|  |  |  |
| --- | --- | --- |
| A picture containing drawing  Description automatically generated |  | Academic Year 2024-2025 |

**Master II**

**Curriculum: Artificial Intelligence**

**Final Thesis**

Healthcare AI

By

**IBRAHIM Mohamad (Ahmad)**

Thesis Director: **KHALIL Hariss**

Academic Supervisors: **HAJJAR Chantal**  
Location: **USJ-ESIB Mansouriyyeh**

Jury Members: **BAKOUNY Youssef, CHAMOUN Maroun**

Date : **25-9-2025**

# Résumé

Les données sont devenues une ressource essentielle, et leur protection est plus importante que jamais au regard du RGPD. Certains ensembles de données sont classés comme sensibles et doivent être sécurisés. Lorsqu’un tel ensemble est utilisé pour entraîner un modèle d’apprentissage automatique, le modèle produit des paramètres. Si ces paramètres sont divulgués, les données sous-jacentes peuvent être compromises. Cela souligne la nécessité d’employer des méthodes de chiffrement, en particulier dans les modèles d’apprentissage fédéré où les paramètres sont transmis de plusieurs modèles locaux vers un modèle global, et où chaque client doit protéger ses paramètres contre les fuites. Cette architecture exige une technique de chiffrement spécialisée connue sous le nom de « chiffrement homomorphe ». Dans cette technique, des opérations mathématiques peuvent être effectuées sur du texte chiffré, appelé « ciphertext ». Dans ce contexte, le « plaintext » désigne les données originales non chiffrées. L’architecture peut utiliser un chiffrement symétrique ou asymétrique, selon l’approche choisie. Cependant, les deux approches employaient initialement la même clé pour tous les clients, créant une vulnérabilité critique : la fuite d’une seule clé compromet tout. Les cryptographes ont adopté une approche asymétrique pour atténuer ce problème, mais leurs méthodes présentent encore des limites non résolues et ne constituent pas des solutions parfaites. Dans ce projet, une approche de chiffrement symétrique a été mise en œuvre pour répondre à ce problème et a été comparée à l’une des solutions asymétriques appelé xMK-CKKS.

Pour l’approche symétrique, la méthode Domingo-Ferrer avec changement de clé a été utilisée. Chaque client possède une clé unique, chiffrant ses paramètres avec des clés différentes de celles des autres. Cela est incompatible avec l’architecture du chiffrement homomorphe, qui exige que tous les textes chiffrés soient chiffrés avec la même clé. Cette limitation rend nécessaire la technique de changement de clé, où des textes chiffrés sous les clés s1, s2, s3… sont convertis en un chiffrement sous s. La solution existante utilisée pour la comparaison est la technique xMK CKKS. Les deux approches ont été simulées à l’aide du cadre SageMath Python sur un ensemble de données médicales, en raison de sa sensibilité, et un ensemble de données synthétiques supplémentaire a été utilisé pour comparaison. Le modèle employé est un modèle d’apprentissage fédéré utilisant une régression logistique pour chaque client.

Les deux approches ont été rigoureusement étudiées et évaluées. De nombreuses différences sont apparues entre elles, mais une distinction clé a été mise en évidence. L’approche symétrique s’est révélée avantageuse à bien des égards car elle empêche le serveur d’effectuer toute opération de déchiffrement, contrairement à la méthode où le déchiffrement a lieu exclusivement côté serveur.

# Abstract

With the rise of AI, data has become a critical resource, and its protection is more important than ever under **GDPR** regulations. Certain datasets are classified as sensitive and must be secured, the medical data for example. When such a dataset is used to train a machine learning (ML) model, the model produces parameters (weights and biases). If these parameters are leaked, the underlying data may be compromised. This underscores the necessity of employing encryption methods, particularly in **federated learning** FL models where parameters (or gradients) are transmitted from multiple local models (“clients”) to a global model (“server”), and each client must safeguard its parameters from leakage. This architecture requires a specialized encryption technique known as “**homomorphic encryption**” (HE). In this technique, mathematical operations can be performed on encrypted text. The architecture can employ either symmetric or asymmetric encryption, depending on the approach. However, both approaches originally employed the same key for all clients, creating a critical vulnerability: the leakage of one key compromise all. Clients are honest but curious, which means they will apply the ML algorithm honestly, but they will extract data from other clients if they have the chance. The authors of this article “PRIVACY-PRESERVING FEDERATED LEARNING BASED ON MULTI-KEY HOMOMORPHIC ENCRYPTION” adopted a new solution called xMK-CKKS based on modifying the primitive CKKS scheme to make it much more suitable to FL by resolving the two main security problems mentioned above resulting from applying HE in a traditional fashion with FL scenario. In this work, the security issues mentioned above, which arise from implementing HE in a traditional manner within a FL scenario, were addressed by integrating the well-known Key Switching (KS) technique with the symmetric Domingo-Ferrer (DF) encryption scheme.

