

DEVELOPMENT OF A SECURED SYSTEM USING GENTRY HOMOMORPHIC ENCRYPTION ALGORITHM

Adeola, O. O., Amao, K. A., Ojo, N. A., Kilani, S. O., Ademola, L. O.
and Abiodun, G. T.

Department of Computer Science, Oyo State College of Agriculture and Technology, Igboora
Corresponding Email: ooadeola@gmail.com, Phone Number: +234 703 326 1184

Abstract

The propagation of the digital technologies has made data security and privacy to be important concerns. The study is focused on how Gentry Homomorphic Encryption Algorithm can be utilised to increase such protection. In declaring encryption as computational variables, the framework by Gentry computationally performs operations without decryption hence maintaining confidentiality during the computation. Complications are raised through traditional encryption schemes that requires that the encryption has to be decrypted before the analysis is conducted, hence it becomes vulnerable. Even though the homomorphic method offers an attractive solution, its actual-life implementation is still narrowed by major computational centre of gravity overheads and scalability aspects. Despite these challenges, the approach still could be relevant to the field of healthcare, finance and the government because real-time and tamper-proof data analysis is a necessity in such settings. In this wise, the paper provides a safe system built with the help of homomorphic algorithm, critically evaluate the failures of current methods of encrypted transmission systems and the direction to find ways to improve privacy. The metrics used for the evaluation of the results give accuracy, precision, recall, and F1-score 99 %. The outcomes prove that the suggested implementation has a strong encryption, decryption, and classification accuracy in the particular scenario. Analyzing the possibility of using homomorphic encryption and outlining the opportunities of introducing it into the modern world of digital environment. The study can advance the future development of data security and privacy norms in industries that require intensive use of data.

Keywords: Algorithm, Data Security, Privacy, Encryption, Gentry Homomorphic

How to Cite: Adeola, O. O.; Amao, K. A.; Ojo, N. A.; Kilani, S. O.; Ademola, L. O. and Abiodun, G. T. (2025). Development of a secured system using gentry homomorphic encryption algorithm. *OYSCATECH Journal of Engineering and Environmental Studies*, 3(1), 29-41. <https://oyscatechpublications.org/>

Introduction

The digital technologies have been spreading; this has increased the apprehension regarding the privacy of data. In particular, the idea of cloud and big-data frameworks is the reminder that there is a need to have secure mechanisms of processing. Ordinary cryptographic systems place the encrypted data in a situation where it might fall at the hands of possible attackers, but since with the help of homomorphic

encryption and, absolutely, Gentry homomorphic encryption, calculations can be executed on encrypted data, decryption is not necessary prior to taking further action.

Gentry homomorphic encryption is one of the instruments to deal with the privacy issues of modern context that was proposed in 2009 and is effective in the increasing realms of IoT, AI, and ML. This framework protects a confidentiality even in the process of

computing since it supports calculations on the encrypted information. With increased reliance on data in most spheres of the global economy, homomorphic encryption by Gentry can become the pillar of privacy in society where access to information is also valued.

The encryptions of Gentry are homomorphic encryptions, which are mathematical equipment that allows the execution of data in an encoded form that does not reveal apparent text. It can be applied to the sphere of healthcare, finance, the military and the government. Introduction of data mining in healthcare can help to examine the records of patients to identify the trend of disease-outbreak and banks could use it to carry out risk analysis and curb fraud. However, a non-insignificant computational overhead of the method is a serious obstacle.

Although hardware and software optimization to homomorphic encryption have been enhanced, homomorphic encryption still consumes higher computing resources compared to the normal cryptographic processes. As a result, adoption in mainstream industries has been quite lower, most of which have been attributed to inherent complexity of the subject or to the specialised skills that it requires. The pressure on regulatory authorities and the need of data-privacy are expected to increase, and in that context, secure mechanisms are needed; a field where encryptions like the one proposed by Gentry (2009) should only gain momentum.

In summary, regarding the research of Gentry, the homomorphic encryption is an important breakthrough in the study of data security that introduced a different way to safely compute data that was encrypted. Despite the issue of computational performance and the lack of mainstream adoption, its possibilities of transformation of the data-driven industries, as well as improving more comprehensive privacy protection, make it the de facto standard in the modern digital age.

Statement of the problem

The increased use of data in the private and the public space has further clashed with the issue of privacy and security. Traditional encryption mechanisms are mostly sufficient in protecting information with storage and transmission and at the same time they require the procedure of decryption any computation or analysis. Such a decryption process creates immense risks because valuable information is not sufficiently secured against possible hacks and attacks. Furthermore, increasing demand of real time data processing and its analysis, especially in the sphere of finance, healthcare and government points to the necessity to provide the system where the presence of the encrypted information will be maintained even during the period of its computational processing. A homomorphic encryption scheme developed by Gentry will provide a possible solution; however, that solution does not come without its challenges considering that its implementation is complex and also that it is costly in terms of its computational expenses alongside the fact that no one by far has adopted its use.

Aim and objectives

The aim of this study is to develop a secured system using Gentry Homomorphic Encryption Algorithm. The objectives include:

- i. Analyzing the limitations of current encryption techniques.
- ii. Implementing the Gentry Homomorphic Encryption Algorithm in a practical scenario.
- iii. Providing recommendations for enhancing data security and privacy using homomorphic encryption.

Significance of the study

Another fundamental work on the modern cryptographic study is homomorphic encryption that was formulated by Craig Gentry in 2009 due to the response to an outstanding issue of data security i.e., maintaining confidentiality throughout the

computation and analysis. Ordinary encryption systems will not be able to offer the same level of protection in such a scenario that information will be subjected to tampering. In case the approach shows itself practicable, homomorphic encryption can be the starting point in the data privacy of such industries as healthcare and finance, thus securing personal health records and financial processes. The insights into the empirical data and the theory created in the course of this study will be later used to develop data-privacy legislation and policies, which will respect the opposing needs of confidentiality and accessibility.

