UNIVERSITY OF TARTU

Faculty of Science and Technology

Institute of Technology

Toghrul Gulmammadov

Post-Quantum Secure Cloud Computation Over Encrypted Data

Bachelor's Thesis (12 ECTS)

Curriculum Science and Technology

Supervisor(s):

MSc Valeh Farzaliyev

Tartu 2024

Post-Quantum Secure Cloud Computation Over Encrypted Data

Abstract:

In recent years, cloud computing has gained significant popularity as businesses increasingly use it for data training purposes. However, this widespread adoption raises privacy concerns, particularly in light of regulations such as the GDPR. Additionally, there is an imminent need to prepare for the potential threats posed by quantum computers, which can easily break most current encryption methods. In response to these challenges, we have undertaken research on "Post-Quantum Secure Cloud Computation Over Encrypted Data" to address privacy issues in a manner resilient to quantum attacks. Our study provides a detailed proof of concept, using Zama's "Encrypted Health Prediction" tutorial as an example, to demonstrate the feasibility and practicality of creating SaaS applications that are secure against quantum threats and capable of training encrypted data in the cloud.

Keywords:

Cloud computing, post-quantum cryptography, fully homomorphic encryption

CERCS: P170 Computer science, numericalanalysis, systems, control

Post-kvantumi turvaliste pilvearvutuste kohta krüpteeritud andmete üle

Lühikokkuvõte: Viimastel aastatel on pilvandmetöötlus saavutanud märkimisväärset populaarsust, kuna ettevõtted kasutavad seda üha enam andmete koolitamiseks. Selline laialdane kasutuselevõtt tekitab siiski muret eraelu puutumatuse pärast, eriti selliste määruste nagu GDPR valguses. Lisaks tuleb peatselt valmistuda kvantarvutite potentsiaalseteks ohtudeks, mis võivad kergesti murda enamiku praeguste krüpteerimismeetodite. Vastuseks nendele väljakutsetele oleme alustanud teadusuuringuid „Post-kvantumi turvaliste pilvearvutuste kohta krüpteeritud andmete üle“, et käsitleda eraelu puutumatuse küsimusi kvantrünnakute suhtes vastupidaval viisil. Meie uuringus on esitatud üksikasjalik kontseptsiooni tõestus, kasutades näitena Zama „Krüpteeritud terviseprognoosi“ õpetust, et näidata, et on võimalik ja praktiline luua SaaS-rakendusi, mis on turvalised kvantohtude vastu ja suudavad koolitada krüpteeritud andmeid pilves.

Võtmesõnad:

Pilvtöötlus, postkvant-krüptograafia, täishomomorfne krüpteerimine

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine

TABLE OF CONTENTS

Unsolved issues

INTRODUCTION

Cloud Computing

Cloud computation, or cloud computing, is the delivery of multiple services via the Internet, from data storage through servers and databases to networking and even software. In other words, instead of companies owning their own computing infrastructure or data centers, they can rent access to anything from applications to storage from a cloud service provider. Probably one of the basic advantages of cloud computing is the ability to scale resources up and down with tremendous elasticity, providing a flexible and cost-effective way of managing IT infrastructure. Such elasticity allows handling changing workloads in an organization flexibly and ensures that an organization pays only for the resources used. Cloud computations are based on the principle of virtualization, that is, the division of one main server into many independent small servers, each running its operating system and application (Armbrust et al. 2010; Mell and Grance 2011).

Cloud computation has the capability to change the ballgame in information technology by providing scalable and on-demand access to computing resources. Several reasons make cloud computations important for business. As the fields do not need to make huge investments in physical hardware and its maintenance, they save significantly on costs. Secondly, cloud services start with the collaboration at the level of data and applications being available from any location with internet connectivity. This capability then generates more mobility for workers and boosts workforce productivity. Finally, cloud computation allows effective disaster recovery and business continuity to be done through robust backup solutions and data redundancy at diverse locations (Buyya et al. 2009). The agility and speed at which cloud services can be provisioned provide organizations with the possibility of innovation at a faster pace, and a response rate and ability to change that would never be possible in a traditional economy (Marinescu 2017).

Privacy is the most significant concern with respect to cloud computing as there is a greater possibility of disclosure and misuse of sensitive data from the client's organization to the third-party cloud. The significantly private information is transferred from client organization to a third-party cloud, and the sensitivity of information tends to be very high because clients do information confidential business, and it makes the clients rely on the security measure and practice of these providers only. Security risks associated with data transfer further include threats of unauthorized access, data breach, and lack of data isolation. The cloud providers, too, are a threat, among other reasons, through access to data in plain text and bring, for example, issues like insider threat and unauthorized data use (Subashini and Kavitha 2011). Moreover,

multi-tenancy may lead to the issue of an accidental data leak in case there is no isolation mechanism provided or an issue of cross-tenant data access problem in the server (Armbrust et al. 2010).

Added to that, general data protection regulations, such as GDPR, make this even more complex for cloud. Organizations need to ensure that cloud providers align with these regulations. These regulation needs strict control requirements in respect of personal data as well as access, erasure, and portability rights (Voigt and Bussche 2017). Regarding specific data centers' location, it means that data in some jurisdictions can be exposed to various standards that differ with government surveillance policies that may put the data into a conflict with local data laws (Pearson 2013). There is a need for organizations, therefore, to consider the policy on security, data management by various cloud providers, and their compliance with various regulations while evaluating them to mention a few.

Compliance with the GDPR (General Data Protection Regulation) matters not only in legal terms, but more so in trust and reputation. Lower bound fines reach as high up as 20 million euros or 4% of global annual turnover, whichever is higher. Going beyond the legal consequences of the regulation, the GDPR argues in favor of best practices related to the management of data that are fair and transpicuous when personal data management is concerned. This produces additional returns when it comes to building trust between the organization and its customers, returns that in this day and age of breaches of data and privacy, can no longer be taken for granted. Still, based on decision of European Commission[1] compliance with the principles of the GDPR may deliver better brand images of the organizations, as consumers today are gradually becoming aware of and concerned about their privacy rights (Voigt and Bussche 2017). Thus, it will be a competitive advantage in itself to implement practices compliant with the GDPR because it reassures people that their personal data is indeed secure and respected.

Cryptography

Cryptography is the science of securing communication and information through the use of mathematical techniques and algorithms. It deals with the conversion of readable data, known as plaintext, into an unreadable format, known as ciphertext, through the use of encryption methods. Therefore, it ensures that it is only possible for the recipient to decrypt and access the original information. Some of the important key concepts of cryptography include encryption, decryption, cryptographic keys, and cryptographic protocols. Cryptography includes various techniques,

[1] https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679

from symmetric-key cryptography (using the same key for both encryption and decryption) to asymmetric-key cryptography, which uses a pair of a public key and a private key for encryption and decryption, respectively (Katz and Lindell 2020).