For the symmetric approach, the DF with Key Switching (**DF-KS**) method was used. Each client possesses a unique key, encrypting its parameters under different keys from the others. This is incompatible with HE architecture, which requires all ciphertexts to be encrypted under the same key. This limitation necessitates the KS technique, where ciphertexts encrypted under keys s1, s2, s3… are converted to encryption under s. The existing solution used for comparison is the **xMK-CKKS** technique, which applies the Learning With Error (“**LWE**”) method. Both approaches were simulated using the SageMath Python framework on a healthcare dataset, due to its sensitivity, and an additional synthetic dataset was used for comparison. The model employed is a federated learning model with a logistic regression model for each client.

Both approaches were rigorously analyzed and evaluated. The DF-KS solution demonstrated several advantages over xMK-CKKS, particularly in terms of execution efficiency, storage, and communication overhead. In addition, DF-KS prevents the global server from performing any decryption operations, whereas xMK-CKKS still allows the server to decrypt aggregated ciphertexts**.**

xMK-CKKS requires all clients to jointly decrypt the aggregated model weights, whereas DF-KS supports independent decryption, allowing progress even if a client is unavailable. However, DF-KS can be vulnerable to known plaintext or ciphertext attacks under certain conditions, while xMK-CKKS remains resistant. Thus, the choice between them depends on the application’s specific security needs and deployment constraints.

**Acknowledgements**

This work would not have been possible without the invaluable supervision, guidance, and insight of **Dr. Hariss** and **Dr. Hajjar**, to whom I extend my profound appreciation. Their expertise and encouragement not only shaped the direction of this project but also greatly enhanced my understanding of the subject.

I am also deeply indebted to my colleagues and close friends for their steadfast support and encouragement during the most challenging stages of this project. Their companionship, constructive feedback, and unfailing willingness to stand by me in difficult moments have been an indispensable source of strength and motivation.

Above all, I owe my deepest gratitude to my parents for their unwavering mental, physical, and financial support. Their constant encouragement, sacrifices, and belief in my abilities have provided the foundation upon which all my efforts rest and have firmly set me on this path.

Finally, I am profoundly thankful to God for blessing me with the perseverance and insight to bring this project to completion.

# Contents

[Résumé 2](#_Toc209884991)

[Abstract 3](#_Toc209884992)

[Contents 5](#_Toc209884993)

[List of figures 6](#_Toc209884994)

[List of tables 7](#_Toc209884995)

[Acronyms 8](#_Toc209884996)

[Symbols 9](#_Toc209884997)

[Chapter 1: Introduction 10](#_Toc209884998)

[1.1 Background 10](#_Toc209884999)

[1.2 Context 11](#_Toc209885000)

[1.3 Purposes 11](#_Toc209885001)

[1.4 Significance, Scope and Definitions 12](#_Toc209885002)

[1.5 Thesis Outline 13](#_Toc209885003)

[1.6 Work Plan 14](#_Toc209885004)

[Chapter 2: Literature Review 15](#_Toc209885005)

[2.1 Historical Background [optional] 16](#_Toc209885006)

[2.2 Topic 1 16](#_Toc209885007)

[2.3 Topic 2 16](#_Toc209885008)

[2.4 Topic 3 16](#_Toc209885009)

[2.5 Summary and Implications 16](#_Toc209885010)

[Chapter 3: Research Design 17](#_Toc209885011)

[3.1 Methodology and Research Design 17](#_Toc209885012)

[3.1.1 Methodology 17](#_Toc209885013)

[3.1.2 Research Design 17](#_Toc209885014)

[3.2 Participants 18](#_Toc209885015)

[3.3 Instruments 18](#_Toc209885016)

[3.4 Procedure and Timeline 18](#_Toc209885017)

[3.5 Analysis 18](#_Toc209885018)

[3.6 Ethics and Limitations 19](#_Toc209885019)

[Chapter 4: Implementation 20](#_Toc209885020)

[Chapter 5: Results 21](#_Toc209885021)

[Chapter 6: Analysis 23](#_Toc209885022)

[Chapter 7: Conclusion 25](#_Toc209885023)

[References 27](#_Toc209885024)

[Appendices 28](#_Toc209885025)

[Some writing guidelines: 29](#_Toc209885026)

# List of figures

# List of tables

# Acronyms

|  |  |
| --- | --- |
| DF  KS  XMK CKKS  SMPC  FL  FEDAVG  HE  LWE  ML  DL  FEDSVG  AI  e.g. | Domingo Ferrer  Key Switch  Multiple Key Cheon Kim Kim Song  Secure Multiparty Computation  Federated Learning  Federated learning that uses mean value for the parameters  Homomorphic Encryption  Learning With Error  Machine Learning  Deep Learning  Federated learning that uses mean value for the gradient  Artificial Intelligence  Exempli Gratia (for example) |
| i.e. | Id est (in other words) |

# Symbols

|  |  |
| --- | --- |
| R  S  Z | Weights of client i  Bias of client I / public key (depends on context)  data size of client i / cipher text i (depend on context)  Security parameter for DF.  Public modulus (DF)  Private modulus (DF)  Security parameter for xMK CKKS (vector dimension)  Rounds  Secret key  Data percentage size  Ring modulus q |

# Introduction

This chapter outlines the background (section ‎1.1) and context (section ‎1.2) of the research, and its purposes (section ‎1.3). Section ‎1.4 describes the significance and scope of this research and provides definitions of terms used. Finally, section ‎1.5 includes an outline of the remaining chapters of the thesis.

