptography and cryptanalysis.

The cryptanalysis part consists of a collection of interactive tools with graphical visualization of the data, designed for a better understanding of attacks on selected classical ciphers. Each demonstrated attack is divided into logical steps. At the moment, it contains five attacks on substitution ciphers:

- Brute-force attack on Caesar Cipher;
- Hill-Climbing attack on simple substitution cipher;
- Friedman test and brute-force attack on Vigenère cipher;
- Friedman test and brute-force attack on an *Autokey cipher*, including the Autokey to Vigenère transformation (see Figure 2);
- Manual and (semi) automated dictionary attack on substitution cipher based on word patterns.

For the transposition ciphers, three attacks are implemented:

- Brute-force attack on Scytale;
- The "moving strips" manual attack on columnar transposition;
- Multiple anagramming method.

[1] The primary goal for the inclusion of the *Education* module to HCPortal was to support an online education in our course Classical Ciphers taught at the Slovak University of Technology in Bratislava. However, we have designed the tools in such a way, that they can be used by other students and the general public (Antal and Zajac, 2021).

[2] https://edu.hcportal.eu

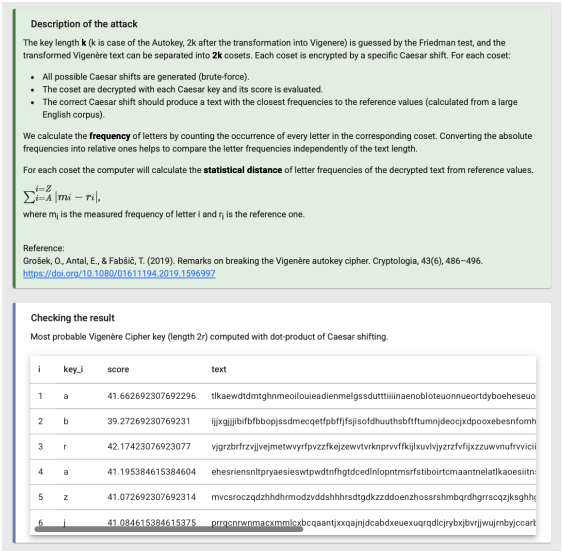


Figure 2: Education module - part of the attack on an Autokey cipher.

The cryptanalysis process is supported with a separate web application called *ManuLab online*[3]. ManuLab online is a software product for statistical analysis of encrypted historical manuscripts. It's the successor of the previous ManuLab Qt application (Antal and Zajac, 2018). The software is designed to display pairs of input files - an image file of ciphertext with the corresponding transcription (see Figure 3). Several functions (called filters) are implemented, which can be chained in the analysis process. The main filters are: frequency analysis of n-grams, Index of Coincidence, Shannon's Entropy, pattern search, text element distances, text modifications, anagram detection, vowel detection, and language guessing methods.

Figure 3: ManuLab Online - example.

In the cryptography part of the education module, general information and a classification of historical ciphers can be found. We present three types of classification based on various criteria for a better understanding of how historical ciphers work. In addition, the visitor can try the encryption and decryption process using various real cipher[4] implementations (see Figure 5), and try some helper methods[5] which are used in our course.

In addition to the real cipher implementation of the education module, a separate web application was created, where custom nomenclator keys can be created by the users (see Figure 4). This module is called *Cipher Creator*[6]. The cipher keys created with this application have both textual (JSON format) and graphical (PDF format) representations, and the website allows the user to encrypt and decrypt custom text messages with the generated nomenclator keys.

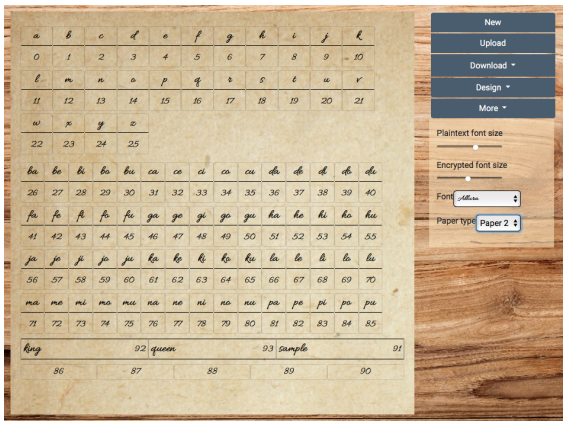


Figure 4: Cipher Creator module - custom cipher key.

A special glossary is accessible[7] from the main portal page, covering definitions of terms related to historical cryptology based on William F. Friedman's Basic Cryptologic Glossary and a discussion from Klaus Schmeh's former Cipherbrain blog (Cipherbrain, 2022).

[3]https://manulab.hcportal.eu

[4]Currently, the real ciphers part cover two Japanese ciphers, the Condenser PBJ cipher machine, and four ciphers from that were used in movies or TV series.

[5]The Bellaso's method of creating numeric permutations from passphrases, and various string-to-pattern conversion methods.

[6]https://ciphercreator.hcportal.eu

[7]https://hcportal.eu/glossary.html

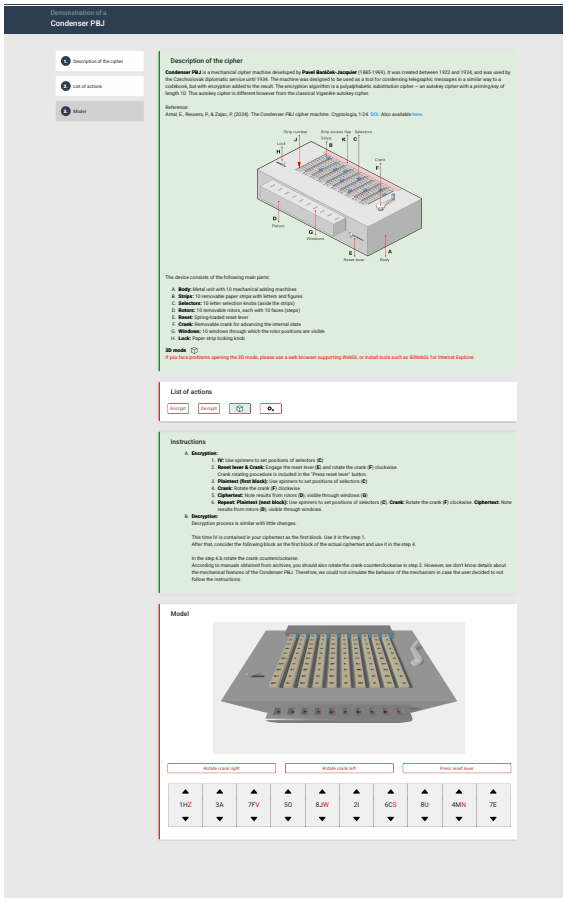


Figure 5: Education module - implementation of the Condenser PBJ cipher machine with a 3D model.

4 Promotion of Historical Cryptology

The second main aspect of the portal is to increase interest in historical ciphers among the general public. By using modern information technologies, a special *Virtual Museum*[8] was created based on the virtual reality (VR) concept.

The museum presents general information about the history of cryptology in a familiar "museum" style. A virtual reality engine for a web browser was used. In this way, materials can be displayed online, even if the user does not have a VR device (Antal and Zajac, 2021). The museum core consists of a static exhibition, which covers a timeline of ciphers and information about ciphers, steganography, cryptanalysis, unsolved cryptograms, and cipher machines (see Figures 6 and 7). In addition, a registered user can create dynamic exhibitions on various topics. These exhibitions allow us to present the following data types: text, image, video, and PDF files.

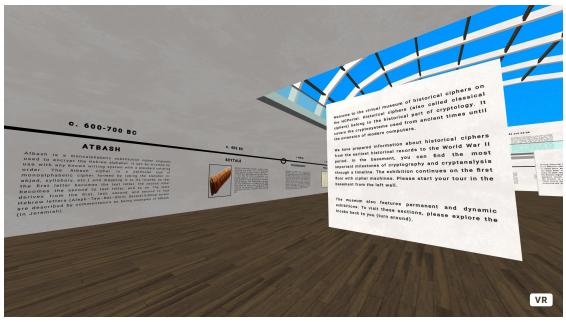


Figure 6: HCPortal - Virtual Museum.

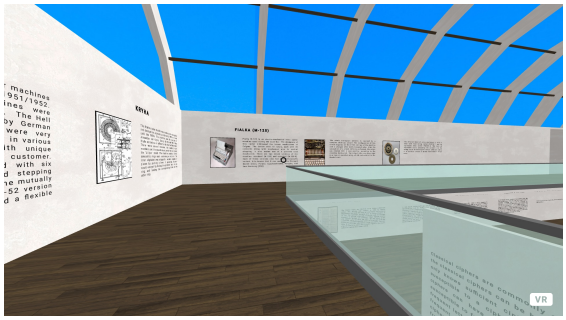


Figure 7: HCPortal - Virtual Museum - cipher machines section.

5 Historical Data Sources

The third important aspect of the portal is to provide a wide range of historical data for researchers and the general public. We collected, processed, and published different databases to support a wide range of research tasks. Contrary to other known databases of materials related to historical cryptology, all our data is publicly available for everyone without a need for registration or other types of special access [9].

For researchers who investigate the history of cryptology and analyse the development of cipher systems, two special databases are available:

- Database of cryptograms and cipher keys[10];
- Database of cryptologists[11].

[8] https://www.museum.hcportal.eu

[9] All the materials from archives and the owners are used with permission or have a public domain license.

[10] https://crypto.hcportal.eu/

[11] https://cryptologists.hcportal.eu

Currently, the database of cryptograms and cipher keys contains 1875 records of cryptograms and 319 records of cipher keys. The database contains, among others, a notable large collection of 988 cryptographic postcards, including Tobias Schrödel's collection (Tobias Schrödel, 2021). The records are supplemented with various metadata and tags (see Figure 8) to allow detailed browsing of the database. We have implemented a modern web interface to browse the database and provide statistics of the records (see Figure 9). Additionally, a timeline of the records is also available in the browsing interface.

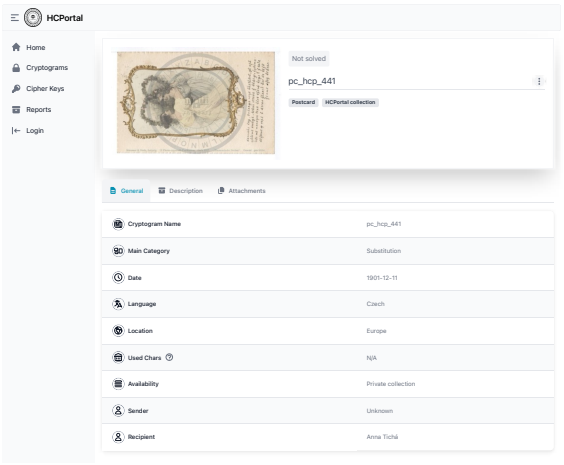


Figure 8: Database of cryptograms and cipher keys - example of a cryptogram record and its metadata.

The database of cryptologists contains 222 records at the moment from various public sources. The records are supplemented with various metadata and tags (see Figures 10 and 11) to allow detailed browsing of the database.

Both system provides content management system for registered users. The records are reviewed by an administrator, and only approved records will be visible in the public database.

For researchers who are working on various computer vision and machine learning tasks related to historical ciphers, a special repository of annotated datasets is available - the *Dataset Builder*[12]. Currently, the database contains 143377 instances of annotations divided into 172 classes (see Figures 12 and 13) (Antal et al., 2026).

This platform provides a wide range of functionality: dataset visualization, view of statisti-

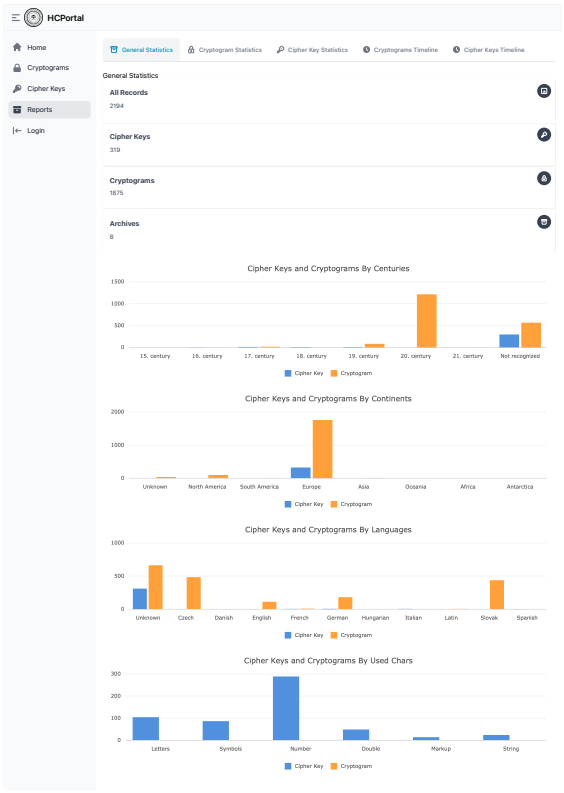


Figure 9: Database of cryptograms and cipher keys - statistics.

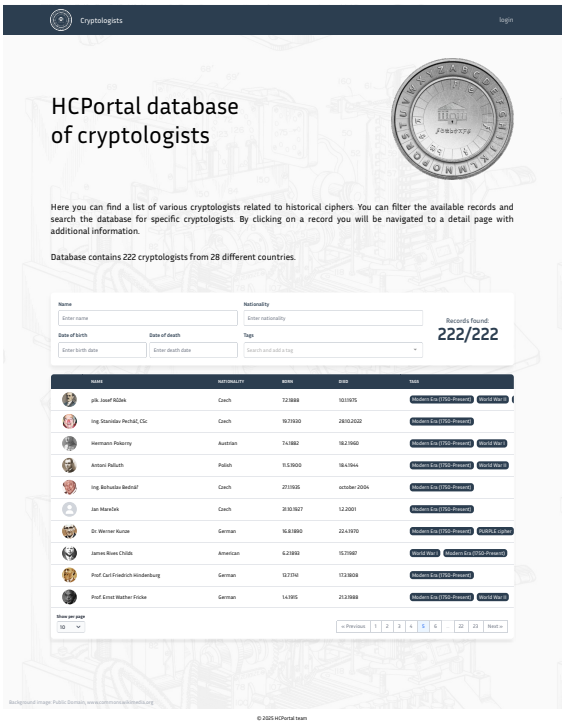


Figure 10: Database of cryptologists.

[12]https://builder.hcportal.eu

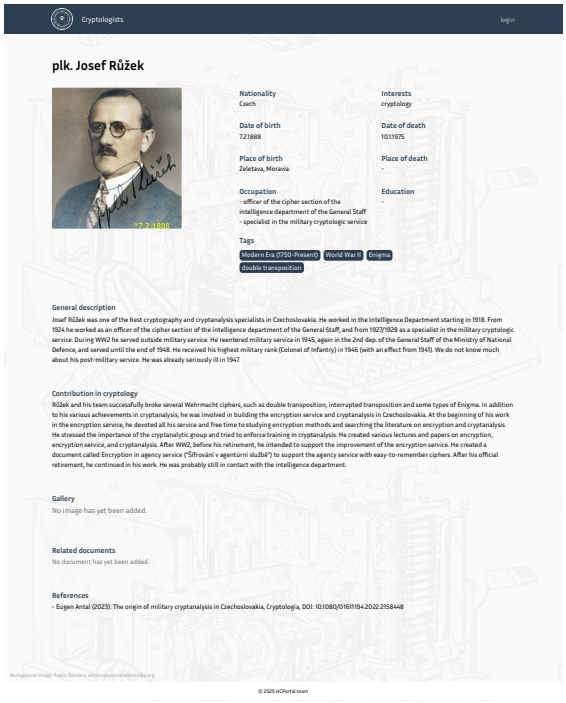


Figure 11: Database of cryptologists - example of a record and its metadata.

cal summaries, and customizable dataset generation (see Figure 14). The users can interactively select the annotation type (polygons and bounding boxes), symbol categories (currently, glyphs or digits are available), define dataset splits, specify target classes, choose individual samples with annotations, and balance the number of annotations across classes. Therefore, they can build a custom dataset that perfectly suits their needs. The datasets can be exported in several widely used annotation formats, including YOLO, COCO, Pascal VOC, and Labelme.

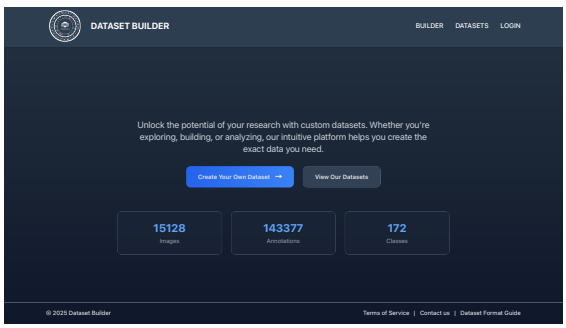


Figure 12: Dataset Builder.

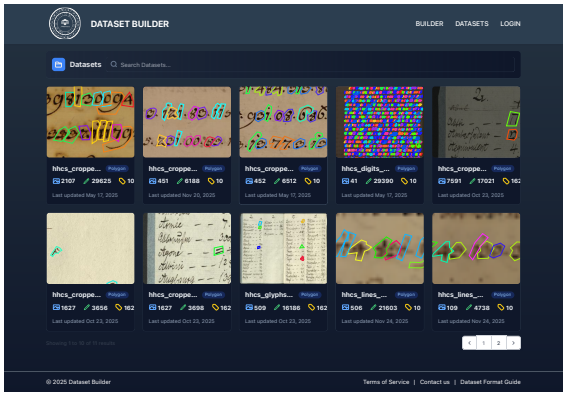


Figure 13: Dataset Builder.

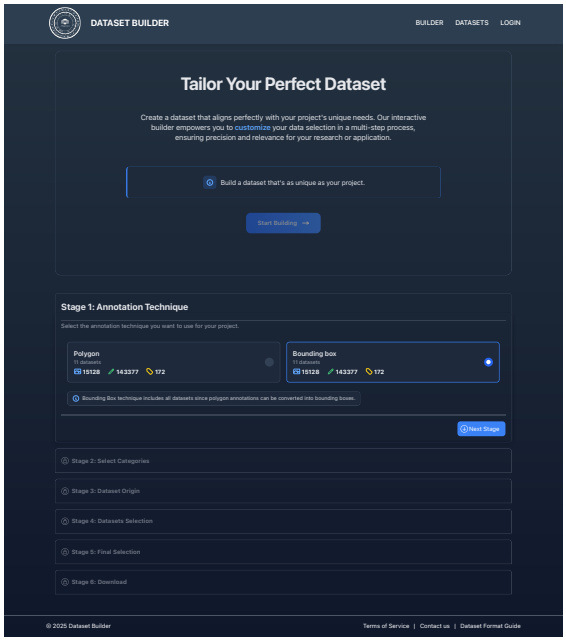


Figure 14: Dataset Builder.

6 Conclusions

There exists a wide range of resources for historical cryptography, including direct and indirect sources of data (such as online archives), online museums and collections, as well as education resources (CrypTool, 2026; Crypto Museum, 2026; dCode, 2026; Megyesi et al., 2020; Virtual Colossus, 2026). The current version of HCPortal represents a middle ground that combines different aspects into a single portal. Our collection of data and tools can support both research and education in the area of historical cryptography.

Acknowledgments

This work was supported by grant VEGA 2/0054/24.

References

Eugen Antal and Pavol Zajac. 2018. ManuLab System Demonstration. In *Proceedings of the 1st International Conference on Historical Cryptology, HistoCrypt 2018*, pages 125–128. Linköping University Electronic Press.

Eugen Antal and Pavol Zajac. 2020. HCPortal Overview. In *Proceedings of the 3rd International Conference on Historical Cryptology, HistoCrypt 2020*, pages 18–20. Linköping University Electronic Press.

Eugen Antal and Pavol Zajac. 2021. HCPortal Modules for Teaching and Promoting Cryptology. In *Proceedings of the 4th International Conference on Historical Cryptology, HistoCrypt 2021*, pages 1–11. Linköping University Electronic Press.

Eugen Antal, Pavol Marák and Filip Mikuš. 2026. HHCS: A Dataset of Cipher Symbol Annotations From Handwritten Historical Encrypted Documents for Machine Learning Tasks. In *IEEE Access*, vol. 14, pages 9226–9240, DOI: 10.1109/ACCESS.2026.3654267.

CrypTool Contributors. 2026. *CrypTool Portal*. `https://www.cryptool.org/en/`

dCode Contributors. 2026. *dCode - The ultimate collection of tools for games, math, and puzzles*. `https://www.dcode.fr/en`

Fränz Friederes. 2026. *Cryptii*. `https://cryptii.com`

Martin Gillow. 2026. *Virtual Colossus*. `https://virtualcolossus.co.uk`

Beáta Megyesi, Bernhard Esslinger, Alicia Fornés, Nils Kopal, Benedek Láng, George Lasry, Karl de Leeuw, Eva Pettersson, Arno Wacker and Michelle Waldispühl. 2020. Decryption of historical manuscripts: the DECRYPT project. *Cryptologia*, 44(6), pages 545–559. Taylor & Francis.

Paul Reuvers and Marc Simons. 2026. *Crypto Museum*. `https://www.cryptomuseum.com/`

Klaus Schmeh. 2022. *Cipherbrain*. `http://scienceblogs.de/klausis-krypto-kolumne`

Tobias Schrödel. 2021. Cryptographic postcards. In *Proceedings of the 4th International Conference on Historical Cryptology, HistoCrypt 2021*, pages 131–136. Linköping University Electronic Press.

Satoshi Tomokiyo. 2026. *Cryptiana* `http://cryptiana.web.fc2.com/code/crypto.htm`

Frode Weierud. 2026. *Crypto Cellar Research* `http://cryptocellar.org/`

CTTS – CrypTool Transcriber and Solver

George Lasry
The CrypTool Project
george.lasry@gmail.com

Abstract

This document describes the CrypTool Transcriber and Solver (CTTS). It enables the transcription, decipherment, and analysis of historical ciphered documents. CTTS provides an end-to-end integrated solution for all those stages, with a convenient graphical user interface. CTTS has been successfully employed to transcribe and decipher dozens of collections of historical ciphered documents.

1 Introduction

Working with historical ciphered documents typically consists of several stages, as illustrated in Figure 1. The stages include the transcription of the cipher symbols in documents, reviewing the transcription and correcting errors, performing initial cryptanalysis with cryptanalysis software (e.g., Kopal, 2019), to recover the symbols assigned to the letters of the alphabet, completing cryptanalysis (recovering the meaning of nomenclature symbols such as nulls, names, words), deciphering the documents with the recovered key, editing (e.g., inserting spaces between words) and analyzing the decrypted text. Before CTTS, those steps could only be performed manually and/or using separate tools, transcription being error-prone and time-consuming, and identifying errors and correcting them cumbersome and inefficient.

This work is highly iterative, as any stage may affect the results of a previous stage, as illustrated in Figure 1. For example, cryptanalysis often helps spot transcription errors. The process may also require collaboration between researchers with diverse skills, such as a codebreaker working on the initial stages, and a historian or linguist working on the later stages.

CTTS is an end-to-end solution that supports iterative work across all stages of a decipherment project. It offers a convenient graphical user interface (GUI). It was developed in 2021 to support the decipherment of dozens of enciphered letters from Mary Queen of Scots (Lasry et al., 2023). Its functionality has been refined over time, and the tool has been improved incrementally to address practical problems encountered when working with those collections, based on user feedback.

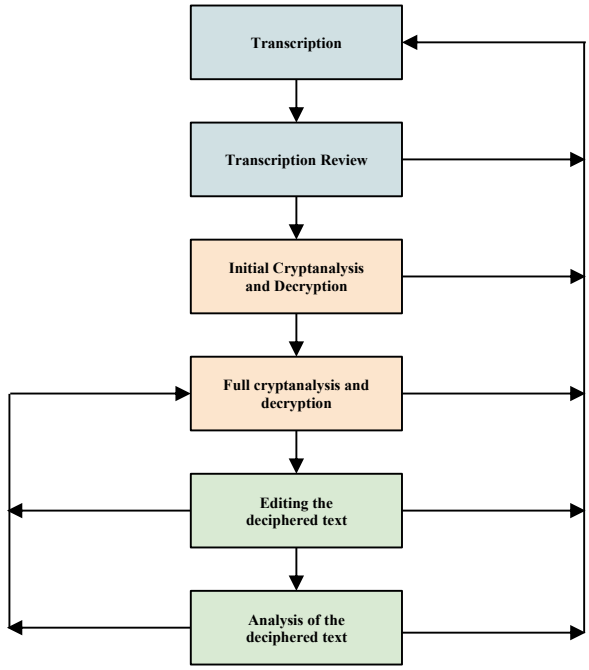


Figure 1. A comprehensive flow for processing historical enciphered documents

CTTS is an open-source desktop GUI application, developed in Java. This application is therefore portable and can be installed on Windows, Linux, or Mac computers.

2 Transcribing with CTTS

CTTS supports the transcription of historical ciphered documents, including symbol segmentation, symbol classification, and transcription review.

Symbol segmentation is performed by surrounding each symbol in a document image with a box, using mouse gestures. CTTS provides

a zoomed-in view of the symbol, enabling refinements to the symbol segmentation, as shown in the bottom part of Figure 2.

Symbol classification consists of assigning segmented symbols to their relevant symbol types. The list of symbol types is shown on the left side (see Figure 2), and new symbol types can be added and labeled. An icon can also be created or imported to represent the type. It is possible to classify the symbols one-by-one using drag-and-drop gestures, or to perform bulk assignments, selecting all unassigned symbols of the same type, as shown in Figure 3.

After the symbols in a document have been transcribed (segmented and classified), a review of the transcription is done either by inspecting each of the types, looking for outliers, or by reviewing the transcription line-by-line. CTTS provides visual cues, showing the segmented symbol near the icon assigned to the symbol type, or all symbols of the same type in the bottom pane, to facilitate spotting and correcting transcription errors.

3 Cryptanalysis with CTTS

CTTS enables automated cryptanalysis of homophonic or monoalphabetic ciphers to recover the meanings of the symbols representing letters (the homophones). If the cipher is not homophonic or monoalphabetic, the output of the transcription may be used as an input to an external cryptanalysis software.

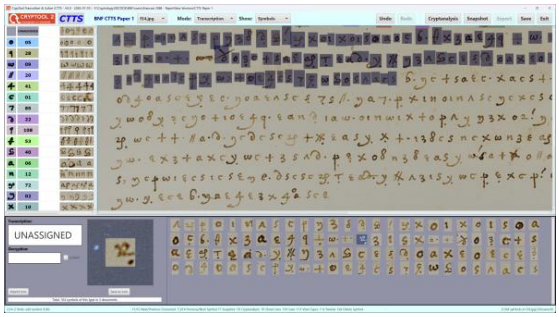


Figure 2. Transcription with CTTS, segmenting the symbols

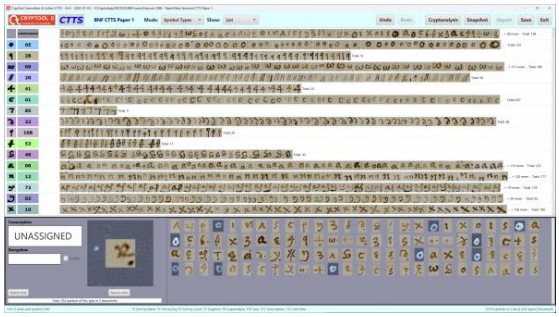


Figure 3. Transcription with CTTS – classifying the symbols

The cryptanalysis algorithm is similar to the method described in Kopal (2019). It relies on simulated annealing and n-gram-based scoring, and can be customized using the following parameters:

- The language: CTTS currently supports French, Italian, German, English, Spanish, Latin, and Dutch.
- Combining pairs of letters into the same entities, such as U/V, which were interchangeable in the early forms of several European languages.
- Ignoring spaces, doubled letters (e.g., the second S in SS), or specific letters, like "h", which was often ignored in Renaissance Italian ciphers.
- N-gram for scoring: 3-grams, 4-grams, 5-grams, or 6-grams.
- The maximum number of homophones per plaintext letter.

After the parameters have been set, automated cryptanalysis starts. CTTS displays the tentative decryption and key as cryptanalysis progresses, updating the results.

CTTS constantly checks the plausibility of decrypted segments by looking for sequences that appear in reference texts in the selected language, highlighting the relevant segments in capital letters, as well as all the symbols that are part of this sequence but may appear elsewhere. In Figure 4, CTTS has highlighted several plausible Middle French fragments, such as "PREVEU" ("expected") and "CE TRAICTE" ("this treaty"), in the first line.