Cryptography has an extremely strong relationship with how privacy issues find solutions. Cryptography develops a secure and private means of sending and storing sensitive information. It provides a foundation for various security protocols, such as Secure Sockets Layer (SSL) and Transport Layer Security (TLS), which are used to protect data transmitted over the Internet. With the help of cryptographic techniques, the data remains private—since it is encrypted, unauthorized access to the data is prevented, which otherwise may be vulnerable to cyber-attacks or data breaches. Additionally, cryptographic practices, such as digital signatures and hash functions, make sure that there is data integrity along with its authenticity, meaning that the data is in its original state without any tempering and is sent by the claimed person (A. J. Menezes, Oorschot, and Vanstone 2001; Schneier 1996). This becomes important for privacy in cyberspace and e-transactions, where data breaches and cyber-attacks are relatively more common nowadays.

RSA (Rivest, Shamir, and Adleman 1978) and ECDSA (Johnson, A. Menezes, and Vanstone 2001) are widely deployed cryptographic schemes that are currently used for securing communications. Quantum computers, which are machines based on the principles of quantum mechanics, have a significant advantage in solving these systems using Shor's algorithm (Shor 1994). Post-quantum cryptography is a new field focused on cryptographic algorithms that are effective against such attacks by a quantum computer. These algorithms are based on hard mathematical problems believed to resist both classical and quantum attacks, such as those in lattice-based cryptography (Ajtai 1996), hash-based cryptography (Merkle 1990), code-based (McEliece 1978), and multivariate polynomial cryptography (Patarin 1996).

The transition to post-quantum cryptography is crucial to safeguarding digital communications and data storage against the threats posed by quantum computers. By implementing post-quantum cryptographic algorithms, we can ensure that encrypted data remains secure, thereby protecting critical systems and sensitive information from future quantum attacks.

In 2009, Craig Gentry pioneered the first Fully Homomorphic Encryption (FHE) scheme, leveraging lattice-based cryptography as its foundational structure (Gentry 2009a). FHE is an advanced form of cryptography that allows computation over encrypted data without data being decrypted; thus, the data is kept secure and private even in use. FHE allows one to properly execute the algebraic operations of interest on ciphertexts, yielding the correct result when decrypted, which would have been obtained when applied to the plaintext.

FHE solves a few of today's most important privacy problems in a world full of data. First,

FHE offers solid data confidentiality during computation; hence, it reduces the connected risks of data breaches and unauthorized access. This is indeed important every time sensitive data has to be sent outside—e.g., for offsite computation—to a third-party cloud service. As a consequence, computation over encrypted data is going to be performed, maintaining both privacy and integrity of the data and increasing adherence to severe data protection regulations such as the GDPR.

1 LITERATURE REVIEW

Recent rapid development in quantum technology brings the security of classical cryptographic systems under severe threat. Since quantum computers have quite an ability to efficiently deal with complex mathematical problems underlying most encryption schemes, therein arises an apparent need for the development of post-quantum cryptographic techniques. The subsequent field of secure cloud computations over encrypted data that is post-quantum has been investigated by the literature review of this document in terms of challenges, solutions, and advancements.

1.1 Post-Quantum Cryptography

Rapid advent of quantum computing has posed a serious threat and, hence, led to the field of post-quantum cryptography research, with almost becoming the need of the hour. There again is a great rush to rediscover encryption strategies that will pose a quantum-proof resilience, being that the hint of quantum computing is placed as a boulder in the path of the already established security strength of conventional cryptographic systems. The recent paper from 2017, "Post-Quantum Cryptography" by Bernstein and Lange (Bernstein and Lange 2017), gives detailed insights into this area, describing the associated issues, latest breakthroughs, and developments regarding the securing of digital communications in an era powered by quantum. It evaluates the pros and cons of these approaches, focusing on criteria like security, efficiency, and quantum attack resilience. The authors further discuss the standardization process being conducted in the domain of post-quantum cryptography, as groups, including NIST, are actively assessing and picking quantum-resistant cryptographic algorithms that are going to be standardized. This process ensures that safe protocols are uniformly adopted and interoperably used in all applications. There again, the specific point at which the authors point is that there is a need for designing an encryption scheme which is, indeed, resistant to attacks—classical, cryptanalysis attacks, along with quantum ones. They discuss the conventional-post transition steps, indicating that a secure path for migrating to the new protocol is a requirement. It really changes the paradigm of using cloud computing to store, process, and access data. It does give rise to a security concern: after all, it is there that highly sensitive information comes to a third-party provider. To enable computations over encrypted data, two types of fully homomorphic encryption schemes have been thoroughly researched in the past: homomorphic encryption and fully homomorphic encryption (FHE). Basic foundations for privacy-preserving cloud computations are built using computationally heavy FHE schemes.

No citations in following subsections. Then remove? - Need to discuss

1.2 Privacy-preserving outsourced computation

One can securely outsource computations to the cloud while keeping data privacy satisfied. There exists several production-ready privacy-preserving computation frameworks based on Secure Multi-Party Computation (Yao 1982). For example, Sharemind, developed by Cybernetica[2], gathers and processes sensitive data of multiple owners (Bogdanov, Laur, and Willemson 2008). Similarly, Carbyne Stack (Becker et al. 2021) is a cloud-native solution ensures maximum security by offering end-to-end encryption for data both during transmission, while stored and in use. Carbyne Stack employs a fundamental client-server architecture to implement MPC, thereby eliminating the need to share data with a third-party intermediary. Other notable frameworks are MP-SPDZ (Keller 2020), Rosetta (Chen et al. 2020) and TF-Encrypted (Dahl et al. 2018) which supports Python interface for Machine Learning applications.

1.3 Case Studies and Applications

Real-world applications of post-quantum-secure cloud computing include secure data analysis in the healthcare domain, financial analytics, and confidential data sharing among organizations. Some of the cases to be shown put into perspective the applicability and effectiveness of the solutions proposed.

A secure framework for a multi-user and multi-owner cloud environment was presented in (Chandel, Yang, and Chakravarty 2020). According to the authors, security, integrity, and privacy of cloud data are the major threat to cloud deployment in a multi-user/multi-tenant cloud environment. They have also designed an algorithm to take care of security issues of the cloud environment and proposed/applied a new amalgamated algorithm of RSA and Ciphertext Policy-Identity Attribute-based Encryption to make the cloud secure. From their work, it was feasible to generate a framework where the Automated Certificate Authority creates two different keys -public and secretive to the owners and the users. Java was used to implement the proposed framework.

Forecast test of the proposed framework was carried out using standard metrics by comparing the metrics which will be produced by Anandand Perumal and Xue and Ren in the year 2019 from the proposed framework. When simulated for different data sizes, the performance of the above was found to be faster and efficient to the EECDH and I-CP-ABE algorithm. From the research, it was found that the proposed algorithm cannot be attacked by the man-in-the-middle. The authors have used the RSA cryptosystem in the model, but it is not quantum safe.