## Background

Advancements in Artificial Intelligence (AI) and Machine Learning (ML) have revolutionized medical healthcare, enabling data-driven patient diagnosis and treatment recommendations [1]. However, integrating privacy-preserving techniques with collaborative learning frameworks like Federated Learning (FL) remains a critical challenge. It is crucial to integrate advanced security solutions with federated learning (FL) to mitigate the risk of data leakage. This research project focuses on leveraging FL with Homomorphic Encryption (HE) [2] for patient classification in diabetes diagnosis, using the SageMath framework with the Domingo Ferrer (DF) [3] scheme combined with Key Switching (KS) for secure computation.

FL allows multiple healthcare institutions or local devices to collaboratively train a shared model without exposing individual patient data [4]. To enhance privacy, HE is integrated into FL to enable computations on encrypted data. However, the current practice of using a single decryption key across all local models presents two significant security vulnerability. The first, clients are considered honest but curious, meaning they will execute machine learning algorithms faithfully but will attempt to extract data if given the opportunity. The second, if any local model is compromised, the entire system is at risk, potentially exposing sensitive patient data and allowing unauthorized access to the global model. This setup also increases the risk of cascading failures in security.

The problematic this project aims to address is the inherent security risks associated with using a single decryption key in FL-HE systems. Specifically, how is it possible to design a secure and efficient mechanism that mitigates these risks without compromising the performance and accuracy of the FL framework?

## Context

With the rapid rise of AI, data has become an indispensable resource whose protection is critical under stringent regulations such as the GDPR [5]. In FL models, where parameters are exchanged between numerous local clients and a central server, the leakage of model weights or gradients can expose sensitive underlying data. Traditional HE enables computation on encrypted data but originally required a single shared key for all clients, creating a serious vulnerability: compromise of one key compromise all. Although asymmetric schemes, the xMK-CKKS for example [6], were introduced to reduce this risk, they still present unresolved shortcomings and typically require server-side decryption and decrypting the aggregated weights enforce that all the clients should be present. This study addresses this problem by implementing a symmetric encryption approach, the DF with Key Switching (DF-KS) method [7], which allows each client to encrypt its parameters under a unique key while maintaining compatibility with homomorphic operations through KS [8]. The approach is compared against an existing asymmetric solution using sensitive healthcare data and a synthetic dataset within a federated logistic regression model. This work seeks to determine whether the symmetric DF-KS method can offer stronger privacy guarantees and operational advantages than current asymmetric alternatives.

## Purposes

This study aims to design, implement, and evaluate a symmetric HE approaches Domingo-Ferrer with Key Switching (DF-KS) [7] to protect model parameters in FL and address vulnerabilities inherent in existing single-key or asymmetric encryption methods. Specifically, it seeks to (i) enable each client to encrypt its parameters under a unique key while preserving homomorphic compatibility through KS [8] , this method will solve both vulnerabilities mentioned in section 1.1 (key compromise and “honest but curios” clients concept) (ii) compare this approach with an established asymmetric scheme, xMK-CKKS [6], using both a sensitive healthcare dataset and a synthetic dataset in a federated logistic regression model, and (iii) rigorously evaluate privacy, security, and computational efficiency. The practical outcomes include a working prototype of a symmetric HE system adapted for FL, comparative performance metrics against an asymmetric method, and guidelines for deploying privacy-preserving encryption in sensitive domains such as healthcare. This research addresses the critical problem that current HE schemes either require a single shared key, risking catastrophic compromise, or necessitate server-side decryption that undermines privacy. Accordingly, the overarching questions guiding this work are whether DF-KS can allow clients to use distinct keys without breaking homomorphic compatibility, whether it can reduce or eliminate server-side decryption compared to asymmetric schemes, how it performs on sensitive data relative to xMK-CKKS in terms of privacy, security, and computational cost, and what best practices can be derived for real-world implementation of privacy-preserving encryption in federated learning.

## Significance, Scope and Definitions

GDPR (General Data Protection Regulation) is a European law issued by the United Nations to protect personal data [5]. Article 9 mentions a special category data, one of them is the health data, which will be the focus in this project. Article 32 mandates the protection of this special category using encryption techniques. This research is significant because it addresses a growing and pressing problem: the privacy and security of sensitive data used in ML, particularly under stringent regulatory regimes such as the GDPR. FL is increasingly employed to train models across decentralized datasets in healthcare and other sensitive domains, yet the exchange of model parameters or gradients between clients and servers poses a major risk of data leakage. The present study fills this gap by implementing and evaluating a symmetric HE schemes with key switching (DF-KS) [7] and comparing it to a state-of-the-art asymmetric approach (xMK-CKKS) [6]. Methodologically, the study contributes by simulating both approaches on real healthcare (the PIMA diabetes dataset) and synthetic datasets within a federated logistic regression framework, providing empirical evidence on privacy, performance, and operational feasibility where the literature remains largely theoretical.

