REPUBLIC OF TURKEY

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Information Technology

PRIVACY PRESERVING HOMOMORPHIC ENCRYPTION IN CLOUD STORAGE

Zahraa Sabbar Lafta LAFTA

Master's Thesis

Supervisor

Asst. Prof. Dr. Muhammad ILYAS

Istanbul, 2022

PRIVACY PRESERVING HOMOMORPHIC ENCRYPTION IN CLOUD STORAGE

Zahraa Sabbar Lafta LAFTA

Information Technology

Master's Thesis

ALTINBAŞ UNIVERSITY

2022

The thesis titled PRIVACY PRESERVING HOMOMORPHIC ENCRYPTION IN CLOUD STORAGE prepared by ZAHRAA LAFTA and submitted on 09/08/2022 has been **accepted unanimously** for the degree of Master of Science in Information Technology.

_____

Asst. Prof. Dr. Muhammad ILYAS
Supervisor

Thesis Defense Committee Members:

Asst. Prof. Dr. Muhammad ILYAS	Faculty of Engineering and Architecture, Altinbas University	_____
Asst. Prof. Dr. Hasan ABDULKADER	Faculty of Engineering and Architecture, Altinbas University	_____
Asst. Prof. Dr. Ali HAMITOGLU	Faculty of Engineering and Natural Sciences, Istinye University	_____

I hereby declare that this thesis meets all format and submission requirements of a Master's thesis.

Submission date of the thesis to Institute of Graduate Studies: ___/___/___

I hereby declare that all information/data presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Zahraa Sabbar Lafta LAFTA

Signature

DEDICATION

I would like to thank Allah Almighty for my success in my studies. I would like to thank very much to my supervisor, Assistant Professor Dr. Muhammad Ilyas for his assistance throughout the research period and for answering many questions and he was the best help for me throughout the study period. Without his time and help, this research would not have been completed and I am very grateful to him.

My dear parents, I know very well that you loved science and study, I am sure that you are now happy with my studies. Thank you for all you have done for me so that I can reach this level of study today.

My husband, I would like to offer you all my thanks, love and appreciation. Thank you for your support, standing by my side, and your great help for me.

My daughters (Sarah, Dimah, Hanna) are the most precious thing that I have in this world. I love you so much. I dedicate this to my family and everyone who stood beside me.

ABSTRACT

PRIVACY PRESERVING HOMOMORPHIC ENCRYPTION IN CLOUD STORAGE

Lafta, Zahraa Sabbar Lafta

M.Sc. Information Technology, Altınbas, University,

Supervisor: Asst. Prof. Dr. Muhammad ILYAS

Date: 08/2022

Pages: 72

The notions of cloud computing and homomorphic encryption were separated by two distinct authorities. In light of earlier research, we sought to give a more comprehensive illustration of the link between the two notions. Our study was conducted with the expectation that other researchers in this field may find it useful. As established, issues over data privacy and security provide a substantial barrier to the spread of cloud computing. The user must encrypt their data before uploading it to the cloud so that their information remains private. If he needs to manipulate the encrypted data, he must first download it to his computer, decrypt it, do the required operations, encrypt it again, and then upload it to the cloud storage. In such a scenario, the client would have no option except to pass over his key to the cloud service provider in order for it to do calculations on his behalf, which is not a safe solution. Homomorphic encryption is a topic that has great promise in the field of cryptography. The use of HE schemes, which allow computations to be performed on encrypted data without exchanging the secret key, has the potential to significantly increase the security of cloud computing. This project aims to determine if it is feasible to maintain symmetric encryption in cloud storage. The purpose of this research is to explain the Cloud's data security flaws and how encryption may be used to mitigate them. Also reviewed were Homomorphic Encryption (HE) and its various implementations. To accomplish this, the following goals will be discussed: Provide a summary of the cloud computing security problems and a definition of cloud computing. Describe the efficacy of classical encryption as a threat

deterrent and the difficulties inherent in its implementation. Demonstrate the principles of Homomorphic Encryption (HE), and then describe some practical uses of the technique.