[2] https://cyber.ee

According to (Awan et al. 2020), assurance in data confidentiality and integrity of the users in cloud basically needs a security model that assures third-party prevention from unauthorized access and assured communication channel. The authors developed a security framework through which cloud users could assure privacy and accuracy in their cloud data. It offers users network usage, data storage, security, and privacy without taking help from the cloud provider. This model enabled authorized and verified users to access cloud data that was suggested to be encrypted through a subset of the AES algorithm. The iFogSim and CloudSim simulators were executed on an integrated development environment that was Eclipse to simulate the suggested model. The simulation results informed us that the suggested model reduces the usage of energy, network, and delay. Thus, the suggested framework enhances security, lowers the consumption of resources, and lowers latency while using cloud computing services. Weaknesses of the Figure study are key distribution issues that the AES cryptosystem possesses.

So far, (Noha, Omara, and Omran 2016) stated that cryptography was the best-known method for data security in a cloud environment. Besides, it recommended that the cryptographic services, used in any cloud environment, are supposed to maintain authorization, availability, confidentiality, integrity, and non-repudiation. Taking the security of data, it recommended using RSA, AES, and SHA256. The disadvantage of this mechanism is that it has a delay working. The RSA cryptosystems compute a significant part of the overhead, which was set because of the use of long keys, and they are breakable by quantum attack. One of the drawbacks of AES is key exchange with AES.

100 respondents carried out studies on the research of the privacy and security fears amid a cloud set in the context of smart campus security. According to the authors, blockchain technology solves the chief issue of expense of cloud computing. Conversely, they said that cloud computing is much cheaper; in turn, blockchain technology is more expensive when it is applied to a chain of objects. They further added that though the challenges of security and privacy are different for different types of businesses, it is safer to store Blockchain accounts and, at that, data integrity, data authentication, data availability, data location, data privacy, data confidentiality, data storage, backup, and recovery, are the chief problems. On the other hand, anonymity and security may be bettered in a cloud atmosphere with the help of Blockchain technology. Further, the problems were concluded when the authors said that the data integrity is compromised due to insufficient encryption, audit control, authentication, and authorization. (Gill et al. 2021)

1.4 Conclusion

Secure cloud computation over encrypted data is potentially one of the most important crossroads between post-quantum cryptography and cloud computing. The consistency and follow-up to the development of strong cryptographic techniques along with effective protocols and practical implementations, such as was reviewed by this literature survey, will manyfold improve this research field. Research efforts in this domain will continue to adapt in pace with changes in quantum technologies and to guarantee the confidentiality, integrity, and availability of sensitive data in cloud computing environments.

2 THE AIMS OF THE THESIS

The objective of this thesis is to research and contribute further to post-quantum cryptographic primitives in a framework, ensuring secure cloud computation over encrypted data. More specifically, regarding quantum computing, those threats and risks take a colossal character against the classical cryptographic systems, mainly because of their quantum-solvable properties towards the mathematical problems on which they are based. The project investigates the specific threats that current encryption schemes face because of quantum computers, research existing post-quantum cryptographic methods, and shows existed novel solutions for enhancing the security of cloud-based computations. More concretely, the present research accepts such a challenge with explaining the perspective of furthering post-quantum secure processing methods of the data in assuring the confidentiality, integrity, and availability of the sensitive information in cloud systems.

The aim of study is being the guide for creating the application which leads to using encrypted data on cloud services. Strict data protection regulations, such as the General Data Protection Regulation (GDPR) within the European Union and the Clarifying Lawful Overseas Use of Data (Cloud Act) in the United States, put strong emphasis on the importance of post-quantum secure cloud computation over encrypted data. Both frameworks put very high demands on data security and privacy, and cryptographic solutions must be developed in a way that guarantees their durability when exposed to possible quantum computer attacks in the future.

3 FHE in the Cloud

The cloud carries everything it offers from the cloud provider to either business organizations or private individuals. The four mainstream types of services in cloud computing are Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS), and Function as a Service (FaaS). All these service models bring in different levels of control, flexibility, and management for various IT and business requirements.

SaaS is a cloud computing model in which applications are hosted and made available for users over the internet. The SaaS model allows its users to access applications via a web browser without the need for purchasing and maintaining software on their local computer. The most common uses of SaaS appear in a variety of various applications, including email, CRM software, and ERP systems. It provides plenty of benefits such as ease of usage, accessibility, and cost-effectiveness—an enormous reason is that businesses are saved from controlling hardware and software infrastructures.

PaaS provides such an environment where customers can develop, run, and manage applications without managing the necessary infrastructure. It provides a general framework for operating systems, development tools, databases, and web servers. PaaS allows developers to focus on coding and developing applications, thereby increasing their productivity and decreasing time to develop software for the market. It supports the various stages of application development from building and testing to deployment and scaling.

IaaS avails virtualized computing resources over the internet. Virtual machines, resources, and networks are under this category. Such a model provides a high level of flexibility to users since they lease and configure hardware and software resources according to specifications. This level is suitable for businesses that need a high level of resource scalability and do not have a capital expenditure for physical infrastructure and its maintenance.

FaaS refers to serverless computing that enables developers to execute code in response to events without taking the responsibility for server management. A cloud provider does the mapping of resources and scales an execution environment automatically based on incoming demand, making it a fully abstracted model of server management. This model is most suitable for applications whose workloads vary over time, since users are charged according to the time their code runs.

In all of the cloud computing types described above the main pitfall is the use of plain customer or application data in the cloud that can cause several problems as I have explained in previous sections. Here, I shortly explain how to avoid such bad practices by the help of FHE for

SaaS applications.

3.1 Software-as-a-Service (SaaS)

It is possible to compute encrypted data without decrypting it with FHE. The section that follows the next section discusses the use of FHE in SaaS commenting on its benefits, technical realization, and its impact on data security and privacy.

FHE provides a solid solution for data privacy in SaaS environments. The most apparent advantage comes from the fact that critical and sensitive information can stay encrypted even in the course of computation and storage. Exactly the opposite happens in the case of the traditional SaaS application, in which data sometimes has to be decrypted in the process, exposing it to possible breaches and unauthorized access. As a result, computations can be done on such sensitive information-as in the case of personal identifiers, financial information, and health records-without exposing it in plaintext, all while achieving today's extremely strict data privacy regulation, such as the EU's General Data Protection Regulation.

Furthermore, FHE establishes the trust between the SaaS provider and the user. The user will not be afraid to use cloud services, where the accessed information may be misused and/or be accessed by unauthorized individuals. This is more significant in respect-based entities, which rely on critical user data such as healthcare, finance, and government services (Acar et al. 2017). In this sense, the adoption of FHE by SaaS providers may act as an element of competitive differentiation and proof to the user and society, in general, of the great commitment to data security and privacy.

FHE will be applied in Saas using encryption schemes that will enable operations on cypher-text messages. The process involves setting the data as a client to the FHE scheme before it is sent to the provider of Saas. The provider will then perform required computations on encrypted data sent to it, and then sent back to the client after performing the computations without decryption, and the results are responses sent back to the client in cyphertext. The work of the client will be the final step of the decryption of the data to plain text output (Gentry 2009b), (Naehrig, Lauter, and Vaikuntanathan 2011).