The scope of this research is deliberately focused on the encryption layer of FL, not on alternative privacy-preserving techniques such as differential privacy or secure multi-party computation. It centres on symmetric DF-KS and asymmetric xMK-CKKS within the context of a two-tier federated architecture (multiple clients, one server) and logistic regression as the base model. The delimitations include using simulated rather than production systems, two specific datasets (one sensitive, one synthetic), and the evaluation of only one symmetric and one asymmetric scheme rather than a broad survey of all possible cryptographic protocols.

Key terms are defined conceptually to guide the reader. “Federated learning” refers to a distributed ML paradigm where models are trained locally on clients and aggregated centrally without sharing raw data. “Parameters” or “gradients” denote the weights, biases, and updates exchanged between client and server during training [4]. “Homomorphic encryption” is an encryption scheme that allows mathematical operations to be performed on ciphertexts without decryption, producing encrypted results that, once decrypted, match the outcome of operations performed on plaintext. “Plaintext” refers to the original, unencrypted data, while “ciphertext” is the encrypted form. “Symmetric encryption” means the same key is used for both encryption and decryption, whereas “asymmetric encryption” uses a key pair (public and private). “Key switching” denotes the process of converting ciphertexts encrypted under one key into ciphertexts encrypted under another key without doing decryption [7], enabling compatibility in systems where multiple keys are used. These conceptual definitions provide the foundation for the operational measures detailed later in the Research Design chapter.

## Thesis Outline

This thesis is structured as follows:

* **Chapter 1 Introduction**: provides background information, context, project objectives, and definitions essential for understanding the project.
* **Chapter 2 Literature Review**: critically examines existing literature relevant to FL, HE, HE challenges, and encryption techniques approaches for this architecture (specifically xMK CKKS).
* **Chapter 3 Research Design**: outlines the chosen methodologies, research procedures, instruments, and analysis strategies used throughout the study (DF-KS integrated with FL).
* **Chapter 4 Implementation**: both DF KS and xMK CKKS theories are explained rigorously in detail with the algorithms of the code implementation.
* **Chapter 5 Results**: presents findings obtained from implementing the proposed methodologies, clearly structured according to research objectives.
* **Chapter 6 Analysis**: displaying a table that highlights main difference between the two approaches.
* A diagram of a diagram

  Description automatically generated**Chapter 7 Conclusion**: summarizes the project's main achievements, addresses its limitations, and provides practical recommendations for future research directions.

Figure 2: project outline

## Work Plan

The Gantt chart outlines the entire 2025 project plan, showing how tasks progress and occasionally overlap to reflect a realistic research workflow. It begins in February with the selection of the dataset and the design of the model, followed by a March focus on DF-KS encryption study. April and May represent a transition from implementation to exploring real-case datasets, while June and July focus on studying and integrating the xMK-CKKS method. August is dedicated to investigating probability attacks, leading into September’s comparative analysis of both approaches. Finally, October centers on writing the master thesis and preparing the presentation. By deliberately overlapping some activities, such as beginning preliminary studies while implementation is underway, the chart illustrates a more dynamic and organic schedule rather than a strictly sequential plan.