Related works

As it is defined in the present-day discourse, data is a crucial strategic resource in a variety of industries, and, at the same time, its explosive development leads to severe security and privacy issues. Protecting confidentiality, integrity and availability have become more urgent than ever especially in the face of growing threats being inflicted in the cyber space and by data breaches. Data safety and secrecy thus are the points of entry which require strong guarding against intrusion, corruption and expropriation. The study by Anderson (2019) is aimed at building reliable distributed systems and it pays much attention to security engineering. Anderson poses the problem of being able to create a system that is reliable despite a myriad of vulnerabilities and has relied on both qualitative and quantitative methods such as case study and theoretical examination that explain how a system can be designed with the support of security mechanisms. The findings suggest that highly integrated security systems are crucial in the loss or risk of loss of information, unauthorised access and data breaches especially in distributed networked environment.

In their discussion on symmetric cryptography, Boura and Dunkelman, (2024) provide a detailed analysis of the Advanced Encryption Standard (AES). The article they

wrote brings forward the problematic aspect of maintaining a strong encryption against the ever-changing attacks techniques. They apply mathematical demonstration and trial to check the cryptography strength of AES. The results draw attention to the resilience of AES to brute force and its high usage because of its cost-effective implementation in either a hardware or software setting.

Brakerski and Vaikuntanathan (2014) discuss approaches towards practical realization of the fully homomorphic encryption (FHE) with the help of lattices. The main issue they solve is that classical approaches to FHE are computationally (in)efficient and therefore not feasible to implement. They propose to utilize Learning With Errors (LWE) problem to devise a more efficient scheme through its utilization. Their practices include an algorithmic enhancement and theoretical study. The findings present that the new method contributes to minimizing the complexity of implementing FHE reduces to some considerable degree, which makes it more viable to use it in secure computation practice.

ElMoudni and Ziyati (2025) conduct a systematic review of cloud data storage security pertaining to the issue that emerges alongside the increased dependency on cloud infrastructure. In their research, they employ an extensive literature review to determine and discuss numerous security problems, such as the information leakage of security breaches, unauthorized entry, and adherence to standard regulations. They come to the conclusion that though cloud storage is very convenient, it must be secured with very strong mechanisms to guarantee integrity and privacy of data. They also propose on future research course, such as improvement on encryption can be done, and control access techniques.

The other area covered by Gentry (2009) is building fully homomorphic encryption on ideal lattices that allows one to compute over encrypted data without decrypting. The issue being addressed is that there are no practically-

used implementations of homomorphic encryption. On the one hand Gentry uses the theory, and on the other hand lattice-based cryptography give a concrete answer. The findings are the landmark of the research since it provides the most viable variant of homomorphic encryption implementations that opens the door to the secure data processing in privacy-sensitive use cases.

In more details, the paper by Green (2021), provides the contemporary means of cryptography and privacy emphasising the complex role of cryptographic applications in its relation to user privacy. In his discourse, Green emphasises that cryptography may be used to safeguard privacy, particularly where there is the question of digital transactions and exchanging information. In the paper theoretical inquiry and case studies are used to draw attention to the issues of maintaining a balance between the cryptographic protection and user privacy. These results suggest a need to develop the cryptographic habits to adhere to the new privacy trends in an ever more interconnected world.

Halevi (2020) explores the process that takes the concept of homomorphic encryption from the theoretical to the practical sphere with regards to the practical application of the technique in the privacy-preserving computations domain. Halevi singles out the problem of the few real-life usages of homomorphic encryption and the complexity that is involved. In an illustrative way, the paper demonstrates by logical experiment and case studies that homomorphic encryption may be used in domains such as secure cloud computing and on privacy-preserving data analysis. Its possibly is noted in the results, but the efforts on efficiency still should be present. Working on improving the efficiency of a conventional ElGamal encryption system and the degree of its security, Hussein and Abdulla (2019) introduce an effective ElGamal cryptosystem. They determine problems such as inefficiencies and weaknesses of the previous versions and offer

a version that can be optimised by refining the algorithms. The approaches that they take are mathematical proofs and performance benchmarking. The findings indicate that their adapted cryptosystem is a promising solution of secure communication systems as their cryptosystem changes signifies a great increase in the speed and security of encryption.

In their comparative paper, Idoko *et al.* (2024) compared the healthcare Data Privacy and Security Rules in the US and Nigeria. Their study deals with the variance in the standards and practise in the data management of healthcare between the developed countries and the developing countries. This paper uses a qualitative approach with the help of such techniques as interviews and document analysis of the difference between regulatory frameworks. Its results are indicative of the fact that despite the US having more developed privacy practises, Nigeria is a country with underlying issues on legal enforcement and adherence. This paper points to the need to improve the standards of regulations to protect healthcare data in Nigeria.

Lewis and Baker (2018) re-examine the data breach that took place in Equifax, which is one of the biggest cybersecurity cases that leaked the personal information of millions of people. In their paper, they established the sources of the breach which revolved around failure by the Equifax to address known vulnerabilities in a timely basis. They also used case study analysis and security audits to bring to the fore the issue that went wrong and offer ideas that can be used to enhance the application of cybersecurity in large firms. The findings have shown that more triggering cybersecurity responses and enhanced incident response are the keys to averting comparable breaches in the future.

In her article, Smith (2021) discusses the role that big data plays in cybersecurity, namely, how the continued expansion of big data in business and government can create new

security issues. According to Smith, data loss, insider attack and data falsification are the threats of big data system. The theoretical and real-world case studies are used in the research. Based on the findings presented, it can be established that although big data analytics can be of value, they need to be complemented with strong security strategies that minimize the vulnerability of sensitive data to exploitation.

Understanding homomorphic encryption

Homomorphic encryption forms a revolutionary cryptographic approach that permits calculations to be carried out on encrypted information. Unlike the normal architectures, whereby decryption has to be performed before processing, this system has the capability of maintaining the confidentiality of any information during its existence. As an example, the sensitive financial data can be processed by the third-party services and the plaintext behind them is still hidden.