Cryptanalysis tries to improve the decryption score. When the results of cryptanalysis are satisfactory, cryptanalysis can be stopped, and the recovered key can be saved. If this is not the case,

cryptanalysis can be stopped, the parameters may be modified, and cryptanalysis restarted.

In rare cases, automated cryptanalysis provides an almost complete and plausible decryption, but in the common case of more complex ciphers, such as those with nomenclatures, the decryption obtained with automated cryptanalysis is fragmentary and requires additional manual cryptanalysis work to refine the recovered key and to recover the meaning of the symbols in the nomenclature that do not represent the letters of the alphabet. This manual process is facilitated by displaying all instances of the symbol type to be identified. By clicking any icon at the bottom of the page, the relevant occurrences of that symbol can be viewed, so they can be interpreted and assigned a decrypted value. Furthermore, any change in the interpretation (decrypted value) of a symbol is automatically propagated to all relevant places, with no need for manual bookkeeping.

4 Editing the deciphered text

Usually, there are no word separations or punctuation in historical ciphers and ciphered documents. Therefore, the initial decrypted text is most often contiguous, making it difficult to read and interpret. Furthermore, any error made by the cipher clerk in the past when enciphering the original plaintext, or any modern transcription or decryption error, may further affect the quality of the decrypted text obtained with CTTS, making it even less readable.

CTTS provides a convenient built-in editing function, as shown in Figure 4, that allows spaces, punctuation, and capitalization to be added.

Figure 4. Editing the decrypted text

By reviewing readable, decrypted text, it is often possible to interpret nomenclature elements that represent proper names or less common words, and to spot transcription or cryptanalysis errors that can be easily fixed, since any correction is immediately propagated without any need for costly bookkeeping.

The edited decrypted text is especially useful to historians or linguists interested in the contents of the enciphered document.

5 Exporting the results of CTTS

The outputs of all stages of the work with CTTS (the transcribed symbols – their positions on the images and their classification, the recovered key, the raw decrypted text, and the edited decrypted text) can also be exported and used as input to other cryptanalytic solvers or to train AI models that aim to automate symbol segmentation and classification.

6 Using CTTS in decipherment projects

CTTS has been successfully used to transcribe and decipher hundreds of historical documents. This includes dozens of collections of undeciphered documents from the Bibliothèque Nationale de France, as well as many other unpublished decipherments by the author, listed in (Tomokiyo, 2022). The owner of the website Cryptiana regularly uses CTTS (Tomokiyo, 2026). CTTS was also critical to the decipherment of a collection of 57 letters from Mary Queen of Scots, written during her captivity between 1578 and 1584, which consisted of almost 200,000 symbols and would have been impossible to transcribe and decipher without this tool (Lasry et al., 2023).

CTTS can also be useful when working with ancient scripts, e.g., in a project aimed at deciphering cryptic Dead Sea Scrolls fragments (Oliveiro, 2025). CTTS has also been used without the built-in cryptanalysis to transcribe non-homophonic ciphers, such as syllabic ciphers, applying an external cryptanalysis tool, then importing the recovered key back into CTTS to complete the decryption (Lasry, 2024).

7 Future capabilities

Other capabilities being planned or considered for future development of CTTS include:

- AI capabilities to automatically transcribe historical documents are being added to CTTS, allowing the automated transcription to be refined and improved with CTTS.

- A web-based version that would enable online collaborative work on artifacts stored in central collections or on Cloud storage.
- Supporting a centralized repository of common cipher symbols, many of which are often shared in historical keys and cipher documents.

8 Conclusion

CTTS is a highly effective tool for deciphering historical ciphers. It is an end-to-end solution that supports all stages of decipherment projects in one place. It has been successfully employed for a multitude of decipherment projects and has been critical in projects involving large-scale collections. The author invites codebreakers and scholars researching historical ciphers to use CTTS and provide feedback to help improve it.

References

CrypTool Project. n.d. CTTS (CrypTool Transcriber and Solver). https://github.com/CrypToolProject/CTTS. Accessed 9 May 2026.

Nils Kopal. 2019. Cryptanalysis of Homophonic Substitution Ciphers Using Simulated Annealing with Fixed Temperature. In *Proceedings of the 2nd International Conference on Historical Cryptology (HistoCrypt 2019)*, Linköping University Electronic Press, 107-116.

George Lasry. 2024. Deciphering Historical Syllabic Ciphers. In *Proceedings of the 5th International Conference on Historical Cryptology (HistoCrypt 2024)*, Linköping University Electronic Press, 147-159.

George Lasry, Norbert Biermann, and Satoshi Tomokiyo. 2023. Deciphering Mary Stuart's Lost Letters from 1578–1584. *Cryptologia*, 47(2):101–202.

Emmanuel Oliveiro. 2025. Cracking Another Code of the Dead Sea Scrolls: Deciphering Cryptic B (4Q362 and 4Q363) through Analysis and Intuition. *Dead Sea Discoveries*, 1:1–27.

Satoshi Tomokiyo. 2022. Decipherment of Hitherto Unsolved Historical Ciphers. https://cryptiana.web.fc2.com/code/GL.htm. Accessed 9 May 2026.

Satoshi Tomokiyo. 2026. Cryptiana: Articles on Historical Cryptography. https://cryptiana.web.fc2.com/code/crypto.htm. Accessed 9 May 2026.

CrypLLM: A Built-in Chat Assistant for CrypTool 2

Nils Kopal
Hochschule Niederrhein – University of Applied Sciences
Krefeld, Germany
nils.kopal@hs-niederrhein.de

Marc Philipp Kray
University of
Siegen, Germany
marc.kray@student.uni-siegen.de

Bernhard Esslinger
University of Siegen, Germany
bernhard.esslinger@uni-siegen.de

Abstract

CrypLLM is a built-in chat assistant (agent) that supports users in creating and analyzing cryptographic workflows ("graphical programs") within CrypTool 2 (CT2). This paper presents the motivation for CrypLLM and describes its integration into CT2. We further discuss several e-learning scenarios in which the agent helps users understand, construct, and troubleshoot cryptographic workflows. Finally, we outline the system architecture and summarize current limitations as well as future directions.

1 Introduction

CrypTool 2 (CT2) is a widely used educational and research platform for experimenting with both classical and modern cryptography (Kopal and Esslinger, 2018). It is part of the CrypTool project, one of the most widely adopted initiatives worldwide for supporting the teaching and learning of cryptography. While CT2 enables powerful and flexible experimentation, it can also be challenging to use: even experienced users may struggle with complex, multi-step cryptographic or cryptanalytic workflows, selecting appropriate parameters, and debugging data flows within the application, whereas beginners may feel overwhelmed by its overall complexity and large scope.

In recent years, many applications and web services have been augmented with so-called AI agents. Such agents typically combine a large language model (LLM) with mechanisms for accessing application state and interacting with it. This is commonly realized through structured context interfaces (e.g., tool APIs) that allow the model to interpret the current state and trigger actions that affect it (OpenAI, 2023; Yao et al., 2023; Schick et al., 2023; Anthropic, 2024).

In the context of CT2, an AI agent can support both experienced users and beginners by explaining the role of individual components and how data propagates through a workflow. The agent can suggest reasonable parameter settings, highlight common pitfalls, and assist in diagnosing issues when results deviate from expectations. By *providing context-aware support directly inside the application*, the agent can reduce reliance on external resources (e.g., tutorials, textbooks, or videos) and smooth the learning curve.

This paper introduces **CrypLLM**, a new addition to CT2. CrypLLM answers questions about cryptographic concepts and helps users interpret the graphical workflow currently open in CT2, thereby lowering the barrier to effective use and learning.

The rest of this paper is structured as follows: Section 2 looks at related work. Section 3 introduces CT2 and CrypLLM, including the workspace model, design goals, and system integration. Section 4 presents representative usage scenarios and the paper is concluded in Section 5.

2 Related Work

Recent advances in large language models have led to the emergence of AI-assisted tools in software engineering and education. Systems such as code assistants, for example GitHub Copilot, provide contextual support within development environments, while intelligent tutoring systems aim to guide learners through domain-specific tasks. (Chen and others, 2021; VanLehn, 2011; Holmes et al., 2019)

In the context of LLM-based agents, prior work has explored tool-augmented models that can interact with external systems via structured APIs (e.g., function calling or tool use). These approaches enable models not only to generate text but also to inspect and manipulate application state. (Yao et al., 2023)

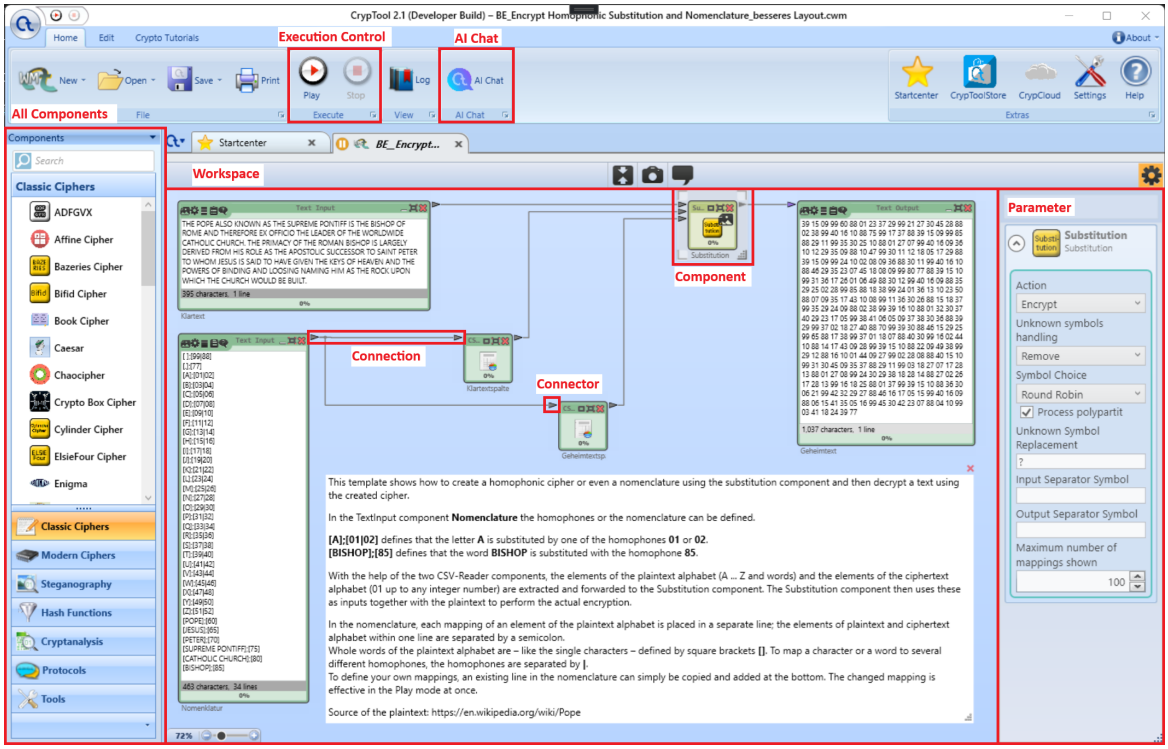


Figure 1: A typical CrypTool 2 workspace. Relevant user interface (UI) elements are marked in red.

Compared to existing approaches, CrypLLM focuses on visual, dataflow-oriented environments for cryptography, where understanding intermediate states and data propagation is crucial. To the best of our knowledge, this integration of a workspace-aware agent into a comprehensive cryptographic education tool has not been explored or implemented before.

3 CT2 and CrypLLM

Application structure: The main parts of CT2 are the Startcenter for navigation, a beginner-oriented Wizard, an Online Help, a set of more than 250 reusable *templates* (ready-to-use graphical programs shipped with CT2), and the *WorkspaceManager* as the core interaction environment (Kopal et al., 2014). Typical cryptographic workflows in CT2 can represent encryption cascades, cryptanalytic pipelines, or complete analysis processes, and are executed via a global run action (Kopal and Esslinger, 2018).

WorkspaceManager and workspace model: The *WorkspaceManager* is the central environment in CT2 where users create and execute workflows (data-flow graphs) (see Figure 1).

The *workspace model* is the internal representation of the workspace as a structured data model. *Components* (functional modules such as ciphers, analysis tools, converters or input/output fields) are connected via *input* and *output* connectors, forming pipelines through which data propagates. Users place components onto the workspace via drag-and-drop and create workflows by drawing *connections* between connectors; these connections are visualized as linking lines that represent the data flow.

Each component provides configurable *settings* that control its behavior (e.g., algorithm variants, parameters, alphabets, or encoding options). Many components (such as the Vigenère Analyzer, the Enigma cipher machine, or the modern hash function Keccak) additionally provide dedicated visualizations through specialized user interfaces (*presentations*), for example to illustrate internal algorithm steps of ciphers or to display the current progress of analysis components (Kopal and Esslinger, 2022).

Data can be inspected at any time, both at connectors and along connections. Workspaces may also contain memo fields and images.

Finally, an execution engine built into CT2 interprets the workspace model and executes the workflow in the defined order. While a workflow is running, many parameters, such as plaintext/ciphertext inputs or key values, can be mod-

ified by the user. This triggers immediate re-execution of the affected components and updates all dependent outputs accordingly. This allows users to observe the effects of their changes in real time, making experimentation more interactive and allowing them to quickly validate hypotheses and explore alternative configurations.

Design goals for CrypLLM: CrypLLM was designed as an agent that supports users in understanding, constructing, and troubleshooting CT2 workflows directly inside the *WorkspaceManager*. Beyond generating responses, it is capable of performing actions within the workspace:

- **Context-aware assistance:** Ground explanations and recommendations in the currently active workspace.
- **User-friendly terminology:** Use the same vocabulary as the CT2 user interface by referring to visible display names and avoiding internal developer identifiers.
- **Actionable guidance:** Provide concise, step-by-step instructions that users can immediately apply (e.g., which component to add, which settings to adjust, or where to connect signals).
- **Low-friction integration:** Reduce context switching by enabling users to ask questions and receive help while working on their workflow, rather than relying on external documentation, tutorials, or videos.
- **AI observation and manipulation:** The agent can observe workspaces without modifying them and can also alter them when requested by the user.

Architecture: CrypLLM follows a modular architecture consisting of four main parts:

- a UI layer that provides the chat panel, thread management, and settings;
- an agent layer that composes the system instructions, user messages, and optional workspace context into a prompt;
- a provider layer that abstracts access to different LLM backends through an OpenAI-compatible API (e.g., the OpenAI API using an API key, or any self-hosted AI backend (e.g. "LMStudio") implementing the same API contract). Currently, we use OpenAI GPT-5.4.
- a tooling layer that exposes accessors for workspace information and manipulation. This separation enables flexible deployment while keeping the agent tightly integrated into CT2.

Integration into CT2: CrypLLM is integrated into CT2 as a built-in chat panel that is available alongside the *WorkspaceManager*. It was developed and integrated into CT2 over approximately six months, including UI integration, prompt/agent design, provider abstraction, and workspace inspection tools. It allows users to create and manage multiple conversation threads. To do so, the agent relies on *tool calls* (OpenAI, 2023; OpenAI, 2025).

Via tool calls, it can determine whether a workspace is currently open, inspect and modify the workspace, including components, connections, values, states, and settings. This enables context-sensitive, workflow-specific guidance.

How to use CrypLLM: To use CrypLLM, users open the integrated chat panel on the right side of the workspace and start a conversation. The LLM used by the agent is displayed in the lower part of the chat panel and can be changed there.

CrypLLM supports both conceptual questions (e.g., explanations of cryptographic principles, grounded in the underlying knowledge of the selected LLM) and procedural guidance (e.g., assembling a workflow step by step, choosing suitable components, configuring component settings, or debugging a workflow). Scenario 2 below illustrates troubleshooting support.

This integration enables users to remain within the *WorkspaceManager* while receiving *just-in-time, context-aware* support.

4 Usage Scenarios

CrypLLM is intended to support users directly inside the *WorkspaceManager*. In the following, we outline three representative usage scenarios.

Scenario 1: Explaining an existing workspace: Here, a user opens an existing CT2 workspace and asks CrypLLM to explain what the workflow does. CrypLLM inspects the current workspace and provides a structured overview of the data flow, the involved components, and the role of important settings. Figure 2 shows an example for a Caesar cipher workspace, where CrypLLM summarizes the pipeline (plaintext $\rightarrow$ cipher $\rightarrow$ ciphertext), explains the purpose of auxiliary inputs such as alphabet and key, and describes how intermediate values propagate through the connections.

Scenario 2: Troubleshooting a broken workflow: When users modify workflows, the result-

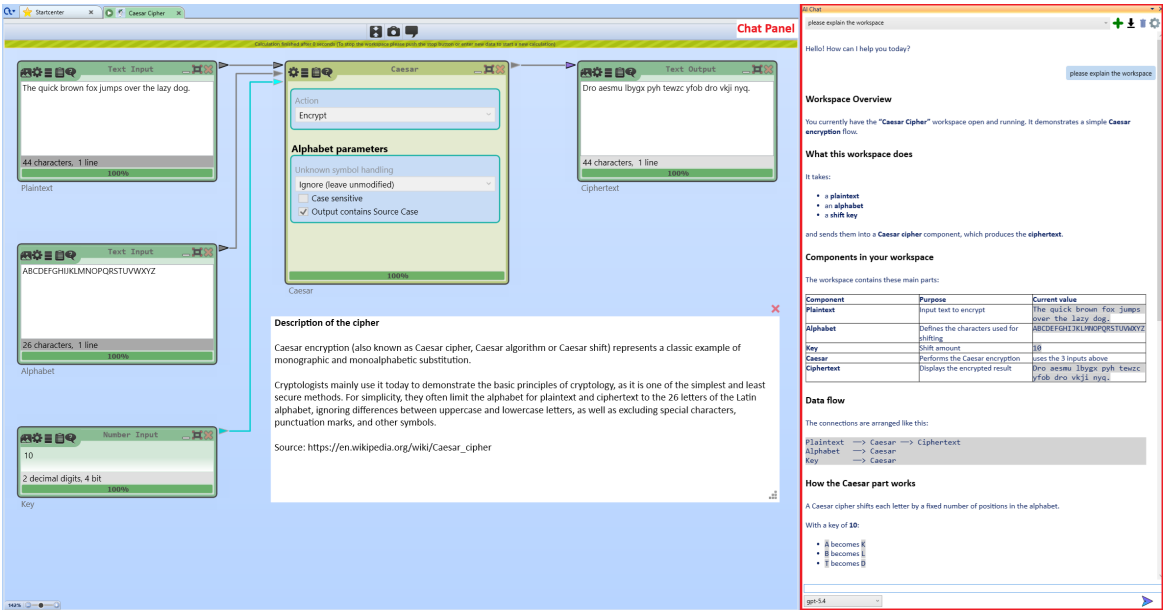


Figure 2: CrypLLM explaining an existing Caesar cipher workspace in CT2.

ing workflows may become invalid or produce unexpected results due to missing or incorrect connections, incompatible data formats, or unintended parameter settings. Figure 3 shows a Vigenère workspace in which the user expects a decryption result, but the output remains empty. CrypLLM inspects the workspace graph and identifies that the mandatory ciphertext input of the decryption component is not connected, and that several connections are inactive. It then provides concrete repair instructions, such as reconnecting the Vigenère encryptor output to the Vigenère decryptor input, ensuring that the connections are active, and re-running the workspace to update all dependent outputs. The user can decide whether to follow the instructions manually or ask the agent to execute them. In addition, CrypLLM can perform quick sanity checks on relevant settings (e.g., mode, key, alphabet) to help users distinguish wiring problems from configuration errors. The agent can also resolve these issues upon request.

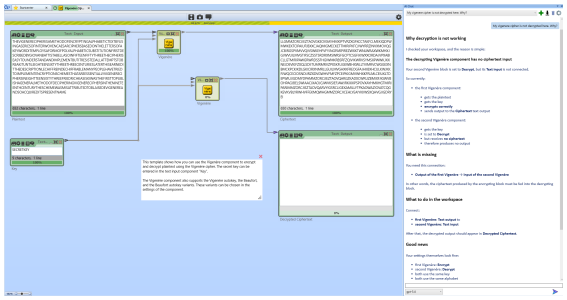


Figure 3: CrypLLM troubleshooting a broken Vigenère workflow.

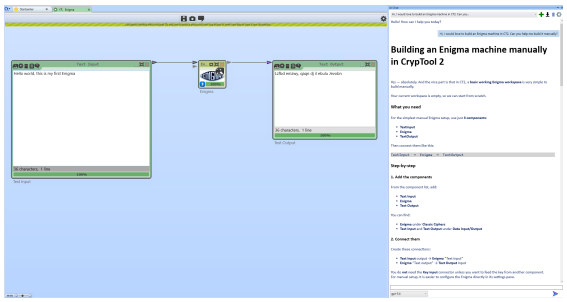


Figure 4: Scenario 3: CrypLLM guiding the construction of a minimal Enigma workflow from an empty workspace in CT2.

Scenario 3: Guided workflow construction from an empty workspace. In an e-learning setting, users often start with an empty workspace and want to build a workflow while learning the underlying cryptographic concept. Figure 4 illustrates such a scenario: The user opens a blank CT2 workspace and asks CrypLLM for instructions to create an Enigma workflow. Without requiring prior CT2-specific explanations from the user, the agent suggests a minimal setup by adding the Enigma component alongside text input and text output components, and by connecting them to form an encryption pipeline. CrypLLM can suggest CT2 components by accessing the component catalog through tool calls, including information such as component names, captions, and tooltips.

From an educational perspective, CrypLLM can reduce cognitive load by providing just-in-time

explanations and guidance. This aligns with principles of scaffolding in learning environments, where learners receive targeted support during problem-solving.

5 Conclusion

CrypLLM demonstrates how LLM-based agents can be tightly integrated into domain-specific tools to provide context-aware, actionable support. By combining conversational interaction with direct access to application state, the system improves usability and lowers the barrier to complex cryptographic workflows.

The presented usage scenarios demonstrate how CrypLLM supports both novice and experienced users by offering just-in-time guidance directly in CT2.

Current limitations include reliance on the underlying LLM's capabilities and reliability, as well as the need for careful terminology control to ensure that the agent's responses align with CT2's user-facing terminology.

Future work will focus on robustness, evaluation, and educational impact.

Acknowledgements

This work has been supported by Riksbankens Jubileumsfond, grant M24-0028: Echoes of History: Analysis and Decipherment of Historical Writings (DESCRYPT).

References

Anthropic. 2024. Introducing the Model Context Protocol. Anthropic News. https://www.anthropic.com/news/model-context-protocol.

Mark Chen et al. 2021. Evaluating large language models trained on code. arXiv:2107.03374. https://arxiv.org/abs/2107.03374.

Wayne Holmes, Maya Bialik, and Charles Fadel. 2019. *Artificial Intelligence in Education: Promises and Implications for Teaching and Learning*. Center for Curriculum Redesign (CCR). Additional copy available at: https://www.consorciosthem.com/wp-content/uploads/2025/02/sthem-ia-07-holmes-fadel-bialik-artificial-intelligence-in-education-promise-and-implications-for-teaching-and-learning-2019.pdf.

Nils Kopal and Bernhard Esslinger. 2018. CrypTool 2 – Ein Open-Source-Projekt zur Kryptologie für Lehre, Forschung, Selbststudium und Experimentieren. In *D·A·CH Security 2018*. https://www.syssec.at/de/veranstaltungen/dachsecurity2018/papers/DACH_Security_2018_Paper_11A2.pdf.

Nils Kopal and Bernhard Esslinger. 2022. New Ciphers and Cryptanalysis Components in CrypTool 2. In *Proceedings of the 5th International Conference on Historical Cryptology, HistoCrypt 2022*, number 188 in Linköping Electronic Conference Proceedings, pages 127–136. Linköping University Electronic Press. DOI: https://doi.org/10.3384/ecp188399.

Nils Kopal, Olga Kieselmann, Arno Wacker, and Bernhard Esslinger. 2014. CrypTool 2.0. *Datenschutz und Datensicherheit (DuD)*. DOI: https://doi.org/10.1007/s11623-014-0274-7.

OpenAI. 2023. Function Calling and Other API Updates. OpenAI Blog. https://openai.com/index/function-calling-and-other-api-updates/.

OpenAI. 2025. Function Calling (Tool Calling) Guide. OpenAI Platform Documentation. https://platform.openai.com/docs/guides/function-calling.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551.

Kurt VanLehn. 2011. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *The Eleventh International Conference on Learning Representations (ICLR 2023)*. https://openreview.net/forum?id=WE_vluYUL-X.

Solving Historical Ciphers with AI:

Analysis of GPT's Capability in Processing and Deciphering Cryptographic Postcards

Eugen Antal **Tomáš Pavuk** **Pavol Zajac**
Slovak University of Technology in Bratislava
Ilkovičova 3, 841 04 Bratislava
Slovakia
eugen.antal@stuba.sk xpavukt@stuba.sk pavol.zajac@stuba.sk

Abstract

In this case study, we focus on the capabilities of ChatGPT for assisting in the research of historical encrypted documents. Our study is based on the dataset of historical encrypted postcards, typically with simple (substitution) ciphers. We split the task into two parts: transcription and interpretation of the postcard image, and then decipherment of the transcribed text. We summarize the current capabilities of (multimodal) AI tools in general, and specifically of GPT-5.2 / GPT-4o that were used as AI tools in our analysis. Following our cryptanalytic pipeline, we show specifically how the AI tool was capable of solving selected typical exemplars from our dataset. We show that GPT-5.2 is already capable of solving most of the required tasks with acceptable quality, and can be useful even for researchers not specifically trained in prompt engineering or artificial intelligence usage.

1 Introduction

Research in historical cryptography requires a significant effort, from collecting historical materials, through their preservation and digitization, as well as for transcription and final decipherment of historical documents. Special tools are developed for computer-assisted transcription of old manuscripts, as well as for decryption of various types of historical ciphers. On the other hand, there is a rapid development of general AI tools and assistants that help with various tasks automatically. In our case study, we focus on whether and how well these general AI tools (focusing on ChatGPT[1]) can help in historical cryptography research.

Classical ciphers and their cryptanalysis can represent a significant and challenging cognitive task, with significant variations across different cipher types. As such, multiple authors have proposed using cryptographic challenges for the evaluation of the AI reasoning methods. AICrypto (Wang et al., 2025) is a benchmark based on CTF-type challenges. CipherBank (Li et al., 2025) is a comprehensive benchmark designed to evaluate the reasoning capabilities of LLMs in cryptographic decryption. The dataset comprises encrypted text from specifically generated and curated plaintext. A benchmark with diverse plaintexts from different domains was also proposed and used for LLM evaluation (Maskey et al., 2025b). There is also ongoing research that tries to evaluate the capabilities of AI tools to solve classical and modern ciphers, from initial exemplification studies (Noever, 2023), to a more complex examination of AI reasoning capabilities (Akilesh et al., 2026) and benchmarking studies (Maskey et al., 2025a).

We are, however, unaware of any systematic study that uses a multimodal dataset comprised of diverse historical data. Furthermore, most of the studies seem to focus on evaluating the reasoning capabilities or direct decryption of provided examples, instead of focusing on the whole pipeline of historical cipher research.

Our aim for this paper is to study the capabilities of (multi-modal) LLMs that can support human researchers in the study of historical ciphers. In the ideal world, the AI tool should be capable of processing the provided raw material (e.g., a scan or photography of a historical document), find regions of interest, transcribe the plaintext and ciphertext parts, translate the plaintext parts, provide statistical and contextual information about the ciphertext part, and possibly even try to solve the encrypted text. Our intention is also to provide an in-

[1]Currently, several multimodal AI models support extended reasoning. However, when we began our research, ChatGPT was the only model suitable for our use-case.

sight into the current capabilities of ChatGPT. We structure this paper more as a tutorial, showcasing existing capabilities, and providing some tips on how to efficiently use the existing AI tools in historical cryptography research.

In Section 2, we introduce our dataset (a collection of cryptographic postcards) and define our cryptanalytic pipeline. Section 3 briefly summarizes the AI-related technological background of this research. The achieved results are described in Section 4. The main contributions, limitations of the analysed models, and future work are discussed in Section 5.

2 The Dataset and the Cryptanalytic Pipeline

The object of our research is a collection of historical encrypted correspondence. The largest, publicly available collection of 988 cryptographic postcards (including Tobias Schrödel's collection (Schrödel, 2021)), is accessible on the HCPortal database[2] of cryptograms and cipher keys (Antal and Zajac, 2020; Antal and Zajac, 2026). These postcards date mainly from the early 20th century, and the encryption systems used are relatively simple and well-known. This makes them ideal as a challenge for processing with AI assistants.