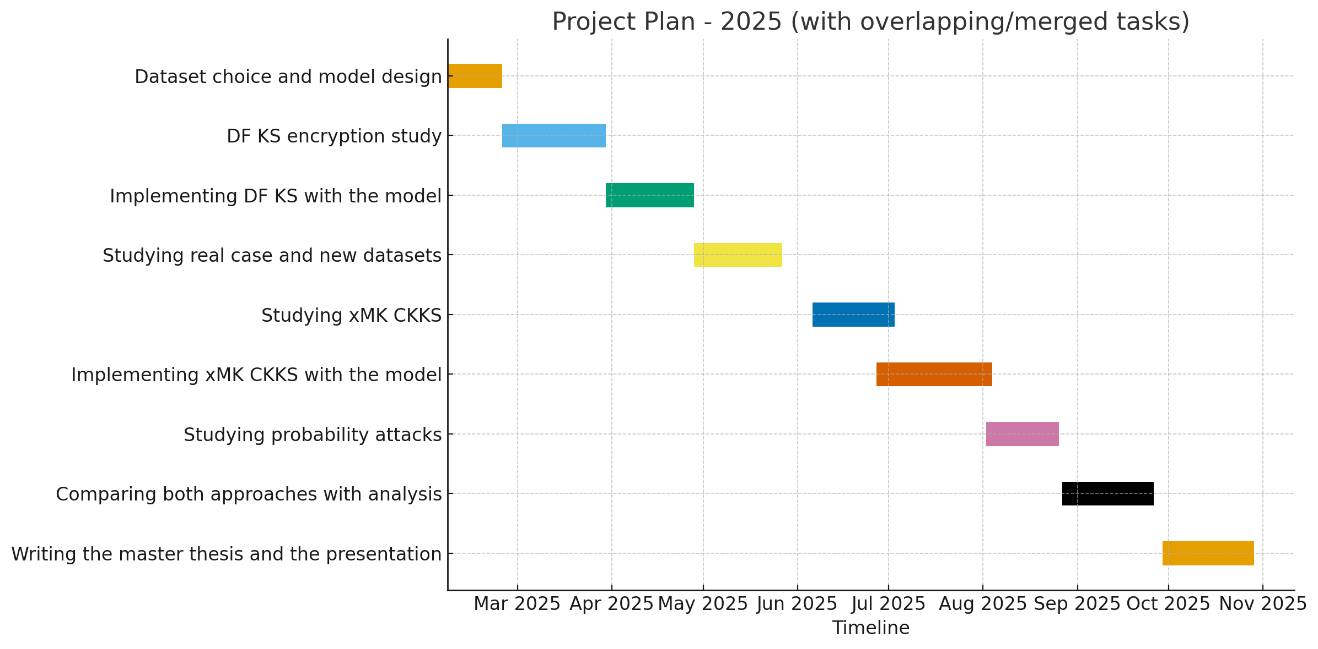


Figure 3:gantt chart

# Literature Review

This chapter begins with a historical background (section ‎2.1) and reviews literature on the following topics: standard architecture (section ‎2.2) this section will describe the standard architecture of FL with HE with it flows; xMK-CKKS (section ‎2.3) this section is critical for the research since this is the method that was studied and used to compare with the proposed solution. Section ‎2.4 highlights the implications from literature and develops the conceptual framework for the study.

## Historical Background

FL as a formal ML paradigm was popularized by McMahan et al. in “Communication-Efficient Learning of Deep Networks from Decentralized Data” (2017) [9], where devices train local models on private data and only share updates (not raw data) to a central server for aggregation. Earlier ideas in distributed and multi‐party learning existed, but McMahan et al. framed key challenges such as non-IID data, limited bandwidth, and client heterogeneity, and proposed the FedAVG algorithm.

The concept of HE dates back decades (e.g. partially homomorphic systems like RSA’s multiplicative property in 1978) [10]. The breakthrough of **Fully Homomorphic Encryption (FHE)**, enabling arbitrary circuits on ciphertexts, was first achieved by Craig Gentry in his 2009 PhD thesis “A Fully Homomorphic Encryption Scheme” (Stanford) [2], introducing the bootstrapping technique to control noise growth. Gentry’s scheme used ideal lattices and recursive “squashing” of the decryption circuit so that the scheme could evaluate its own decryption homomorphically.

The **Learning With Errors (LWE)** problem was introduced by Oded Regev in 2005 in his conference paper “On Lattices, Learning With Errors, Random Linear Codes, and Cryptography. [11]” In that work, he formalized the problem of recovering a secret vector from noisy linear equations and showed that solving (average‐case) LWE is a post-quantum scheme that promises to be safe against quantum computers. Over time, the LWE assumption became foundational for lattice-based cryptography, underpinning public-key encryption, key exchange, and FHE schemes. Variants such as Ring-LWE and Module-LWE augment LWE with algebraic structure to gain performance while retaining hardness under similar reductions.

In lattice‐based (especially LWE / Ring-LWE) FHE schemes, **Key Switching** [8] is a technique that transforms a ciphertext encrypted under one secret key (or a higher‐degree form) into a ciphertext under a (usually simpler) target key, without full decryption. To enable this, the scheme publishes key switching hints or evaluation keys that encrypt components of the original secret under the target secret. During evaluation, a partial homomorphic “decryption” is applied using those hints to rewrite the ciphertext.

## FL integrated with HE

A diagram of a mathematical model

Description automatically generatedThe concept of **FL** aligns from GDPR regulations, which prohibit personal data from being transferred outside specific regions. In this approach, each region maintains a local model (client) trained on its own dataset of a certain size. Clients train their models locally and then send their parameters (or gradients) to a global model (server). The global model performs a mathematical operation on the received data, typically a simple mean or a weighted average, and sends the aggregated results back to the clients. The clients then retrain their models using the updated weights and biases. This process is repeated for *R* rounds, where *R* is determined similarly to other hyperparameters (such as epochs, batch size, and learning rate) in neural network problems.

Figure 5: FedAVG architecture

The repetition of *R* rounds is necessary due to variations in data distribution across clients. For example, one client may produce entirely positive outputs, while others may generate more evenly distributed outputs. Repeating *R* rounds ensures that learning occurs from all clients, in a manner analogous to the backpropagation process in machine learning.

When clients send their parameters (weights and biases) to the server, the process is known as **FedAVG** [4], which is the approach adopted in this research. Alternatively, if clients send gradients instead, the process is called **FedSGD** [9], which has proven useful in neural network models because it reduces the need to transmit large numbers of parameters to the server.

However, FL itself presents a significant vulnerability: when clients send data to the server, there is a risk of data leakage. This highlights the importance of incorporating HEto enhance data security.

HE is a cryptographic architecture that enables mathematical operations—specifically addition and multiplication, to be performed directly on ciphertext:

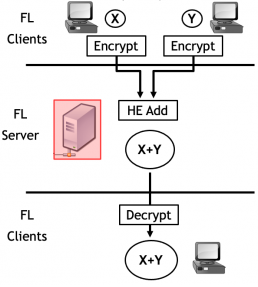


Figure 6: homomorphic encryption architecture (visit [developer.nvidia.com](https://developer.nvidia.com/blog/federated-learning-with-homomorphic-encryption/))

A fundamental requirement of HE is that plain-texts must be encrypted under the same key. Consequently, in symmetric encryption schemes, all clients generally share the same secret key, which the server must also use. In contrast, the more common approach is asymmetric encryption, where clients encrypt plaintext using a shared public key, and a corresponding private key is used for decryption. In this study, this method was briefly explored using TenSEAL and the built-in CKKS encryption scheme; however, it was not included in the comparative analysis or results.