First of all, privacy cannot be added on top of existing software. Instead, the software should be developed with privacy in mind from the beginning. Thus, SaaS developers need to posses certain cryptography skills. Considering the recent advancement in the field, hopefully, this need will be evaded in the near future. That's why current FHE suppliers design solutions such that

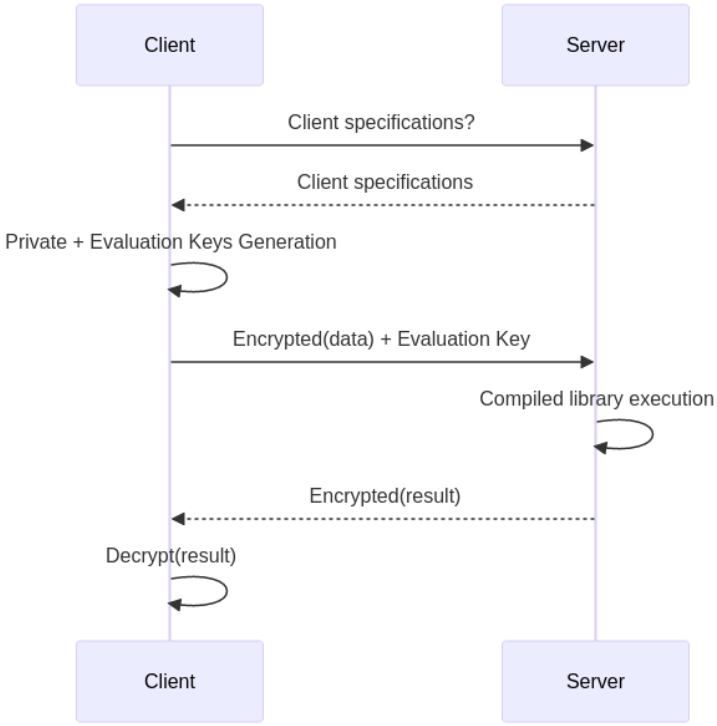

Figure 1. Client-Server Architecture by Zama

they can be used by anyone. For instance, Zama[3], a french start-up specialized in developing FHE and cryptography, has several tools and libraries that can be used now and require a little background in advanced cryptography.

Secondly, the development and the deployment of SaaS application should be treated separately. The development stage consists of carefully writing SaaS application homomorphically, such that it would support computations over encrypted values. The complexity of the application also determines FHE parameters for key generation. In FHE, there are three different key types which are private, public and evaluation keys. Only the last one shall be stored in SaaS application, hence in the cloud. Thus developers shall use well-organized cloud storage to host FHE parameters and evaluation keys.

Finally, the deployed application has to provide stored encryption parameters to the end users. From now on, I will refer to the end user as client. Upon receiving specified parameters, the client generates unique FHE keys and uploads only evaluation key to the server, i.e, SaaS application running on the cloud. After this moment, the client shall only send encrypted personal data to the server in order to protect the data from any misuse. Furthermore, the homomorphic application can process encrypted data and return results back to the client. At last, the client may decrypt to see the result of computation. The client-server communication is summarized in the following

[3] `https://www.zama.ai`

sequence diagram(Figure 1) provided by Zama[4].

[4] https://github.com/zama-ai/concrete/blob/main/docs/core-features/fhe_basics.md

4 EXPERIMENTAL PART

On the following part, I am going to share detailed explanation of the real life project designed by Zama. The project is designed for health-care system, which takes the inputs as symptoms and returns outputs as predicted illness. The source code and deployed application is available on `https://huggingface.co/spaces/zama-fhe/encrypted_health_prediction`.

4.1 MATERIALS AND METHODOLOGY

The following tools and dataset are utilized in this particular project.

4.1.1 Python3

Python3 is a high-level interpreted programming language that is interpretable, known for its qualities of readability and versatility. Released in December 2008, it is the improved successor to Python 2, with several main changes that increased the incompatibility with the previous version. Python3 is more focused on the clarity of syntax and on readability, which is a productivity element and is easy to learn in the environment for a beginner and professional in the discipline. Some of the main features include a very comprehensive standard library, support for different programming paradigms such as object-oriented programming, procedural programming, and functional programming. It also supports dynamic typing and automatic memory management (Van Rossum and Drake 2009).

Python 3—most notably, NumPy, pandas, and scikit-learn—are major libraries for scientific computing and data analysis. Python 3 has tremendously influenced software development, data science, and machine learning. Further integration with web frameworks, such as Django and Flask, has made it increasingly popular in web development (Lutz and Safari 2013).

4.1.2 Concrete

The Zama Technologies Concrete library (Zama 2022a) is designed for the potential realization and optimization of TFHE (Chillotti et al. 2020). The aim is to provide a library that serves as a toolkit with the needed tools and frameworks for computing on encrypted data, such that at every point in the processing life cycle of the data, its privacy is guaranteed. Concrete library is practical, efficient, and friendly. It is formulated with the latest advances in research in the area of cryptosystems to bring applicability of FHE in the real world.

This is extremely valuable in situations where data privacy is very important, for example,

in healthcare, finances, and government services. Allowing computation on the encrypted data eliminates decryption of sensitive information, thereby reducing the risk of data breach or authorization.

4.1.3 Concrete ML

The Concrete ML (Zama 2022b) is developed as an extension of the Concrete library for generic machine learning applications. It brings in fully homomorphic encryption in machine learning models, which then makes it possible to conduct computations over encrypted datasets while keeping the data private. This will allow data scientists to train and deploy machine learning models without the risk of the data being leaked and used without consent

Concrete ML is compatible with the majority of famous machine learning tools, including scikit-learn and PyTorch. Concrete ML harnesses the strength of TFHE and the strengths of machine learning so as to tackle critical issues around privacy in data-intensive applications that require secure processing and analysis techniques.

4.1.4 Gradio

Gradio (Abid et al. 2019) is an open-source Python library built for developing user interfaces for machine learning models, making it very simple. It offers interfaces for user input with the possibility of visualization of model predictions to get interested in machine learning models, all in real time. Gradio allows the input of different kinds of data, including text, images, audio, and video, so it can be flexible for different applications.

Gradio, with its ease of use and integrated ability to work with popular machine learning frameworks like TensorFlow and PyTorch, has really been helpful for data scientists and researchers. It helps prototype and deploy machine learning models very quickly so that accessibility and user engagement are improved.

4.1.5 FastAPI

FastAPI[5] is a modern, fast (high-performance) web framework for building APIs with Python 3.7+ based on standard Python type hints. FastAPI is designed to be easy to use and maintain while delivering high performance with the asynchronous programming paradigms. FastAPI is based on top of Starlette for the web parts and Pydantic for the data parts.

[5] `https://fastapi.tiangolo.com/`

FastAPI is pretty performant, as much as Node.js and Go, for high-demand applications. It also includes automatic interactive API documentation; it's better for developer experience and productivity. A couple of features in FastAPI make it best for class in RESTful API development and microservices.

4.1.6 Uvicorn

Uvicorn[6] is an ultra-fast ASGI server implementation, built on uvloop and httptools. Uvicorn is built to be really quick and highly efficient in running asynchronous web applications, for instance, created using either FastAPI or Django Channels. Performance advantages for Uvicorn come from using uvloop, which is a very fast implementation of the event loop, and httptools, an efficient HTTP parser.