Please be aware that because the dataset is accessible online, there's a possibility that it may have been (in some way) included during the training of the investigated large language models. However, in most cases, the transcriptions or solutions to the cryptograms are not available in the database, and it's not easy to obtain them from the images without additional processing. Moreover, there is no available procedure for solving the problems being examined.

2.1 Analysis of the Dataset

We systematically analysed the available digitized images with a focus on the used cipher system and on the used characters (ciphertext alphabet). Because the postcards may contain the ciphertext on the picture side, address side, or on both sides, we decided to manually process the collection per individual scanned image, where at least a portion of the ciphertext is present.

From the resulting 1010 images, the vast majority (836x) were encrypted using a substitution cipher (mostly monoalphabetic). The substitution ciphers found can be further divided based on the character sets into the following categories: numbers only[3] (160x), letters only (79x), symbols only[4] (241x) and mixed alphabet (356x). The few transposition ciphers (10x) were mainly grammatical (mirror/reverse writing) or geometrical (text written in columns). The steganographic techniques (15x) were various, such as anamorphic writing, text hidden under the stamp, etc. The shorthand category contains classical shorthand messages and other *shorthand-like* ciphers. Similar to Schrödel (2021), we will leave them out of further analysis, because we were unable to clearly identify them. Morse codes are not ciphers; however, many times presented on the auction sites as codes/ciphers. The *other* category contains 10 (possibly) abbreviation ciphers, and 1 image, where some letters were omitted from the message. In 21 cases, we were unable to clearly determine the category.

2.2 Definition of our Cryptanalytic Pipeline

According to the results of our analysis (see Table 1), the covered cipher types are relatively easy to solve. A simple substitution solver can cover the decipherment of the vast majority of the postcards.

In more detail, we selected three main categories of postcards for further experiments (based on the difficulty of the solving process):

- A numeric substitution. The cipher key is constructed based on the lexicographical order of the plaintext alphabet letters. This is the easiest cipher type/task.
- A Pigpen cipher. Medium difficulty because the cipher is graphical/geometrical, and different patterns exist, but most of them are known. If the letters are randomly assigned to the Pigpen grids, the cipher decryption task is equivalent to the hard difficulty category.
- A general simple substitution (with symbols/letters/mixed alphabets). The hardest difficulty as the analysis usually requires statistical and pattern analysis.

Our main objective is not the decryption itself, but rather the study of the interpretation, transcription, and analysis capabilities of multimodal AI

[2] https://crypto.hcportal.eu/

[3] In many cases, the letters were replaced with numbers based on their lexicographical order.

[4] From which 61 are Pigpen.

	Used characters				
Cipher type	Number	Letter	Symbol	Mixed	N/A (other)
Substitution	160	79	241	356	
Transposition		10			
Steganography					15
Shorthand					100
Morse code			17		
Other		11			
Unidentified					21

Table 1: Cryptographic postcards statistics.

tools. Processing of the postcards requires visual analysis and transcription before the decipherment attempt. In this regard, we focus on the following tasks:

- Transcription of numeric substitution ciphers — processing handwritten numbers.
- Transcription and interpretation of the Pigpen cipher — processing symbols.
- Transcription of substitution ciphers that use mixed alphabets.
- Interpretation of transposition and steganographic techniques — processing letters and graphic shapes.
- Analysis of the transcribed text.

We acknowledge the fact that most of the ciphers used in our dataset are trivial and relatively easy to decipher either manually or using dedicated software. This is, however, the ideal situation to use AI tools in their cryptanalysis, for offloading and centralizing the set of tasks itself. The AI-powered analysis, interpretation, and transcription could help amateur codebreakers, enthusiasts, or postcard collectors without any cryptologic background, and without the need to find or construct specific tools for specific tasks.

While we would like to automate the entire task, our experimental pipeline consists of two main parts:

- Processing the scanned image (analysis, interpretation, and transcription).
- Processing the transcribed text (decipherment).

This helps us to assess the parts separately, while in practice, it is easy to connect these two parts to form a full analytic pipeline.

3 AI Tools and How to Use Them

The term Large Language Model (LLM), in a popular sense, has become synonymous with a chat interface driven by a generative transformer implementing a statistical representation of a large language model inside a complex neural network, such as ChatGPT. Moreover, the term LLM has been conflated with the term Artificial Intelligence (AI). Even if imprecise, we will also call these and similar tools and APIs either AI tools or simply LLMs.

Recent work distinguishes between *general-purpose LLMs* and *Large Reasoning Models* (LRMs) (Srivastava and Yao, 2025). General-purpose models, such as GPT-4o, handle diverse tasks through single-pass inference without explicit multi-step explanation. In contrast, reasoning models such as OpenAI o1 and DeepSeek R1 allocate additional computation to chain-of-thought processing, decomposing complex problems into intermediate steps.

Multimodal models are capable of processing multiple input modalities: text, images, and documents within a unified architecture (Lin et al., 2025). This capability enables direct analysis of scanned materials without requiring separate preprocessing or transcription steps.

Prompt engineering (Schulhoff et al., 2024) is a technique of crafting specific prompts to the chat interface of a generative transformer in such a way that the desired results have a high(er) chance to occur as a response of the transformer. Prompt engineering can also optimize the outputs of the reasoning models (Srivastava and Yao, 2025). Moreover, LLMs themselves can be tasked to optimize prompts for a given task, a technique where the model reformulates the user's intent into a more effective prompt for itself or another model, or even tailors it based on a prompting strategy.

Common prompting strategies include: *Zero-shot*, *Few-shot*, and *Chain-of-thought* (CoT) prompting (Schulhoff et al., 2024).

Interactions with LLMs can be *single-turn*, where the model produces a solution in response to a single prompt, or *multi-turn*, where the solution emerges through iterative refinement based on user feedback. In multi-turn interactions, feedback may include corrections, hints, or additional constraints that guide the model toward the desired output.

Modern LLM platforms extend the base text-generation capabilities through integrated *tools*. These include code interpreters for executing code, web search for retrieving external information from allowed or specified sources, file input for processing uploaded files that the model cannot handle natively, and more. Tools transform pure text inference into assisted analysis, where the model can delegate specific subtasks to specialized components and build its context from their outputs.

Visual processing in multimodal models can operate in different modes (Lin et al., 2025). *One-pass visual perception* processes the image in a single forward pass without iterative refinement. *Active visual perception* involves iterative processing where the model revisits the image, considers alternative interpretations, and integrates visual evidence with contextual reasoning, optionally leveraging tools such as a code interpreter to preprocess the image (e.g., adjusting contrast or orientation). The latter is particularly relevant, as handwritten characters may be ambiguous.

LLMs can be accessed either through a conversational *chat interface* or programmatically via an *API* (Application Programming Interface). While chat interfaces are designed mostly for interactive use, APIs additionally enable automated and batch processing of requests. APIs also enable more detailed configuration of LLM's system prompt or its response, which allows for advanced orchestration of multiple LLMs and custom tools for the LLM to execute.

3.1 ChatGPT Overview

ChatGPT is a conversational AI system developed by OpenAI, powered by models from the GPT (Generative Pre-trained Transformer) family.

The platform provides access to multiple model variants and reasoning configurations, as well as integrated tools relevant to the tasks considered in this paper. In particular, multimodal capabilities allow the direct analysis of scanned or photographed historical documents, while advanced reasoning modes support exploratory statistical analysis, hypothesis generation, and cryptanalytic reasoning.

This subsection provides an overview of the operating modes, available model variants, and access mechanisms relevant to our experimental setup. The description is intended to clarify the technical context in which the experiments were conducted and to support the reproducibility and interpretation of the results presented in later sections.

Prior work demonstrates that prompt formulation significantly affects the behaviour of generative transformer models (Schulhoff et al., 2024), and that prompt engineering remains effective even for models equipped with explicit reasoning capabilities (Srivastava and Yao, 2025). Consequently, model variant, operating mode, and prompt design are treated as experimentally relevant variables rather than fixed implementation details.

3.1.1 Subscription Tiers

ChatGPT is offered through multiple access tiers that differ in usage limits, availability of reasoning modes, and access to integrated tools. At the time of the experiments, the available tiers included Free (with and without login), Go, Plus, Pro, Edu and Business (OpenAI, 2026a; OpenAI, 2026b). These distinctions are relevant for reproducibility, as certain reasoning modes and tools are restricted or rate-limited depending on the access level.

In particular, logged-in Free users can manually activate extended reasoning via the input interface, whereas anonymous users rely solely on automatic mode selection. Higher-tier plans provide increased usage limits and access to advanced reasoning configurations.

Table 2 presents the available models in ChatGPT. Recent research emphasizes a distinction between general-purpose large language models and models explicitly optimized for extended reasoning (Srivastava and Yao, 2025). While all modern transformer-based models exhibit some degree of implicit reasoning, reasoning-optimized variants allocate additional computation to multi-step inference. This distinction is particularly relevant for cryptanalytic tasks, which often

require hypothesis generation, iterative refinement, and validation rather than direct pattern completion.

Model	Ctx	Out
GPT-5.2 Auto	128–196k	32k
GPT-5.2 Instant	128k	32k
GPT-5.2 Thinking	196k	32k
GPT-5.2 Pro	196k	32k
Legacy models		
GPT-5.1 Inst./Think.	128k/196k	16k
GPT-5 Inst./Think.	128k/200k	16k
GPT-5 Think. mini	200k	16k
GPT-4o	128k	16k
GPT-4.1	1M	32k
o3	200k	100k
o4-mini	200k	100k

Table 2: Maximum context and output tokens for ChatGPT models (not API). Ctx: Context window (tokens), Out: Maximum output tokens.

GPT-5.2 is the current flagship model, optimized for professional knowledge work including creating spreadsheets, presentations, writing code, perceiving images, and handling complex multi-step projects. It demonstrates state-of-the-art performance in long-context reasoning, vision tasks (error rates cut roughly in half on chart reasoning and software interface understanding), and tool calling (OpenAI, 2026c).

3.1.2 Operating Modes

GPT-5.2, GPT-5.1 and GPT-5 models operate in different modes (auto, instant, thinking, pro) that control reasoning depth (OpenAI, 2026d).

The choice of operating mode is particularly consequential for tasks involving ambiguity or multi-step reasoning. Cryptanalytic analysis of historical material often requires evaluating multiple plausible interpretations of partially degraded or handwritten symbols prior to decryption. Operating modes that emphasize extended reasoning are therefore more appropriate for such tasks, whereas faster modes are primarily designed for single-pass processing.

3.1.3 Integrated Tools

ChatGPT provides a set of integrated tools (code interpreter, web search, file input) that extend its base text-generation capabilities. These tools are natively available through the chat interface and may be enabled or disabled depending on the task and access tier.

These tools are conceptually distinct from the underlying language model and can substantially alter the nature of a task by transforming it from pure inference into assisted analysis. Consequently, their usage is treated as an explicit experimental factor in this study and is reported separately where applicable.

3.1.4 Model Selection

For the experiments presented in this paper, we selected GPT-5.2 in its Thinking mode as the primary model. This choice was motivated by two principal considerations: (1) GPT-5.2 represents the current flagship model with state-of-the-art performance in vision tasks and long-context reasoning, both critical for historical document analysis. (2) The selected configuration is accessible to free-tier users, ensuring that our experimental setup can be reproduced without requiring a paid subscription.

To assess the contribution of extended reasoning capabilities, we additionally evaluated GPT-4o, a legacy non-reasoning model that lacks chain-of-thought functionality. This comparison allows us to isolate the impact of explicit reasoning on cryptanalytic performance and the overall progression of LLMs.

Furthermore, given evidence that prompt engineering can influence model performance even in reasoning-enabled systems (Srivastava and Yao, 2025), we systematically evaluate both zero-shot prompts and optimized prompt formulations. This design allows us to separate inherent model capabilities from sensitivity to task framing.

Table 3 summarizes the configurations proposed for our experiments.

Model	Reasoning	Mode	Tools
GPT-5.2	Yes	Thinking	Enabled
GPT-4o	No	—	Enabled

Table 3: Selected GPT model configurations.

4 Results of Our Experiments With AI Tools

We evaluated the two selected GPT models (see Table 3) on the two main logical parts of our pipeline, defined in Section 2.2.

4.1 Image Processing Capabilities

First, we focused on the analysis of the GPT models' image processing capabilities of the scanned images.

4.1.1 Numeric substitution

In the first experiment, we analysed whether it is possible to successfully transcribe handwritten numbers. We evaluated the address side of the *pc_hcp_191* postcard[5] from HCPortal. It contains twelve lines of ciphertext encrypted with a simple substitution using numbers as ciphertext characters. Moreover, each number is separated by a dash, and individual words are also separated by a comma. We prompted the models to transcribe the text from the postcard without providing any further details.

GPT-4o transcribed the text with a lot of transcription errors and without noticing the commas. Therefore, the result is only partially usable in further decipherment attempts. Communication 1 is accessible from the Appendix.

GPT-5.2 handled the transcription quite well, including the word separators. It made only a few mistakes, only in the areas where the handwriting was not clear or covered by other printed parts on the postcard. Communication 2 is accessible from the Appendix.

This experiment showed that there is no need for any prompt engineering for the transcription of handwritten numbers. Both models identified the used cipher without asking for it, and *GPT-5.2* achieved an almost correct transcription. We recommend using it for this task.

4.1.2 Substitution with symbols and mixed alphabet

The real challenge, however, remains whether we can also transcribe texts where various symbols and glyphs were used. Transcription of a ciphertext containing such a large variety of strange symbols is not trivial.

First, we focused on an easier cipher instance – on a Pigpen cipher, where a regular (known) Pigpen grid was used. We evaluated the address side of the *pc_hcp_135* postcard[6] from HCPortal. It contains six lines of ciphertext encrypted with a classic Pigpen cipher and one additional row above the receiver's address. To read the ciphertext correctly, it is necessary to flip the postcard upside down. We asked the models to transcribe the postcard.

While *GPT-4o* transcribed the address (handwritten plaintext) without problem, the cipher symbols were transcribed incorrectly. The model tried to transcribe it to some Unicode graphical symbols with similar shapes as used in the Pigpen cipher; however, the transcription, the number of transcribed symbols, and the number of lines are incorrect. The postcard's correct orientation was also not identified. Communication 3 is accessible from the Appendix. Even when we constructed a detailed prompt with specific instructions, it was unable to correctly interpret and transcribe the text. The only property identified correctly was the flipped orientation of the postcard. Communication 4 is accessible from the Appendix.

GPT-5.2 handled the transcription much better (after specially asking for the transcription of the ciphertext). The model tried to transcribe and represent the cipher symbols with Unicode shapes (differently from *GPT-4o*). It tried to construct the shapes with their base shape and with an additional bullet (if it was required). It also recognized whether the bullet is inside or after the shape. The encoding looks more precise and is almost correct (with a few errors). However, the problem with wrong image orientation is still present. Next, we specified in the prompt that only Pigpen characters were used, asked to check the image orientation, and informed the model about the transcription errors. Except for one additional bullet and one mistake in the first line, the transcription was correct. The result is satisfying and can be potentially used in cryptanalysis. Communication 5 is accessible from the Appendix.

Next, we focused on a harder cipher instance – on a substitution cipher using a mixed alphabet of letters, numbers, and symbols. We evaluated the address side of the *cpc_1919-02-07_USA_Dubois_USA_Springville* postcard[7] from HCPortal.

GPT-4o was unable to correctly transcribe the ciphertext, whether we used simple or detailed instructions in the prompt. Communications 6, 7, and 8 are accessible from the Appendix.

GPT-5.2 handled the transcription much better, but not without errors. Mostly, it mistaken number 4 (or a symbol with a similar shape to number 4) as 7, number 3 (almost closed upper circle in number 3) as 9, etc. Despite these errors, we can use the partially correct transcription to (at least partially)

[5] https://crypto.hcportal.eu/dashboard/cryptograms/1510
[6] https://crypto.hcportal.eu/dashboard/cryptograms/1454
[7] https://crypto.hcportal.eu/dashboard/cryptograms/944

solve the message. Communication 9 is accessible from the Appendix.

We can conclude that a ciphertext containing uncommon symbols is problematic to transcribe in general. The transcription success depends on the symbol set used and on the quality of the input image.

4.1.3 Transposition (standard letters with unconventional text-writing direction)

In this experiment, we focused on analysing the transcription capabilities of handwritten text, which was written in an unconventional manner. We evaluated the address side of the *pc_hcp_508* postcard[8] from HCPortal. It contains relatively easy-to-read handwritten text written in columns from left to right, instead of rows (similarly to the encryption process of the Scytale cipher), followed by three columns of similar characters, rotated by 90 degrees. The words were separated with a separator character (X). Moreover, for some columns, the writing style (orientation of the letters) at the end of the column was changed. The address and name of the recipient are also written in a special way. We prompted the models to identify what is on the input image without any additional context, followed by a second prompt, where we specified to transcribe the whole postcard and describe every component.

The *GPT-4o* transcribed most of the vertical columns correctly, including the recipient details, even the details from the post stamp. However, it was unable to read the rotated columns located at the bottom of the image. Communication 10 is accessible from the Appendix. In general, the model can handle unusual text writing directions with a couple of errors. The biggest problem for this model is to deal with the rotated part of the text.

We repeated the experiment with the *GPT-5.2* model. The result was almost perfectly correct. This model sequentially crops, adjusts, and analyses the image parts to achieve the best results. The model also identified that letter X was used as a word separator and can identify typos in the written text (made by the scribe). After we specified that some lines were rotated by 90 degrees, the model rotated and successfully read the rotated lines (correctly), too. Communication 11 is accessible from the Appendix. This model is very suitable for analysing and transcribing handwritten text, even with different writing directions or when the text writing direction is changed halfway. There was no need for special prompts (except for the text rotation). The achieved result is acceptable with only minor typos.

4.1.4 Steganography (anamorphic writing)

In this experiment, we focused on analysing the image transformation capabilities of the models. We evaluated the picture side of the *pc_hcp_140* postcard[9] from HCPortal. It contains a special text writing style/ text deformation technique – called anamorphic writing. The meaningful text is visible (in most cases) only if the image is stretched. The text on the postcard is also rotated by 180 degrees.

We prompted the *GPT-4o* model to transcribe the image with a hint that it may be rotated. The model was not able to identify the used technique nor interpret it correctly. Communication 12 is accessible from the Appendix. In the next step, we used a more sophisticated prompt, including detailed instructions. The model decided to use a series of geometric transforms (non-uniform scaling, rotations, and optional blur + thresholding) in an attempt to recover readable characters. The proposed result was still incorrect. Communication 13 is accessible from the Appendix.

The *GPT-5.2* model was also unable to identify the writing technique used and to transcribe the text without providing detailed instructions. Communication 14 is accessible from the Appendix. Next, we used the extended prompt with the instructions. After the model was unable to OCR the whole text successfully, it started to segment the image into word-sized chunks and repeated the OCR. We obtained only a partially correct result, despite the model presenting the result with a very high confidence. Communication 15 is accessible from the Appendix. Despite the specific instruction in the previous prompt, we were able to obtain the correct transcription only after we extracted the transformed image produced by the model during its reasoning process and used it as input (still rotated by 180 degrees). The model was able to identify and transcribe the correct text without the need for additional details. Communication 16 is accessible from the Appendix.

[8] https://crypto.hcportal.eu/dashboard/cryptograms/1827

[9] https://crypto.hcportal.eu/dashboard/cryptograms/1459

4.2 Decipherment

As per our pipeline definition, the initial step is to obtain the transcription of the ciphertext from the postcard, which will then enable us to proceed with the decipherment process. In the following, we present decipherment attempts of selected substitution ciphers (in cases where we were able to obtain a suitable transcription in the previous step).

4.2.1 Numeric substitution

We continue our experiment with the *pc_hcp_191* postcard. Because the *GPT-4o*'s transcription was not correct, we were also unable to obtain the correct decryption even if the model correctly guessed the used cipher type and key. Only part of the result (a few words) was correct.

We prompted the *GPT-5.2* model to decipher the numbers after receiving the transcription. The model successfully identified[10] the used cipher type and key. The decipherment was successful, and only a few small mistakes occurred. The model was also able to correct most of its transcription errors during the decipherment (probably based on the context analysis of the resulting text). Communication 17 is accessible from the Appendix.

This experiment showed that only *GPT-5.2* is suitable for cryptanalysis of this cipher type.

4.2.2 Substitution with symbols and mixed alphabet

Because only *GPT-5.2* was able to successfully transcribe the *pc_hcp_135* postcard, we focused only on this model. First, we prompted the model to transcribe the Pigpen cipher. After the initial transcription, we asked the model to correct the (few) errors and decipher the ciphertext. It was also necessary to specify that the text was flipped. The model was unable to process our request without providing additional information. It provided us with the following options:

- Specify if the sender used the standard Pigpen alphabet.
- Provide if the stand-alone dots between groups are meant to be spaces.
- Provide some cribs.

[10] As an A1Z26 cipher: 1 = A, 2 = B, ..., 26 = Z (hyphens = letters, commas = word breaks).

We decided to provide the correct key as an input image, instead of selecting one from the previous option. The resulting plaintext was fully correct. Communication 18 is accessible from the Appendix. We also tried a different approach – we instructed the model to search for the key. At the beginning, the model summarized various known Pigpen cipher keys (mainly from Wikipedia); however, it was not able to find/identify the correct key and provided an incorrect result. When we asked to use a different key, it found the correct Pigpen variant from the dCode website. In this phase, it even tried to correct the misspelled words. Communication 19 is accessible from the Appendix. Then we tried a new approach with a more detailed initial prompt. When the model asked us for user input, we instructed the model to try to brute-force the cipher key. Unfortunately, it was unable to find the correct key in this way. We can deduce from our experiments that if we don't provide the possible cipher key, or it can't be found over the internet (we still need to specifically ask the model to search for it), or a randomly assigned Pigpen grid was used, this cipher type is not solvable with GPT. Communication 20 is accessible from the Appendix.

In our last experiment, we tried to decipher the substitution cipher with a mixed alphabet. The *cpc_1919-02-07_USA_Dubois_USA_Springville* postcard was transcribed only partially. Despite this fact, we decided to try to decipher it with *GPT-5.2*. We continued in the previous communication and asked the model to solve the transcribed text. It was able to correctly decrypt five words. From the result, it is possible to manually obtain the whole solution. Communication 21 is accessible from the Appendix.

5 Conclusions

In this work, we explore the capabilities of ChatGPT in assisting with the processing, interpretation, and decipherment of historical encrypted documents. We evaluated two multimodal GPT models: GPT-4o and GPT-5.2. Our case study has shown that GPT-5.2 is suitable to process and solve various historical ciphers directly from an image input (of a cryptographic postcard).

Our approach uses visual and reasoning features of ChatGPT, and not its ability for text completion. Even though some of the used examples could

have been seen by the model during the training, it is clear from the achieved results (reasoning traces and various mistakes) that the model isn't affected by the online database and really was supported by long context reasoning, multimodality (visual processing), and tool calling.

With correct prompting, the AI tool is able to automatically apply transcription, image interpretation, and cryptanalytic tasks. We were able to successfully transcribe or interpret most of the investigated postcards, as well as decipher their contents. Moreover, in the case of easier instances, any special prompt engineering was completely unnecessary. Note that the older model, GPT-4o, is behind GPT-5.2 in every aspect, and was only rarely useful for our tasks.

One notable drawback of the GPT-5.2 model is its extended duration in achieving the required outcomes during our experiments, especially when it is performing image processing operations. For example, when processing Pigpen experiments, the initial prompt took approximately 15–25 minutes to process, with subsequent prompts often taking 30 minutes or more. This was mainly due to the model's reasoning/thinking and subsequent correction of its own mistakes (mainly image processing tasks). In some cases, the context window proved insufficient, leading to timeouts before reaching the final output.

For optimal results when using AI tools for historical cipher analysis, we recommend following these guidelines:

- **Image orientation.** Explicitly instruct the model to check and correct image rotation before processing. Printed elements (e.g., "POST CARD", address fields) can serve as reference points, but note that the ciphertext may have a different orientation than the printed text, so the model should try multiple orientations.
- **Contextual framing.** Provide as much context as possible: specify that the input is a postcard or personal letter, mention expected conventions such as salutations, closings, period-typical abbreviations, common phrases, and likely language. The more contextual information provided, the better the transcription and decipherment accuracy. If the cipher type is known, specify it explicitly.
- **Distinguish cursive from symbols.** Stylized cursive handwriting with flourishes is often misinterpreted as a symbolic cipher. Explicitly state that the text is handwritten in Latin script to prevent this confusion.
- **Manual reading.** The observed default behaviour of the model is to use automated OCR/computer vision, which often fails for symbolic ciphers and cursive text. Explicitly instruct the model to skip OCR and read characters manually from the start. The model can still use code tools for image enhancement, cropping, or frequency analysis.
- **Systematic processing.** For ciphertexts with many symbols or characters, instruct the model to process row-by-row or column-by-column rather than attempting one-shot transcription. Additionally, instruct the model to count and verify the number of characters per line to ensure the transcription matches the image.
- **Symbol inventory.** Request the model to create an explicit legend or inventory of all distinct symbols identified in the ciphertext. This provides the model with a consistent base for symbol interpretation throughout the transcription and allows the user to inspect and correct the model's symbol recognition. Alert the model to easily confused symbols: mirrored shapes, rotational variants ("6" vs. "9", "d" vs. "p"), and similar corner orientations.
- **Known plaintext.** If any portion of the plaintext is known, provide it to the model — this dramatically improves decipherment accuracy.
- **Cipher-specific considerations:**
  - *Numeric substitution* (lexicographic): Minimal prompting is sufficient; transcription errors often self-correct during decoding.
  - *Pigpen*: Only standard keys are recognized; variant keys must be provided or the model instructed to search for alternatives. Verify the model's key interpretation, as symbol orientations are frequently confused.
  - *Transposition*: Explicitly specify reading direction (columnar vs row-based).

– *Symbol/mixed substitution*: Higher error rates due to glyph variety and similar-looking characters; row-by-row processing and iterative correction recommended. Context exhaustion and timeouts are common for complex instances.

Final research notice: This article was constructed as a case study, with the aim to help in automating specific tasks in historical cipher research. In addition to the examples mentioned above, we have conducted other experiments that are not included in this article. Our recommendations are based on the experience gained from all our experiments. We do not provide a full statistical evaluation of current AI capabilities, and we have not examined the capabilities of other AI models than GPT-4o and GPT-5.2. Furthermore, we are aware that it is appropriate and worthwhile to conduct a more systematic comparison with other models as well. Therefore, we are working on an expansion of our research, where we are comparing GPT with other AI systems, such as Claude or Gemini. The achieved results will be published in a future publication.

Acknowledgments

This research was supported by the project "Artificial intelligence for encrypted handwritten document processing", "09I05-03-V02-00031" of call "Support of research projects aimed at digitization of the economy in TRL levels 1-3", Call. No. 09I05-03-V02, managed by the Research Agency and funded by The Recovery and Resilience Facility of the Slovak Republic.

References

S Akilesh, Rajeev Sekar, R Abinaya, K Palani Thanaraj, and C Christopher Columbus. 2026. Zero-shot chain-of-thought reasoning for cryptanalysis of popular ciphers. In *2026 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, pages 1–6.

Eugen Antal and Pavol Zajac. 2020. HCPortal Overview. In *Proceedings of the 3rd International Conference on Historical Cryptology HistoCrypt 2020*, pages 18–20. Linköping University Electronic Press.

Eugen Antal and Pavol Zajac. 2026. HCPortal: Ten Years of Development. In *Proceedings of the 9th International Conference on Historical Cryptology HistoCrypt 2026 (accepted)*.

Yu Li, Qizhi Pei, Mengyuan Sun, Honglin Lin, Chenlin Ming, Xin Gao, Jiang Wu, Conghui He, and Lijun Wu. 2025. CipherBank: Exploring the boundary of LLM reasoning capabilities through cryptography challenge. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5929–5965, Vienna, Austria. Association for Computational Linguistics.

Zhiyu Lin, Yifei Gao, Xian Zhao, Yunfan Yang, and Jitao Sang. 2025. Mind with eyes: from language reasoning to multimodal reasoning. *arXiv preprint arXiv:2503.18071*.

Utsav Maskey, ZHU Chencheng, and Usman Naseem. 2025a. Benchmarking large language models for cryptanalysis and side-channel vulnerabilities. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 19849–19865.

Utsav Maskey, Chencheng Zhu, and Usman Naseem. 2025b. Benchmarking large language models for cryptanalysis and side-channel vulnerabilities. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 19849–19865, Suzhou, China. Association for Computational Linguistics.

David Noever. 2023. Large language models for ciphers. *International Journal of Artificial Intelligence & Applications*, 14:1–20.