Keywords: Cloud Computing, Fully Homomorphic Encryption, AES, IT, HE, Date Security**.**

TABLE OF CONTENTS

Pages

ABSTRACT ..... **vi**

TABLE OF CONTENTS ..... **viii**

LIST OF TABLES ..... **x**

LIST OF FIGURES ..... **xi**

ABBREVIATIONS ..... **xii**

LIST OF SYMBOLS ..... **xiv**

1. INTRODUCTION **1**

1.1 BACKGROUND 1

1.1.1 HE APPLIED TO CLOUD SECURITY 4

1.2 PROBLEM STATEMENT 6

1.2.1 Cryptography Drawbacks and Benefits 7

1.3 CONTRIBUTIONS AND OBJECTIVES 8

1.4 SIGNIFICANCE OF THE STUDY 9

1.5 RESEARCH QUESTIONS 9

1.6 THESIS ORGANIZATION 9

2. LITRITAURE REVIEW **11**

2.1 INTRODUCTION 11

2.2 PREVIOUS WORKS 11

3. MATERIALS AND METHODS **19**

3.1 CLOUD COMPUTING 19
3.1.1 Key Features of Cloud Computing 20
3.1.2 Cloud Computing Architecture 21
3.1.3 Service Delivery Models 25
3.2 CLOUD NETWORK SECURITY 26
3.2.1 Service Providers 28
3.3 SECURITY THREATS AND COUNTER MEASURES 31
3.3.1 Homomorphic Encryption and DNS Hijacking 31
3.3.2 Threat Path Processing 35
3.3.3 Virtual Machines VM 36
3.3.4 Consumer Platform 38
3.3.5 Storage Space 40
4. SIMULATION AND RESULTS 43
4.1 SYSTEM SETUP 43
4.2 IMPLEMENTATION OF PAILLIER ALGORITHM 47
4.3 TEST RESULTS 50
5. CONCLUSIONs AND FUTURE WORK 54
5.1 CONCLUSIONS 54
5.2 FUTURE WORK 55
REFRENCES 57

LIST OF TABLES

Pages

Table 1.1: Existing Homomorphic Encryption Cryptosystems' Characteristics [4] 5

Table 2.1: Summary of the literature review 18

Table 3.2: DNS, NTP and DHCP purposes in clouds [18] 34

Table 4.1: Paillier encryption and some variants that will later be used for comparison 49

Table 4.2 Evaluation of different dataset for the same encryption scheme 53

LIST OF FIGURES

Pages

Figure 1.1: Basic architecture of cloud computing [69] 1

Figure 2.1: Cryptography in clouds [3]..... 3

Table 3.1: Characteristics of cloud computing [2]..... 21

Figure 3.1: Cloud Computer architecture [16]..... 22

Figure 3.2: Elements of Cloud computing structure [12] 23

Figure 3.3: Enterprise Content Management (ECM) in cloud computing (CC) [15]..... 25

Figure 3.4: Risk factors in CC according to [12]..... 27

Figure 3.5: Combined ISP in CC [15] 29

Figure 3.6: DNS hijacking through HSTS traffic in CC [33] 32

Figure 3.7: Role of VM ware in clouds [12]..... 37

Figure 3.8: General view of the consumer and provider interaction platform in clouds [7] 39

Figure 3.9: Storage mechanism in CC [17] 41

Figure 4.1: General view of the proposed system [9]..... 44

Figure 4.2: Python 3 libraries needed for the execution 48

Figure 4.3: Encryption time comparison of the proposed V1 algorithm and other variations 51