A diagram of a graph

Description automatically generatedThis method introduces two critical risks: first, since all clients share the same key, compromising a single client’s key exposes the data of all other clients. The second risk, The clients are considered **honest-but-curious**; that is, they execute the machine learning algorithm faithfully but may attempt to infer or extract private information if the opportunity arises. To mitigate such risks, this standard architecture has been adapted in various ways by using a technique called xMK-CKKS (see Section 2.3 and 2.4).

Figure 7: FedAVG integrated with HE architecture

## MK CKKS

To explain about xMK CKKS, one must know before about MK CKKS and LWE techniques.

Consider *n* the dimension of **LWE** for a device , the vectors and are public, the vector (consists of very small numbers) and the number are generated randomly and kept in secret.

Consider the linear system:

This linear system consists of *N* equations with *N + 1* unknowns ( and vector ), which leads to infinitely many solutions and therefore no unique solution. Although this may initially seem like a nonsensical mathematical construct, it has a clear interpretation in cryptography. Here, represents the private key; knowing this value allows the system to be solved (that is, to decrypt the ciphertext). The solution of the system corresponds to the plaintext. Meanwhile, , which represents the public key, is used to encrypt the plaintext. This concept will become clearer when discussing MK CKKS. [7]

The Multi-Key CKKS (**MK CKKS**) scheme is a homomorphic encryption method based on the Learning With Errors (LWE) problem [6], specifically designed for Federated Learning models. In this scheme, each element in the vector is a polynomial from the ring , and the system has a dimension denoted by . The public key is represented as the private key as from , and corresponds to the noise vector drawn from the error distribution over Z. is a public number from . Let denote the plain text to be encrypted. As previously mentioned, .In this case is defined in the ring . The **encryption of**  under the private key for device is expressed as the ciphertext

(notice ), and from . **The decryption of**  using will be given by the scalar product:

Neglecting terms with , the result will be:

This method has an **additive homomorphism** property [6]. Let and be the ciphertexts of the plaintexts and from the devices and . The addition of the ciphertexts is defined by = (), it can be decrypted using .

Our team extended this architecture to demonstrate the **multiplicative homomorphism** property; however, the proof remains incomplete and could be addressed in future work.

Given and . The multiplication of ciphertexts could be defined as: . This may be decrypted using . Neglecting

However, these vectors have a dimension of 5, whereas it should be 3. This highlights the need for re-linearization, which is left for future work.

This architecture introduces a potential threat to the ciphertext, as the clients are *honest but curious*—they may exploit any opportunity to extract data from one another, which presents the importance of the xMK CKKS architecture [6].

## xMK CKKS

**xMK-CKKS** is like MK-CKKS, with one main difference: clients share a portion of the key and collaboratively decrypt the ciphertext through a server. In this architecture, clients use aggregated public keys along with their own private keys for encryption. However, decryption can only be performed by the server using all private keys collectively (**after aggregation**) [6]. This architecture follows the same setup as MK-CKKS, except that the public key used is denoted as . The number of devices is .

The plaintext of a device is **encrypted** as . (let )

This ciphertext is impossible to encrypt using one secret key , but it does possess **an additive homomorphic property** [6]. Let be the sum of the ciphertexts .

This term can be **decrypted** using which is called “the share”. The decryption of is:

As above, the team discussed **the multiplicative homomorphic** property and left the rest of the proof for future work. This may be given by the linearization of the vectors:  and (in case of two devices i and j only).

## Summary and Implications

While xMK CKKS may be resistant to plaintext–ciphertext attacks, it has a significant limitation: only the server can encrypt the aggregated ciphertext, requiring clients to trust the server not to leak sensitive information. In contrast, the method proposed in this research eliminates that trust requirement — the server is unable to perform any decryption, and each client holds their own key to decrypt the ciphertext independently.

# Research Design

This chapter describes the design adopted by this research to achieve the aims and objectives stated in section ‎1.3 of Chapter 1: design an encryption technique that will solve key compromise vulnerability, and honest but curios clients concept. Section ‎3.1 discusses the methodology DF-KS used in the study, the stages by which the methodology will be implemented, and the research design; section ‎3.2 details the participants in the study; section ‎3.3 lists all the instruments to be used in the study and justifies their use; section ‎3.4 outlines the procedure used and the timeline for completion of each stage of the study; section ‎3.5 discusses how the data was will be analyzed; finally, section ‎3.6 discusses the ethical considerations of the research and its potential problems and limitations.

## Methodology and Research Design

### Methodology

The DF integrated with KS technique is discussed in the article “An Efficient FHE Scheme to Secure Cloud Computing” [7]. This section elaborates on the theoretical principles of the technique and examines its integrability within a federated learning framework.

The following list is the setup for DF scheme [3] adopted in the project:

* Security parameter
* Private modulus with .
* Public modulus .
* The dimension of the cipher text . Many values were tried for this parameter. .
* The plain texts . Consider there are two clients for the sake of this study.
* The secret key such that exists and

The number is the scale factor parameter specifically used for this project.