OpenAI. 2026a. Chatgpt enterprise and edu — models & limits. Accessed: 2026-01-12.

OpenAI. 2026b. Chatgpt pricing. Accessed: 2026-01-12.

OpenAI. 2026c. Introducing gpt-5.2. Accessed: 2026-01-12.

OpenAI. 2026d. Using the latest model (gpt-5.2) — api guide. Accessed: 2026-01-12.

Tobias Schrödel. 2021. Cryptographic postcards. In *Proceedings of the 4th International Conference on Historical Cryptology HistoCrypt 2021*, pages 131–136. Linköping University Electronic Press.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, Pranav Sandeep Dulepet, Saurav Vidyadhara, Dayeon Ki, Sweta Agrawal, Chau Pham, Gerson C. Kroiz, Feileen Li, Hudson Tao, Ashay Srivastava, Hevander Da Costa, Saloni Gupta, Megan L. Rogers, Inna Goncearenco, Giuseppe Sarli, Igor Galynker, Denis Peskoff, Marine Carpuat, Jules White, Shyamal Anadkat, Alexander Miserlis Hoyle, and Philip Resnik. 2024. The prompt report: A systematic survey of prompting techniques. *CoRR*, abs/2406.06608.

Saurabh Srivastava and Ziyu Yao. 2025. Revisiting prompt optimization with large reasoning models — a case study on event extraction. *arXiv preprint arXiv:2504.07357*.

Yu Wang, Yijian Liu, Liheng Ji, Han Luo, Wenjie Li, Xiaofei Zhou, Chiyun Feng, Puji Wang, Yuhan Cao, Geyuan Zhang, Xiaojian Li, Rongwu Xu, Yilei Chen, and Tianxing He. 2025. AICrypto: A Comprehensive Benchmark For Evaluating Cryptography Capabilities of Large Language Models. *CoRR*, abs/2507.09580.

Appendices

Communication 1:
https://chatgpt.com/share/697aabad-25a0-8013-8c7f-25fd063f12ba

Communication 2:
https://chatgpt.com/share/697aaa83-fadc-8013-9227-2a54b85d78ff

Communication 3:
https://chatgpt.com/share/697bd6c1-1254-8013-a2ec-e67ec44e2e97

Communication 4:
https://chatgpt.com/share/697be07c-b008-8013-9a5e-28bb474db70c

Communication 5:
https://chatgpt.com/share/697e416f-aef4-8013-a808-e869369bfedf

Communication 6:
https://chatgpt.com/share/697c090f-aab0-8013-8838-0fa9da2a391f

Communication 7:
https://chatgpt.com/share/697c094e-c9dc-8013-a09a-ab04a603e041

Communication 8:
https://chatgpt.com/share/697c08c6-338c-8013-b472-cc750c2fafde

Communication 9:
https://chatgpt.com/share/697c1060-6540-8013-8745-7780ea71c41d

Communication 10:
https://chatgpt.com/share/697a91a9-e728-8013-b137-b85a43844515

Communication 11:
https://chatgpt.com/share/697a9a94-6b78-8013-ae27-5d2d3c32c26b

Communication 12:
https://chatgpt.com/share/697b53dd-9e28-8013-9e8a-ec74cd753da5

Communication 13:
https://chatgpt.com/share/697b513e-8658-8013-b7f7-117f8da8dc92

Communication 14:
https://chatgpt.com/share/697b553c-7ce0-8013-a071-82bc172cad6a

Communication 15:
https://chatgpt.com/share/697b5187-2fa4-8013-b3dd-8fce1c0b87e0

Communication 16:
https://chatgpt.com/share/697aa80f-7694-8013-8c68-c99926be0b11

Communication 17:
https://chatgpt.com/share/697aaa83-fadc-8013-9227-2a54b85d78ff

Communication 18:
https://chatgpt.com/share/697c036c-ac9c-8013-b390-8c621ccae5b0

Communication 19:
https://chatgpt.com/share/697c0ae7-4c0c-8013-9814-8557d24ce43c

Communication 20:
https://chatgpt.com/share/697c1272-f884-8013-b2af-965babecbb50

Communication 21:
https://chatgpt.com/share/697c1060-6540-8013-8745-7780ea71c41d

Exploring the Automatic Alphabet Identification of Images of Handwritten Ciphers

Alejandra Reinares
Giuseppe De Gregorio
Alicia Fornés
Computer Vision Center
Universitat Autònoma de Barcelona
{gdegregorio, afornes}@cvc.uab.cat

Beáta Megyesi
Stockholm University, Sweden
beata.megyesi@ling.su.se

Abstract

Historical encrypted manuscripts often use invented or heterogeneous alphabets, making alphabet identification a necessary but traditionally manual first step prior to transcription and decryption. This work explores the use of unsupervised computer vision methods to automate this task without requiring labeled data. We propose a pipeline that segments characters from cipher manuscripts, groups them into clusters of visually similar symbols using unsupervised methods, and compares those clusters against a reference database of known alphabet symbols to identify the most likely underlying writing system. Experiments show that the method can correctly identify the alphabet when a handwritten alphabet is available, but performance degrades when handwritten symbols are compared against printed alphabets, with handwriting style dominating shape similarity. These results highlight the importance of realistic handwritten reference alphabets.

1 Introduction

Historical encrypted manuscripts were frequently written using invented alphabets that mixed digits, Latin or Greek letters, graphical signs like alchemical or zodiac symbols, and even completely invented symbols, often combined with diacritics. Consequently, the first step, prior to the transcription and decryption of the encrypted documents, consists of identifying the underlying cipher alphabet. This task, similar to script identification (Bashir et al., 2022; Khan et al., 2024) and alphabet comparison (Chen et al., 2021), has been traditionally manual, time-consuming, and has required specialized expertise.

Several handwritten text recognition (HTR) techniques have been applied for transcribing ciphers (Souibgui et al., 2023; Fornés et al., 2024), but there are two requirements: first, we must identify the alphabet, including deciding the transcription labels for each symbol/character, and second, it is necessary to provide labeled examples of that type of alphabet to train HTR models. However, the unique nature of ciphers, with invented alphabets and few labeled data, makes supervised deep learning methods unsuitable.

This work explores the use of unsupervised methods for the identification of the alphabet by comparing characters of an input undeciphered document against a database of known alphabets.

While the current reference database comprises only six alphabets, whose visual differences are often sufficient for a human expert to distinguish at a glance, the proposed approach is designed with scalability in mind. As cipher research progresses and databases of known alphabets grow, manual inspection becomes increasingly impractical. An automated pipeline that can be extended to larger and more heterogeneous reference databases would significantly reduce the expert effort required at this preliminary identification stage. Furthermore, integrating this step into a fully automated cipher analysis pipeline removes a manual bottleneck that would otherwise prevent end-to-end processing of large document collections.

2 Methodology

The proposed method consists of creating a reference database of alphabets, segmenting characters from the input historical cipher manuscripts, and finding their correspondence with the known alphabets.

2.1 Data Sources and Alphabet Creation

To evaluate the pipeline, we distinguish between the Experimental Cipher Dataset, i.e., the handwritten documents to be identified and the Reference Alphabets Database, i.e., the ground truth used for comparison.

The Experimental Cipher Dataset consists of images of handwritten documents sourced from the Decode database[1] (Megyesi et al., 2019; Héder and Megyesi, 2022). Six cipher collections were used (given their names followed by the record ID and image number in the Decode database): ASV (Record ID 19, Images 112 and 114), Copiale (Record ID 2, Images 42 and 51, corresponding to pages 076 and 095), Ramanacoil (Record ID 1564, Images 6975 and 6985), Vatican (Record ID 5, Images 66 and 67), Zodiac (Images 340, 408-1, and 408-2), and the Runicus collection taken from the Codex Runicus[2] (pages 30 and 97).

To create the Reference Alphabets Database, a dedicated folder was established for each of the six alphabets: Arabic (147 items), Copiale (90), Cuneiform (252), Greek (109), Latin (62), and Runic (81). Within each folder, individual image files represent every character in that alphabet.

While the Copiale alphabet database was sourced from prior work by (Knight et al., 2012), the remaining alphabets were generated programmatically. Appropriate fonts were sourced from Google Fonts, and a Python script iterated through the defined Unicode blocks for each alphabet, rendering them into individual character images. Finally, non-letter punctuation symbols were manually identified and removed from each set. There is a significant difference between Copiale and the other alphabets. While the symbols in most databases were generated using a font, the ones in the Copiale database were produced from handwritten samples. Given that the encrypted documents used for comparison are also handwritten, this difference may affect the final results. The differences are illustrated in Figure 1 for Arabic and in Figure 2 for Copiale.

[1]https://descrypt.org

[2]https://www.e-pages.dk/ku/579/

Figure 1: Example of the created Arabic alphabet.

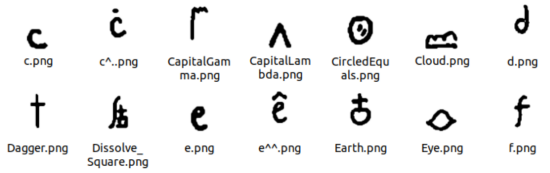


Figure 2: Example of the Copiale alphabet.

2.1.1 Preprocessing and Segmentation

The handwritten documents were preprocessed by applying adaptive binarization, with the specific method selected per collection based on qualitative evaluation: Adaptive Gaussian thresholding (OpenCV) was applied to ASV, Ramanacoil, and Vatican; Sauvola thresholding (window size 125) to Copiale and Zodiac; and Niblack thresholding (window size 125, k=0.8) to Runicus. To ensure that the texts align horizontally and vertically, deskewing was applied when necessary. Line segmentation was performed by horizontally grouping connected components, while discarding very small elements, inspired by the approach described in (Jindal and Ghosh, 2023).

2.1.2 Character segmentation

Three character segmentation methods were implemented and applied to all document images: Connected Component Analysis (CCA) (He et al., 2017) and Vertical Projection Profile (VPP) (Postl, 1986), and the combination of both. These are described below.

a) **Connected Component Analysis with diacritic merging:** The CCA algorithm processes the binary image to identify connected pixel regions, computing a bounding box and pixel area for each component. Following extraction, a merging and filtering step is applied, linking diacritics to their nearest vertically aligned base character.

b) **Vertical Projection Profile:** The VPP algorithm analyzes a line image by computing a one-dimensional histogram of pixel intensity

across image columns. This vertical projection is smoothed using a Gaussian filter to reduce noise. Local minima in the smoothed curve are interpreted as candidate character boundaries, and a dynamic threshold is applied to suppress insignificant minima. The remaining cut points define vertical bounding boxes for each symbol. Since segmentation operates on full vertical slices, diacritics aligned with their base characters are inherently included, eliminating the need for post-processing merges.

c) **CCA+VPP:** This hybrid approach combines connected component analysis with VPP-based segmentation. The method first applies standard CCA to detect connected components and merges diacritics with their nearest vertically aligned base symbols. Components whose width exceeds a threshold computed as the median width of all CCA-detected components in the line (suggesting touching or overlapping characters) are subsequently re-segmented using the VPP module. In this step, the vertical projection profile of the region is analyzed to identify valleys of whitespace that correspond to natural split points between glyphs.

Qualitative analysis identified Connected Component Analysis (CCA) as the most effective segmentation method. Although CCA can struggle with connected characters, VPP and CCA+VPP tend to over-segment many characters.

2.1.3 Clustering

Each segmented character image is converted to grayscale and resized with white padding to preserve the aspect-ratio to the 224×224 dimensions expected by the pre-trained deep convolutional neural network, ResNet-50 (He et al., 2016), used as a feature extractor. For this purpose, the original classification output layer is removed and replaced by an identity function. The network outputs a 2048-dimensional feature vector from the final global average pooling layer. Although ResNet-50 was originally trained on natural images, its pre-trained weights are sufficient for feature extraction without additional fine-tuning.

Two clustering algorithms are then applied:

a) **K-means Clustering:** It partitions the feature space into a predefined number of clusters K, which can be adjusted depending on the cipher. It initializes K centroids at random and iteratively alternates between assigning each data point to its nearest centroid and updating centroid positions until convergence. This process groups visually similar symbols, yielding candidate character classes. In all experiments, K was set to 200, providing sufficient granularity to capture the symbol variety present across the tested cipher collections.

b) **Hierarchical Clustering:** It employs a bottom-up strategy to construct clusters, using Ward's linkage method to minimize the increase in within-cluster variance at each merge. The algorithm iteratively merges the pair of clusters that results in the smallest variance increase until all samples are merged, producing a dendrogram that captures cluster relationships at multiple similarity levels. Unlike K-means, the number of clusters is not specified in advance; instead, the dendrogram is cut at a chosen distance threshold (height) to obtain the final clustering. In our implementation, a range of 16 distance thresholds evenly spaced between 0.5 and 2.0 in the ResNet-50 feature space is evaluated. Each configuration is assessed using the silhouette score, which measures intra-cluster cohesion and inter-cluster separation, and the threshold yielding the highest silhouette score is selected as the optimal cut point. This optimization selects the threshold that yields well-separated and compact clusters, balancing overly coarse groupings that merge distinct symbols against overly fine partitions that fragment coherent character classes.

Next, a quality control step evaluates the resulting clusters based on two criteria: cluster size and internal consistency. Clusters with few elements (typically five) are discarded since they are unreliable or they contain rare symbols. Clusters with high intracluster variance (low compactness) are also discarded for being too heterogeneous (different symbol types or poorly defined groupings).

2.1.4 Identification

This step determines which known alphabet in the database most closely corresponds to the symbols in the cipher manuscript. We compare feature vectors of cipher symbols with the feature vectors

of characters from the known alphabets. Using cosine similarity, each cipher symbol is matched against all characters across the known alphabets.

The best-matching alphabet for each symbol is first determined, and a histogram is constructed to count the number of symbols in each cipher cluster assigned to each alphabet.

To identify the most probable alphabet, a weighted scoring scheme is applied. Cluster sizes are normalized using a square-root transformation, to mitigate the disproportionate influence of very large clusters while preserving the contribution (with moderate weight) to smaller ones. For each alphabet, weighted scores are computed based on these adjusted cluster contributions and subsequently normalized into percentages. The alphabet with the highest normalized weighted score is selected as the most likely match. In addition to a single best prediction, this methodology produces a detailed breakdown of cluster-level contributions to the final decision.

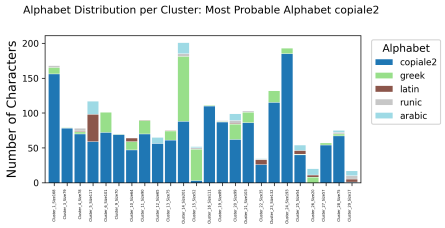
3 Results

Results from the K-means clustering

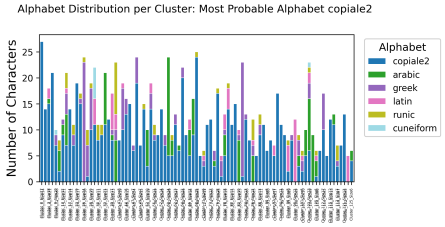
The results using K-means clustering are not satisfactory. In particular, the system fails to associate handwritten digits in the ciphers with the printed digits in the Latin reference dataset. In the case of the ASV cipher, the Latin score reaches only 2.06%. Instead, the system incorrectly favors Copiale (77.59%), apparently prioritizing handwriting style over symbol content. For the Vatican digit cipher, the Latin score is similarly low at 2.40%. This behavior suggests that the clustering stage may be problematic, as symbols with visual similarities to Greek characters appear to have been grouped together, further reducing the contribution of the Latin alphabet.

Results from the hierarchical clustering

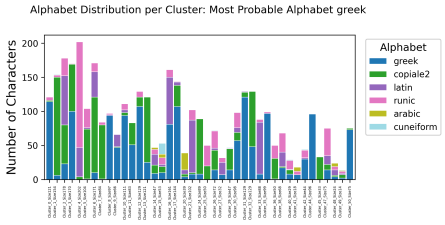
The histograms obtained for each manuscript are shown in Figure 3. These results demonstrate a consistent bias toward the Copiale alphabet across nearly all tested manuscripts. We attribute this to the handwritten nature of the cipher documents, contrasting with the font-generated reference alphabets; Copiale is the only handwritten reference alphabet in our database. For Runicus, the system assigned the (printed) Runic alphabet only 4.02%, favoring the handwritten texture of Copiale. Similarly, the Vatican and ASV ciphers (which use



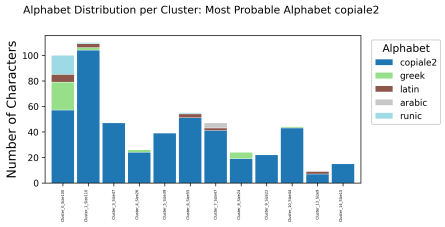
(a) ASV-France



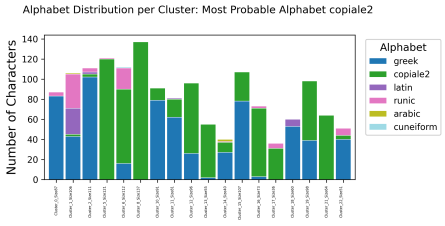
(b) Copiale



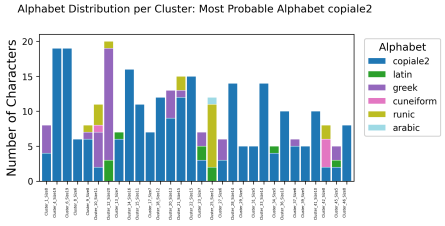
(c) Ramanacoil



(d) Runicus



(e) Vatican



(f) Zodiac

Figure 3: Alphabet Identification Results using the Hierarchical Clustering. Each bar in the horizontal axis corresponds to one cluster.

Latin digits) scored only 2.40% and 2.72% for Latin, respectively. This suggests that the extracted features encode handwriting style more strongly than character shape. The feature-space distances are smaller between different handwritten styles than between handwritten and printed versions of the same character.

Ciphers whose alphabets are absent from the Reference Alphabet Database, such as Zodiac, would ideally yield low-confidence scores or a mixture of candidate alphabets. Instead, Zodiac is misclassified as Copiale with a 75.98% score, illustrating the dominance of handwriting style in the matching process.

For Ramanacoil, the system identifies Greek as the closest match (40.96%). Unlike other cases, the distribution plots reveal several clusters dominated by Greek symbols. In particular, clusters 12, 16, and 31 show a clear preference for Greek over Copiale. This indicates that distinctive shape characteristics were sufficient to outweigh handwriting texture.

Copiale is the only cipher with the correctly identified alphabet (confidence of 66.27%). However, the Copiale cipher contains many examples of Latin letters. Despite this, the system does not favor the Latin alphabet, instead assigning higher scores to Arabic (11.12%), likely because their more complex shapes produce a larger number of incidental visual matches than the simpler printed Latin characters.

Finally, large "super-clusters" remain problematic despite square-root normalization; for instance, single clusters in Runicus and ASV contributed 47.02 and 55.49 points, respectively, to the final scores.

4 Conclusions

We have explored the use of unsupervised image processing methods for automatically identifying the alphabet of a cipher manuscript. The experiments showed that the use of realistic handwritten reference alphabets is necessary for accurate identification. Future work will focus on this aspect.

Acknowledgments

This work has been partially supported by Riksbankens Jubileumsfond, grant M24-0028 (Echoes of History: Analysis and Decipherment of Historical Writings, DESCRYPT), the Spanish project PID2024-157778OB-I00 (SUKIDI) from the Ministerio de Ciencia e Innovación, the Departament de Cultura of the Generalitat de Catalunya, and the CERCA Program / Generalitat de Catalunya. Alicia Fornés acknowledges financial support for her general research activities from ICREA under the ICREA Academia (Departament de Recerca i Universitats de la Generalitat de Catalunya).

References

Rumaan Bashir, SMK Quadri, and Kaiser J Giri. 2022. Script identification: a review. *International Journal of Information Technology*, 14(1):459–473.

Jialuo Chen, Mohamed Ali Souibgui, Alicia Fornés, and Beáta Megyesi. 2021. Unsupervised alphabet matching in historical encrypted manuscript images. In *Proceedings of the 4th International Conference on Historical Cryptology, HistoCrypt 2021*, pages 34–37. Linköping University Electronic Press.

Alicia Fornés, Jialuo Chen, Pau Torras, Carles Badal, Beäta Megyesi, Michelle Waldispühl, Nils Kopal, and George Lasry. 2024. Icdar 2024 competition on handwriting recognition of historical ciphers. In *International Conference on Document Analysis and Recognition*, pages 332–344. Springer.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Lifeng He, Xiwei Ren, Qihang Gao, Xiao Zhao, Bin Yao, and Yuyan Chao. 2017. The connected-component labeling problem: A review of state-of-the-art algorithms. *Pattern Recognition*, 70, 04.

Mihály Héder and Beáta Megyesi. 2022. The DECODE database of historical ciphers and keys: Version 2. In *Proceedings of the 5th International Conference on Historical Cryptology, HistoCrypt 2022*, pages 111–114. Linköping University Electronic Press.

Amar Jindal and Rajib Ghosh. 2023. Word and character segmentation in ancient handwritten documents in devanagari and maithili scripts using horizontal zoning. *Expert Syst. Appl.*, 225:120127.

Tauseef Khan, Md Saif, and Ayatullah Faruk Mollah. 2024. Music: A novel multi-scale deep neural framework for script identification in the wild. *IEEE Access*, 12:166955–166976.

Kevin Knight, Beáta Megyesi, and Christiane Schaefer. 2012. The copiale cipher. In *ACL Workshop on Building and Using Comparable Corpora (BUCC)*.

Beáta Megyesi, Nils Blomqvist, and Eva Pettersson. 2019. The DECODE database collection of historical ciphers and keys. In *Proceedings of the 2nd*

International Conference on Historical Cryptology, HistoCrypt 2019, pages 69–78. Linköping University Electronic Press.

Wolfgang Postl. 1986. Detection of linear oblique structures and skew scan in digitized documents. In *Proc. Int. Conf. on Pattern Recognition*, pages 687–689.

Mohamed Ali Souibgui, Pau Torras, Jialuo Chen, and Alicia Fornés. 2023. An evaluation of handwritten text recognition methods for historical ciphered manuscripts. In *Proceedings of the 7th International Workshop on Historical Document Imaging and Processing*, pages 7–12.

The BACK IN TIME Project

A User-Centred Platform for Encrypted Historical Documents

Cécile Pierrot
Inria, Nancy, France*
cecile.pierrot@inria.fr

Camille Desenclos
Université de Picardie
Inria, Nancy, France*
camille.desenclos@u-picardie.fr

Michaël Mera
Inria, Nancy, France*
michael.mera@inria.fr

Benjamin Kiessling
Inria, Paris, France
benjamin.kiessling@inria.fr

Gaspard Damoiseau-Malraux
Sorbonne Université, CNRS, LIP6
& Inria, Nancy, France*
gaspard.dama@lip6.fr

Hassen Aguili
Inria, Paris, France
hassen.aguili@inria.fr

Thibault Clérice
Inria, Paris, France
thibault.clerice@inria.fr

* is the short for:
Université de Lorraine
CNRS, Inria, LORIA
F- 54000 Nancy, France

Abstract

Since the Late Middle Ages, sensitive correspondence (political, diplomatic, military, etc.) has frequently been encrypted. Thousands of these documents now lie in archives and libraries, yet they remain largely beyond the reach of humanities researchers. The sheer volume and complexity of these materials render both manual cryptanalysis and manual analysis impractical, if not impossible. Although recent advances in historical cryptanalysis and artificial intelligence have enabled partial automation, existing tools remain ill-suited to the needs and practices of humanities researchers. The BACK IN TIME project was established in late 2024 to address these challenges. It seeks an alternative approach to automation that places the end user – namely, the humanities researcher – at the centre of the system's design. In this paper, we articulate our vision for a user-centred assistive platform intended to bridge the current gap: we highlight how the guiding principles identified by the project team can be used to combine computer vision, cryptanalysis, and historical expertise into a user-centred design that makes historical cryptanalysis both accessible and compatible with established scholarly practices. Initial implementations will focus on Western European scripts and languages, as well as on manually encrypted documents (up to the early 20th century), reflecting the team's expertise and easier access to these documents.

1 Introduction

While working with historical documents, especially historical correspondence, humanities researchers may encounter encrypted materials. These documents are unfortunately scattered and poorly identified within archives and libraries, and no systematic study of their volume and nature has yet been conducted. Existing works and projects suggest there may be thousands of encrypted documents, and correspondence of all types may have been encrypted, from diplomacy to intimacy (Láng, 2018; Héder and Megyesi, 2022; Tomokiyo, 2018). However, the presence of ciphertext does not necessarily prevent access to a document's content. Many encrypted documents have been preserved alongside the recipient's decryption, such as plaintext annotations in the margins, between lines, or on separate sheets.[1] In

[1] No systematic study has been conducted on decryption. According to our research, although it is unlikely that more than 1-2% of the encrypted documents never had a decryption (primarily intercepted letters with unsuccessful cryptanalysis), a significant portion (10-15%) has not been preserved

cases where plaintext is missing, the content may still be recovered (Desenclos and Lasry, 2025): it is possible to use the cipher key, provided the relevant key has both survived and been correctly identified. However, matching ciphertexts to their cipher keys requires significant time, extensive knowledge of archival repositories, and often depends on chance, as well-identified and preserved cipher keys are uncommon.[2] Even when keys are available or recovered, the subsequent manual decryption remains laborious and time-consuming,[3] particularly when handling larger or multiple corpora.

When part of the corpus contains both ciphertext and the corresponding plaintext, an alternative for humanities researchers is to reconstruct the cipher key themselves and then decrypt the remaining ciphertext. Although applied in several works (Stix, 1934 1936; Greenough, 1997; Monts de Savasse et al., 2004; Domnina, 2018; Pierrot et al., 2025; Benavent, 2025), this approach is also highly time-consuming, may require prior knowledge of cipher key structures, and poses a real problem when the corpus is large and/or when there are only a few short ciphertexts/plaintexts. Besides, it becomes outright impossible in the presence of a ciphertext alone (without a cipher key or plaintext) or of a ciphertext with some short plaintext but a complex or unknown encryption system.

Some tools (e.g., CTTS, CrypTool) can already assist manual cryptanalysis depending on the length and complexity of the ciphertext, while some works have investigated how to match cipher keys with ciphertexts (Pettersson and Megyesi, 2019) or ciphertexts with their plaintext (Bruton and Megyesi, 2025). In each case, the process remains a complex combination of automated techniques that are guided and supplemented by human intervention, and require expert knowledge. A first attempt to create a global pipeline was carried out by the DECRYPT project (Héder et al., 2024); however, this effort could not be completed. Despite their role as primary providers of encrypted documents and architects of historical research questions, humanities researchers still face a fragmented ecosystem of tools and methods that, save for a few highly specialised experts with extensive cross-disciplinary knowledge, they cannot use due to a lack of technical computer skills and unfamiliarity with historical ciphers.

To enable humanities researchers to access the content of encrypted documents and better understand encryption practices and systems, an alternative approach is required. Since its inception in late 2024, the BACK IN TIME project has investigated the benefits of designing an automated yet user-centred cryptanalysis process. Alongside the development of tools for automated decryption, the core of the BACK IN TIME project is the realisation of a user-centred pipeline that enables the end-user – the humanities scholar – to process encrypted documents autonomously. In this paper, we reassert the importance of this goal and outline our findings about the core principles that will ultimately enable the development of such a pipeline.

2 Goals

The BACK IN TIME project aims to address both the significant technical challenges presented by automated historical document decryption (technical tools) and the prerequisites for autonomous usability by humanities researchers (user-centred pipeline) simultaneously. The latter aspect is the main methodological basis of the project, driven by the fact that such collaboration is a fundamental precondition for the very creation of valid, usable and used research tools. Humanities researchers are indeed uniquely positioned to further the development of automated decryption by providing relevant ciphertexts, high-quality, accurate and structured historical metadata (such as date, language, origin, and historical context), and valuable feedback both on the transcription (palaeography) and on the reconstructed plaintext (historical and linguistic accuracy). To encourage this complementary expertise, the project values developing a platform that serves as a collaborative hub for end-users and the diverse communities involved in the automation process.

2.1 The Platform's Design Principles

Along with studying existing software and developing new tools for automated decryption, we identified the following core principles for a plat-

alongside or close to its decryption (plaintext) and requires cryptanalysis.