Figure 4.4: Performance of the proposed method after parallelization 52

ABBREVIATIONS

IT : Information Technology

CC : Cloud Computing

AES : Advanced Encryption Standard

IaaS : Infrastructure as a Service

PaaS : Platform as a Service

SaaS : Software as a Service

HE : Homomorphic Encryption

EDM : Electronic Document Management

IVS : Integrity Verification Service

FHE : Fully Homomorphic Encryption

DES : Data Encryption Standard

VCM : Virtualized Cryptography Machine

WSNs : Wireless Sensor Networks

SHE : Secure Hardware Extension

FDMK: Fully Dynamic Multi-Key

NC : Network Computing

CRM : Consumer Relationship Management

ECM : E-commerce, enterprise Content Management

ERP : Enterprise Resource Planning

ERM : Enterprise platforms. Risk Management

APIs : Application Programming Interfaces

SRP : Secure Remote Password

HSTS : Strict-Transport-Security

CSP : Content-Security-Policy

SRI : Sub resource Integrity

HTTPS : Hypertext Transfer Protocol Secure

CDN : Content Delivery Network

OLAP : Online Analytical Processing

DBMS : Database Management System

LIST OF SYMBOLS

γ : gamma function

$\wp$: Particularly functions

Λ : lowercase

Z* : integers number

1. INTRODUCTION

1.1 BACKGROUND

A significant shift has occurred in people's day-to-day lives as well as in almost all spheres of human knowledge as a direct result of the lightning-fast pace at which information technology (IT) has developed in the last few decades. The proliferation of different access methods, such as mobile devices, the development of various technologies, such as virtualization, and the growing demand from users for new systems and services that are adapted to the new trends in the market were the fuel that led to the emergence of a new paradigm known as cloud computing (CC). [69]

Cloud computing is the inevitable progression of conventional datacenters, which differentiate themselves from one another by providing computational resources such as storage, processing, and managed applications by means of standardized web services [69]. Cloud service providers provide their clients access to computer infrastructure, software platforms, and applications in the vast majority of situations rapidly, without the involvement of any bureaucracy, and with the mere signature of a service contract.

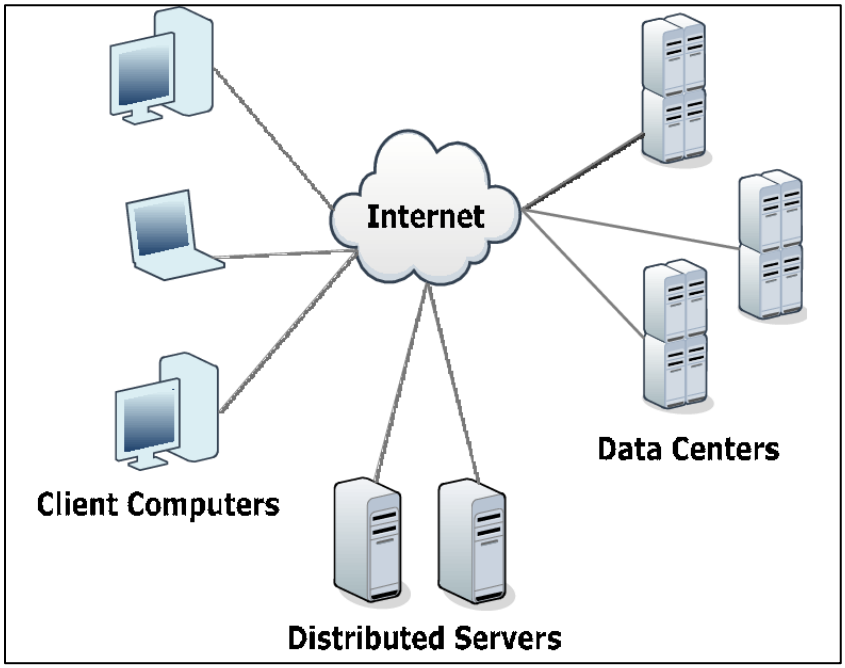

Figure 1.1: Basic architecture of cloud computing. [69]

Is an example of a storage service that is included in the portfolio of services marketed by CC platforms from large international corporations? Are examples of storage services that belong to the portfolio of services provided by cloud providers the data storage service is one of the most common services provided by cloud providers. Cloud data storage services are lightning-fast, remarkably affordable, and remarkably scalable. On the other hand, there are concerns over the dependability of the data, since even the greatest service providers might experience downtime. For many businesses, the primary barriers to using cloud services are concerns around security and privacy.