To begin with, is the weight generated by the client , which is a real number. is the new aggregated weight:

Which will be done with ciphertexts on the server side.

represents the relative size of data of client (data point):

However, the DF technique works only with integers. For this reason, the encryption will proceed with and . In this section, the DF-KS technique will be proven as a homomorphic technique. Scaling the weights and data point won’t affect this homomorphic property, and this can be proven by the reasoning below:

Dividing each term by in will lead to:

On the other hand:

Which means dividing by will lead to the same result.

Let and . and must belong to . From here, the plain texts will be addressed as and the ciphertext as . The DF **encryption** is given by two steps:

1. is randomly decomposed into elements [] such that
2. The encryption of under the key is given by the cipher text:

Which can also be expressed in a polynomial way:

The **decryption** procedure is performed by multiplying the term of by to get . The plain text will be retrieved by .

The second step of the encryption is **modified** by using a vector of secret invertible keys instead of powers of one invertible key (which was ). The cipher text in this case becomes:

It can be expressed as a multi-variable polynomial:

The decryption after this modified step is performed by multiplying term of by to get . The plain text will be retrieved by .

Both the original and the modified schemes exhibit **homomorphic properties**. Since the objective of this work is both theoretical and comparative, homomorphic property will be discussed for both schemes to highlight their respective characteristics. However, as only the modified scheme was implemented, the formal proof and experimental validation will focus exclusively on this version.

Original version: Consider two cipher texts and the encryption of the plain texts and respectfully under the **same** key . The **addition property**:

Decrypting this number under r (multiplying term by ) will result with , then can be retrieved from

For the **multiplication property**: multiplying by will result in the same terms as multiplying by (polynomial multiplication, not term by term) then multiplying the term by . The degree of is . As a proof consider and . and . (same terms in multiplication process)

Modified version: Consider two cipher texts and the encryption of the plain texts and respectfully under the **same** key . The **addition property**:

To decrypt this number, it must be multiplied by which will result with Then can be retrieved.

For the **multiplication property**:

Terms of the same coefficient must be added together (regrouped) (reminder that the operation is a multiplication between two multivariable polynomials where all variables have the degree 1 as the highest degree). For example: and become (factorizing).

The degree of is = . This is discussed in the book “Ideals, Varieties, and Algorithms” [12]. To decrypt , an extended version of the key of dimension is used:

In the federated learning setting, each client encrypts its weights and biases using a unique encryption key. This approach mitigates key compromise risks and addresses the honest-but-curious threat model. However, it introduces a new challenge: the loss of homomorphic properties, as each client employs a distinct key. To overcome this limitation, KS is utilized [8].

### Research Design

Outline the research design (e.g., quantitative, qualitative). If quantitative, spell out the independent, dependent and classificatory variables (and sometimes formulate an operational statement of the research hypothesis in null form so as to set the stage for an appropriate research design permitting statistical inferences). If qualitative, explain and support the approach taken and briefly discuss the data gathering procedures that were [will be] used (observations, interviews, etc.)

## Participants

Give details of who were [will be] the participants in your study (including, if applicable, sample type and size, reasons for the number selected and the basis for selection).

## Instruments

List and briefly describe all the instruments (e.g., tests, measures, surveys, observations, interviews, questionnaires, artefacts) [to be] used in your study for data collection and discuss their theoretical underpinnings, that is, justify why you used [will be using] these instruments. So that the line of argument is not broken, it is useful to place copies of instruments in Appendices to which this section can refer.

## Procedure and Timeline

Outline the procedure across and within the techniques [to be] used in your study for collecting and recording data. This could include how, when (in what order) and where the instruments were [will be] administered (for example, field, classroom or laboratory procedures, instructions to participants or distribution of materials) and how the data was [will be] recorded. Include the rationale for the procedures used. If the study was [is to be] done in stages, give a timeline for the completion of each stage.

## Analysis

Discuss how the data was [will be] processed and analysed (e.g., statistical analysis, discourse analysis). This section needs to link the analysis of the research to the methods and demonstrate why this is the best approach to analysis. For qualitative research, justification needs to be provided for methods such as coding and dealing with divergent data. For quantitative research, justification of the choice of statistics and the expected results that they will provide [confirmation document] should be described. There should be enough detail for the reader to replicate the analysis. For example, “NVivo or SPSS will be used” is not adequate. Rather, the approach to coding, including how categories were [will be] derived and validated, how the data was [will be] structured, and specific analytical techniques applied, should be included.

## Ethics and Limitations

Outline the ethical considerations of the research and any [potential] problems and limitations (weaknesses), as well as any [anticipated or actual] threats to the validity of the results.

# Implementation

This part could be made of one or more sections. You should:

* Explain how the solution blocks are implemented.
* Specify the technical environment, used resources (hardware and software).
* Point out techniques, tools, and skills used (algorithms, mathematical development, programming, simulators, etc.)
* Provide required resources for the implementation
* Give details about eventual prototype and/or simulation
* Mention difficulties/obstacles encountered during the implementation phase and how you managed to overcome them.