[2] Many cipher keys are actually matched to their ciphertexts thanks to cryptanalysis. (Kopal and Waldispühl, 2021; Desenclos and Lasry, 2024)

[3] According to the experiments we conducted, a researcher unfamiliar with historical ciphers will need, on average, 1 hour and 45 minutes to decrypt one fully encrypted page (approx. 30 lines)

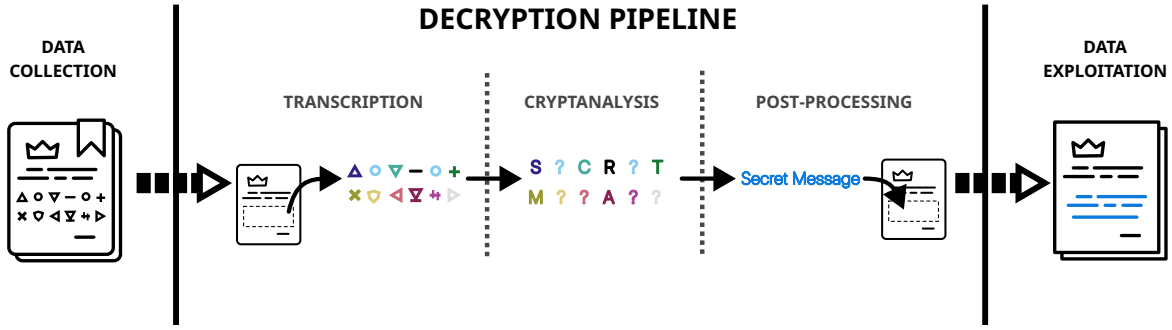


Figure 1: The standard pipeline for the processing of encrypted documents.

form centred on end users.

End-to-End: A significant barrier to the widespread adoption of computational methods for processing encrypted documents is the fragmentation of available tools, each designed to address only isolated stages of the processing pipeline. Furthermore, some specific components, such as ciphertext character recognition, remain important research challenges and are not yet fully automated tasks (Souibgui et al., 2023; Fornés et al., 2024). While this disjointed landscape may be navigable for experts already well-acquainted with historical cryptography and its necessary processing steps, it presents an almost insurmountable obstacle for novice and occasional users.

To address this fragmentation, we suggest integrating our tools and providing a single user-facing interface that spans the entire pipeline. This end-to-end design ensures that users can concentrate on their primary domain of expertise, specifically the collection of historical documents and the interpretation of reconstructed plaintext, which are situated at the input and output sides of the pipeline (see Figure 1). The complexity of the intermediate steps should be abstracted away to create a unified workflow designed specifically for humanities researchers (without a technical background).

User-centred: While a single interface already enhances consistency, it must also provide a clear and intuitive workflow for a wide target audience without prior knowledge in historical cryptography. In practice, this workflow must transpose the specialised terminology and abstract away algorithmic details. An informal survey conducted among humanities researchers (history, literature) has shown that successful use of such a platform and its integration within the humanities ecosystem requires tools that are grounded in both vocabulary and methodology in existing scholarly norms, tools that provide accessible and introspectable outputs at every stage of the transcription and cryptanalysis process. Within the BACK IN TIME platform, cryptanalysis algorithms and computer vision AI models, for example, will be handled internally and not exposed to the end-user. Of course, this is not to say that the user will be passive during the process, but rather that the feedback will be requested in terms that match the user's background, e.g. *Are the following characters correctly identified?* or *Is the following text correct in the language of the message?* among others.

User-in-the-loop: Designing a fully automated process for the decryption of historical documents poses significant challenges for both computer vision and cryptanalysis. At the time of writing, no such comprehensive system exists. To address these challenges, the user will need to be fully involved, assisting and guiding the process. Rather than waiting until the end of each pipeline step to query the user's (dis)approval, it is critical to design a workflow that actively and regularly collects feedback throughout the entire processing pipeline. It allows for a more fine-grained and dynamic parameterisation of the pipeline, as highlighted in subsection 3.3. However, while designing such a workflow requires abstracting away the technical intricacies of computer vision and cryptanalysis, it is equally important to let users understand the role of each functional block. This enables them to better identify error sources and apply their expertise (linguistic, historical) to redirect the process in the right direction when needed.

Modular: Given the extreme heterogeneity of encrypted historical documents, the assumption that a single algorithm or AI model could universally address such diversity is unrealistic. As such, the workflow is built upon the foundational principle of modularity. Rejecting a monolithic architecture with fixed implementations and general models, the platform is designed to integrate a diverse array of interchangeable modules. Each module targets a specific situation, such as a particular encryption system, secondary security devices (nulls, nomenclature, etc.), ciphertext character sets, plaintext script and language, allowing the workflow to be dynamically configured to suit the specific requirements of each document or corpus.

Resilient: Despite its recent growth, there are still gaps in the current research that will inevitably lead to failures on the platform. Due to a lack of metadata or knowledge, some types or parts of encryption systems may not be correctly identified by the automated process or, due to the rarity or complexity of some encryption systems, no generic cryptanalysis algorithm would work on them. Some encrypted documents present interlinear plaintext that the computer vision algorithms might not correctly recognise during segmentation, or ciphertext characters that the algorithms might not correctly identify. As such, there will be documents that cannot be successfully handled by any of the modules provided on the BACK IN TIME platform. Rather than trying to solve everything within our funded time, we only address the most recurring challenges (ciphertext character sets and encryption systems) and design a platform to gracefully handle failure cases. In case of failure, the type of problem encountered should be automatically reported to the project team, and the particular use case will be analysed with the user to identify its specificities. In the longer term, a dedicated expert module that solves the problem could be developed to increase the platform's coverage.

Open Source, Free, Transparent: We strongly believe that only an open-source and freely accessible ecosystem can foster durable, sustained progress in research. Open science is not merely a matter of transparency, but a prerequisite for reproducibility, collective validation, and long-term impact, especially in interdisciplinary fields where methods and data must be scrutinised, adapted, and extended by diverse communities. By releasing the source code developed throughout this project, we aim to facilitate reuse, critical assessment, and incremental improvement across all kinds of expertise (history, computer vision, cryptanalysis, etc.). Beyond providing open access to the code, we plan to host a free instance of the platform in order to lower the barrier to entry for users who lack the technical infrastructure or expertise to deploy such tools themselves.

Responsible Data Management: Beyond mere compliance with the European General Data Protection Regulation, we need to establish a rigorous data management policy designed to ensure users maintain absolute stewardship over their materials. The platform should operate on a strict non-retention basis by default: all uploaded documents remain the exclusive property of the user and are neither stored, redistributed, nor utilised by the project without authorisation.

Conversely, researchers may choose to contribute their research artefacts (transcriptions, cryptanalysis, recovered cipher keys) to the community, but only through a conscious, explicit opt-in mechanism. This commitment to user autonomy is critical for building trust, particularly given that archival materials may be subject to restrictive reproduction and dissemination rights. Through these principles, the project ensures that the user remains the sole arbiter of how their data is shared and valued.

2.2 Community-Driven Development and Open Challenges:

User-centred or not, the pipeline needs to rely on technical expertise from diverse technical fields. But, if the BACK IN TIME project involves the development of specialised software tools, these tools actually aim to simplify the creation of expert modules, allowing the specific problem-solving strategies of computer scientists and cryptographers to be integrated directly into the historian's workspace. By lowering the barrier to technical contributions, we aim to transform the platform into a nexus where these distinct forms of expertise can coalesce into a shared methodology. We posit that the complexity of historical, especially automated, cryptanalysis cannot be resolved solely through the research methodology of a single domain. Instead, it requires a synergistic ap-

proach that converges the distinct inputs of cryptanalysts, computer vision experts, and historians into a unified operational framework.

To encourage and support this collective dynamic, we drew inspiration from existing initiatives that use challenges to leverage external expertise to advance some research questions and challenges in the specific field of historical cryptography.[4] These challenges have proved their value. In the BACK IN TIME project, we intend to propose different kinds of challenges, such as code-breaking tasks on synthetic data, to calibrate cryptanalysis algorithms more precisely. However, our main focus will be on encouraging humanities researchers to submit their own challenges (e.g. an encrypted document that they cannot decrypt) to the platform. We hope to achieve several goals in this way: bringing humanities researchers to the platform (as active contributors and not as passive users) even before the release of the full automation pipeline, enabling the continuous refinement of transcription and cryptanalysis algorithms (multiple examples and valuable feedback), and addressing specific cryptographic challenges (unknown or rare encryption systems, rare ciphertext characters, etc.). Moreover, these challenges will enable researchers to identify the most effective solutions for specific tasks within our automation decryption pipeline.

By fostering collaboration around well-defined problems, the challenges will help verify that the pipeline remains aligned with the state of the art and users' needs while benefiting from the collective intelligence of its diverse members. Moreover, by prioritising the construction of open, community-driven infrastructures, we ensure that methodological advances are not lost when the project ends, but remain accessible and verifiable. This shared ecosystem is essential to the field's long-term sustainability.

2.3 Studying the Gap in Current Research

As outlined in subsection 2.1, the platform must be designed with resilience and modularity in mind, to avoid enforcing a *one-size-fits-all* model and instead to make processing errors explicitly visible and identifiable. In such a framework, a failure is indeed an informative signal, rather than a technical glitch. In its current state, history of cryptology can only provide an incomplete taxonomy of encryption practices and systems. Therefore, failures of proven algorithms are more generally hints of limitations of the taxonomy itself, rather than of the effectiveness of said algorithms.

Failures indicate that the material deviates from expected norms, whether in its visual structure (rare or badly formed ciphertext characters), cryptographic logic, or language (encryption errors, dialect, etc.). Hence, this signal serves as a heuristic for discovery, highlighting the necessity for a dual research effort: the development of more robust computational methods capable of handling outliers, and deeper historical research to better describe and categorise the vast diversity of extant sources.

This diagnostic approach is particularly critical given that current computational methods are rarely systematically evaluated against the full diversity of historical material. This lack of comprehensive benchmarking is understandable, stemming from the scarcity of representative datasets and the genuine heterogeneity of real-world encryption practices. By systematically capturing these failure cases, the platform enables a finer-grained analysis of algorithmic boundaries. It allows researchers to relate performance not just to abstract metrics but to concrete document properties, such as ciphertext character sets or layout characteristics, thereby transforming implicit algorithmic limitations into explicit and actionable research questions.

3 User-centred Pipeline

The platform we envision aims to provide a strict separation of concerns between the interactions exposed to the end-user and the decisions involving domain-specific automation tools and algorithms. To guarantee this separation, we structure the platform using a layered architecture, as shown in Figure 2.

3.1 User Interface

At the very top of the layered architecture, the user interface displays a unified workflow decoupled from the underlying algorithms and expert modules involved in the automated decryption process. This will let the user focus on the various

[4] Many websites propose cryptanalysis challenges (e.g., `https://mysterytwister.org/`) or unsolved cryptograms (e.g., `https://crypto.hcportal.eu/dashboard/cryptograms`). More recently, challenges dedicated to Handwriting Recognition of Historical Ciphers have also arisen (Fornés et al., 2024).

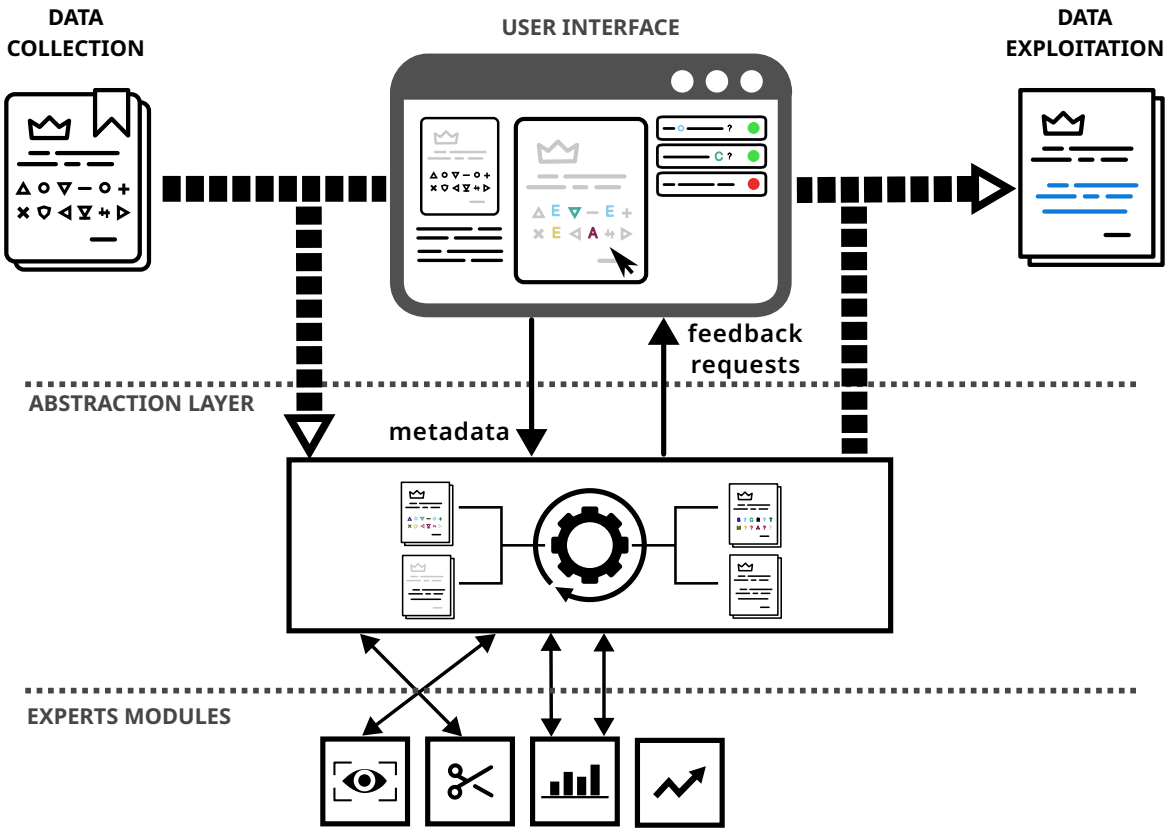


Figure 2: The layered architecture for the user-centred platform.

outputs rather than the mechanisms that drive the transcription and cryptanalysis. For instance, in terms of the final result, it should not matter to the user which type of AI model, which cryptanalysis algorithm or combination of algorithms has been used to perform the decryption. On the contrary, it should matter to the user to get classification both of the ciphertext characters and of the encryption system, as well as to know whether the plaintext has been reconstructed by pure cryptanalysis, by matching an already stored cipher key or by matching existing plaintext.

More specifically, the user interface integrates the entire pipeline and presents it to the user as an iterative workflow: Starting from a digitised version of the encrypted document (e.g. personal photo, online digitised documents), the document's text is progressively extracted (segmentation then transcription), cryptanalysed and refined into a final usable plaintext alongside the reconstructed cipher key. This iterative workflow is sustained by the user, who responds to feedback requests from the underlying layers, allowing the platform to gradually refine the result while collecting additional metadata. At each stage and iteration, the user will also have the option to obtain specific, workable outputs to independently analyse the ciphertext characters (classification, frequency statistics, visualisation of usage) as well as the encryption system (reconstructed cipher key and parameters).

3.2 Expert Modules

The collection of expert modules resides at the bottom of the layered architecture, serving as the platform's foundational engine. The primary role of an expert module is to encapsulate specific technical logic required to solve a standalone task, such as performing cryptanalysis on a specific cipher type or segmenting lines of text from a noisy image. These modules are typically implemented as wrappers around existing methods developed by domain experts, such as cryptologists or computer vision specialists, integrating diverse external software into a unified ecosystem.

To ensure seamless integration, each module must adhere to a standardised interface. However, because no single algorithm can handle the vast diversity of historical documents and encryption methods, strict scope declarations are mandatory.

These declarations act as machine-readable contracts that specify the exact prerequisites for the module's operation. For instance, a cryptanalysis module must explicitly state which encryption systems it supports (e.g., homophonic ciphers) and the valid ranges for key parameters. Similarly, an image processing module designed for handwriting recognition might restrict its scope to specific script types, such as Roman numerals or abstract symbols, while rejecting printed text.

This modularisation presents a significant opportunity for standardisation. By forcing domain experts to formalise the capabilities and limitations of each tool via scope declarations, the platform creates a definitive list of criteria for algorithm evaluation. Furthermore, as these modules are deployed against real-world data, their scope definitions can be iteratively refined, narrowing the gap between theoretical capability and practical application.

3.3 Abstraction Layer

The abstraction layer functions as intelligent middleware connecting the high-level user interface (subsection 3.1) with the low-level expert modules (subsection 3.2). Its primary responsibility is orchestration: it must interpret the current state of the document processing, identify blocking issues (such as unknown or uncommon encryption systems, poorly recognised or overlooked ciphertext characters, unintelligible reconstructed plaintext), and dynamically select the most appropriate expert modules to resolve them.

This selection process relies heavily on the scope declarations provided by the underlying modules. The abstraction layer matches the metadata and features of the encrypted document against the available modules to determine eligibility. For a simple encrypted document (easily identifiable encryption system, one or a few security devices), this may involve only a linear sequence of calls: segmentation, followed by recognition, followed by cryptanalysis, etc. However, for a complex encrypted document (with a partially or non-identified encryption system or multiple security devices), the abstraction layer employs sophisticated resolution strategies. It may act as a competitive orchestrator, invoking multiple cryptanalysis modules in parallel to compare their outputs, or it may apply a fallback strategy, trying a computationally expensive algorithm only after a faster heuristic has failed.

Crucially, the abstraction layer is also responsible for data synthesis. It must harmonise the raw, often technical outputs from various modules into a coherent structure that the User Interface can render. By presenting synthesised and readable results to the user for validation, the abstraction layer creates a feedback loop: if the user rejects a result, the layer can interpret this as a constraint update, triggering a new search for alternative modules or requesting specific input to unlock advanced processing paths.

4 Related Work

The BACK IN TIME project is neither the first nor the only one dedicated to the processing of encrypted historical documents. Key initiatives paved the way for identifying major methodological challenges and research questions and facilitated the development of a few pioneer tools. While the insights and outcomes of these efforts have deeply informed our conceptual framework, BACK IN TIME does not intend to duplicate or repackage existing tools. Instead, to ensure the end-user remains the focal point of the global pipeline, the project explores an alternative design strategy. This entails the development of novel tools from the ground up, while strategically incorporating and adapting established algorithms, e.g., eScriptorium's layout analysis functionality (Kiessling, 2020), and capitalising on recent technical advancements in machine learning, newly combined cryptanalysis algorithms, as well as the latest work on the history of cryptology to overcome the limitations identified with the current research landscape.

The following overview examines initiatives such as the DECRYPT (Megyesi et al., 2020) and DESCRYPT (Megyesi et al., 2025) projects as well as existing tools and environments, including CrypTool2[5] for cryptographic experimentation, Cryptool transcriber and Solver (CTTS)[6] for manual transcription and cryptanalysis, and eScriptorium[7] (Kiessling et al., 2019) and Transkribus[8] for large-scale transcription. By analysing these contributions, we identify their respective strengths and limitations, thereby clarifying how our ap-

[5] https://www.cryptool.org/en/ct2/
[6] https://www.cryptool.org/en/ctts/
[7] https://escriptorium.inria.fr/
[8] https://www.transkribus.org/fr

proach builds upon and diverges from the existing landscape.

Historical Cryptanalysis Projects. Since the mid-2010s, a few initiatives have emerged to advance the computational analysis of encrypted historical documents, most notably the DECRYPT project (Megyesi et al., 2020). These efforts have significantly advanced the study of history of cryptology and structured the field of historical cryptography by developing and integrating a wide array of tools and resources. As does BACK IN TIME, the DECRYPT project addresses the primary stages of processing encrypted documents, with research activities ranging from data collection (via the DECODE database) to cryptanalysis. The final phase of DECRYPT (Héder et al., 2024), as well as the workflow of the new DESCRYPT program[9], envisions linking their various tools and resources into a unified pipeline. However, this integrated workflow has not yet been fully realised (Héder et al., 2024). Furthermore, the proposed architecture assumes a more linear pipeline, applying modules one after another, which contrasts with the iterative and exploratory process often required by humanities researchers. The BACK IN TIME team pursues a very different strategy. By prioritising a unified platform from the outset, we aim to demonstrate that placing a user-centred, iterative workflow at the centre of the architecture yields genuine benefits for developing new tools and integrating them into research practices.

Further, even though DECRYPT has undeniably demonstrated the feasibility of applying computational methods to historical ciphers, particularly by developing algorithmic approaches for cryptanalysis and establishing experimental frameworks for transcription, its software ecosystem remains primarily designed for expert use. Key tools provide robust environments for cryptanalysis but are distributed as standalone executables requiring specific technical environments. For instance, CrypTool Transcriber & Solver (CTTS) is a Java-based desktop application that requires command-line interaction and manual configuration, with documentation hosted on GitLab, a platform unfamiliar to most humanities scholars. Similarly, tools for semi-automatic transcription, such as Transcripttool,[10] impose comparable technical barriers. This overhead, combined with interfaces that presuppose prior knowledge of cryptography, e.g., requiring the user to select specific cipher parameters before analysis, renders these tools, on their own, unsuitable for humanities researchers whose primary objective is recovering document content rather than acquiring expertise in historical cryptography. However, to ensure compliance with all the platform's design principles (subsection 2.1) and to integrate the latest technical and methodological advances, new tools (transcription and cryptanalysis) will be designed for the BACK IN TIME pipeline.

Community Resources and Datasets. Part of the DECRYPT ecosystem, the DECODE database (Héder and Megyesi, 2022) is a unique example of a collaborative and searchable database of encrypted documents and cipher keys. By making such materials available, it has enabled a more systematic analysis of encryption practices and systems across time (Megyesi et al., 2021). Such work will not be integrated within BACK IN TIME: the aims are different – the humanities researchers are directly providing encrypted documents they need to decrypt and/or analyse – and DECODE is already widely acknowledged as the main repository for encrypted documents and cipher keys. Similarly, large-scale historical corpora such as HistCORP[11] (Pettersson and Megyesi, 2018) play a crucial role by offering curated resources that support comparative and contextual analyses in historical research. Although neither DECODE nor HistCORP will be integrated into the BACK IN TIME architecture, they provide a foundational data layer that complements the data our team already collected, resulting in a diverse corpus for training and evaluating our transcription and cryptanalysis tools.

Transcription Platforms for Historical (non-encrypted) Documents. Platforms such as eScriptorium (Kiessling et al., 2019) and Transkribus have reshaped the field of digital humanities, unlocking large-scale transcription of handwritten manuscripts. Driven by advances in deep learning, they have significantly lowered the barrier to producing accurate transcriptions from digitised

[9] https://descrypt.org/ The DESCRYPT project reuses and extends the tools and expertise of DECRYPT to cover rare scripts and low-resource texts.

[10] https://github.com/decrypt-project/transcripttool

[11] https://www2.lingfil.uu.se/person/pettersson/histcorp/

handwritten texts.

However, the specific nature of encrypted documents introduces fundamental challenges beyond the scope of standard transcription platforms. First, modern handwritten text recognition architectures rely heavily on linguistic modelling to compensate for visual ambiguity[12] an approach that fails when applied to ciphertexts designed specifically to maximise entropy and eliminate predictable lexical patterns. Second, ciphertext characters can differ significantly from cleartext scripts in their visual composition, necessitating the creation of substantial new ground-truth datasets to train models on non-standard character sets. Finally, whereas standard models depend on fixed character inventories and closed classification layers, symbolic ciphertext characters often operate as open or semi-open sets, which some have approached by returning to basics and character classification (Siglidis et al., 2024).

Consequently, while we cannot use transcription platforms out of the box, BACK IN TIME adopts a hybrid strategy. It integrates the eScriptorium infrastructure specifically for its state-of-the-art layout analysis and line segmentation capabilities (the first stage of the transcription pipeline). However, it replaces the subsequent text recognition block with its own specialised modules designed to handle the visual complexity and linguistic entropy of encrypted script.

5 Conclusion

The BACK IN TIME project certainly addresses already-identified issues, such as the transcription of encrypted documents, their cryptanalysis, and the development of an automated decryption pipeline. Some initiatives already dedicate resources to some of these issues, but offer a fragmented, hardly usable ecosystem for humanities researchers. However, beyond designing new tools, it plans to explore a different, user-centred approach by revising the classical pipeline and rebuilding each stage to fit the needs of the end user: humanities scholars. We hope this will reinforce automation efforts while being progressively integrated into the toolbox of humanities researchers who can benefit from and could more often contribute to historical cryptography. While this project does not intend to solve every kind of cryptogram and will first focus on the most commonly used ciphertext character sets and encryption systems, it aims to make continuous improvements using a federated approach that brings together all researchers working with encrypted historical documents. The ultimate goal is to enable humanities researchers to engage directly with cryptographic sources at scale, thereby transforming encrypted documents from marginal curiosities into fully exploitable historical evidence.

[12] TrOCR (Li et al., 2023) includes a language transformer after a visual one, PyLaia (Tarride et al., 2024), while older, provides connector for small language models KenLM (Heafield, 2011).

Aknowledgement

This work received government funding managed by the French National Research Agency (ANR) under France 2030, reference number ANR-24-RRII-0002, named *Back In Time* project and operated by the Inria Quadrant Programme (PIQ).

References

Júlia Benavent. 2025. La escritura secreta en Europa en el siglo XVI. *Estudios románicos*, 34:11–189.

Micaella Bruton and Beáta Megyesi. 2025. From Statistics to Neural Networks: Enhancing Ciphertext-Plaintext Alignment in Historical Substitution Ciphers for Automatic Key Extraction. In *Proceedings of the 8th International Conference on Historical Cryptology, HistoCrypt 2025*, pages 18–26. Linköping University Electronic Press.

Camille Desenclos and George Lasry. 2024. An early French digit cipher: deciphering a letter from the King of France to the Duke of Nevers (1592). In *Proceedings of the 7th International Conference on Historical Cryptology (HistoCrypt 2024)*, pages 46–56. Tartu University Library.

Camille Desenclos and George Lasry. 2025. Cryptanalytic and historical challenges with unidentified encrypted documents from the early modern era. In *Proceedings of the 8th International Conference on Historical Cryptology, HistoCrypt 2025*, pages 41–51. Linköping University Electronic Press.

Ekaterina Domnina. 2018. Nicodemo Tranchedini's Diplomatic Cipher: New Evidence. In *Proceedings of the 1th International Conference on Historical Cryptology (HistoCrypt 2018)*, pages 3–7. Tartu University Library.

Alicia Fornés, Jialuo Chen, Pau Torras, Carles Badal, Beáta Megyesi, Michelle Waldispühl, Nils Kopal, and George Lasry. 2024. ICDAR 2024 Competition on Handwriting Recognition of Historical Ciphers. In *Document Analysis and Recognition - ICDAR 2024*, pages 332–344. Springer Nature Switzerland.

H. Paul Greenough. 1997. Cryptanalysis of the Swedish HC-9. A Known-Plaintext Approach. *Cryptologia*, 21(4):353–367.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197. Association for Computational Linguistics.

Mihály Héder and Beáta Megyesi. 2022. The DECODE Database of Historical Ciphers and Keys: Version 2. In *Proceedings of the 5th International Conference on Historical Cryptology (HistoCrypt 2022)*, pages 111–114. Tartu University Library.

Mihály Héder, Alicia Fornés, Nils Kopal, Ferenc Szigeti, and Beáta Megyesi. 2024. Supporting Historical Cryptology: The Decrypt Pipeline. In *Proceedings of the 7th International Conference on Historical Cryptology (HistoCrypt 2024)*. Tartu University Library.

Benjamin Kiessling, Robin Tissot, Peter Stokes, and Daniel Stokl Ben Ezra. 2019. eScriptorium: An Open Source Platform for Historical Document Analysis. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 2, pages 19–19. IEEE Computer Society.

Benjamin Kiessling. 2020. A modular region and text line layout analysis system. In *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 313–318. IEEE.

Nils Kopal and Michelle Waldispühl. 2021. Deciphering three diplomatic letters sent by Maximilian II in 1575. *Cryptologia*, 46(2):103–127.

Benedek Láng. 2018. *Real Life Cryptology: Ciphers and Secrets in Early Modern Hungary*. Amsterdam University Press.

Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. 2023. TrOCR: Transformer-Based Optical Character Recognition with Pretrained Models. In *Proceedings of the AAAI conference on artificial intelligence*, pages 13094–13102.

Beáta Megyesi, Bernhard Esslinger, Alicia Fornés, Nils Kopal, Benedek Láng, George Lasry, Karl De Leeuw, Eva Pettersson, Arno Wacker, and Michelle Waldispühl. 2020. Decryption of historical manuscripts: the DECRYPT project. *Cryptologia*, 44(6):545–559.