As a result, it is essential that CLOUDs be able to offer ways to ensure the integrity of data that has been saved while also guaranteeing the data's confidentiality. Keeping data in outsourced environments is not always reliable because it makes users dependent on the availability and integrity provided by the service providers. This is true despite the fact that CC provides users with flexibility and relieves them of the burden of worrying about the complexity of managing the storage infrastructure. However, despite these benefits, keeping data in outsourced environments is not always reliable. In this scenario, structural or human failures that could corrupt or allow the leakage of stored data, as well as, in the worst-case scenario, the deliberate deletion of this data, can be omitted from their owners for extended periods of time. This is especially true in circumstances in which the data in question are rarely accessed. In light of the information presented above, it is essential that precautions be taken to ensure, from the perspective of the customer, that the integrity and confidentiality of data that are kept in the cloud be maintained [2].

Applying cryptographic solutions, such as the Advanced Encryption Standard (AES) cryptographic algorithm, is one method that may be used to attain confidentiality. However, there is still difficulty involved in assuring honesty. There are a number of research papers that focus on verifying the integrity of files. As a consequence, these researchers propose a variety of methods to enable clients to verify the integrity of their own files. Some of these methods include the utilization of homomorphic asymmetric cryptography, the application of cryptographic hash algorithms, the utilization of challenges, and third-party auditing. The examined ideas, although highlighting a variety of methods for validating the integrity of files, have features that make it challenging to use them in contexts other than the one for which they were designed.

This is the case even though the mechanisms have been identified. In consideration of the applications of the suggested solutions in the field of asymmetric cryptography as well as the high consumption of network bandwidth caused by the need to download the complete file's content prior to carrying out the verification process In addition, none of the offered solutions include a system that seeks to proactively react to faults that have been found in a particular service provider in order to simplify the process of verifying the remainder of the files that are kept inside. Another issue that was not discussed in the articles that were reviewed was the waste of computing resources that occurred when the researchers checked files that were kept with providers that had an outstanding track record, particularly in terms of integrity and availability. In this context, the architecture that was proposed in this work also uses trust concepts to perform load balancing of integrity checks.

This prioritizes the verification of files stored in clouds that have already failed and saves computational resources for clouds that have historically provided a quality storage service. In other words, the architecture that was proposed in this work uses trust concepts to perform load balancing of integrity checks.

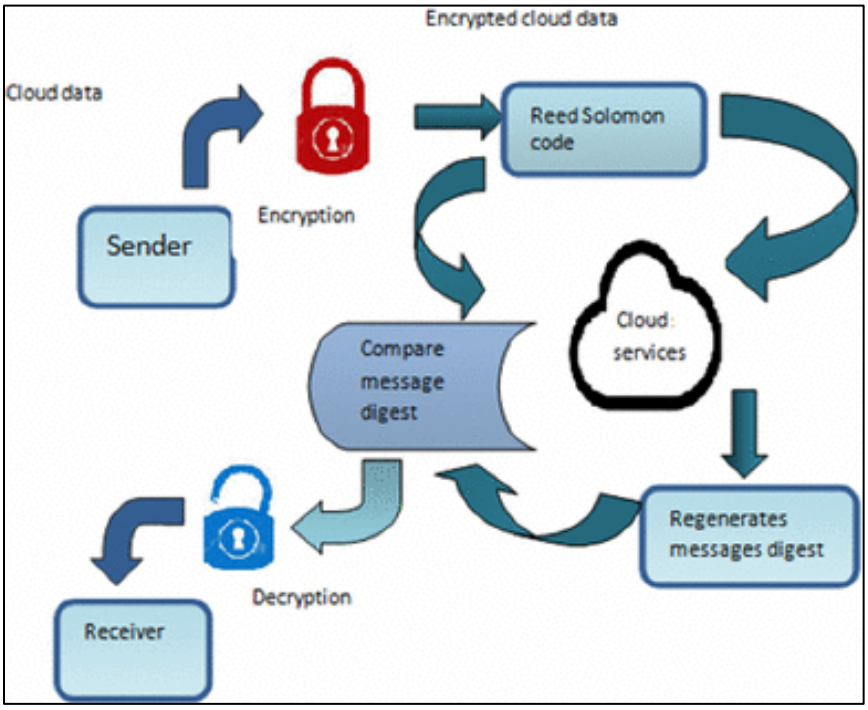

Figure 2.1: Cryptography in clouds [3]