This part is not:

* An installation manual (keep it to appendices)
* User’s guide (keep it to the appendices)
* A work journal
* A listing (only put pertinent code sections if necessary)

# 

# Results

Chapter 4 details all the results of your study. You can put some analysis of the results here, but generally just the results are presented, without interpretation, inference, or evaluation (which will be in Chapter 5). The results should be linked inextricably to the design – describe what happened factually and unemotively. However, in certain historical, case-study and anthropological investigations, factual and interpretive material may be interwoven rather than being presented as “findings”.

Include a paragraph at the beginning of the Results chapter outlining the structure of the chapter. The results should be reported with respect to furnishing evidence for your research question(s) as outlined in Chapter 1. Thus, you might choose to use headings that correspond to each main question of your hypothesis/objectives from Chapter 1 and/or your theoretical framework from Chapter 2. Or you might organise your results in terms of the stages of the study (if applicable).

Present the findings/results in tables or charts when appropriate.

# Analysis

Chapter 5 contains a full discussion, interpretation and evaluation of the results with reference to the literature. This chapter can also include theory building.

As with the previous chapters, include a paragraph at the beginning summarising the structure of the chapter. Organise the chapter in terms of the objectives of the study and/or the theoretical framework. For each objective, discuss the results with reference to the literature, for example, the similarities/differences to the findings in the literature review. Develop theory or models from this comparison and evaluation.

It can be useful to check your literature and try to find a place for as much of the literature as you can. If you find that a section of your literature can not be used in this chapter, it may be useful to consider the pertinence of this literature and reduce the space in the literature chapter given to it.

Thus your research outcomes are tied together in relation to the theory, review of the literature, and rationale.

# 

# Conclusion

This chapter contains conclusions, limitations, and recommendations – so what is the theory? Where to from here? What are the practical implications? Discussion of where the study may be extended.

Again, the chapter should begin with a summary paragraph of the chapter structure. The opening section(s) of the chapter should be a brief summary of everything covered so far. Follow this with your conclusions. This is the “so what” of the findings – often the hypothesis/research question(s) restated as inferences with some degree of definitive commitment and generalisability, and the raising of new and pertinent questions for future research. You could include a final model of the theory.

It can be useful to use the purposes from Chapter 1 as an organising structure for this chapter. The chapter should also include a discussion of any limitations of the research, and should end with your final recommendations – practical suggestions for implementation of the findings/outcomes or for additional research.

References

|  |  |
| --- | --- |
| [1] | U. -. ESIB, "Masters in AI Project - Healthcare AI," Mar Roukoz, 2025. |
| [2] | C. Gentry, A FULLY HOMOMORPHIC ENCRYPTION SCHEME, Stanford University, 2009. |
| [3] | J. Domingo-Ferrer, "A Provably Secure Additive and Multiplicative Privacy Homomorphism," in *ISC 2002 (the Information Security Conference)*, Sao Paulo, 2002. |
| [4] | D. P. E. Z. E. B. W. Malik Muhammad Qirtas, "Privacy Preserving Loneliness Detection: A Federated Learning Approach," in *IEEE International Conference on Digital Health (ICDH)*, Barcelona, 2022. |
| [5] | E. Union, *General Data protection Regulation (GDPR),* Brussels: Official Journal of the European Union, 2026. |
| [6] | S.-A. N. S. S. X. L. Jing Ma, "Privacy-preserving Federated Learning based on Multi-key Homomorphic Encryption," p. 16, 2021. |
| [7] | A. E. S. M. C. Khalil Hariss, "AnEfficient FHE Scheme to Secure Cloud Computing," in *16th International Joint Conference on e-Business and Telecommunications*, Prague, 2019. |
| [8] | Z. B. a. V. Vaikuntanathan, "Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages," in *CRYPTO 2011 — 31st Annual International Cryptology Conference*, California, 2011. |
| [9] | E. M. D. R. S. H. B. A. y. A. H. Brendan McMahan, "Communication-Efficient Learning of Deep Networks," in *the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)* , Florida, 2017. |
| [10] | A. S. L. M. A. Ronald L. Rivest, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," in *AFIPS National Computer Conference*, New york, 1978. |
| [11] | O. Regev, "On Lattices, Learning with Errors, Random Linear Codes, and Cryptography," in *37th Annual ACM Symposium on Theory of Computing (STOC)*, Baltimore, 2005. |
| [12] | J. L. a. D. O. D. A. Cox, Ideals, Varieties, and Algorithms 4th edition, Cham, Switzerland: Springer, 2015. |

Appendices

* + 1. Title

Start each appendix on a new page. Place appendices in the same order as they are referred to in the body of the thesis. That is, the first appendix referred to should be Appendix A, the second appendix referred to should be Appendix B, and so on. Appendix formatting can be different to the main document.

# Some writing guidelines:

* Never forget to provide citation whenever you paraphrase and/or summarize someone else's ideas or you directly quote someone’s words. This is discrediting for the team and constitutes **plagiarism**.
* Information should be presented in a logical and interesting way.
* Avoid misspelling and/or grammatical errors (use an Auto-correct tool).
* A report of excellent quality should feature formatting strengths for good legibility such as uniform format using the provided template, consistent structure (headings, sections, etc.), consistent numbering, clear/labeled figures/ tables/ graphs/ equations, good referencing, etc.