Uvicorn is used in most production deployments of ASGI-based web applications because it provides peak performance as well as the ability to minimize outages. Versatile by supporting HTTP/1.1, WebSockets, and HTTP/2, Uvicorn is flexible to the various needs of web applications. LaTeX is a very versatile and useful tool for many needs of scientific users.

4.1.7 Dataset

The Disease-Prediction-from-Symptoms [7] dataset is highly beneficial when attempting to build machine learning models aiming at diagnosing diseases when reported symptoms are given. The availability of this data provides the ability to train predictive models, and it would contain a normalized set of symptoms and labeled diseases.

The dataset includes two CSV files, one for labeled training data and one for testing, each containing 133 columns where the first 132 columns represent symptoms and the last column indicates the prognosis of 42 different diseases.

4.1.8 Supporting resources

To improve the clarity of the text, sections have been revised, edited, and paraphrased by using ChatGPT. Specifically, Introduction and Materials parts of the research.

[6] https://www.uvicorn.org/

[7] https://www.kaggle.com/datasets/kaushil268/disease-prediction-using-machine-learning

4.2 RESULTS

It is straightforward to build the application. After the source code is cloned, it is enough to run a few command line instructions given in README to launch user-friendly application interface on the browser:

```
python3 -m venv .venv
source .venv/bin/activate
pip3 install -r requirements.txt --ignore-installed
python3 app.py
```

The workflow of the project consists of 4 steps, which are selecting the chief complaints, encrypting the data, running the FHE evaluation and decrypting the prediction.

1. Select chief complaints

   As it is shown on the Figure 2, there are 9 groups of complaints, which consists of different symptoms see the Figure 3 and Figure 4. After choosing all of your symptoms, you are submitting your complaints.

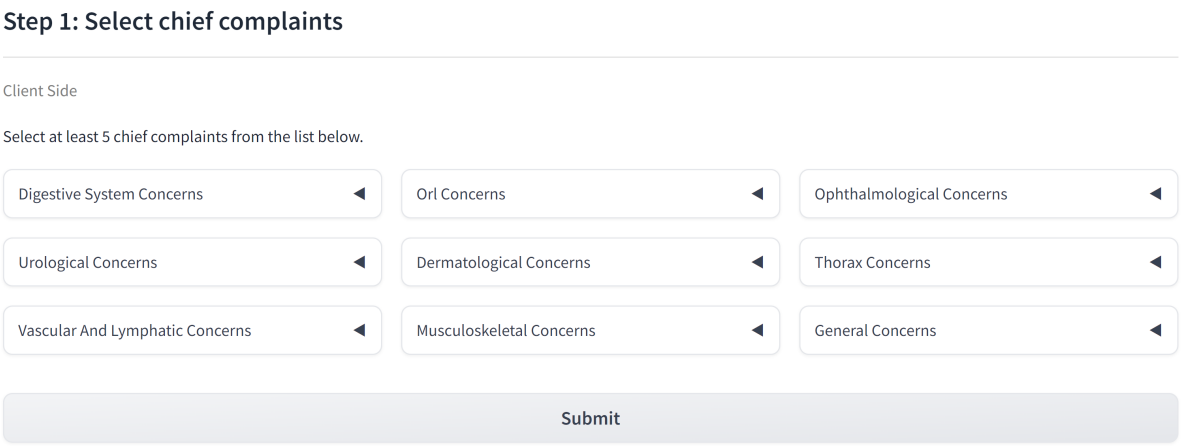

Figure 2. Step1.1

2. Encrypt data

   In FHE schemes, secret encryption/decryption keys are generated for the client to encrypt and decrypt their data. Additionally, a public evaluation key is created, allowing external entities to perform homomorphic operations on the encrypted data without decrypting it. The evaluation key is then sent to the server for further processing.

   First, private and evaluation keys are genenerated. Then data is encrypted using the private secret key. The encryption result should look like Figure 5. "User Symptoms Vector"