Béata Megyesi, Crina Tudor, Benedek Láng, and Anna Lehofer. 2021. Key Design in the Early Modern Era in Europe. In *Proceedings of the 4th International Conference on Historical Cryptology (HistoCrypt 2021)*, pages 121–130. Tartu University Library.

Beáta Megyesi, Alica Fornés, Mihaly Héder, Raphaela Heil, Nils Kopal, Benedek Láng, Rune Rattenborg, and Michelle Waldispühl. 2025. Decipherment of Historical Manuscripts with Unknown or Rare Writings: The DESCRYPT Project. In *Proceedings of the 8th International Conference on Historical Cryptology, HistoCrypt 2025*, pages 101–105. Linköping University Electronic Press.

Jacques de Monts de Savasse, Yves Soulingeas, and Stéphane Gal. 2004. *L'Europe d'Henri IV. La correspondance diplomatique du Secrétaire d'Etat Louis de Revol (1558-1593)*. Presses Universitaires de Grenoble.

Eva Pettersson and Beáta Megyesi. 2018. The HistCorp Collection of Historical Corpora and Resources. *Digital Humanities in the Nordic and Baltic Countries Publications (DHN2018)*, 1:306–320.

Eva Pettersson and Beáta Megyesi. 2019. Matching Keys and Encrypted Manuscripts. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 253–261. Tartu University Library.

Cécile Pierrot, Gaspard Damoiseau-Malraux, Paul Mekhail, Olivier Chaline, and Ludovic Perret. 2025. A Caribbean Directory-based Encryption during the American War of Independence. In *Proceedings of the 8th International Conference on Historical Cryptology, HistoCrypt 2025*, pages 122–132. Linköping University Electronic Press.

Ioannis Siglidis, Nicolas Gonthier, Julien Gaubil, Tom Monnier, and Mathieu Aubry. 2024. The learnable typewriter: a generative approach to text analysis. In *International Conference on Document Analysis and Recognition*, pages 297–314. Springer, ICDAR.

Mohamed Ali Souibgui, Pau Torras, Jialuo Chen, and Alicia Forné s. 2023. An Evaluation of Handwritten Text Recognition Methods for Historical Ciphered Manuscripts. In *Proceedings of the 7th International Workshop on Historical Document Imaging and Processing*, page 7–12. Association for Computing Machinery.

Franz Stix. 1934–1936. Die Geheimschriftenschlüssel der Kabinetskanzlei des Kaiser. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen Philologisch-Historische Klasse*, 1–2:207–226 and 61–70.

Solène Tarride, Yoann Schneider, Marie Generali-Lince, Mélodie Boillet, Bastien Abadie, and Christopher Kermorvant. 2024. Improving Automatic Text Recognition with Language Models in the PyLaia Open-Source Library. In *International Conference on Document Analysis and Recognition (ICDAR 2024)*, pages 387–404. Springer Nature Switzerland.

Satoshi Tomokiyo. 2018. Cryptiana: Articles on Historical Cryptography.

Statistical Tests for Randomness on a Typewritten Key Stream Extracted With Computer Vision and Classified With a Convolutional Neural Network

Floe Foxon
University of Texas at Austin / USA
University of Leeds / UK
ff5384@eid.utexas.edu

Abstract

For a key stream to be cryptographically secure, it must be sufficiently random (i.e., unpredictable). This study tested the randomness of a set of typewritten, WW2-era German diplomatic key stream tables. Character objects were extracted from images of the tables using computer vision, and a bespoke convolutional neural network (convnet) was trained to classify these objects as digits (from 0–9). The convnet had a mean cross-validated testing balanced accuracy of 93.7% (standard deviation: 0.7%). $N = 74,979$ digits were extracted and classified from the images. Randomness was tested with the arithmetic mean, chi-squared, runs, and Monte Carlo pi tests; the key stream failed all four tests with 95% confidence. One digit appeared to be over-represented, and two others under-represented in the tables. Analysis suggests that the under-represented digits may be a simple artefact of computer vision error/bias, but the over-represented digit did not appear to have resulted from computer vision and/or classification error/bias. Reference streams generated with the Mersenne Twister and Linux OS entropy passed all four tests. WW2-era German diplomatic key stream tables may have lacked randomness. The extent to which this could potentially be exploited by cryptanalysts is unknown.

1 Introduction

Random numbers are used extensively in cryptography, for example in one-time pads (OTPs). Under the usual OTP assumptions including 'true' key randomness and no key re-use, text encrypted with the OTP cannot, in theory, be decrypted by any means except by using the very same key stream that encrypted it. Modern computer systems can, in theory, provide strong (or at least, sufficient) randomness quickly and efficiently. For example, the Linux operating system generates pseudo-random numbers by collecting data in the form of noise from the computer's environment/hardware ('entropy'), such as time between keystrokes (Nakov, 2019), and transforming these data into seeds for cryptographic random number generators. Historically however, the process of generating random numbers (without computers) was more time-consuming, less efficient, and potentially less cryptographically secure. As a purely demonstrative example, one could continuously re-roll a 10-sided die labelled 0–9, but even with many dice this process would be extremely time-consuming, and the dice could be manufactured or rolled in such a way that is biased, and the resulting key stream may contain patterns that could be exploited by cryptanalysts (at least in theory).

Nevertheless, key streams *were* generated manually and used extensively in the 20th century. From a cryptographic perspective, it is interesting to test the extent to which these key streams were random. If tables of historical key streams are found to lack randomness, then this could imply a degree of insecurity in texts encrypted with the tables.

Because historical key stream tables were typewritten on paper, the first task in analysing these tables for randomness is to read the key streams into a computer. Since the tables consist of thousands of digits, digitizing them by hand would be extremely slow and prone to human error, but tools exist to automate the process, namely computer vision (to detect and segment characters on the page) and classifiers (to determine which digit the character represents). Once digitized, the second task is to test the key stream for randomness, for which many methods have been developed (Foley, 2001;

Kenny, 2005).

The aim of this study was to digitise historical key stream tables and test them for randomness, investigating bias both in the physical tables and in their digitization.

2 Methods

2.1 Data Source

The historical key stream tables used in this study come from German diplomatic tables from around the period of the Second World War, which were preserved in the United Kingdom's National Archives. As shown in Figure 1, the particular set of tables used in this study consist of thousands of digits prepared in groups of five, with 13 columns per page, dozens of rows per page, and multiple pages. The method used to generate these particular tables is unknown to the author, as is the purpose of the tables and their possible relation to the broken German OTP system described by Filby (1995).

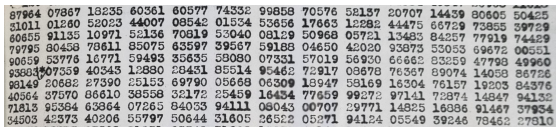


Figure 1: Example image of the historical key stream tables used in this work. Image courtesy of Sir Dermot Turing and the National Archives.

2.2 Pre-Processing

Colour images of the tables were taken with a smartphone and saved in PNG format. The images were cropped as best as possible to contain only digits. These cropped image files were read into a Jupyter Notebook using the Open Source Computer Vision (cv2) Python package (2025). The imported images were converted to black-and-white, with adaptive Gaussian thresholding used to remove noise that might impact subsequent steps.

2.3 Computer Vision for Detection and Segmentation

Characters (digits) were identified by their coordinates in each pre-processed image using the Python-tesseract package; a python wrapper for Google's Tesseract optical character recognition engine (2024).

2.4 Digit Classification

Although Tesseract itself provides a classification (i.e., digit label) for each identified character, it was quickly realised that this pre-trained classifier performed poorly on the tables used in this study, e.g. by confusing the digits '3' and '5' (see Section 3). Therefore, it was necessary to build a bespoke classifier trained on these specific tables (as opposed to other texts).

For this purpose, a convolutional neural network (convnet) was built in Python using the TensorFlow (2025) and Keras (2015) packages. This convnet used a simple, feed-forward (sequential) architecture with three hidden convolution layers (each with a convolution window size of 3; and with 32, 64, and 128 respective filters; and each followed by pooling layers), as well as a dropout layer to prevent over-fitting. Batch sizes from 16 to 128, and epoch numbers from 50 to 300 were tested. The model was evaluated with three-fold cross-validation to estimate the generalisation performance. Balanced accuracy (as opposed to standard accuracy) was used as the evaluation metric to account for class imbalance (some digits appeared more frequently than others).

To train and evaluate the convnet, a dataset of $N = 651$ manually-labelled characters was created. This dataset included noise 'characters' (i.e., blobs detected by Tesseract representing ink/paper artefacts rather than real digits) which were treated as a separate, 11th class/category. After evaluation, the convnet was fitted to this entire dataset, and the final trained/fitted convnet was applied to all available images of the tables to create a final key stream dataset to be tested for randomness.

2.5 Testing for Randomness

Four tests for randomness were used, testing different aspects of the key stream:

1. **Arithmetic Mean ($\bar{x}$) Test:** In the limiting case, the frequencies of truly randomly-generated digits from 0–9 should be equal. The theoretical arithmetic mean of a truly random key stream is then $\bar{x} = \frac{0+1+2+3+4+5+6+7+8+9}{10} = 4.5$, where each digit has equal weight due to equal frequencies. To test whether the centrality of a key stream of length n is consistent with this theoretical value with 95% confidence, the observed arithmetic mean $\hat{\bar{x}}$, standard deviation s, and 95% confidence interval

$\hat{\bar{x}} \pm t_c \frac{s}{\sqrt{n}}$ of the key stream can be calculated, where t_c is the t-value corresponding to the significance level $\alpha = 0.05$ and degrees of freedom $\mathrm{df} = n - 1$. If the 95% confidence interval contains the theoretical value, then the arithmetic mean test is passed with 95% confidence, in support of the randomness of the key stream; otherwise, it is failed.

2. **Chi-Squared (χ^2) Test:** The expected (theoretical) frequencies for n total digits 0–9 generated randomly are all $f_e = \frac{n}{10}$ (a discrete uniform distribution). Pearson's chi-squared test can be used to compare the distribution of observed frequencies f_O in a key stream of length n to the expected frequencies f_E. The test statistic is given by $D = \sum_i \frac{(f_{\mathrm{O},i} - f_{\mathrm{E},i})^2}{f_{\mathrm{E},i}}$, where the sum is over the $k = 10$ digits 0–9. This test statistic is compared to a chi-squared distribution with $\mathrm{df} = k - 1$. If the p-value of the test is $\geq \alpha = 0.05$, then the chi-squared test is passed with 95% confidence, in support of the randomness of the key stream; otherwise, it is failed. Whereas the arithmetic mean test can be passed without equal frequencies (e.g. if the digits 0 and 9 are equally inflated), the chi-squared test detects unevenness in the frequencies. This test used the SciPy Python package (Virtanen et al., 2020). It can be thought of as an extension to decimal sequences of the Frequency (Monobit) Test for binary sequences in the NIST Statistical Test Suite (Bassham et al., 2010).[1]

3. **Runs Test:** A random key stream should not contain trends, i.e. long runs of values all above/below the median; or cyclic effects, i.e. regular oscillations about the median. A runs test can be used to test for the presence of these effects in a key stream. If the p-value of the test is $\geq \alpha = 0.05$, then the runs test is passed with 95% confidence, in support of the randomness of the key stream; otherwise, it is failed (Bradley, 1968, Chapter 11). This test used the statsmodels Python package (Seabold and Perktold, 2010). It can be thought of as an extension to decimal sequences of the Runs Test for binary sequences in the NIST Statistical Test Suite (Bassham et al., 2010).

4. **Monte Carlo Pi Test:** A circle of radius $r = 1$ and area $A_\circ = \pi r^2 = \pi$, and a square of side length $a = 2$ and area $A_\square = a^2 = 4$ are centered on the origin of a Cartesian coordinate system. The ratio between these shapes' areas is $\frac{A_\circ}{A_\square} = \frac{\pi}{4} \leftrightarrow \pi = \frac{4A_\circ}{A_\square}$. n points are placed randomly in the two-dimensional space. The number of points that fall inside the circle $n_\circ \propto A_\circ$. The number of points that fall inside the square $n_\circ + n_{\circ'} \propto A_\square$, where $n_{\circ'}$ denotes the number of points that fall inside the square but *not* the circle. In the limiting case, $\pi = \lim_{n\to\infty} \frac{4n_\circ}{n_\circ + n_{\circ'}}$, and an estimate for π for some finite n is given by $\hat{\pi} = \frac{4n_\circ}{n_\circ + n_{\circ'}}$ with approximate 95% confidence interval $\hat{\pi} \pm z_c \sqrt{\frac{\pi(4-\pi)}{n}}$, where z_c is the z-score corresponding to the significance level $\alpha = 0.05$. A key stream can be divided into successive strings of six digits (e.g. 160930, 717903, ...), which are then divided by $1{,}000{,}000$ to produce random numbers from 0.00000 to 0.999999 (e.g. 0.160930, 0.717903, ...). Successive pairs of these numbers are treated as coordinates, e.g. $(x = 0.160930, y = 0.717903)$. If the distance between this coordinate and the origin $d = \sqrt{x^2 + y^2} <= r$, then $n_\circ$ is incremented by one, else if $d > r$ then $n_{\circ'}$ is incremented by one. $\hat{\pi}$ and its 95% confidence interval are then calculated with the equations above. If the 95% confidence interval contains the true value of $\pi = 3.14159...$, then the Monte Carlo pi test is passed with 95% confidence, in support of the randomness of the key stream; otherwise, it is failed (Li and Nakano, 2022; Nakade, 2025).

2.6 Reference Streams

Two reference key streams with lengths equal to the historical tables were produced with the following methods:

1. Mersenne Twister (MT): the reproducible random number generator used in the 'random' Python package (Matsumoto and Nishimura, 1998; Python Software Foundation, 2025a).

2. Cryptographically Strong Pseudo-Random Number Generator (CSPRNG): the non-reproducible random number generator used

[1] For a similar example using the normal distribution, see Tomášek et al. (2021).

in the 'secrets' Python package, which "provides access to the most secure source of randomness that your operating system provides" (Python Software Foundation, 2025b).

All four tests for randomness were applied separately to the historical key stream and reference streams.

3 Results

For the convnet, 100 epochs and a batch size of 32 resulted in the optimal mean cross-validated testing balanced accuracy (93.7%, with a standard deviation of 0.7%) without evidence of overfitting (perfect mean cross-validated training balanced accuracy, which was observed for more epochs). By way of comparison, the pre-trained Tesseract classifier (configured for single digits) achieved a balanced accuracy of just 21.7% on the objects it extracted from the training image. This demonstrates the superiority of a bespoke convnet for classification of obscure text images whose properties are unlike those Tesseract was trained on.

The tables contained $N = 83,390$ digits total. Tesseract extracted $N = 81,252$ objects from all historical key stream images combined, of which the final convnet classified $n = 74,979$ as digits from 0–9, and $n = 6,273$ as noise which were removed for subsequent analyses.

Table 1 shows the results of the randomness tests for each key stream. Both reference streams (MT and CSPRNG) passed all four tests, while the historical key stream failed all tests.

Test	**H**	**MT**	**CSPRNG**
$\bar{x}$	Failed	Passed	Passed
$\hat{\bar{x}}$ (95% CI)	4.228 (4.207, 4.248)	4.511 (4.490, 4.532)	4.492 (4.472, 4.513)
χ^2	Failed	Passed	Passed
p-value	< 0.001	0.307	0.415
Runs	Failed	Passed	Passed
p-value	< 0.001	0.148	0.152
π	Failed	Passed	Passed
$\hat{\pi}$ (95% CI)	3.319 (3.279, 3.360)	3.133 (3.092, 3.174)	3.149 (3.108, 3.189)

Table 1: Randomness test results with 95% confidence. H means historical key stream tables. MT means Mersenne Twister. CSPRNG means cryptographically strong pseudo-random number generator.

Figure 2 visualises the Monte Carlo π test results. There appears to be a small cluster of co-ordinates from the historical key stream tables around $x \approx 0.3$, $y \approx 0.3$ (see Section 4).

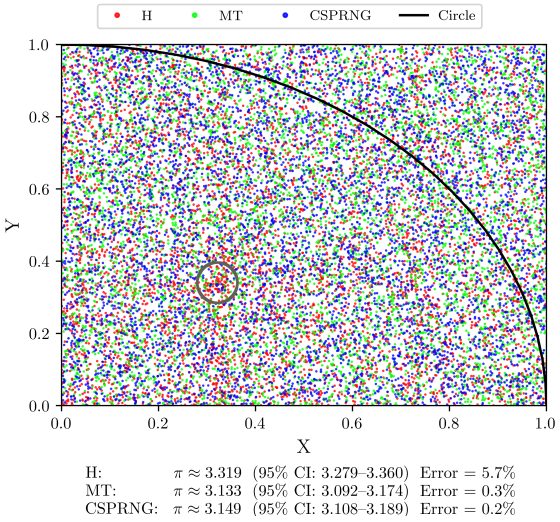


Figure 2: Monte Carlo estimates for the constant π. $N = 6,248$ points derived from $N = 74,976$ digits per method (six digits per co-ordinate, two co-ordinates per point). The dark grey ring around $(x \approx 0.3,\ y \approx 0.3)$ is a visual indicator of the cluster of co-ordinates from the historical key stream tables.

4 Discussion

This study found statistical evidence for lack of randomness in a digitization of historical key stream tables used by diplomats in WW2-era Germany. The digitized key stream's arithmetic mean was too low. Inspection of the raw frequencies reveals that this is due to over-representation/inflation of the digit 3, and under-representation of the digits 8 and 9. The over-representation of the digit 3 in particular ($n = 9,894$ compared to the theoretical $n = 7,498$) explains the cluster observed in Figure 2, as well as the failings of both the χ^2 and π tests. The failing of the runs test also implies that the digitized key stream contains trends, but the extent to which these trends could be exploited by cryptanalysts is unknown.

The major limitation of this study is that bias could have been introduced in the digitization process for the historical key stream tables; if the computer vision and/or the convnet digit classification under/over-extracted/classified one or more digits, this in turn could have biased the results of the randomness tests. E.g., if the computer vision systematically failed to recognise the digit '3'

(which had the highest measured frequency), or the classifier systematically mistook the digit '8' (which had the lowest measured frequency) for the similar-looking digit '3', then it may be erroneous to conclude that the historical key stream lacked randomness.

Bias was investigated by manually counting the frequency of the digits 3, 8, and 9 in the training image, and comparing these to (1) the number of Tesseract-extracted digits from this image; and (2) the number of convnet-classified digits in this image. There were $n = 64$ digit 3s, $n = 57$ digit 8s, and $n = 61$ digit 9s in the training set image. Tesseract extracted $n = 59$ digit 3s, $n = 44$ digit 8s, and $n = 58$ digit 9s intelligibly (remaining digits in the training set were extracted by Tesseract but with additional digits in the same extraction, e.g. in one case, '34' was extracted as a single object. These were labeled manually as noise). Thus, Tesseract under-represented the number of digit 3s, 8s, and 9s by five, thirteen, and three counts, respectively in the training set image. This likely explains the apparent under-representation of digit 8s (and possibly digit 9s, though to a lesser extent) in the historical tables (when computer vision and the convnet were applied to all available images), but does not explain the apparent *over*-representation of digit 3s in the historical tables, since the biasing effect from Tesseract would work in the opposite direction. The convnet classified $n = 60$ digit 3s, $n = 44$ digit 8s, and $n = 58$ digit 9s from the Tesseract extractions; thus, the convnet is unlikely to have introduced substantial bias as these are identical or very similar to the number of Tesseract extractions for each digit. Thus, the observed under-representation of digit 8 in the historical key stream may be a simple artefact of computer vision error, but the over-representation of digit 3 may not.

In conclusion, computer vision and convolutional neural networks can be used to digitize historical key streams. The particular digitized key stream in this work, the purpose of which is unknown, appeared to lack randomness in basic statistical tests. This finding may raise real doubts about the security of the key stream from a modern cryptographic perspective, but does not alone prove practical cryptanalytic exploitability. Bias introduced by computer vision and the convnet cannot be ruled out for some digits but appears less likely for others.

Acknowledgments

The author thanks Sir Dermot Turing for providing the images of the historical key stream tables and some references used in this work, the National Archives for preserving the original documents, and three anonymous reviewers for helpful comments and suggestions. Code is available in an online GitHub repository: `https://github.com/FloeFoxon/Statistical-Tests-for-Randomness-With-Computer-Vision-and-a-Convnet`

References

Lawrence E. Bassham, Andrew L. Rukhin, Juan Soto, James R. Nechvatal, Miles E. Smid, Stefan D. Leigh, M. Levenson, M. Vangel, Nathanael A. Heckert, and D. L. Banks. 2010. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*. National Institute of Standards and Technology, Gaithersburg, MD. `https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=906762`.

James V. Bradley. 1968. *Distribution-free statistical tests*. Prentice-Hall, Englewood Cliffs, NJ.

Percy William Filby. 1995. Floradora and a unique break into one-time pad ciphers. *Intelligence and National Security*, 10(3):408–422. `https://doi.org/10.1080/02684529508432310`.

Louise Foley. 2001. *Analysis of an On-line Random Number Generator*. Trinity College Dublin. `https://www.random.org/analysis/Analysis2001.pdf`.

Charmaine Kenny. 2005. *Random Number Generators: An Evaluation and Comparison of Random.org and Some Commonly Used Generators*. Trinity College Dublin. `https://www.random.org/analysis/Analysis2005.pdf`.

Keras Developers. 2015. *Keras*. `https://keras.io`.

Rongpeng Li and Aiichiro Nakano, 2022. *Calculating Pi with Monte Carlo Simulation*, pages 1–18. Apress, Berkeley, CA. `https://doi.org/10.1007/978-1-4842-8185-7_1`.

Makoto Matsumoto and Takuji Nishimura. 1998. Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. *CM Transactions on Modeling and Computer Simulation*, 8(1):3–30. `https://doi.org/10.1145/272991.272995`.

Apurva Nakade. 2025. Estimating π. In *Monte Carlo Methods*. `https://apurvanakade.github.io/Monte-Carlo-Methods/chapters/estimation/estimating_pi.html`.

Svetlin Nakov. 2019. Secure random generators (csprng). In *Practical Cryptography for Developers*. https://cryptobook.nakov.com/secure-random-generators/secure-random-generators-csprng.

opencv-python Developers. 2025. *opencv-python 4.12.0.88*. https://pypi.org/project/opencv-python/.

pytesseract Developers. 2024. *pytesseract 0.3.13*. https://pypi.org/project/pytesseract/.

Python Software Foundation. 2025a. *random — Generate pseudo-random numbers*. Python Software Foundation, Wilmington, DE. https://docs.python.org/3/library/random.html.

Python Software Foundation. 2025b. *secrets — Generate secure random numbers for managing secrets*. Python Software Foundation, Wilmington, DE. https://docs.python.org/3/library/secrets.html#module-secrets.

Skipper Seabold and Josef Perktold. 2010. statsmodels: Econometric and statistical modeling with python. In *Proceedings of the 9th Python in Science Conference*, pages 92–96. https://doi.org/10.25080/Majora-92bf1922-011.

TensorFlow Developers. 2025. *TensorFlow 2.20.0*. Zenodo. https://doi.org/10.5281/zenodo.16852354.

Pavel Tomášek, Hana Tomášková, and Jakub Rak. 2021. Chi-square of Pseudorandom Number Generator of Normal Distribution in C++17. *TEM Journal*, 10(4):1495–1499. https://doi.org/10.18421/TEM104-01.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272. https://doi.org/10.1038/s41592-019-0686-2.

A Medieval Czech Penitential Prayer Behind the Cryptographic Enigma of Santa Maria La Nova?

Cosimo Palma
University of Pisa, Italy
cosimo.palma@phd.unipi.it

Louie Helm
RockstarResearch.com, USA
louiehelm@protonmail.ch

Abstract

This paper presents renewed attack on the encrypted inscription housed in the Turbolo Chapel of the Neapolitan Church of Santa Maria La Nova. Following the cryptanalytic hypothesis of monoalphabetic substitution and the multilingual leads explored in related works, the present contribution investigates Old Czech, a candidate not yet systematically tested against the cipher. Through a combination of n-gram-driven automated decryption via AZdecrypt and LLM-assisted segmentation of the resulting *scriptio continua*, a candidate reading has been obtained, pointing to a medieval Czech penitential prayer. Although a definitive solution cannot be provided without the support of philologists and cryptologists versed in Czech and related languages, the convergence of statistical, lexicographic and thematic evidence constitutes to date the strongest cue for this long-standing historical puzzle.

1 Introduction

Decrypting historical ciphers can reveal otherwise inaccessible aspects of the past. The importance of this activity is reflected in initiatives such as the DESCRYPT project (Megyesi et al., 2025), whose aims include the systematic collection and cataloguing of encrypted historical sources. Among the many unsolved historical ciphers that still await a convincing solution, one particularly striking case is found in Naples.

The ciphered epigraph of the Turbolo Chapel at Santa Maria la Nova has been the subject of two previous investigations.

The first contribution (Palma, 2023) demonstrated that the epigraph overwhelmingly displays the statistical signature of a monoalphabetic substitution cipher, while the second (Palma et al., 2025) strengthened the hypothesis that some glyphs may function as homophones and that the plaintext could involve multiple languages. The only suggestive partial results from Palma et al. (2025) was the identification of the Greek sequence ΒΑΣΙΛΕΙΑ ΝΑΥΣΙΚΑ ΣΧΕΡΙΑ in one row of the inscription (Nausicaa is renown as the legendary princess of the Kingdom of Scheria, the island where Odysseus is shipwrecked in Book 6 of the *Odyssey*), alongside ̓ΑΓΟΡΕ΄Υ(-) ΓΟΝΥ ΔΈ ̔ΡΗΤΎΡ ̓ΑΓΝΌ (*... to harangue on one's knees, but rather as a venerable rhetor*) in another. However, these partial mappings could not be extended consistently to the rest of the text, leaving open the possibility of multilingualism or an entirely different plaintext language.

In the present work, we focus on a lead which emerged from a convergence of independent observations. The normalized IC of the inscription aligns closely with that of Old Czech corpora spanning the 13th–20th centuries, as reported in Table 3. A diagnostic analysis on the epigraph's pigments, based on UV-induced fluorescence, infra-red imaging and pigment sampling, confirmed that the inscription dates to the 16th century at the latest, with repainting interventions beginning much earlier than previously assumed (Falcucci, 2018). This dating range would be fully compatible with a text in Old Czech, a language which moreover possesses deep historical ties to the Franciscan Observant milieu that permeated Santa Maria la Nova.

The structure of the paper is as follows: Section 2 recapitulates the relevant preliminary data; Section 3 describes the decryption strategy and presents the main candidate readings; Section 4 discusses the thematic convergence and its statistical plausibility; Section 5 concludes with the study limitations, open questions and envisioned future work.

2 Preliminary data and revised transcription

The frequency analysis and Index of Coincidence (IC) calculations from the previous works (Palma, 2023) slightly differ from the values presented in Table 3, showing a variance of approximately 0.25. This variation stems from a re-transcription aimed at covering the entire epigraph (Figure 3, Appendix), undertaken to extend the decryption coverage. The false-color rendering, in particular, helped disambiguate glyphs whose identification was previously uncertain, leading not only to additions but also to revisions of some previously assigned mappings.

Given the exploratory nature of the IC in any decipherment approach, this does not constitute a significant issue; however, inexperience during the initial analysis led to an overestimation of the values' significance, and consequently the first observed difference between the IC of the inscription and that of medieval Czech was judged to be excessive. For this reason, an attack on medieval Czech was long overlooked.[1]

Language / Corpus	IC
Old Hungarian (Erdy Codex)	2.10
Old Romanian (16th-18th centuries)	1.98
Old Albanian (Buzuku Missale)	1.93
Ancient and Koiné Greek	1.89
Latin	1.87
Old Hungarian (Hussite Bible)	1.85
Old Slovenian	1.69
Old Church Slavonic	1.63
Old Czech (16th–20th c.)	**1.56**
Old Czech (13th–16th c.)	**1.55**
Inscription	**1.53**
Old Bohemian (Kralice Bible)	1.40

Table 1: Index of Coincidence values of main historical candidates. The inscription's IC matches the Old Czech (13th–16th c.) corpus most closely.

For the n-gram model driving the AZdecrypt attack, only texts dated between the 13th and 16th centuries were used, collected from the online repository *HistCorp* (Pettersson and Megyesi, 2018). Two separate IC measurements were performed: one on the Kralice Bible alone (1.40) and one on the broader medieval Czech corpus (1.55), which proved to be the closest match to the IC of the inscription among all the languages investigated.

[1]This metric should be regarded as indicative only: the brevity of the ciphertext renders the computed value little more than an estimate, and the partial damage to the epigraph, with all the peculiarities this entails, further increases the volatility of the IC.