1.1.1 HE APPLIED TO CLOUD SECURITY

Even though the above definition of cloud computing is meant to be a modern explanation of cloud computing, the definition makes no mention of the security of cloud-stored data. In light of this, it is evident that cloud computing lacks adequate security, privacy, and transparency. It is not sufficient for a cloud provider to provide Infrastructure as a Service (IaaS), Platform as a Service (PaaS), or Software as a Service (SaaS) if they cannot also guarantee that the customer's data will be safeguarded against unwanted access and kept confidential. Protecting the Cloud entails safeguarding the processing (calculations) and storage of personal or business data that occurs outside of the organization's control (that is, outside of the organization). This expression is used in relation to cloud computing (databases hosted by the Cloud provider).

On the cloud platform supplied by cloud providers like as IBM, Google, and Amazon, many enterprises' virtualized storage and treatment areas may coexist on the same server. To guarantee the safety of the data flowing from each of the firms, security and confidentiality problems must be addressed. In order to ensure the security of data storage and processing while using a modern cryptographic technique, it is required to adhere to specific conditions. These include the time required to reply to client queries and the volume of encrypted data that will be kept on a cloud server. When you entrust a third party with the administration of your data, you abdicate some responsibility for maintaining its security and ensuring that it conforms to existing requirements.

It is understandable that security specialists would fear the potential consequences of a terrorist attack. Therefore, you must have entire faith in your cloud service provider. The usage of cloud computing reduces overhead expenses and simplifies access to accessible resources, which increases the profitability of a corporation. According to our method, we would encrypt data before moving it to the cloud provider, but we would need to decrypt it before making any modifications. In the past, it was impossible to encrypt data, rely on a third party to keep it secure, and do remote computations on it. Currently, all of these are feasible. Homomorphic Encryption cryptosystems must be used to enable the Cloud provider to perform actions on encrypted data without decrypting the data first [8]. When data is sent to a cloud service, the service's operations and data storage are encrypted using industry-standard methods. Encrypting the data before to transferring it to a cloud storage provider was the first and most critical step. In contrast, the last component must decrypt

the data before to every operation. Before doing the required computations, the Cloud provider will need the client's private key in order to decode the encrypted data.

This might compromise the security of the data and the user's right to privacy while it is stored on the cloud. Thanks to the approach presented in this article, we will be able to run operations on encrypted data without first decrypting such data. The outcomes will be the same as if we had dealt with the raw data directly. In a system that utilizes homomorphic encryption (i.e., one that does not need decryption), only the individual who has access to the secret key may conduct operations on encrypted data. When the result of any operation is decrypted, it is same as if the computation had been performed on the raw data. This is true regardless of the procedure undertaken. This concludes the ninth cycle of defining [9]. A homomorphic encryption is one that can be determined from Enc(a) and Enc(b), where f (a, b) may be: +, without the need for the private key. This encryption method is also known as reversible encryption. According to Paillier [10] and Goldwasser-Micalli [11], one of the Homomorphic encryptions that may be assessed on raw data is additive homomorphic encryption. This kind of homomorphic encryption just inserts the unencrypted data.

Table 1.1: Existing Homomorphic Encryption Cryptosystems' Characteristics [4]

	Homomorphic Encryption Cryptosystems					
Characteristics	**RSA**	**Paillier**	**El Gamal**	**Goldwasser-Micali**	**Boneh-Goh-Nissim**	**Gentry**
Platform	Cloud Computing	Cloud Computing	Cloud Computing	Cloud Computing	Cloud Computing	Cloud Computing
Homomorphic Encryption type	Multiplicative	Additive	Multiplicative	Additive, but it can encrypt only a single bit	Unlimited number of additions but only one multiplication	Fully
Privacy of data	Is ensured in communication and storage processes	Is ensured in communication and storage processes	Is ensured in communication and storage processes	Is ensured in communication and storage processes	Is ensured in communication and storage processes	Is ensured in communication and storage processes
Security applied to	Cloud Provider Server	Cloud Provider Server	Cloud Provider Server	Cloud Provider Server	Cloud Provider Server	Cloud Provider Server
Keys Used by	The client (Different keys are used for encryption and decryption)	The client (Different keys are used for encryption and decryption)	The client (Different keys are used for encryption and decryption)	The client (Different keys are used for encryption and decryption)	The client (Different keys are used for encryption and decryption)	The client (Different keys are used for encryption and decryption)