Features and advantages of Homomorphic Encryption

- i. Privacy-Preserving Computation: Homomorphic encryption keeps medical records or financial transactions encrypted during the computing processes and disclosure is avoided even when the computing operations are made in insecure systems such as cloud infrastructure.
- ii. Operational Flexibility: The scheme does not restrict any complex mathematical operations, such as addition or multiplication, to encrypted data, making it appropriate to many applications, ranging to secure voting systems or privacy-preserving machine learning.
- iii. Scalability: The design is also massive in that it supports large datasets, and hence can be used as an advantage by companies carrying out massive datasets with sensitive data.

- iv. Enhanced Security: Lack of decryption limits the field of the possible attacks upon the potential break-in which is especially important in some specific industries like healthcare where data security is the main priority.

Materials and method

The present study has the methodology as follows; utilization of systematic method of collection of data, critical analysis of the drawbacks of the current existing system and an explanation of the alternative existing proposed system with the diagrams supporting the same.

Method of data collection

Data for this work were gathered from the various sources.

Data sources

The considered data belongs to the library of the computational Cryptography that has long been a benchmark of the sphere of data protection and other areas of application.

First, the current work reprocesses the dataset and takes it through a series of pre-analytical activities, i.e., cleaning, transformation, and enrichment.

Feature extraction: This will be carried on the reprocessed database, as it is an essential part of the data preprocessing stage and allows modeling the data more precisely and getting better data insights to know more about the potential patterns that may somehow present within the given data.

Mathematical Foundations

Homomorphic algorithms allow to avoid decryption only after those cryptographic calculations can be performed on encrypted data. They provide on a fundamental level mathematical language of privacy-preserving computing. The given mathematical notations would have general form of homomorphic algorithm as follows:

Let an encryption scheme G be considered to refer to the plaintext input and $G(i)$ shall be considered as encryption of i . That homomorphic property can be defined as:

i. Addition Homomorphism. When such schemes are admitting addition it can be represented as:

$$G(i + j) = G(i) \oplus G(j) \quad (1)$$

in which, $\oplus$ denotes the operation given in the encryption domain.

ii. Multiplication Homomorphism. In support of schemes that support multiplication we have:

$$G(i * j) = G(i) \otimes G(j) \quad (2)$$

in which $\otimes$ is the operation as defined in the encryption page.

iii. General Homomorphic encryption scheme. The encryption scheme G is said to be homomorphic when the relation below is satisfied:

$$G(f(x)) = f(G(x)) \quad (3)$$

In that case we have $f(x)$ as any possible operation allowed on the original text x and $f(G(x))$ to be the same on inside the encrypted world.

Such property is essential to privacy-preserving computation, since it also allows working directly on the encrypted data, which protects sensitive information against attacks during processing.

Homomorphic Lattices

This homomorphic encryption scheme is a type of encryption that uses lattices as their conceptual grid on which to perform secure computations on secure data. Under this paradigm, the scheme of encryption is modeled in a lattice identified as L . Such a scheme may be represented by the following equations:

1. Encryption

Let d be the plaintext message and (r) be a random noise term. The encryption (G) of a message (d) under a public key (pk) is given by:

$$[G(pk, d) = (b_1, b_2) = (a \setminus b.pk + r, d + a \setminus b.ext\{noise\}) \setminus] \quad (4)$$

where $(a \setminus)$ is a lattice vector, and the noise term ensures security by adding uncertainty to the encryption.

2. Homomorphic Operation

$$[G(pk, d_1 + d_2) = (b_1, b_2) + (b_3, b_4) = (b_1 + b_3, b_2 + b_4) \setminus] \quad (5)$$

For multiplication, the operation is represented as:

$$[G(pk, d_1 \setminus b.d_2) = (b_1, b_2) \setminus b.(b_3, b_4) = (b_1 \setminus b.b_3, b_2 \setminus b.b_4) \setminus] \quad (6)$$

where (d_1, d_2) are the plaintexts and (b_1, b_2, b_3, b_4) are the ciphertext. Homomorphic property is naturally achieved by the lattice structure thus allowing one to perform cryptographic functions on encrypted messages without having to decrypt them.

Bootstrapping

This is a process applied in Fully Homomorphic Encryption (FHE) in order to counter the intrusion of noise in the ciphertexts and, in turn, to maintain the accuracy of computation. Practically, the technique re-encrypts the ciphertext so as to reinstate the performance ability once a set of homomorphic operations have been undertaken. The mathematical model can be formalized as the following one:

1. Homomorphic Operations Noise Implosion

The ciphertext develops statistical noise after every operation of homomorphism, be it addition or multiplication.

$$\setminus ext\{Noise\}(b) = ext\{Noise\}(b) + ext\{Noise\}(op) \setminus] \quad (7)$$

where (b) is the ciphertext and (op) represents the homomorphic operation

2. Bootstrapping

Bootstrapping process involves performing homomorphic decryption of the ciphertext, a re-encryption process and thereafter reduction of the noise that has accumulated.

Mathematically it is described in the following way:

$$\begin{aligned} & \setminus [\\ & \quad G(pk, d) \\ & \quad \rightarrow ext\{Decrypt\}(b) \\ & \quad \rightarrow ext\{Re-encrypt\}(b') \\ & \quad \rightarrow G(pk, d) \\ & \setminus] \end{aligned}$$

The operation can be viewed as:

$$\begin{aligned} & \setminus [\\ & \quad ext\{Bootstrap\}(b) = \\ & \quad ext\{Reencrypt\}(b', ext\{ReduceNoise\}(b))(8) \\ & \setminus] \end{aligned}$$

Reduced noise term in homomorphic encryption determines that consecutive homomorphic operations are only subjected to minimal ciphertext quality degradation.

Evaluation

Evaluation metrics forms a relevant tool in assessing the performance of machine-learning models and algorithm. The choice of the appropriate metrics depends on the task as well as on goals of the analyses under consideration.

- i. Accuracy: Accuracy refers to a ratio between the availability of accurately classified instances to the total number of instances. This metric may not be misleading in the case of balanced datasets, but they may be misleading in imbalanced datasets.
- ii. Precision: Precision is the percentage of correct positive prediction of the total positive prediction and thus it is the ability of the model to be accurate in forecasting the positive cases.

iii. Sensitivity (Recall): Recall, sometimes also called sensitivity, is a ratio of true positive predictions to all the positive cases presenting the capability of the model to identify all the positive cases.

iv. F1 score: F1 score balances both precision and recall because it is the average between the two scores. This statistic is especially useful in the circumstances where there are non-uniform distributions of classes.