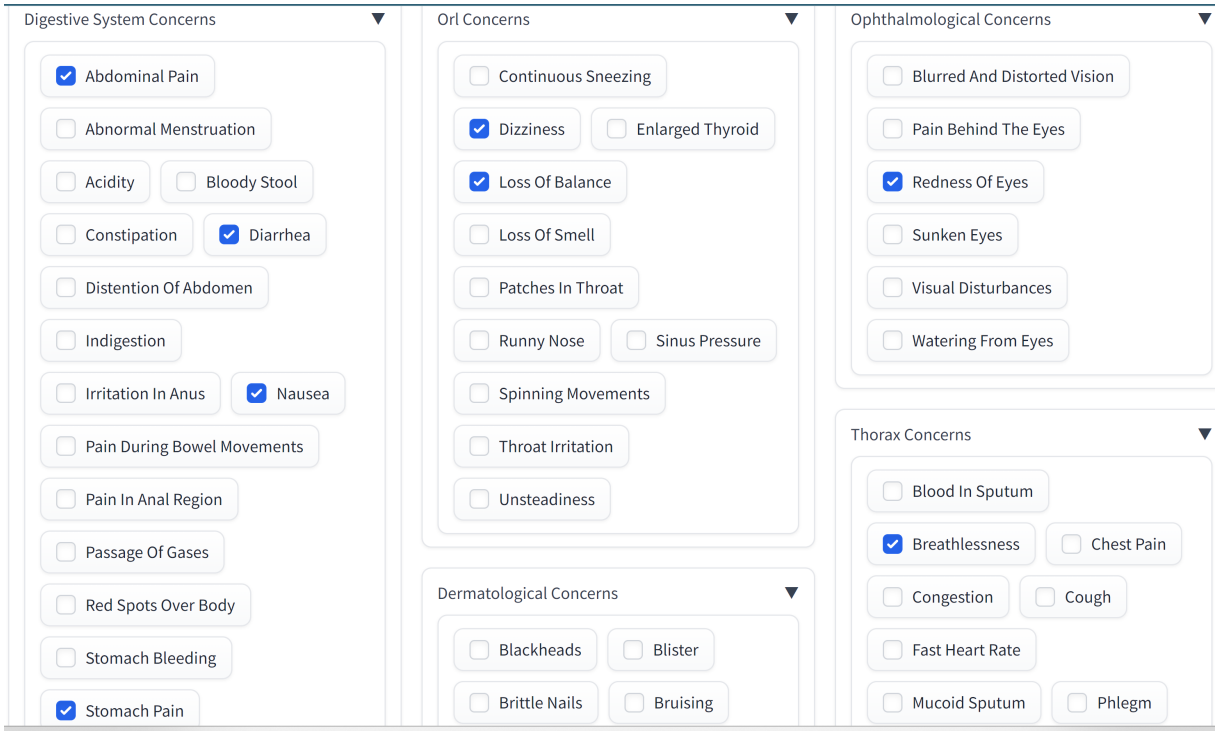

Figure 3. Step1.2

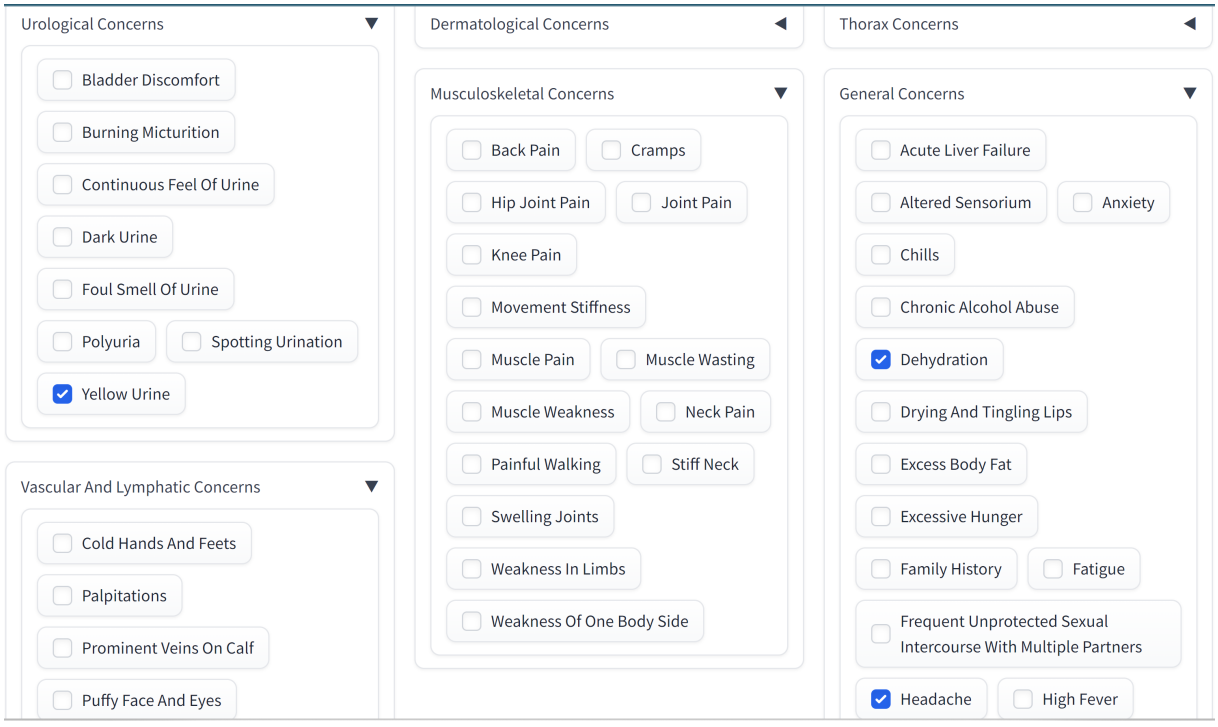

Figure 4. Step1.2

shows the chosen complains by patient, and "Encrypted Vector" is the encrypted version of it.

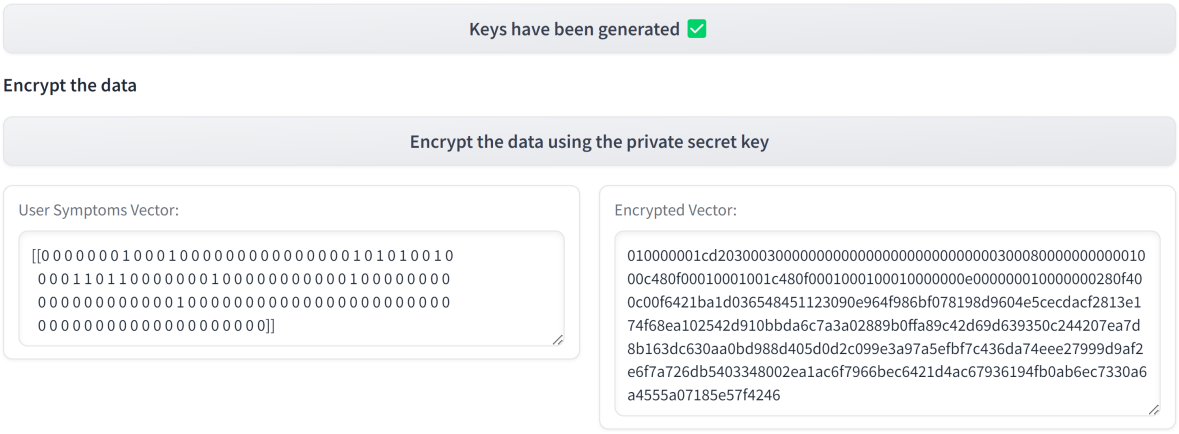

Figure 5. Step 2

At this point, we are sending encrypted symptoms data to the server side which is a SaaS application that predicts possible diseases based on symptoms. Figure 6

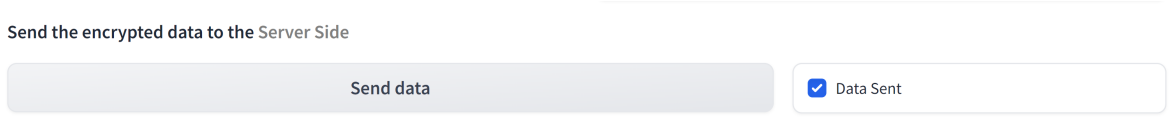

Figure 6. Step 2.3

3. Run the FHE evaluation

   The data which has been sent to the server passes through Logistic Regression model that has already been trained on given dataset. It should be noted that, in this application example, the SaaS provider may have access to the raw training and test data and design best scoring machine learning model. The Concrete ML framework supports many classical machine learning and some modern deep learning models under the hood. These models are designed to carry out computations over encrypted inference data.

   We are running the FHE evaluation and then see the "Total FHE Execution Time" in Figure 7

4. Decrypt the Data

   On the last step, we are receiving the data from the server side, and the decrypting it with the unique private key and after that our output will be shown, Figure 8