3 Decryption strategy

The decryption process was carried out in two successive phases.

3.1 Phase 1: Automated n-gram attack

The 5-gram model generated from the medieval Czech corpus[2] was loaded into AZdecrypt (AZdecrypt, 2023; Palma, 2023) (version 1.25), and the solver was run in both *Substitution* mode and *Substitution + Nulls & Skips* mode, selecting the entropy level as "1". The *temperature* parameter has been set to 700. The input file, unlike the one used for the attacks in Palma et al. (2025), contains not only the most legible section of the epigraph, corresponding to roughly 26 lines, but also the less readable portions, rendered through ASCII-formatted special characters (numbers, capitalized and accented letters, symbols), thus bringing the total number of lines to 39 (the first two lines were excluded due to almost complete illegibility).

The output of the automated attack produced a *scriptio continua*, a continuous string of Latin characters without word boundaries, as is to be expected for a text originally carved in glyphs not separated by spaces.

The process described above produced many candidate readings; we take in consideration only two: the first, referred as "Hospodiný" (from the opening word of the first) and second "Litování" (from the opening word of the second). Both readings emerged from the same AZdecrypt input, processed in different stages: the former represents the result after circa 20 minutes of computation, the latter after circa 1 hour and 20 minutes. They converge on a single genre: a medieval penitential prayer in the Czech language. In the following, we will focus chiefly on the final one, "Litování".[3]

[2]The input file, the corpus, the n-gram models and the scripts used are available in the project repository: `https://github.com/Glottocrisio/MariaLaNova`.

[3]It is noteworthy to highlight that both words originate from a string of ASCII special characters inserted in place of a damaged portion of the inscription. Therefore, both words cannot be used as concrete results. However, "Hospodin" will be mentioned again in the following as it appears in another not-deleted portion.

3.2 Phase 2: LLM-assisted segmentation

To segment the continuous output into Old Czech words and phrases, two independent Large Language Models (LLMs) were employed: *Claude Opus 4.6* (Anthropic) and *Gemini 3.1 Pro* (Google). Each was provided with the raw letter strings and instructed to identify plausible Old Czech phrases. This process was iterated multiple times, producing slightly varying segmentations at each pass but with a stable thematic core:

- An address to God using the archaic vocative *Hospodin*;
- Penitential vocabulary: sin (*hřích / hříeše*), guilt (*vinná*), repentance (*litování*), mercy (*milost / milosrdenství*);
- Body-soul contrast: carnal life (*tělesně žil*) versus spiritual illumination (*svíce / světlost*);

The rationale for using LLMs lies in their capacity to recognise morphological patterns across languages for which dedicated NLP tools may be unavailable or unreliable for historical varieties. It is important to stress that the n-gram models driving the AZdecrypt solver and the LLMs performing the segmentation are methodologically independent: AZdecrypt optimises for statistical fit to the corpus, while the LLMs segment on the basis of lexical and morphological plausibility.

Figure 1: A glimpse of the epigraph transcription (excerpt of four fully readable lines, 15–18), as performed in November 2025. Below each line the encoding in machine-readable characters as shown in Figure 2 (top).

Despite the undeniable superiority of "Hospodiný" compared to outputs from other candidate languages, what really encouraged us to pursue the Old Czech lead with more conviction are some striking patterns in the glyph-character mapping, as presented in Figure 2. Furthermore, this first attempt is showcased to provide a detailed account for the whole genesis of the proposed solution (reading ""Litování"").

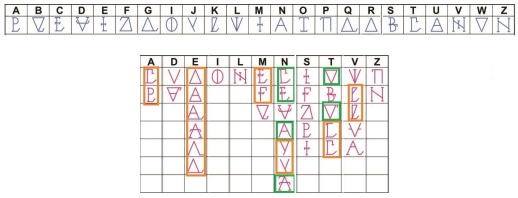


Figure 2: Glyph-character encoding (top) and decryption key (bottom) of the inscription, reading "Hospodiný". It can be noted, for example in columns "T", "N" and "E", that visually similar epigraph glyphs cluster under the same Latin-character column headers, a coincidence supporting the assumption that homophony might be implemented by means of allographs. It can be noted that only about half of the alphabet features in the output.

The better fleshed-out decryption key for "Litování" (Table 8) does not present as many, although some correspondences with the "Hospodiný" reading are to be observed.

4 Quantitative assessment of the candidate reading

A legitimate concern is whether this thematic coherence might be an artefact of the decryption process itself– either of the n-gram model, of the LLMs, or of both. We address this concern on three fronts.

First, the Old Czech corpus used for n-gram generation is not monothematic. The Czech National Corpus project, maintained at Charles University in Prague, has documented that the diachronic section of the corpus (DIAKORP) spans texts from the end of the 13th century onward, covering administrative, literary, religious and legal registers (Křen, 2020; Kučera et al., 2015). The Kralice Bible itself, included in our primary source, encompasses the full diversity of biblical genres–narrative, prophetic, legal, epistolary, sapiential and lyrical. Were the n-gram model biased toward penitential vocabulary, this bias would have to arise from the entire corpus, which includes far more military, historical and legal n-grams than devotional ones.

Second, AZdecrypt's hill-climbing algorithm is language-agnostic: it maximises the log-probability of the decrypted text against the n-gram model without any semantic awareness. It cannot "choose" to produce penitential content

rather than, say, a legal charter or a chronicle passage. The fact that the optimised output, when segmented, resolves into penitential vocabulary is therefore an emergent property of the cipher itself.

Finally, one could argue that only a strictly mono-alphabetic substitution ought to justify the similarity of the Index of Coincidence for both inscription and candidate language: as better observable in the Appendix (see Section B) a full decryption has not been achieved yet, and in this phase limited by 5-grams well under full coverage, the presence of homophony in the solution is unavoidable. For a more grounded statistical analysis of the obtained results, we take into account *texts*, formatted and cleaned up according to a shared *normalisation*, which are then compared using *heuristic metrics*.

Texts. The inscription is the latest transcription (615 alphabetic characters after normalisation). The candidate plaintext is the 40-row AZdecrypt output, with the LLM-assisted segmentation yielding 305 tokens. The Old Czech reference is the concatenation of three DIAKORP volumes covering the 14th–15th centuries (1,112,323 characters, 237,384 tokens after stripping metadata and Roman-numeral section markers): *Lékařství neznámého františkána* (1440–1460), *Milíčovský sborník modliteb* (1350–1400), and *Životy svatých otců* (1400–1450) (Kučera et al., 2015).

Normalisation. For all character-level metrics we adopt a single normalisation: lower-cased, NFD-decomposed and diacritic-stripped, keeping only the 26-letter Latin alphabet $\{a,\ldots,z\}$. This places the cipher transcription (which has no diacritics) on the same alphabet as the corpora (whose diacritics would otherwise inflate their entropy and deflate their IoC), and matches the working alphabet used internally by AZdecrypt. The same normalisation is applied to every text.

Heuristics. Five measures are computed on the candidate plaintext and benchmarked against the DIAKORP Old Czech corpus (13th to 15th century): (i) Shannon entropy H (Shannon, 1951), the average information per character; (ii) the Index of Coincidence (Friedman, 1922), sensitive to letter-frequency concentration; (iii) average word length; (iv) the Zipfian rank-frequency exponent s and goodness-of-fit R^2 (Zipf, 1949), fitted by ordinary least squares on $\log_{10} f(r) = \log_{10} C - s \log_{10} r$; and (v) a bootstrap calibration of (s, R^2) at matched sample size. Measures (i) to (iii) use the 26-letter normalisation (diacritics stripped, case folded). The Zipfian fit uses whitespace tokens after removing Roman-numeral section markers (*i, ij, iij, …*). We adopt OLS rather than maximum likelihood (Clauset et al., 2009) for direct comparability with the bootstrap, which is computed identically on observed and resampled data. For the bootstrap we draw $B = 10{,}000$ contiguous windows of $N = 305$ tokens (preserving local thematic coherence) from two pools: the combined DIAKORP corpus (broad register) and DIAKORP-29 alone (prayer-book register), reporting the empirical 95 % interval and a two-sided p-value for the candidate's (s, R^2).

Source	H	N
Inscription cipher	4.235	615
Candidate plaintext	4.067	2,669
DIAKORP combined	4.226	1,112,323
Uniform random ($H_{\max}$)	4.700	—

Table 2: Shannon entropy H (bits/char) on the 26-letter normalisation.

The cipher and the DIAKORP corpus agree on H to within 0.01 bit/char, consistent with a monoalphabetic substitution preserving the single-character entropy of the plaintext. The candidate sits slightly lower, as expected for a short, thematically focused text in which a few content words concentrate probability mass. All three values lie well below the $\log_2 26 = 4.700$ bit ceiling, replicating the non-randomness diagnostic of Palma (2023).

Source	raw IoC	norm. IoC
Inscription cipher	0.0628	1.633
Candidate plaintext	0.0752	1.956
DIAKORP combined	0.0610	1.587
Uniform random	0.0385	1.000

Table 3: Index of Coincidence (normalised IoC = IoC·26).

The cipher's IoC (1.633) matches the corpus (1.587) within natural variation, consistent with substitution of an Old-Czech-like plaintext and replicating the language-identification result of Palma et al. (2025) on the re-cleaned transcription. The candidate sits above both, in the direction predicted by genre concentration: the mirror image of the entropy displacement (lower $H \Leftrightarrow$ higher IoC).

Source	avg. length	N tokens
Inscription cipher	N/A	—
Candidate plaintext	3.13	305
DIAKORP combined	4.63	237,384

Table 4: Average word length (chars/token) after removing Roman-numeral markers; the inscription is in *scriptio continua* (*N/A*).

Arguably, the 1.5-character gap between candidate and corpus reflects the fact that LLM-assisted segmentation over-splits in weakly-resolved lines (about 30 % of solution tokens are 1–2 characters: *i, a, sě, na, ne*).

Source	V	N	s	R^2
Candidate plaintext	140	305	0.705	0.871
DIAKORP combined	28,437	237,384	0.901	0.992
Canonical Zipf	—	—	1.000	—

Table 5: Zipfian fits (V: vocabulary, N: tokens). The corpus fit uses the top-1,000 ranks; the candidate's $V = 140 < 1000$, so full and top-V fits coincide.

The candidate's $R^2 = 0.871$ is solidly within the natural-language range; gibberish strings typically yield $R^2 \in [0.4, 0.6]$ at this sample size (Ferrer-i Cancho and Elvevåg, 2010). The exponent $s = 0.705$ is shallower than the asymptotic canonical value of 1, but the relevant benchmark at $N = 305$ is not the asymptote: it is the empirical distribution of s for matched-size corpus windows, since finite-sample effects depress s even for genuine Old Czech text.

Null distribution	s	R^2
DIAKORP combined (broad register)		
mean	0.471	0.829
95 % CI	[0.352, 0.661]	[0.731, 0.913]
DIAKORP-29 only (prayer book)		
mean	0.430	0.822
95 % CI	[0.327, 0.544]	[0.737, 0.886]
Candidate (observed)	**0.705**	**0.871**

Table 6: Bootstrap distributions of (s, R^2) from $B = 10{,}000$ contiguous 305-token windows.

The candidate's R^2 lies comfortably inside both 95 % intervals, while s lies above the upper 97.5 % percentile of both, indicating greater lexical concentration than a typical short slice of either pool.

	s		R^2	
Null	in CI?	p	in CI?	p
DIAKORP combined	no	0.011	yes	0.352
DIAKORP-29 only	no	$< 10^{-4}$	yes	0.198

Table 7: Position of the candidate's $(s, R^2) = (0.705, 0.871)$ in each null; p two-sided.

4.1 Joint reading

The five measures point in a single direction. Tables 2 and 3 establish that the cipher's letter-frequency profile is indistinguishable from Old Czech on this normalisation, and the candidate sits near the corpus on both, consistently displaced toward greater concentration. Table 4 shows shorter average word length, attributable to over-segmentation. Tables 5 to 7 show that the candidate follows the rank-frequency law of natural language ($R^2 = 0.871$, well inside the empirical 95 % interval), with a steeper slope than the corpus null (two-sided $p = 0.011$ broad, $p < 10^{-4}$ prayer-book).

Three independent statistics, namely single-character entropy, Index of Coincidence, and Zipfian exponent, thus converge: the candidate is natural-language Old Czech with measurably greater lexical concentration than a generic short slice of the corpus, the quantitative signature of a tightly thematic short text. This provides a quantitative correlate for the thematic convergence already established at the lexical level.

The $\sim 10^{-4}$ p-value against the prayer-book null does not indicate that the candidate is unlikely to be Old Czech prayer text; it reflects that the inscription is considerably shorter and more thematically tight than an arbitrary 305-token window of a 250-page prayer-book, while LLM-assisted segmentation over-produces single-character function-word tokens. Both effects inflate the share of high-frequency types and steepen the Zipf slope, while leaving the R^2 of the fit well inside the natural-language range.

4.2 Historical plausibility

Santa Maria la Nova was, from the late 15th century, one of the principal seats of the Franciscan Observants in Southern Italy. Saint James of the Marches (Giacomo della Marca, d. 1476), who had led extensive missions in Bohemia and Hungary against the Hussites, was closely associated with this church and was buried in its cloister

Glyph	Input	Freq.	Lito.
	a	49	e
	b	21	v
	c	30	e
	d	24	z
	e	10	s
	f	23	l
	i	78	i
	j	7	c
	k	11	n
	l	24	t
	m	65	n
	n	30	i
	o	16	v
	p	19	u
	q	15	a
	r	46	a
	s	19	e
	t	33	m
	u	44	e
	v	15	l
	w	3	c
	z	5	n

Table 8: Decryption key for the ‘Litování’ reading (reflecting Fig. 2 for the glyph-character encoding). Entries are sorted alphabetically by the input glyph; only inputs in the range a–z are listed. The first column shows the original inscription glyph, the second its AZdecrypt input symbol, the third its frequency, and the last the proposed Old Czech plaintext letter. Ten of the twenty-two input characters are left out from the solution, showing a still too high degree of homophony due to the limitations of the adopted n-grams model.

(the incorrupt remains were transferred in 2001 to Monteprandone, his native place, where it is visible to the public today). The presence of a Bohemian text in this specific chapel is therefore not historically anomalous: it would represent a “linguistic ex-voto”, a devotional inscription in the language of a land reconverted from heresy, brought to Naples by a Bohemian legate or friar accompanying one of the many diplomatic and religious exchanges between the Aragonese court and Central Europe.

Whether this represents an ex-voto preserved in convent archives and subsequently encrypted as an academic exercise, or (as historical evidence suggests is plausible) the spiritual legacy of a descendant of the Bohemian Brethren (founded around 1457 as heirs to the Hussite movement) who chose to inscribe their heterodox faith in a secure form, the fundamental significance remains unchanged.

More in general, a penitential prayer appears in a church because it constitutes an act of devotion, yet it is encrypted because the shame of confessed guilt proves too intimate to leave exposed.[4] The ciphered text, in this instance, conceals the message not from the Lord, but from human eyes.

5 Future Work and Conclusion

The methodological setup employed here, namely AZdecrypt for n-gram-driven substitution search, paired with an LLM for segmentation and gloss, has proven sufficiently robust: parallel runs against alternative candidate languages, conducted as internal controls and not reported here, yielded no spurious convergence: when the target language is falsely specified, the pipeline returns no coherent output rather than fluent noise; even single words are hard to recognise, let alone any semantic connection among them. The Old Czech reading, by contrast, has been independently confirmed as recognisable Medieval Czech by two native speakers (see Acknowledgements and Appendix).

Furthermore, the Slavic language family deserves a more articulated treatment. We tentatively advance the hypothesis that the Old Czech words and structures recovered here indicate a correct approach but might still point to a slightly displaced target language. Several high-confidence reconstructions have direct Old Church Slavonic counterparts (*milost* / *milostĭ*; *litovánie* / *litovanije*; *slíbil* / *sŭljubŭ*; *věci* / *vešči*; *dávání* / *dajanije*; *tísni* / *těsnĭ*), and the apparent vocalic over-abundance in the segmented output is consistent with a transcription tradition in which long vowels were rendered through duplicated or alternate graphemes. A natural next step is therefore to extend the reference corpus beyond DIAKORP, incorporating Old Church Slavonic devotional material and testing whether the substitution key stabilises under a Slavic continuum rather than a strict Old Czech target.

Both threads converge on the recurrent limiting factor for n-gram-based decryption of histori-

[4] A similar conjecture was formulated nine years ago, even before inception of any cryptological attack. See `https://storienapoli.it/2017/05/12/santa-maria-la-nova/`.

cal inscriptions: corpus size. Under-resourced languages produce sparse n-gram tables, and the substitution search inherits that sparsity as ambiguity in the key.

We are currently developing a complementary pipeline in which an LLM, conditioned on an attested Old Czech lexicon and on thematic prompts derived from extant literature, generates synthetic but grammatically and stylistically controlled Old Czech text; the synthetic output is then used to enrich the n-gram tables consumed by AZdecrypt. Preliminary results will be reported in a forthcoming publication.

The present investigation has produced the most semantically coherent and linguistically plausible candidate decryption of the Santa Maria la Nova epigraph to date. Two independent readings, obtained through a combination of n-gram-optimised substitution and LLM-assisted segmentation, converge on a medieval Czech penitential prayer addressed to *Hospodin* (the Lord), featuring vocabulary of sin, repentance, mercy and divine illumination. The morphological features of the identified Old Czech forms point to a dating between the late 14th and mid-15th century, consistent with the physical dating of the inscription.

Acknowledgements

The authors warmly thank Dr. Paolo Bonavoglia and Mr. Francesco Pastore for their continuous support and pertinent criticism, and Martin Lednický and Dr. Pavol Zajac for confirming the fruitfulness of the Slavic lead. Dr. Zajac kindly provided an annotated commentary on the clear-text and insightful remarks on the plausibility of further Slavic languages (see appendix and section 5).

We also thank the anonymous reviewers for their valuable suggestions, and Dr. Cécile Pierrot for her help in improving the readability and structure of the camera-ready version.

References

AZdecrypt. 2023. Azdecrypt. Accessed: 2024-12-01.

Aaron Clauset, Cosma Rohilla Shalizi, and M. E. J. Newman. 2009. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703.

Claudio Falcucci. 2018. Relazione diagnostica sull'iscrizione della Cappella Turbolo, Santa Maria la Nova, Napoli. Unpublished diagnostic report. Analysis based on UV-induced fluorescence, infrared reflectography and pigment sampling.

Ramon Ferrer-i Cancho and Brita Elvevaåg. 2010. Random texts do not exhibit the real Zipf's law-like rank distribution. *PLOS ONE*, 5(3):e9411.

William F. Friedman. 1922. The index of coincidence and its applications in cryptanalysis. *Riverbank Publication*, 22.

Karel Kučera, Anna Řehořková, and Martin Stluka. 2015. DIAKORP: Diachronic corpus of Czech, version 6.

Michal Křen. 2020. Czech National Corpus in 2020: Recent Developments and Future Outlook. In *Proceedings of the 8th Workshop on Challenges in the Management of Large Corpora*, pages 52–57. European Language Ressources Association.

Beáta Megyesi, Alicia Fornés, Mihály Héder, Raphaela Heil, Benedek Láng, Nils Kopal, Rune Rattenborg, and Michelle Waldispühl. 2025. Decipherment of historical manuscripts with unknown or rare writings: The DESCRYPT project. In *Proceedings of the 8th International Conference on Historical Cryptology, HistoCrypt 2025*, pages 101–105, Poznan, Poland. Tartu University Library.

Cosimo Palma, Yll Rugova, and Paolo Bonavoglia. 2025. A new attack on the mysterious inscription of Santa Maria La Nova. In *Proceedings of the 8th International Conference on Historical Cryptology (HistoCrypt 2025)*, pages 106–110, Poznan, Poland. Tartu University Library.

Cosimo Palma. 2023. Encrypted epigraphy – the case of a mysterious inscription in the Neapolitan church of Santa Maria La Nova. In *Proceedings of the 6th International Conference on Historical Cryptology, HistoCrypt 2023*, pages 139–147. Linköping University Electronic Press.

Eva Pettersson and Beáta Megyesi. 2018. The HistCorp collection of historical corpora and resources. In *Proceedings of the Digital Humanities in the Nordic Countries 3rd Conference*, volume 2084 of *CEUR Workshop Proceedings*, pages 306–320, Helsinki, Finland. CEUR-WS.org.

C. E. Shannon. 1951. Prediction and entropy of printed English. *Bell System Technical Journal*, 30(1):50–64.

George Kingsley Zipf. 1949. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Addison-Wesley, Cambridge, MA.

A AI-powered segmentation and translation: Reading 1 (“Hospodiný”)

L.	Raw output	Segmented Medieval Czech	English translation
1	hospodinymsve**ms**vice	Hospodin **m**ým **mn**ě více	The Lord to me mine more
2	a**e**iveduchov**v**zklade**v**	a i ve duch**u** v **zá**kladě	and also in the spirit in the foundation
3	**a**teri**z**tens**i**tvae**i**est	-těři- ten- -stv**í** **j**est	this status is
4	vinne**it**s**s**ve**em**stan**t**i	vinně- své- stání	guilty, my standing
5	nim**i**roze**s**sii**esny**mil	-ním roz- s **ním** mil-	with him, with him in grace
6	izet**s**ei**i**sevsezi**v**e**iu**	-ize sě i se vše živ-	that all that lives
7	m**s**v**u**dusisme**t**i**e**zsv**ez**	m- v du**chu** s m**n**ě též své-	in the spirit with me also his own
8	smil**s**t**s**e**s**timmsavezm	s mil**o**stí **s**- m**n**ě **s**ě **v** m-	with grace to me in the…
9	issls**m**si**s**v**e**ne**a**v**i**est	-íš- **mn**ě si vně věst	is known outside to lead
10	ze**ai**setajtesamismet	že sě tajte sami s m**n**ě t-	that they hide themselves with me
11	**ie**z**i**sme**z**evs**im**eseivs	-ěž s m**n**ě vše sě i vš-	with me all and all
12	e**e**sestn**z**a**i**vseenie**e**s	-é **j**est na vše **n**ení **s**ě	it is over all, is not
13	se**e**zna**s**ve**s**ne**m**ime**s**ed	sě zná vše **v** m**é**m sě d-	all is known in my…
14	**in**e**a**vinnasivmstvta**e**	vinná si v m**ém** -stv**í** ta-	guilty in my status
15	**n**tilen**i**se**d**siv**e**se**a**mn	-ti lene si v sě mn**ě**	rests in me
16	senia**n**nastnisvelase	sení na st**á**ní s velase-	standing with the greatness
17	nilnesv**e**siniasvelne	-nil nes- v síni a s vel- ne-	carrying in the hall and with great
18	sea**n**sies**l**evma**e**nsti**t**	sě **v** sí**ni** v m**é**m -stv**í**-	in the hall in my status
19	ets**nl**ei**easie**nevm**i**ne	-et sě **m**ně v **j**mě-	to have the name
20	t**n**siase**d**mtitnsv**v**ate	-t s**ě** **mn**ě m**n**íti **v** svaté	to think in the holy
21	sinela**ns**stisevtves**n**	sín**i** **n**a st**á**ní sě v tvé s-	hall upon the standing in your
22	**e**navnia**e**naen**s**en**ii**ts	-na v ní a na **sě** není s-	in it and on itself is not
23	**e**net**s**etanem**s**meitesl	-né té **n**a **m**ně **j**mě**j**te sl-	have on me…
24	v**n**iesanetvesme**t**evsi	vně s**ě** n**a** tvé s m**n**ě v sí-	outside on your with me in the hall
25	amsesa**tn**enznesszi**as**	-ám sě s- **j**enž nes**e** s **j**a-	I am he who carries with
26	ilntva**enut**i**nt**smzasv	**s**ilno**u** tv**o**u **v**- s mz**dou** sv-	strong your… with reward
27	**e**nabojichsevzites**e**n	na bojích sě v **v**ítě**z**n-	in the battles in the victory
28	stv**upr**ivets**ans**lnesa	-ství v- -ěts- s nesa	status… carrying
29	**r**skychusesinei**aeev**z	-ských u sě **v** sí**n**i i v-	in the hall and
30	**es**a**s**ke**i**ea**s**z**s**enadlit	… sě na dlit-	on the delay
31	varisini**e**snymi**etiaz**	vaří **v** síni s n**ím** i-	in the hall with him
32	sei**a**sete**zn**iechuprot	sě i- -ě- **hř**iechu prot**o**-	for the sin, therefore
33	o**z**enesi**te**si**t**imi**t**ibl	-o nes- **s**ě ti- mi- bl-	carrying…
34	is**z**me**an**m**ee**z**bychhrie	-íš mně **že** bych hřie-	to me that I would sin
35	si**s**mi**iz**ese**ne**zleskyn	-ši m- **s**ě sě z lesk**u** n-	from the shine
36	a**s**ie**a**mae**es**ieses**aiel**	a sě m**n**ě sě sě s-	and to me…
37	**a**en**n**imsetiseprotoze	-ením sě ti sě protože	because
38	senesemnnzetaemnost	sě nese mn**ě** že ta**j**emnost	it carries to me that the mystery
39	**e**ivse	i vše	and all

Table 9: Complete segmentation and translation of Reading 1 (“Hospodiný”). Dashes at line beginnings indicate word continuations from the previous line. In the raw-output column, **bold** marks letters dropped during segmentation (present in the raw AZdecrypt output but absent from the aligned Old Czech reading); in the segmented Old Czech column, **bold** marks letters differing from the raw output at the aligned position. The large fraction of bolded letters reflects the high sampling temperature deliberately used here: as one of the earliest readings of the inscription, an exploratory regime was preferred over a conservative one to probe the space of plausible Old Czech renderings, before later, longer AZdecrypt runs (e.g. Reading 2, “Litování”). Line numbering follows the AZdecrypt output and is not aligned with the input file shown in Figure 1.