System design

A conceptual model is given below which defines the flow of interactions of the system users and system component which make up the whole system. This framework helps to carry out the prototype in a systematic way.

System requirements

- i. An operating system of windows 8 or above.
- ii. JupyterLab graphical user interface and notebook-based environment development.
- iii. Python execution environment that lets one run Python programs.

Hardware Requirements

Minimum Pentium Cpu architecture.

- i. 2GB RAM/ or more.
- ii. Hard disk of 50 Gb or more.

System Modeling

The system that is explained in this write-up is based on three libraries:

- i. The Cryptography and the embedded homomorphic dependencies
- ii. Fernet in the generation of encryption functions, as well as encryption keys.
- iii. Tkinter in the Graphical user interface (GUI).

The main innovation in the system is the use of homomorphic algorithm, equipping the data encryption operations. The given homomorphic encryption method forms a paradigm shift in the secure management of sensitive data with regard to existing issues in data confidentiality and utility.

Homomorphic key

The diagram in figure 3.1 below illustrates the application process of homomorphic encryption with primary application as pursued with the help of a homomorphic key. Within this scheme, the key that is homomorphic allows performing of the

encrypted data (i.e., addition and multiplication) without decryption. The encrypted data retains the confidentiality of the data and during the process of computation secure operations are made in the ciphertext itself.

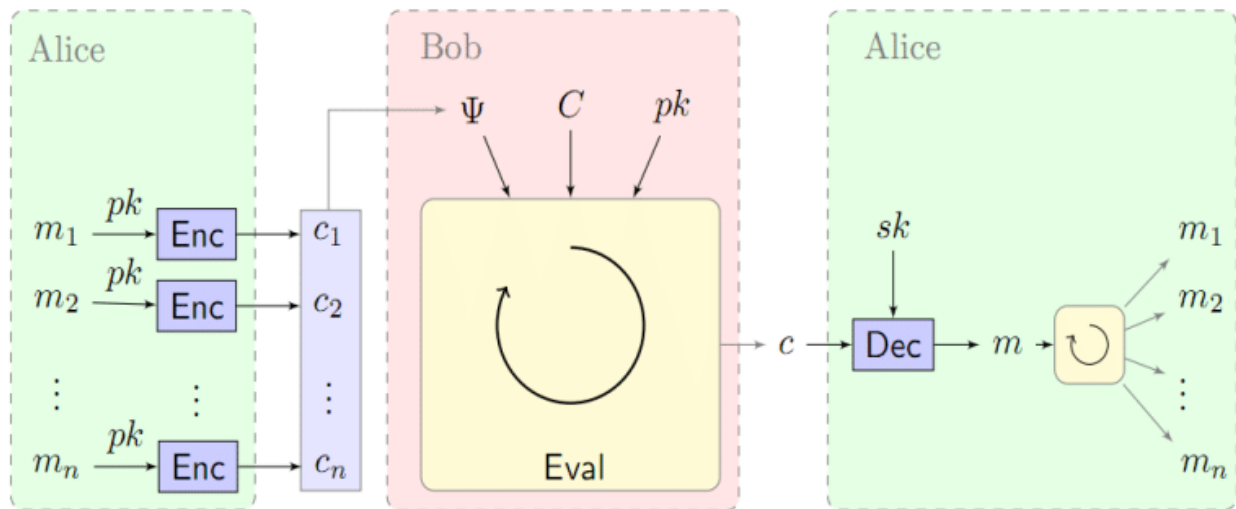


Fig. 3.1 Homomorphic key example (Source: Benzekki et al., 2016)

System flowchart

The system shown in the figure 3.2 below defines the steps of sequence in a scheme of homomorphic encryption. It starts with an encryption run of plaintext data with the use of homomorphic key. The cipher is then

subjected to computation either through addition or multiplication operations and the output is recovered by the procedure of decryption. The data will be encrypted all through the stages thus ensuring that the whole procedure is secure.

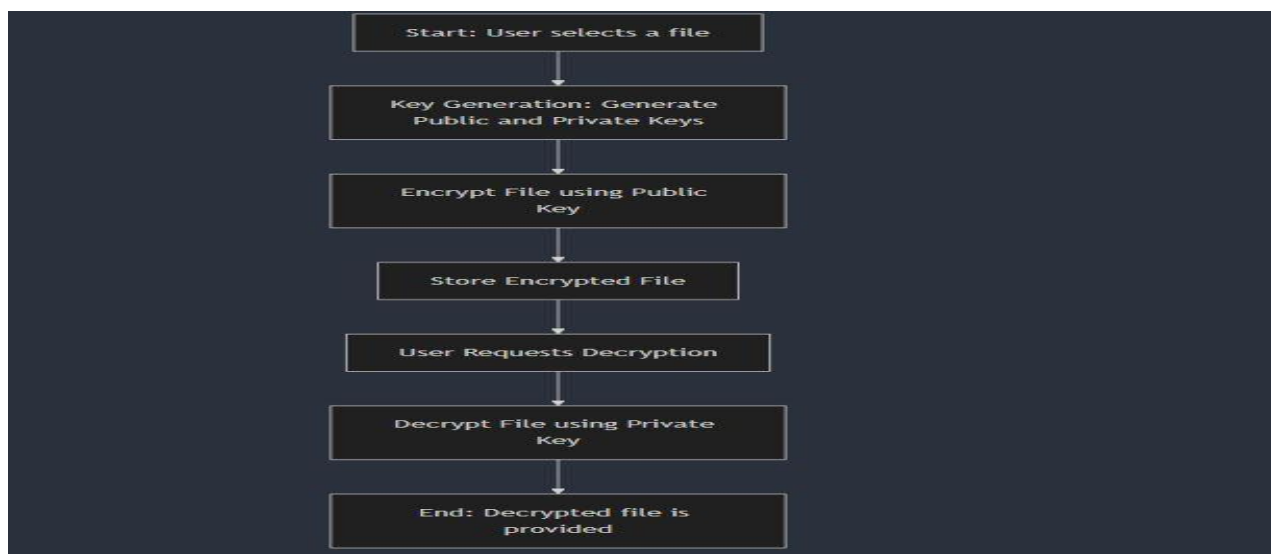


Fig. 3.2 System flowchart

System architecture

Herein, there is a system architecture diagram, Fig 3.3, that aims to depict the architecture of homomorphic encryption implementation. The diagram represents the relationship between the following constituent elements: data input, encryption modules, processing system that is characterized with privacy and confidentiality at every phase of data transfer.