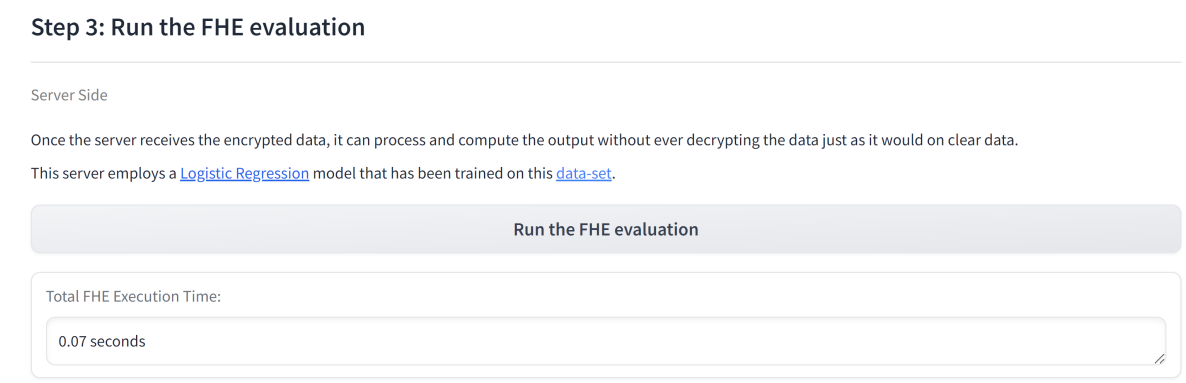

Figure 7. Step 3

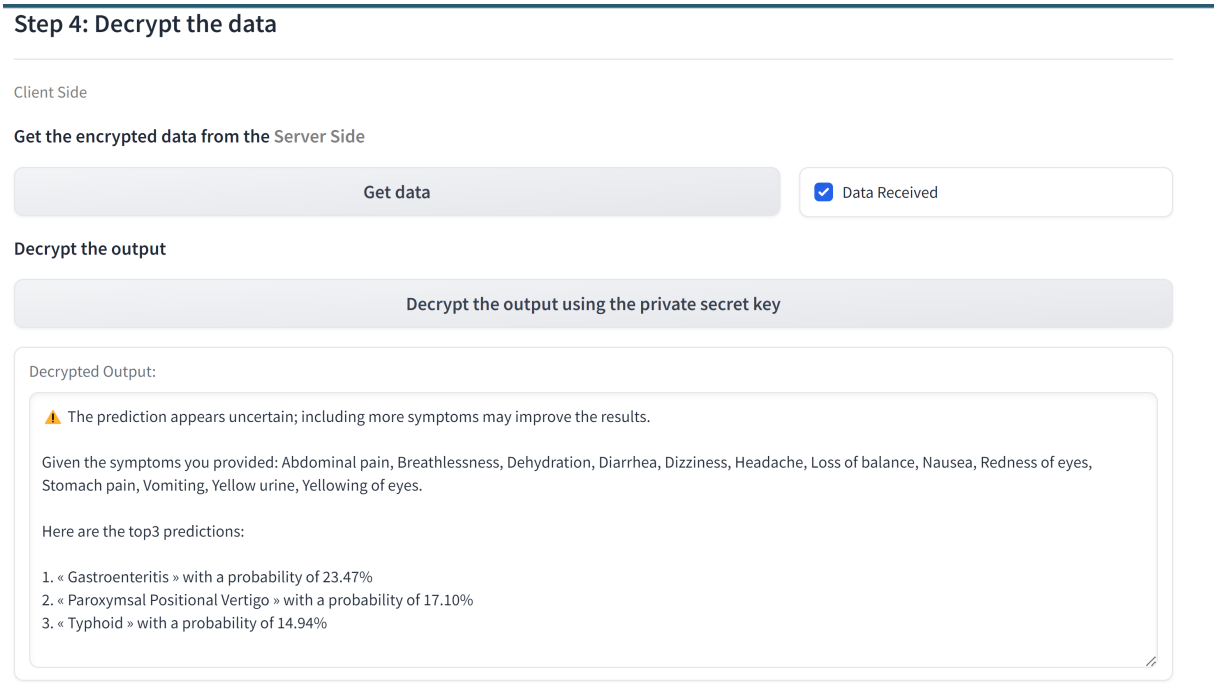

Figure 8. Step 4

4.3 DISCUSSION

The preceding sections serve as a demonstration that it is feasible to develop real-world SaaS applications utilizing FHE. The workflow has been provided for those interested in replicating such applications. Additional examples can be found in Zama's official Github repositories[8]. FHE has reached a level of maturity suitable for SaaS applications, evidenced by the reasonable execution times. To achieve more intricate projects and obtain finer results, further investment of time into the project is necessary. The feasibility of the development process has been articulated.

[8] https://github.com/zama-ai/

SUMMARY

The primary aim of this thesis was to enhance post-quantum cryptographic primitives within a framework designed to ensure secure cloud computation over encrypted data. The research delved into the substantial risks that quantum computing poses to traditional cryptographic systems due to their vulnerability to quantum algorithms. By examining these risks, the study evaluated existing post-quantum cryptographic techniques and proposed innovative solutions to bolster the security of cloud-based computations.

The research specifically concentrated on developing methods for post-quantum secure processing to safeguard the confidentiality, integrity, and availability of sensitive information within cloud environments. The goal was to provide guidance for creating applications that handle encrypted data on cloud platforms, in adherence to stringent data protection regulations like the GDPR and the Cloud Act. The outcomes of this research confirm the practicality of developing SaaS applications with enhanced data security using post-quantum secure Fully Homomorphic Encryption. As a concrete example, I have used Zama's "Encrypted Health Prediction" demo tutotial. This advancement addresses future threats posed by quantum computing and contributes to the evolution of more secure cloud computing frameworks.

References

Armbrust, Michael et al. (Apr. 2010). "A View of Cloud Computing". In: *Commun. ACM* 53, pp. 50–58. DOI: 10.1145/1721654.1721672.

Mell, Peter and Timothy Grance (2011-09-28 2011). *The NIST Definition of Cloud Computing*. en. DOI: https://doi.org/10.6028/NIST.SP.800-145.

Buyya, Rajkumar et al. (2009). "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility". In: *Future Generation Computer Systems* 25.6, pp. 599–616. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2008.12.001. URL: https://www.sciencedirect.com/science/article/pii/S0167739X08001957.

Marinescu, D.C. (2017). *Cloud Computing: Theory and Practice*. Elsevier Science. ISBN: 9780128128114. URL: https://books.google.az/books?id=O9smDwAAQBAJ.

Subashini, S. and V. Kavitha (2011). "A survey on security issues in service delivery models of cloud computing". In: *Journal of Network and Computer Applications* 34.1, pp. 1–11. ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2010.07.006. URL: https://www.sciencedirect.com/science/article/pii/S1084804510001281.

Voigt, Paul and Axel Bussche (Jan. 2017). *The EU General Data Protection Regulation (GDPR): A Practical Guide*. ISBN: 978-3-319-57958-0. DOI: 10.1007/978-3-319-57959-7.

Pearson, Siani (Jan. 2013). "Privacy, Security and Trust in Cloud Computing". In: pp. 3–42. ISBN: 978-1-4471-4188-4. DOI: 10.1007/978-1-4471-4189-1_1.

Katz, J. and Y. Lindell (2020). *Introduction to Modern Cryptography*. Chapman & Hall/CRC Cryptography and Network Security Series. CRC Press. ISBN: 9781351133029. URL: https://books.google.az/books?id=zwIPEAAAQBAJ.

Menezes, Alfred J., Paul C. van Oorschot, and Scott A. Vanstone (2001). *Handbook of Applied Cryptography*. CRC Press. URL: http://www.cacr.math.uwaterloo.ca/hac/.

Schneier, Bruce (1996). *Applied Cryptography*. 2nd. Wiley. ISBN: 0471117099. URL: http://www.amazon.com/Applied-Cryptography-Protocols-Algorithms-Source/dp/0471117099/ref=sr_1_1?ie=UTF8&qid=1287063315&sr=8-1.

Rivest, R. L., A. Shamir, and L. Adleman (Feb. 1978). "A method for obtaining digital signatures and public-key cryptosystems". In: *Commun. ACM* 21.2, pp. 120–126. ISSN: 0001-0782. DOI: 10.1145/359340.359342. URL: https://doi.org/10.1145/359340.359342.

