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)

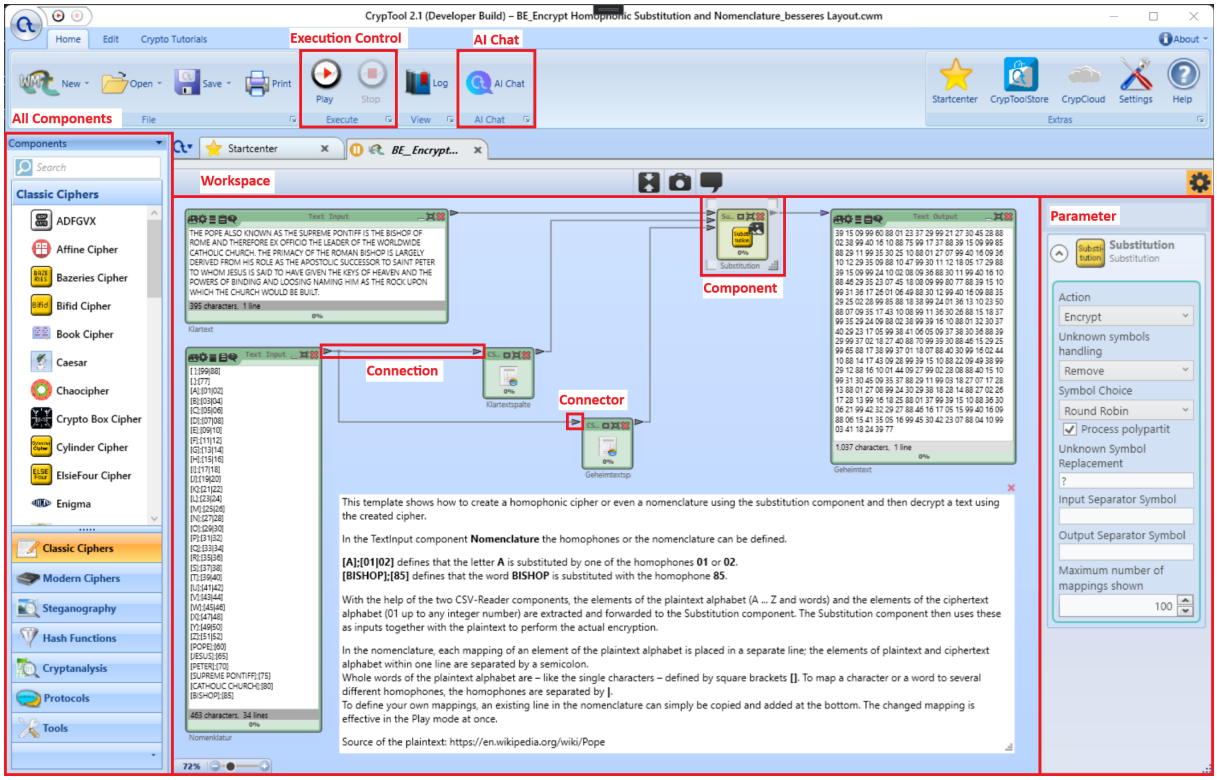

Figure 1: A typical CrypTool 2 workspace. Relevant user interface (UI) elements are marked in red.

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-

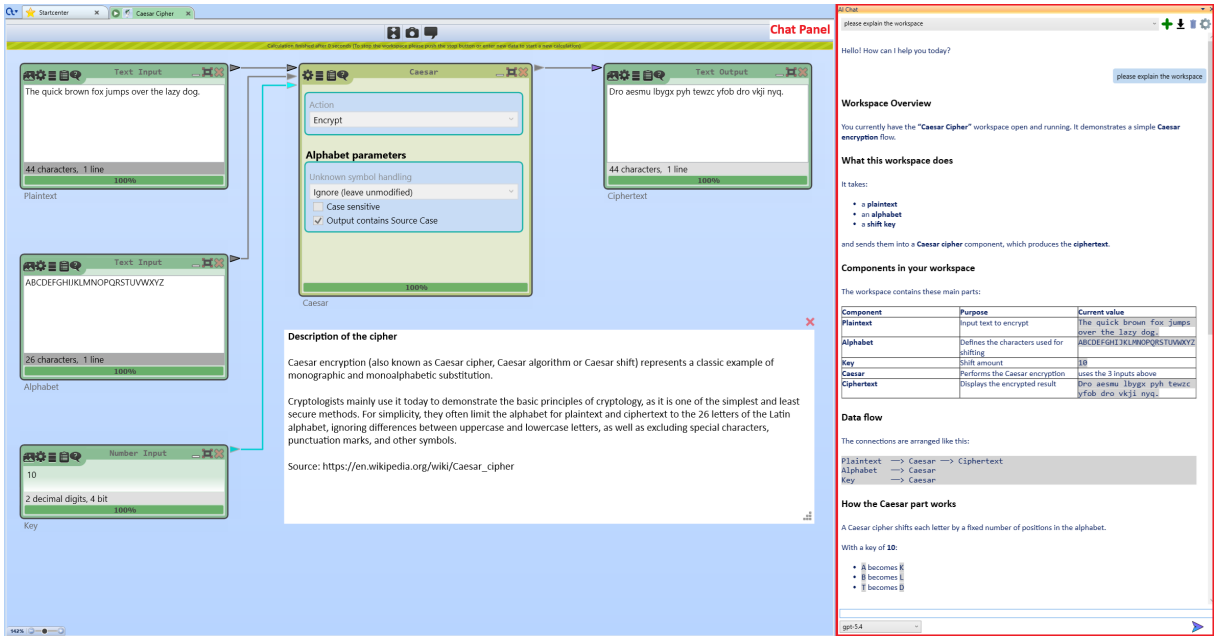

Figure 2: CrypLLM explaining an existing Caesar cipher workspace in CT2.

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.

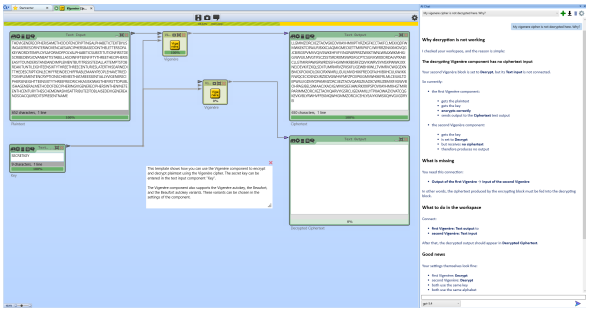

Figure 3: CrypLLM troubleshooting a broken Vigenère workflow.

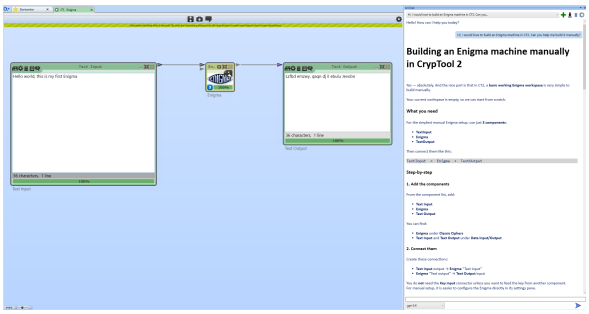

Figure 4: Scenario 3: CrypLLM guiding the construction of a minimal Enigma workflow from an empty workspace in CT2.

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.