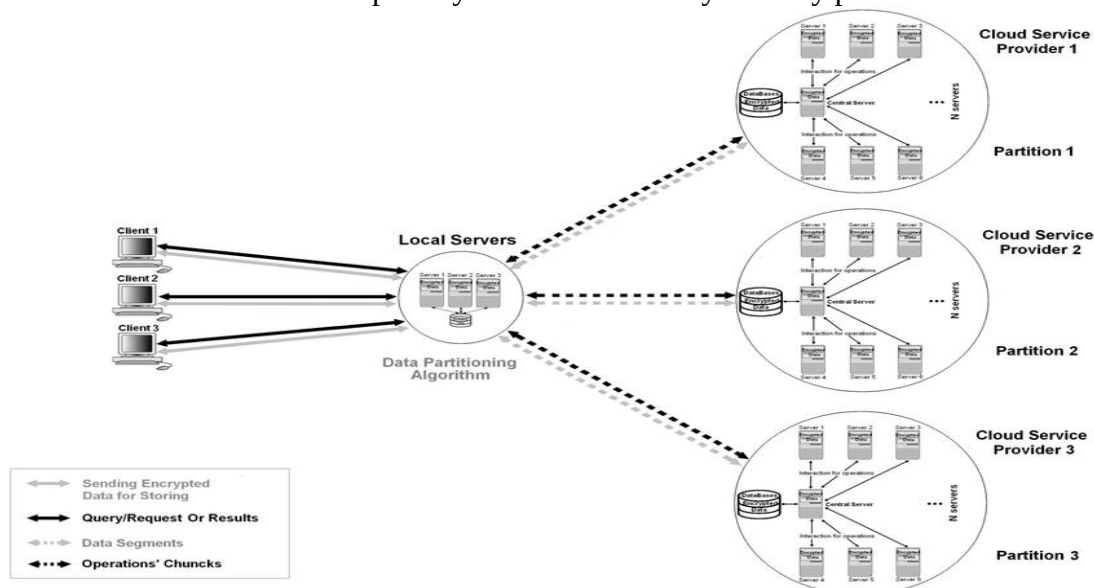


Fig. 3.3 System Architecture (Source: Benzekki et al., 2016)

Results and discussion

The deliverable of this step is a set of source code files, which are supplemented with the documentation that improves the readability of code. The steps towards the implementation of the system were as presented below:

- Training the Model: used the training data to train the machine-learning model. This involved provision of the data to the model and optimizing the parameters so as to reduce the difference between the expected and the actual results.
- Model testing: On training of the model, the very same process was applied on the testing data to evaluate performance. The test inputs were used upon the trained model.
- Performance assessment: Calculated a variety of assessment measures, to ascertain the performance of the model on the test dataset.

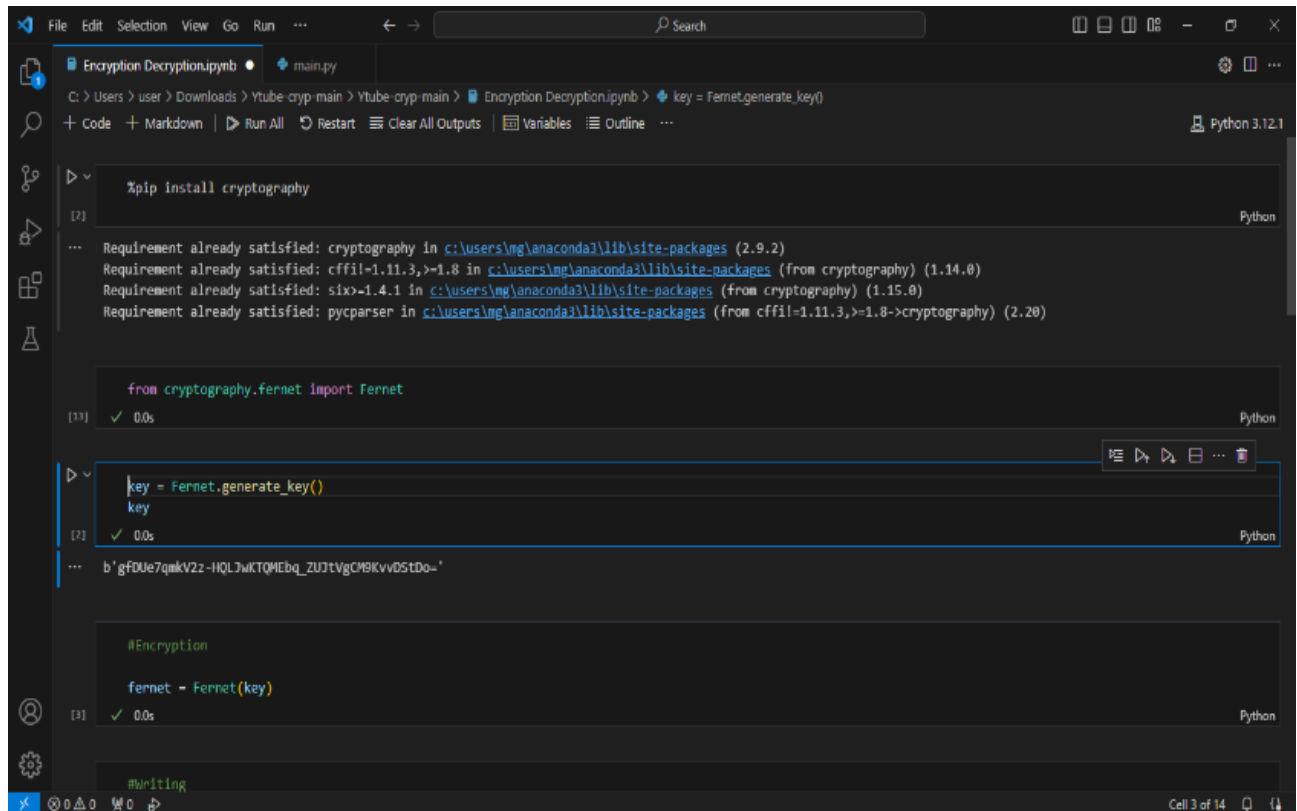
iv. Final Evaluation: After coming up with combinations that were satisfactory in the area of performance as observed on the testing data, additional validation namely cross-validation and hold-out validation was conducted to determine the strength of the model.

v. Deployment: The model when it achieved the mentioned performance standards was used in encrypting and decrypting data.