Johnson, Don, Alfred Menezes, and Scott A. Vanstone (2001). “The Elliptic Curve Digital Signature Algorithm (ECDSA).” In: *Int. J. Inf. Sec.* 1.1, pp. 36–63. URL: http://dblp.uni-trier.de/db/journals/ijisec/ijisec1.html#JohnsonMV01.

Shor, P.W. (1994). “Algorithms for quantum computation: discrete logarithms and factoring”. In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.

Ajtai, M. (1996). “Generating hard instances of lattice problems (extended abstract)”. In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. STOC ’96. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, pp. 99–108. ISBN: 0897917855. DOI: 10.1145/237814.237838. URL: https://doi.org/10.1145/237814.237838.

Merkle, Ralph C. (1990). “A Certified Digital Signature”. In: *Advances in Cryptology — CRYPTO’ 89 Proceedings*. Ed. by Gilles Brassard. New York, NY: Springer New York, pp. 218–238. ISBN: 978-0-387-34805-6.

McEliece, R. J. (Jan. 1978). “A Public-Key Cryptosystem Based On Algebraic Coding Theory”. In: *Deep Space Network Progress Report* 44, pp. 114–116.

Patarin, Jacques (1996). “Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms”. In: *Advances in Cryptology — EUROCRYPT ’96*. Ed. by Ueli Maurer. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 33–48. ISBN: 978-3-540-68339-1.

Gentry, Craig (2009a). “Fully homomorphic encryption using ideal lattices”. In: *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*. STOC ’09. Bethesda, MD, USA: Association for Computing Machinery, pp. 169–178. ISBN: 9781605585062. DOI: 10.1145/1536414.1536440. URL: https://doi.org/10.1145/1536414.1536440.

Bernstein, Daniel J. and Tanja Lange (Sept. 2017). “Post-quantum cryptography”. In: *Nature* 549.7671, pp. 188–194. ISSN: 1476-4687. DOI: 10.1038/nature23461. URL: https://doi.org/10.1038/nature23461.

Yao, Andrew C. (1982). “Protocols for secure computations”. In: *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, pp. 160–164. DOI: 10.1109/SFCS.1982.38.

Bogdanov, Dan, Sven Laur, and Jan Willemson (2008). “Sharemind: A Framework for Fast Privacy-Preserving Computations”. In: *Proceedings of the 13th European Symposium on Research in Computer Security - ESORICS’08*. Ed. by Sushil Jajodia and Javier Lopez. Vol. 5283. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 192–206. ISBN: 978-3-540-88312-8.

Becker, Sebastian et al. (2021). *Carbyne Stack*. DOI: 10.5281/zenodo.8192460. URL: https://carbynestack.io.

Keller, Marcel (2020). "MP-SPDZ: A Versatile Framework for Multi-Party Computation". In: *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. CCS '20. Virtual Event, USA: Association for Computing Machinery, pp. 1575–1590. ISBN: 9781450370899. DOI: 10.1145/3372297.3417872. URL: https://doi.org/10.1145/3372297.3417872.

Chen, Yuanfeng et al. (2020). *Rosetta: A Privacy-Preserving Framework Based on TensorFlow*. https://github.com/LatticeX-Foundation/Rosetta.

Dahl, Morten et al. (2018). *Private Machine Learning in TensorFlow using Secure Computation*. arXiv: 1810.08130 [cs.CR].

Chandel, Sonali, Geng Yang, and Sumit Chakravarty (2020). "RSA-CP-IDABE: A Secure Framework for Multi-User and Multi-Owner Cloud Environment". In: *Information* 11.8. ISSN: 2078-2489. DOI: 10.3390/info11080382. URL: https://www.mdpi.com/2078-2489/11/8/382.

Awan, Ijaz Ahmad et al. (2020). "Secure Framework Enhancing AES Algorithm in Cloud Computing". In: *Secur. Commun. Networks* 2020, 8863345:1–8863345:16. URL: https://api.semanticscholar.org/CorpusID:221509498.

Noha, Fatma Omara, and Nahla Omran (Apr. 2016). "A Hybrid Hashing Security Algorithm for Data Storage on Cloud Computing". In: DOI: 10.13140/RG.2.1.4103.3844.

Gill, Sajid et al. (Oct. 2021). "Security and Privacy Aspects of Cloud Computing: A Smart Campus Case Study". In: *Intelligent Automation and Soft Computing* 31, pp. 117–128. DOI: 10.32604/iasc.2022.016597.

Acar, Abbas et al. (Apr. 2017). "A Survey on Homomorphic Encryption Schemes: Theory and Implementation". In: *ACM Computing Surveys* 51. DOI: 10.1145/3214303.

Gentry, Craig (2009b). "Fully homomorphic encryption using ideal lattices". In: *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*. STOC '09. Bethesda, MD, USA: Association for Computing Machinery, pp. 169–178. ISBN: 9781605585062. DOI: 10.1145/1536414.1536440. URL: https://doi.org/10.1145/1536414.1536440.

Naehrig, Michael, Kristin Lauter, and Vinod Vaikuntanathan (2011). "Can homomorphic encryption be practical?" In: *Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop*. CCSW '11. Chicago, Illinois, USA: Association for Computing Machinery, pp. 113–124. ISBN: 9781450310048. DOI: 10.1145/2046660.2046682. URL: https://doi.org/10.1145/2046660.2046682.

Van Rossum, Guido and Fred L. Drake (2009). *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace. ISBN: 1441412697.

Lutz, M. and an O'Reilly Media Company Safari (2013). *Learning Python, 5th Edition*. O'Reilly Media, Incorporated. URL: https://books.google.az/books?id=LWM6zQEACAAJ.

Zama (2022a). *Concrete: TFHE Compiler that converts python programs into FHE equivalent*. https://github.com/zama-ai/concrete.

Chillotti, Ilaria et al. (Jan. 2020). "TFHE: Fast Fully Homomorphic Encryption Over the Torus". In: *Journal of Cryptology* 33.1, pp. 34–91. ISSN: 1432-1378. DOI: 10.1007/s00145-019-09319-x. URL: https://doi.org/10.1007/s00145-019-09319-x.

Zama (2022b). *Concrete ML: a Privacy-Preserving Machine Learning Library using Fully Homomorphic Encryption for Data Scientists*. https://github.com/zama-ai/concrete-ml.

Abid, Abubakar et al. (2019). *Gradio: Hassle-Free Sharing and Testing of ML Models in the Wild*. arXiv: 1906.02569 [cs.LG].

Appendix

I. Glossary

NON-EXCLUSIVE LICENCE TO REPRODUCE THESIS AND MAKE THESIS PUBLIC

I, **Toghrul Gulmammadov**,

*(*author's name)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Post-Quantum Secure Cloud Computation Over Encrypted Data,

   *(*title of thesis)

   supervised by Valeh Farzaliyev.

   *(*supervisor's name)

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Toghrul Gulmammadov

27/05/2024