B AI-powered segmentation and translation: Reading 2 (“Litovaní”)

R.	Raw output	Segmented Medieval Czech	English translation
1	it[ova]niussviznivnesa	**L**itování u **sě** v**y**zniv nesa **já**	bearing a confession of repentance; I
2	**ia**svtis**m**zai**vl**nstiass**v**e	v tís**ni** za **milo**sti as ve **vó**li	in anguish cry out for mercy; in
3	**z**v[**idel**liby]sesli	b**y kd**y sě slí**bil tě pro**tože	the will, would he ever promise
4	i**iv**[**y**roze**nst**]v**i**isivd[**ov**]	víš i v d**á**vání **i** ti těli s	thee, for thou knowest in giving, and
5	vani**ti**i[t]el[ismy]slis**v**ci**v**ia	myslí s**úc**í an ta úsilno val	to the body, with the mind’s candle; that
6	**n**ntau**l**s**e**iln[o]valsvicenen	svíce není **ú**s**ilné** slně silným	one rolled the candle; it is not
7	iu**e**ilnslne**v**sil[ny]maenie	a e nie ne ná **sě** ve **vše** své	hard for the strong; nor hates
8	nenae**si**veve**s**svevecisveav**i**	věci své a v úle a v **s**at st	itself in all its own; own things, the will
9	**a**u**i**lea**u**vat**niesanaipnian**	na s a ži **sě** i n v **n**aučen**í** a	and in [unclear]
10	**inecuei**stnasazi**ai**n**sv**aa	slzě i ei vání s mieti mžen **sě**	strives also in teaching,
11	ucen**n**aslz**a**eieivanismie	tělěnu sě i **mě**l**i** li mi sáhls	and tears and grace in having;
12	timzens**i**telenuseimei	i ti na sě **mě** s**n**aží emženu	blinking to the body, they too had; whether
13	limisa[h]lsitina**e**semna[**s**]i	s**víc** tělesně žil má spál**ení**	thou touched me, they strive for me; having
14	emzenusc**iv**telesnezilma	měnil má niem s ni**mi** li tě	lived more carnally; my burning
15	spalimi**en**tilmaniemsni**en**	sě a m**ilo**s**ti**vě n m s le **i** a	he changed; he holds it; with them, whether
16	liteseaml[o]is**eeu**nmsleas	sě nie s měne m z s **v sí**n**i**	thee, and mercifully, in clarity
17	eniesmene[m]zsi**vc**siseni[**h**]**eei**	**Hospodin t**i sě níže v **n**ěm**ž** lež	with memory; in the hall the Lord
18	[**o**]z**ntis**eve**ni**me**l**ezsn**le**i**nat	s ní**m** na tě te**m** n měz sět mie	stoops to thee; in whom thou liest, upon thee
19	etenmezsetmieez**see**mleci	**sě** ze mlecí i e na že sě ne	in dark, it dims for me, from grinding;
20	ienazesenevsemsa**e**lea**sie**a**n**	všem sá le a e a sě e t sie e	and that, not to all in the hall;
21	**ltmienseeaustins**aseetsi	ne l sě vzsím učí snu i e ni	and themselves [unclear];
22	eenelsevzsimucisn**l**uien	l žet sě až je **ež** mě n**e** s**nad**	I rise; to teach the dream,
23	ilzetseaz[j]e**si**mens**e**nes	ne snáze **potk**á svém snu s ie	to lie until it come; perhaps no ease
24	naze[t**ehd**a]s[vem]snusi**ee**	ním **sě** s **světl** os**t**i s věv všem	meets me; in my dream
25	nimss[o**liby**]s[**e**v]**s**evvsemn	n**alé**zalo **sě** ti a spěšně ni ma	with light, with all that was found;
26	lza**se**[lo]s[tia]s[pe]snenima	ie a s nu **a** ale s nie sie e	to thee and swiftly…
27	ie**e**asnualesn[i]esieen	nu na krásli **i** otev**řeš o**či e	and now; but with her [the soul]
28	una[kra]sli[o]te[v]**s**[e]s**ni**ci	na hříeše **ech** usneš ie ně ve	in beauty; thou shalt open thine eyes
29	en[ahri]ese[**jd**]usnesienev	**mí**sto **sr [milosr]denství** sě	to the sins; thou shalt fall asleep
30	e**u**m[**upo**]s[to**by**]s[e**za**]**su**azan	**v**ázání vání **u** v **šík**ovi sě sílu	in the place of mercy, bond of grace;
31	ivanivs[**cie**]vis[es]ilue**ee**	e e mě a a st **i**ch **pros**iteli ne	striving for strength,
32	meaas[t**oby**ch]s[**ev**]elinei	i i na s na **a** za úsi**lní sě** z	…me; of those who pray, not
33	inasnazau**l**si[**kra**]szesli**e**e	**sě** s lie e e e sě nie s ně ně	upon us; and for the diligent…
34	eeeseni[e]snene**i**alsseezmi	a**l**e s sě e z mi e na sě i ve	not with him but with himself, from me;
35	enaseive[ho]s[podu]s[i]svemini	Hospodu si své **e** mi nie m s mu	and in the Lord his own; to him
36	emsmuivseaesz[adu]siitsne	i vše a e s žádu **ú**sící t s ně	and all with desire, desiring from them

Table 10: Complete segmentation and translation of Reading 2 (“Litování”), with continuous-flow segmentation performed through *Claude Opus 4.6*. In the **Raw output** column the per-line decryption is reproduced verbatim, position-aligned to the inscription (lines 3–38): **bold** letters mark divergences from the reading, square brackets enclose positions where text is unreadable (as displayed in Figure 3). The rows in fuchsia represent the decryption of lines 15–18 (see Figure 1). The **Segmented Medieval Czech** column gives the fixed Old Czech reading flowed evenly across the rows (it does *not* mirror the raw-output line breaks); here **bold** marks the characters added or modified with respect to the raw output. The English column glosses the reading token by token.

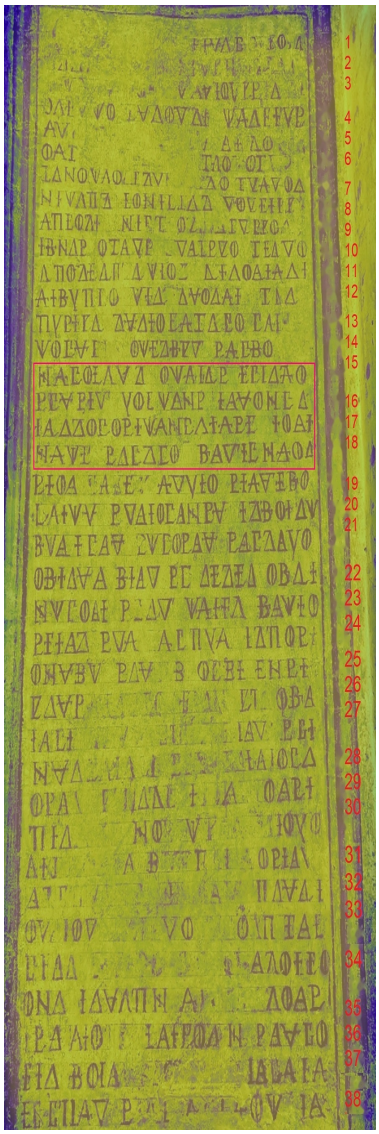
C Human validation of Reading 2 (“Litovaní”) automatic segmentation

For a validation of the AI-supported segmentation and translation, we have consulted Dr. Pavol Zajac, a researcher at the Slovak University of Technology in Bratislava, expert in historical cryptography and familiar with the Czech language. He confirmed that several text segments were indeed readable as Old Czech while expressing reservations about the overall coherence of the text and the attempted translation. With his permission, we include his annotated commentary in this appendix.

R.	Raw output	Segmented Medieval Czech	English translation
1	litovaniusevyznivnesa	**Litovániu sě vyzniv nesa**	*Litovániu*: repentance [dative]; *sě vyzniv*: possibly malformed grammar, "final stage of a sound or process, to taper off"; *nesa*: unknown, possibly related to carrying
2	iavtisnizamilostias	**já v tísni za milosti** (continues l. 3)	*já*: I; *v tísni*: in anguish; *za milosti*: for grace (or "in grace")
3	vevolibykdyseslibil	**[a s]vé vóli by kdy sě slíbil**	*a své vóli*: and my will; *by kdy sě slíbil*: would ever have promised myself [to thee, l. 4]
4	teprotozevisivdavani	**tě, protože víš i v dávání**	thee, because thou knowest also in the giving
5	ititelismyslisvcici	**i ti těli s myslí súcí cí**	*i ti těli*: and those bodies; *s myslí súcí*: with mind sound; *cí*: probably runs into l. 6
6	antauseilnovalsvice	**[cí]antauseilnoval svíce**	*svíce*: candles (may attach to neighbouring line)
7	neniusilneslnesilnym	**není úsilné slně silným**	*není*: is not; *úsilně*: related to working hard; *slně*: not a real word, fragment of many Czech words; *silným*: [to] strong [dat. pl.] or [with] strong [instr. sg.]
8	aenienenaesivevessve	[unclear]	
9	vecisveaviauileavsat	[unclear]	
10	niesanaipnianinecuei	[unclear]	
11	stnasaziseinvnauceni	**[unclear] v naučení**	*v naučení*: in teaching
12	aslzaeieivanismieti	[unclear]	
13	mzensitelenuseimeili	[unclear]	
14	misahlsitinaesemnasi	**mi sáhls [unclear]**	*mi sáhls*: [you] touched me / mine
15	emzenuscivtelesnezil	**[-em] ženu [sciv] tělesně žil**	*ženu*: woman [acc.]; *[sciv]*: unknown, possibly "with whom"; *tělesně žil*: bodily lived
16	maspalimienilmaniem	**ma spálí [mienil?] [maniem]**	*ma spálí*: will burn me; *mienil*: wanted to
17	snimiliteseamilostive	**s nimi [líté] [se] a milostivé**	*s nimi*: with them; *líté*: cruel/furious; *se*: verbal particle, ungrammatical here; *a milostivé*: and merciful
18	nmsleiaseniesmenemzs	[unclear]	
19	vsinihospodintiseniz	**v síni Hospodin [tiseniz]**	*v síni Hospodin*: in the hall of the Lord (ungrammatical: should read *v síni Hospodině*)
20	vnemzlezsnimnatetem	**v němž lež s ním [natetem]**	*v němž*: in whom; *lež*: a lie; *s ním*: with him; *[natetem]*: unclear
21	nmezsetmiesezemleci	[unclear]	
22	ienazesenevsemsaelea	[unclear]	
23	sieanltmiensemaustin	[unclear]	
24	easeetsieenelsevzsim	[unclear]	
25	ucisnuienilzetseazje	**učí snu [ienilzetseazje]**	*učí snu*: [he] teaches a dream; remainder unclear
26	ezmenesnadnesnazepotka	**[-e] změně snad nesnáze potká**	*změně*: a change [dat./loc.]; *snad*: perhaps; *nesnáze potká*: [he] will meet problems (if *s* from next line is read in: *nesnáze potkáš* – you will meet problems)
27	svemsnusieenimsesvetl	**[svemsnusieenim] se světl-**	*se světlosti své*: with its brightness (continues l. 28)
28	ostisvevvsemnalezalos	**-osti své [vvsemnalezalos]**	continuation of l. 27; remainder unclear
29	tiaspesnenimaieasnua	[unclear]	
30	alesniesieenunakrasli	[unclear]	
31	iotevsesocienahriese	[unclear]	
32	echusnesienevemistosr	[unclear]	
33	denstvisevazanivaniuv	[unclear]	
34	sikovisesilueemeaast	[unclear]	
35	ichprositelineiinasna	**[unclear] prositeli**	*prositeli*: "those who beg" in a church context, but contextually out of place
36	azaulsilniszeslieeeee	[unclear]	
37	eseniesneneialsseezmi	[unclear]	
38	enaseivehospodusisve	**[enasei] ve Hospodu si své**	*ve Hospodu*: "in [the] Lord", ungrammatical (should read *v Hospodu* or *v Hospodinu*, cf. l. 19); *si své*: verbal particle + "[his] own", verb missing
39	eminiemsmuivseaeszadu	[unclear]	
40	usicitsne	[unclear]	

Table 11: Line-by-line segmentation and translation of the candidate Old Czech plaintext from the AZdecrypt 40-rows output, not aligned to input lines, retouched to match more Czech words while avoiding polyphones. Bold tokens in the middle column are the reconstructed Old Czech reading. The input-text in fuchsia corresponds to the decryption of the transcription as performed in Figure 1.

D The mysterious Inscription of Santa Maria La Nova



123456789 cmDrc mkiAr 1
CEFGHIJKLMNOPRSU 2
nlXYZhip bndmibka r 3
irebli tdribfm lnrceba 4
db$%&’()*+ uefi 5
ino,-./:;<=soni io>? 6
orzilni@cfD[\] fi ownbir 7
zmlrpf civmToqf wibcmak 8
npcifm zmco if^_tqakiu 9
mszra iouba wbuoaoi ocrbi 10
rpifcqp qlmic gmriumnrm 11
nmsjpai lmq rdirnm orq 12
pjamkq fdrcicnorci tna 13
litdam ilcfskx antsi 14
vntieqbf ilnmqa ctmrni 15
atdamf jioluva mudivtr 16
enrfitiamlnztrmnat mium 17
vnlc artfti sufmt vuir 18
amiu tukc{d nbjei amnBcsi 19
Tdmld abumituvad mfsimrl 20
slumtud altiaud autfuji 21
ismrdu smub at rcfcr isqm 22
vltium atqx lnmcr sulei 23
acmuf abd ntpjn mfpiam 24
ivdsl ard|s itsm czae 25
krda}˜¢£¤ ¥§© kp isu 26
mnte «¬®°± óô ubo atm 27
vdreañµ ¶ïð »¼ mumitr 28
iaur kprfa m½u iuam 29
pmq¾¿À viÁlaÂ Ã miji 30
umÄÅÆÇu sÈÉp mu iamub 31
uptÊËÌ ÍÎÏÐ ÑÒÓ prdqm 32
ibrmib ÔÕÖbi ×Øifpcua 33
tcrr ÙÚÛÜÝÞ ßàufimci 34
izr mrdrpv náâã da fiua 35
cau umiä mumairz audti 36
cmr siouåæ çèéê ë outnmn 37
ct tpnb arc dìíî nil mu 38

Figure 3: On the left, the artifact under investigation, located in the Turbolo Chapel of the Neapolitan Church of Santa Maria La Nova. False-color rendering applied to enhance contrast, which highlights deleted characters in the background, corroborating the *scriptio continua* assumption. In the center, the full transcription of the inscription. On the right, the machine-readable input for AZdecrypt. First two lines, in italic, were removed from the input file due to almost complete illegibility. In fuchsia, lines transcribed as in Figure 1.

Enigma-Fusion: Connecting Digital Twin and 3D-Printed Reconstruction

Joanna Strobel
Friedrich-Alexander-Universität Erlangen-Nürnberg
joanna.strobel@fau.de

Felix Schmutterer
Friedrich-Alexander-Universität Erlangen-Nürnberg
felix.schmutterer@fau.de

Noah Lewis
Friedrich-Alexander-Universität Erlangen-Nürnberg
noah.lewis@fau.de

Abstract

This paper presents the design and realization of a fully functional, 3D-printed replica of the Enigma cipher machine, coupled with an interactive digital twin via a hardware–software interface. Beyond the acquisition of practical and transferable skills, the project enabled student-participants to develop a thorough understanding of the machine's mechanical operation, encryption principles, and historical relevance. The combined physical and digital system supports an illustrative and transparent demonstration of the Enigma's internal processes, making its cryptographic functionality accessible and comprehensible to a broad audience.

1 Project Framework and Conditions

The Enigma reconstruction project was carried out as a student project as part of a degree programme in computer science. The seminar component focused on the machine's historical role, as well as on the fundamentals of its cryptographic operation.

It required each student to research and present on core topics, including the Enigma's mechanical design, the principles of encryption and decryption it implements, historical context and significance, and, of course, Allied cryptanalysis efforts — most notably the work carried out at Bletchley Park. These presentations formed the academic groundwork for the practical phase (Bauer, 2000; Welchman, 2023; Bruderer, 2020).

After the seminar component, the practical one followed, which consisted of several parts. On one hand, the participants built a 3D-printed, fully working Enigma I reconstruction. On the other hand, a digital twin of the Enigma machine, EnigmaTwin, was built. Finally, the digital twin was connected to the 3D-printed Enigma machine via an interface enabling the keys from the hardware version as input to its digital twin.

An overall goal of the project was to present the final result to a wider audience in the form of a presentation and demonstration at a local science event.

2 Educational Goals and Academic Focus

A central objective was to enable the participants to develop a deep conceptual understanding of encryption mechanisms, using the Enigma machine as a historically significant and technically rich example. By studying its rotor-based cipher system, plugboard configuration, and operational procedures, the students explored how electromechanical encryption devices embody fundamental principles that still influence modern cryptography (Bauer, 2000; Bruderer, 2020).

Equally important was the historical perspective. Through their seminar work, the students examined the Enigma's decisive role in World War II, its impact on secure military communication, and the extraordinary cryptanalytic efforts at Bletchley Park that contributed to the Allied victory. This dual focus on technical function and historical consequence encouraged the participants to understand encryption not merely as a mathematical problem, but as a phenomenon with profound cultural and geopolitical implications (Welchman, 2023).

A second major objective involved strengthening practical, hands-on skills. Working with contemporary technologies such as 3D-printing introduced the students to digital fabrication processes, materials selection, tolerances, and iterative prototyping. They also practiced basic electronics and wiring techniques while assembling a functional machine. These activities required continuous experimentation, troubleshooting, and collab-

orative problem-solving — core competencies in computer science and engineering education.

Finally, the project was intentionally structured to foster teamwork, communication, and analytical thinking. From coordinating research for seminar presentations to planning the construction workflow, the students were required to collaborate, negotiate design choices, and reflect on their methods. In combining historical research, cryptographic theory, and modern fabrication, the project encouraged them to engage with the broader intersection of history, technology, and information security — and to appreciate how these domains inform one another (Hubwieser, 2007).

3 Reconstructed Enigma

The reconstructed Enigma is a physical reconstruction of an Enigma machine using additive manufacturing (3D-printing) for the case and selected mechanical parts. To enable correct operation, electrical functionality is required, which includes wiring and battery power, so that it can operate independently of laboratory mains power.

The build follows previous reconstructions inspired by the design of Hochschule der Medien Stuttgart which were then updated with the help of materials provided by Deutsches Museum, Munich (Wiest, 2021; Deutsches Museum, 2022). It was modified to suit available materials, safety considerations, historical accuracy, and learning objectives.

The main step is the fabrication of mechanical components of the Enigma replica through 3D-printing. The complete workflow includes file preparation, material selection and calibration. During the process, refinement of parts is an important step to ensure proper fit and durability.

In parallel, the electrical system is built. It includes the connection of micro-switches, LEDs, and rotors using wires. Additionally, the individual wiring links between letters in the rotors and the reflector are implemented. This setup creates a current circuit that can reproduce the machine's lampboard output. Verifying signal flow and debugging the circuitry forms an essential part of the later construction process.[1]

Once assembled, shown in Figure 1, the replica achieves its intended functionality: pressing a key illuminates the corresponding substituted letter, accurately reflecting the internal wiring logic of the Enigma's cryptographic mechanism.

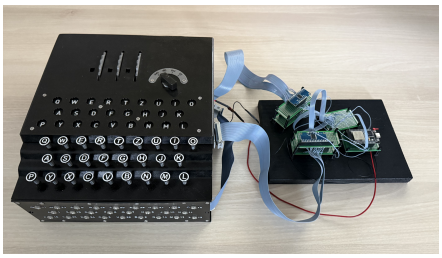


Figure 1: 3D-printed replica of the Enigma machine. On the right side is the hardware part of the interface between the replica and EnigmaTwin.

4 EnigmaTwin

EnigmaTwin is an interactive application representing a digital twin of an Enigma machine. It includes a 3D visualization with a functioning logic. This software allows viewing the machine from different angles, changing its transparency, and observing the different parts and mechanism during usage. After setting configurations, messages can be encoded or decoded using the keyboard as input. As an additional feature, the path of the current can be visualized during the encoding of a letter, allowing a better understanding of the inner mechanisms. The ability to visualize current and mechanics in an interactive way is one of the strong contributions of this paper that existing simulators do not offer (Gillow, 2021; Enigma Machine Emulator, 2019; Palloks, 2025).

Logic. The logic is the core of the program, since it is the foundation for a correct message encryption. This subsection covers the most important parts of it, providing pedagogical transparency for a possible replication of the project.

We represent the letters $a, b, ..., z$ by the integers $0, 1, ..., 25$ defining a set $\mathscr{L}$. Addition and subtraction of a letter by an integer k is performed modulo 26. Let the plugboard p, the rotors r and the reflector u be the bijective functions:

$$p : \mathscr{L} \to \mathscr{L}; r_{i,n} : \mathscr{L} \to \mathscr{L}; u : \mathscr{L} \to \mathscr{L} \qquad (1)$$

$r_{i,n}$ represents the ith rotor that has a rotation $n \in \mathscr{L}$. Plugboard p is modeled as a pair-wise mapping table including the entire set $\mathscr{L}$. A letter can be linked either to itself or to a different letter, depending on the plugging of the plugboard cables.

[1] Before their participation, the students completed practical hands-on training in the university workshop, including soldering and the use of measurement instruments.

The rotation of the wiring n of a rotor $r_{i,n}$ combines the rotation of the digit ring, the so called ring setting $\omega \in \mathscr{L}$, and the rotation of the whole rotor $\psi \in \mathscr{L}$:

$$n = \psi - \omega \tag{2}$$

ψ is incremented every time that the rotor is rotated. The frequency of rotor rotations is determined by the rotor's position. The right rotor turns at every encryption of a new letter. The two other rotors rotate when the rotation ψ of their right neighbor equals specific values (Smart, 2016). At these rotation values, the notch of those neighbors is facing towards the back of the Enigma, allowing the pawls to rotate the rotor and its neighbor. This also realizes the double-stepping mechanism of the middle rotor.

Every rotor $r_{i,n}$ has its table t_i with unique encryption, where its values correspond to the wiring of the rotor (Bauer, 2000). Using t_i, the function of a rotor $r_{i,n}$ that corresponds to the encryption of a letter λ in the forward direction, coming from the plugboard, is defined as:

$$r_{i,n}(\lambda) = t_i(\lambda + n) - n \tag{3}$$

To calculate the output letter backwards, coming from the reflector, Equation 3 is adapted to $r^b_{i,n}$:

$$r^b_{i,n}(\lambda) = \{j - n | t_i(j) = \lambda + n\} \tag{4}$$

where $j \in \mathscr{L}$.

The reflector u is implemented as a table of encryption combinations resembling UKW-B (Bauer, 2000).

All in all, the function E that represents the whole encryption process of the Enigma is:

$$E(\lambda) = p(r^b_{\alpha,n_0}(r^b_{\beta,n_1}(r^b_{\gamma,n_2}(u(r_{\gamma,n_2}(r_{\beta,n_1}(r_{\alpha,n_0}(p(\lambda)))))))))) \tag{5}$$

where λ is the input letter, and α, β and γ are three different rotors each having a rotation n_0, n_1 and n_2.

Creation and Assembly of parts. To create the digital twin, the stl files from the 3D-printed parts are incorporated. Missing parts, such as cables, screws, switches, and the battery, are designed in Blender. All parts are first assembled in Blender and then transferred to Unity for further processing. Further implementation details are provided in Section 6.

User Interface. The application also includes an interface for control. Users can zoom, rotate, and move around the Enigma using their mouse. The main view offers different features. There are two small windows showing the given input and the corresponding encryption. There are also buttons to reset the view and configurations of the Enigma. Furthermore, two check boxes enable connection to the 3D-printed Enigma or visualization of the current flow. Two sliders change the transparency of the Enigma and the speed of the current visualization. Additionally, one button opens a menu to set rotors α, β and γ, their configurations ψ and ω, and plugboard-cable combinations. Changing these settings triggers the creation of plugboard cables or animations around the rotors, as well as configurations in the logic are adjusted.

Animations. The application features several animations that visualize the operation of the Enigma. One animation represents the illumination of the encoded letter by simulating a glowing lamp. Another animation depicts the mechanical interaction of the keyboard: when a key is pressed, the corresponding button moves downward, causing the actuator bar to rotate around its central axis. This rotation lifts the pawls, reflecting the mechanical response of the original machine. These animations are implemented using Unity's built-in animation system. In addition, the rotors are visualized through a set of dedicated animations. When a rotor is assigned to a specific position within the encryption pipeline, it is placed accordingly. Ring settings are represented by rotating the digit ring by the required offset ω. Rotating the entire rotor by a defined angle simulates the stepping mechanism.

Transparency. The cover can be made transparent with a slider. To achieve transparency, the opacity value α of the materials is decreased using Unity-specific elements called MaterialPropertyBlocks.

Plugboard cables. Setting a letter combination for the plugboard triggers the creation of a new cable. Using the positions of the two letters as start and end, a cubic Bezier curve is calculated. Subsequently, a circle is determined at discrete points of the Bezier curve using the vector between the point before and after as the normal vector. The points of all circles are then connected with triangles to form a mesh.

Visualization of the current. Inspired by an explanation video (Owen, 2021), the current can be visualized. During the visualization, all cables are set to transparent, and many parts of the Enigma have a high transparency for a better visualization. While the logic encrypts the letter, it stores the names of the different, necessary cables. Starting from the battery, the visualization uses these cables to trace each one in turn, showing the current's path. The coloring of the path is completed by iterating over the texture coordinates of the cables, and changing the colors directly in the fragment shader. The tip of the path is colored in light blue, while the remaining part of the already visualized path is colored in magenta. When the visualization passes the light bulb, it turns on. An additional blooming effect is added so that the path is better visible. This visualization allows the user to clearly see the path of the current, especially in the rotors, as shown in Figure 2.

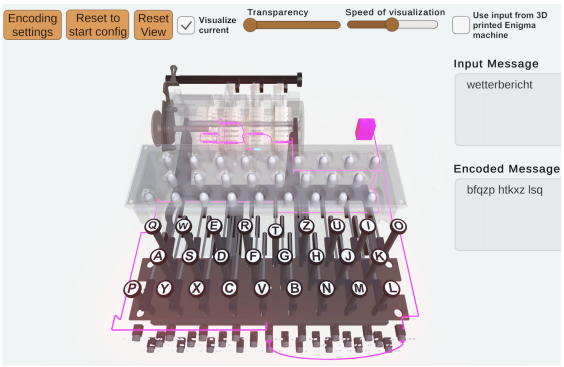


Figure 2: EnigmaTwin with current flow visualization when a key is pressed and a plugboard cable is used. The cover is invisible.

5 Connection between Software and Hardware

Another major contribution of this paper is the connection between the 3D-printed Enigma and EnigmaTwin. To figure out the pressed key, the current has to be measured at two different positions: at the cables connecting switches and plugboard, and switches and light bulbs. Between the switches and the plugboard, current can only be measured at the cables of the input and output letter. This detail leaves only two options as the input key. When the current in the cables leading to the light bulb is measured, only the cable of the output letter shows current flow. Combining the results by a logical 'and not' determines the input.

Realization of Interface. The required hardware components are four 16:1 multiplexers and one microcontroller. The multiplexers are connected to the 52 cables from the two places where current is measured as input. Steering inputs as well as outputs are directly linked to the microcontroller. The microcontroller iterates over the inputs from the multiplexers to collect the information on which cable current flows. The logical combination of the cables is then computed by the microcontroller. If it identifies a key that is pressed, it sends this information to EnigmaTwin via WiFi. The application processes the information in real-time and uses it as new input.

6 Implementation Details

The 3D-printed parts were printed in a BambuLab X1 Carbon printer with PLA filament. The application EnigmaTwin was developed in Unity using Version 2022.3.62f2 (Unity Technologies, 2022) and missing parts were created in Blender Version 4.4 (Blender Foundation, 2024). For the interface, Arduino IDE Version 2.3.6 was used to implement the program that was transferred to the microcontroller ESP32-WROOM-32 (Arduino, 2025).

7 Conclusion

After successful completion, the outcome of this project is a fully working 3D-printed Enigma machine, which is connected via an interface with its digital twin, the EnigmaTwin. This connection allows pressing a key on the hardware Enigma and using that letter as input for the digital twin. Consequently, EnigmaTwin is an application that visualizes the inner workings and the current flow of an Enigma machine during encryption.

Among achieving different soft and practical skills, this project allowed the participants to get a deep understanding of the functionality and history of the machine. Using this result, an audience can interactively explore the encryption process from inside and outside the machine, offering an illustrative way to explain the Enigma. To ensure educational impact, EnigmaTwin has already been integrated into the seminar "History of Computing" at FAU as well as into guided tours of the ISER (Informatik Sammlung Erlangen).

Acknowledgments

The authors would like to thank Simon Wiest (HDM) for sharing his experience and results.

References

Arduino. 2025. *Arduino IDE*. Version 2.3.6. URL: https://www.arduino.cc/en/software.

Friedrich L. Bauer. 2000. *Entzifferte Geheimnisse: Methoden und Maximen der Kryptologie*. pages 112-120, 3rd rev. and ext. edition. Springer, Berlin and Heidelberg.

Blender Foundation. 2024. *Blender*. Version 4.4. URL: https://www.blender.org/.

Herbert Bruderer. 2020. *Meilensteine der Rechentechnik. Band 2: Erfindung des Computers, Rechnerbau in Europa, Weltweite Entwicklungen, Zweisprachiges Fachwörterbuch, Bibliographie*. pages 126-132, 3 edition. De Gruyter, Oldenburg.

Deutsches Museum. 2022. *Durchleuchtet: Die Geheimnisse der Chiffriermaschinen*. URL: https://www.deutsches-museum.de/museum/aktuell/durchleuchtet-die-geheimnisse-der-chiffriermaschinen.

Enigma Machine Emulator. 2019. *Enigma Machine Emulator*. URL: https://www.101computing.net/enigma-machine - emulator/.

Martin Gillow. 2021. *Virtual Enigma*. URL: https://enigma.virtualcolossus.co.uk/.

Peter Hubwieser. 2007. *Didaktik der Informatik. Grundlagen, Konzepte, Beispiele*. pages 15-19, 67-71, 3 edition. Springer, Berlin and Heidelberg.

Jared Owen. 2021. *How did the Enigma Machine work?* YouTube. URL: https://www.youtube.com/watch?v=ybkkiGtJmkM.

Daniel Palloks. 2025. *Enigma-Simulation in Javascript/HTML*. URL: https://people.physik.hu-berlin.de/~palloks/js/enigma/.

Nigel P. Smart. 2016. *Cryptography Made Simple*. page 136. Springer, Cham.

Unity Technologies. 2022. *Unity*. Version 2022.3.62f2. URL: https://unity.com/.

Gordon Welchman. 2023. *The Hut Six Story: Breaking the Enigma Codes*. pages 7-28, 195-252. M&M Baldwin, Cleobury Mortimer.

Simon Wiest. 2021. ENIGMA R.D.E.: Die berühmteste Chiffriermaschine der Welt für den Schulunterricht – hergestellt aus dem 3D-Drucker. *Datenschutz und Datensicherheit - DuD*, 45:298–302.