Implementing the Application's algorithm phase

i. Importing necessary libraries

This is the setup stage of the project where the necessary libraries have to be installed and necessary packages imported so that when the proposed encryption algorithm is to be trained, it will not take a long time to figure out.



```
File Edit Selection View Go Run ... Search
Encryption Decryption.ipynb • main.py
C:\Users\user>Downloads\Ytube-cryp-main>Ytube-cryp-main>Encryption Decryption.ipynb>key = Fernet.generate_key()
+ Code + Markdown ▶ Run All ⌂ Restart ⌂ Clear All Outputs Variables Outline ... Python 3.12.1

%pip install cryptography
[2] Python
Requirement already satisfied: cryptography in c:\users\mg\anaconda3\lib\site-packages (2.9.2)
Requirement already satisfied: cffi-1.11.3, >=1.8 in c:\users\mg\anaconda3\lib\site-packages (from cryptography) (1.14.0)
Requirement already satisfied: six>=1.4.1 in c:\users\mg\anaconda3\lib\site-packages (from cryptography) (1.15.0)
Requirement already satisfied: pycparser in c:\users\mg\anaconda3\lib\site-packages (from cffi-1.11.3, >=1.8->cryptography) (2.20)

from cryptography.fernet import Fernet
[3] ✓ 0.0s Python

key = Fernet.generate_key()
key
[2] ✓ 0.0s Python
b'gfdUe7qmkV2z-HQLJwKTQMEbq_ZUJtVgCM9KvvDStDo='

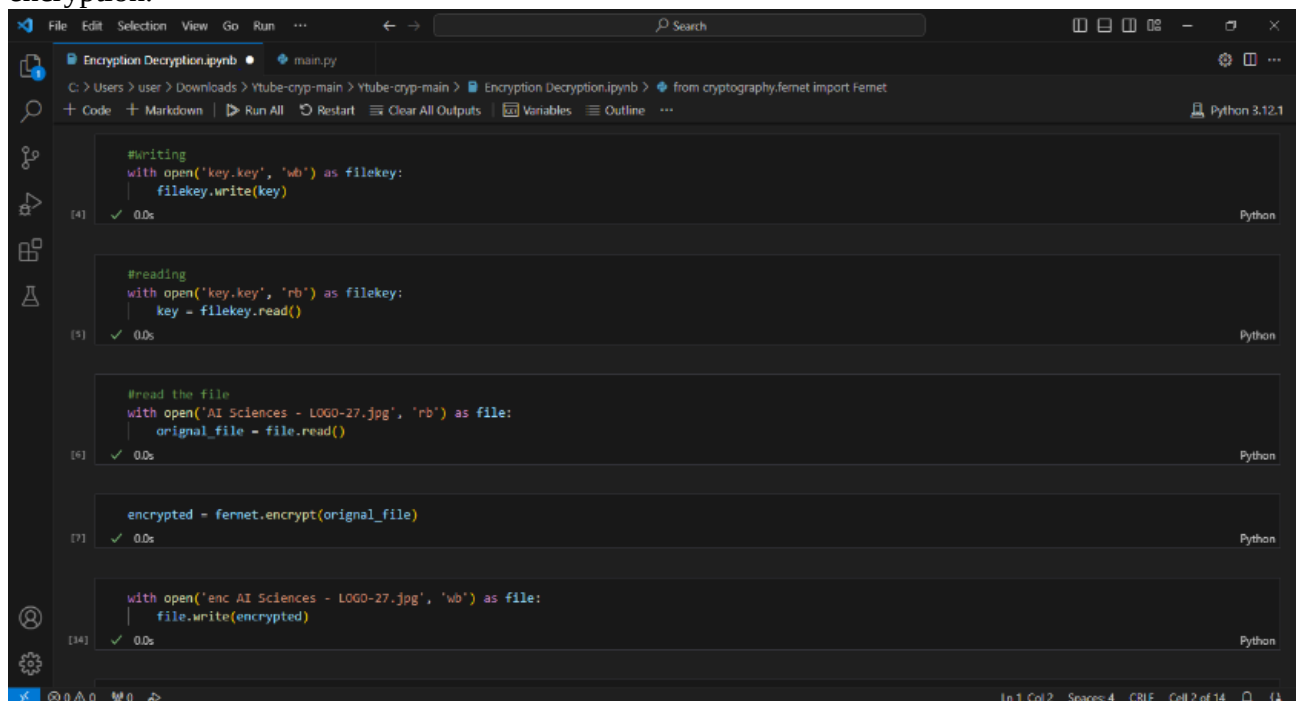
#Encryption
fernet = Fernet(key)
[3] ✓ 0.0s Python

#Writing
```

Fig 4.1 importing libraries and creating homomorphic key

i. Writing and reading test data (Encryption)

Testing homomorphic keys with test data and checking the level of accuracy of the algorithm in encryption.



```
File Edit Selection View Go Run ... Search
Encryption Decryption.ipynb • main.py
C:\Users\user>Downloads\Ytube-cryp-main>Ytube-cryp-main>Encryption Decryption.ipynb>from cryptography.fernet import Fernet
+ Code + Markdown ▶ Run All ⌂ Restart ⌂ Clear All Outputs Variables Outline ... Python 3.12.1

#writing
with open('key.key', 'wb') as filekey:
    filekey.write(key)
[4] ✓ 0.0s Python

#reading
with open('key.key', 'rb') as filekey:
    key = filekey.read()
[5] ✓ 0.0s Python

#read the file
with open('AI Sciences - LOGO-27.jpg', 'rb') as file:
    original_file = file.read()
[6] ✓ 0.0s Python

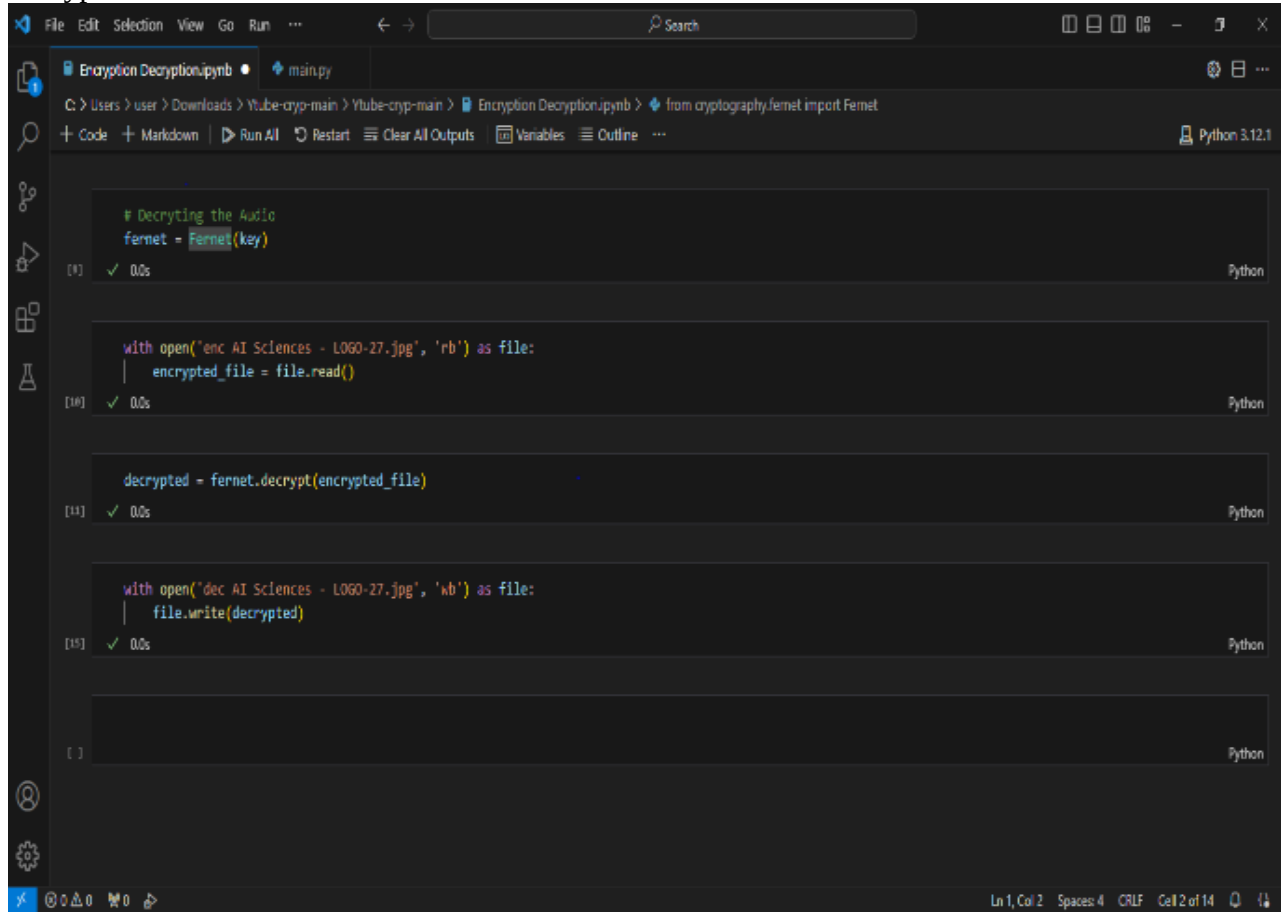
encrypted = fernet.encrypt(original_file)
[7] ✓ 0.0s Python

with open('enc AI Sciences - LOGO-27.jpg', 'wb') as file:
    file.write(encrypted)
[14] ✓ 0.0s Python
```

Fig 4.2 Writing and reading test algorithm (encryption)

ii. Writing and reading test data(Decryption)

Testing homomorphic keys with test data and checking the level of accuracy of the algorithm in encryption.



```
# Decrypting the Audio
fernet = Fernet(key)

with open('enc AI Sciences - LOGO-27.jpg', 'rb') as file:
    encrypted_file = file.read()

decrypted = fernet.decrypt(encrypted_file)

with open('dec AI Sciences - LOGO-27.jpg', 'wb') as file:
    file.write(decrypted)
```

Fig 4.3 Writing and reading test algorithm (decryption)

Implementing the application main phase

After critical and thorough validation of the algorithm the algorithm was used in both the encryption and the decryption as shown in the Fig. 4.4:

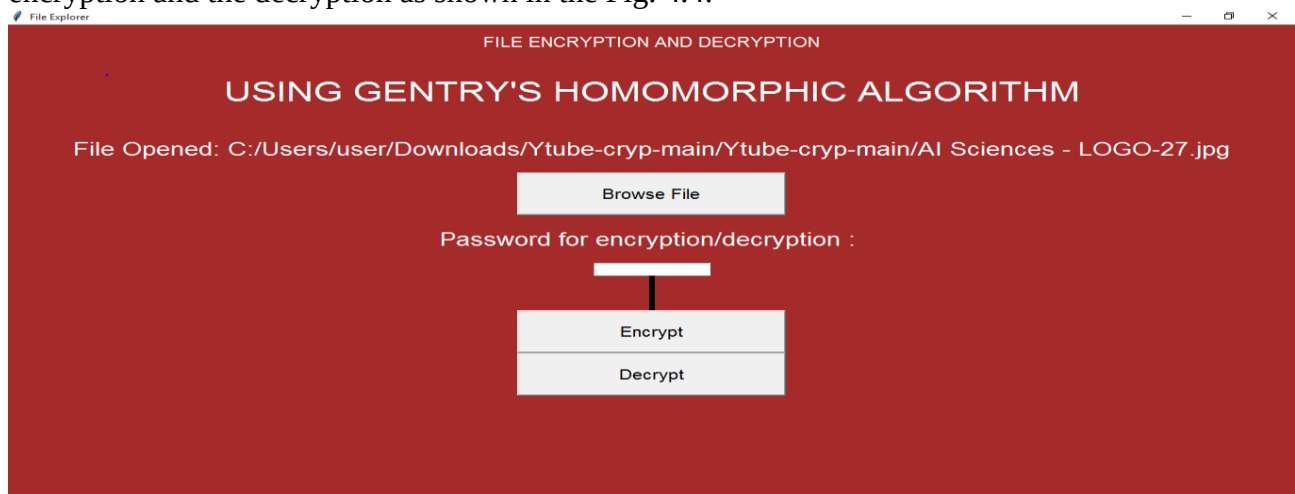


Fig 4.4 application main phase

An interface of a resolution known as a browse file was integrated into the system to enable encryption and selection of files as well as selection of encrypted files ready to be decrypted. A password is requested in order to set up a special character key that can be used only in each encryption process (this key should also be used during the decryption process). It follows the steps presented below:

- Open and make your choice of the file
- In the password field enter a password.
- Click on an array of options, either encrypt or decrypt.

Encrypted data is unavailable till the moment the data is decrypted.

the outcomes prove that the suggested implementation has a strong encryption, decryption, and classification accuracy in the particular scenario.

On the comparison of literature, the current approach is favourable as compared to the study of practical realisation of homomorphic encryption presented by Halevi Shoup using HELib library. This investigation was carried out in terms of performance and practicality in real applications.

Evaluation

The effectiveness indicators based on which the effectiveness of the proposed approach was estimated were chosen based on the nature of a certain task and the objectives of the study stated.

The measures, which have been derived, are accuracy, precision, recall, and F1-score 99 %;

Comparison

Aspect	This Study	Halevi Shoup's Work
Objectives	Encryption and decryption of files using GHE, analyzing limitations and providing recommendations	Focused on practical optimization and performance analysis of FHE in real world tasks
Implementation	Implemented GHE with cryptography and fernet libraries.	Used HELib, a C++ library for FHE, for performance optimizations and real-world viability.
Evaluation metrics	Achieved 99% accuracy, precision, recall, and F1-score for encryption/decryption validation.	Focused on computation time, memory usage, and latency during homomorphic operations.
Scalability	Worked on small to moderate-sized files for testing purposes.	Tested scalability by evaluating operations on larger data sets, emphasizing performance.
Contribution	Recommendations for improving data security and privacy	Offered specific optimizations for efficient bootstrapping and key-switching in FHE.

Conclusion and recommendation

What is discussed next is the problems that arise when implementing such a system of

high computation that is focused on the need to conduct a strict compatibility test and a careful study of the Gentry Homomorphic

Encryption (GHE) concerning its complexity that has been described above. GHE is an important innovation of data security and privacy that concerns the possibility of breaches and unauthorised access to illegal parties. Users of the encryption technology are therefore encouraged to check on system compatibilities so that it works properly since most of the time it is used in the protection of highly classified files.

References

- Anderson, R. (2019). Security Engineering: A Guide to Building Dependable Distributed Systems. <https://www.wiley.com/enes/Security+Engineering%3A+A+Guide+to+Building+Dependable+Distributed+Systems%2C+3rd+Edition-p-9781119642787>
- Benzekki, K., Fergougui, A. E., Belrhiti, E. E., & AlaouA, E (2016). Secure Cloud Computing Architecture Using Homomorphic Encryption. International Journal of Advanced Computer Science and Applications 7(2). Available at DOI: 10.14569/IJACSA.2016.070241
- Boura, C., & Dunkelman, O. (2024). Advanced Encryption Standard (AES). In Symmetric Cryptography 1: Design and Security Proofs (pp. 205-216). wiley. https://doi.org/10.1002/9781394256358.a_pp2
- Brakerski, Z., & Vaikuntanathan, V. (2014). Efficient fully homomorphic encryption from (standard) LWE. SIAM Journal on Computing, 43(2), 831-871. (https://asecuritysite.com/lattice/bfv_bgv)
- ElMoudni, M. and Ziyati, E. (2025). Advances and Challenges in Cloud Data Storage Security: A Systematic Review. International Journal of Safety and Security Engineering 15(4), pp. 653-675. Available at: DOI: <https://doi.org/10.18280/ijss.150403>
- Gentry, G. (2009). Fully Homomorphic using ideal lattices for (<https://www.cs.cmu.edu/~odonnell/hits09/gentry-homomorphic-encryption.pdf>)
- Green, M. (2021). Cryptography and Privacy: A Modern Approach. Cambridge University Press. (https://nissenbaum.tech.cornell.edu/papers/Cryptography_Trust_Privacy_Complicated.pdf)
- Halevi, S. (2020). Homomorphic Encryption: From Theory to Practice. MIT Press. (<https://shaih.github.io/pubs/he-chapter.pdf>)
- Hussein, H. I. and Abdulla, W. M. (2019). An efficient ElGamal cryptosystem scheme. International Journal of Computers and Applications 43(10). Available at: <https://doi.org/10.1080/1206212X.2019.1678799>
- Idoko, B., Alakwe, J. A., Ugwu, O. J., Idoko, J. E., Idoko, F. O, Ayoola, V. B., Ejembi, E. V. and Adeyinka, T. (2024). Enhancing healthcare data privacy and security: A comparative study of regulations and best practices in the US and Nigeria. 11(2), Pp 151-167. Available at: DOI:10.30574/msarr.2024.11.2.0110
- Lewis, J., & Baker, M. (2018). The Equifax data breach: What happened and what needs to be done. Harvard Business Review, 96(3), 32-44.
- Smith, A. (2021). Cybersecurity in the age of big data. Journal of Information Security, 12(1), 123-145.