1.2 PROBLEM STATEMENT

The need of preserving the truthfulness of one's information has grown even more pressing in recent years as the use of electronic document management (EDM) has become more widespread not only in public administration organizations but also in the commercial sector. With the advent of the usage of the documents in physical medium (paper), they were replaced by solely electronic documents, the validity, temporality, and categorization of which in regard to confidentiality are backed by the law that is in force today [4].

The duration for the protection and conservation of electronic documents is called their temporality. This term, which is set in the law, may range anywhere from five to thirty years in length, depending on the nature of the document and the information included within it. The need to access a document after it has been processed and filed, despite the fact that law mandates that its custody be maintained, is very uncommon. This is particularly true if more than a year has passed after the document was first filed.

The amount of backup copies may easily approach 10 gigabytes in size depending on the number of documents that are processed in the organization as well as the substance of those documents. Therefore, the use of clouds for the purpose of their storage may be a workable option, taking into consideration the potential of minimizing the need for the business to make investments in its information technology infrastructure. However, in order to make use of clouds, it is required to guarantee that the integrity of the data that are stored can be continuously checked. This is primarily necessary to satisfy legal requirements. Current works on verifying the integrity of cloud-stored files offer interesting solutions for the domains for which they were made, but they fall short in some ways, such as high computational cost, use of homomorphic asymmetric cryptography, or need to retrieve the file as a whole to perform the verification. Due to the shared integrity verification key, you must trust IVS; and need to retrieve the file in its entirety to perform the verification In light of the issues that are still present and those that have been brought to light, it is imperative that research be conducted on the technologies that are currently available, and that a proposal be developed for a protocol that would enable users of files to permanently monitor the integrity of their files through the use of a service that was specialized in integrity checks and was provided by third parties, and that would have a low computational cost and be capable of redundancy [5].

1.2.1 Cryptography Drawbacks and Benefits

Currently, data only exists as bits and bytes owing to the globalization of network infrastructure. Computer systems and open communication channels of all types are utilized to store, process, and send digital information.

As a result of the high value of information, attackers are focusing their efforts on computer systems and open communication channels in an attempt to either steal information or disrupt important information systems. Using modern encryption, it is now possible to prevent unwanted users from accessing sensitive data while still allowing authorized users to see the data. In this chapter, we will analyze the use of cryptography, its limitations, and its potential applications. Numerous advantages of utilizing cryptography in today's digital world, the necessity of using encryption cannot be overstated.

It offers the four services listed below, which are the most basic in information security: It is feasible to prevent unauthorized exposure or access to one's information and communications by using different kinds of encryption. Using cryptographic methods such as MAC and digital signatures, it is possible to safeguard information against forgery and spoofing. This is possible because to the security offered by these approaches. Cryptographic hash functions play a crucial role in ensuring that users can trust the validity of the data they save. The non-repudiation service provided by the digital signature protects against the possibility of a dispute arising from the sender disputing that they sent the communication. All of these key services have been provided by cryptography, which has made it possible to conduct business across computer networks in a very effective and efficient way.

A study of the disadvantages of utilizing cryptography even while the four most important aspects of information security are crucial, there are a vast number of other concerns that might reduce the use of information. Even an authorized user may have difficulty acquiring access to highly encrypted, genuine, and digitally signed material at a crucial moment in the decision-making process. A malicious user may disable a computer system or network, preventing it from running correctly. Cryptography cannot ensure the high availability that is regarded as one of the most crucial components of information security. Additional measures must be taken to protect against threats such as denial of service and total system failure. Selective access control is one of the most

essential needs for information security that cryptography cannot provide. For the same reason, administrative controls and processes must be implemented. It is challenging for cryptography to guard against the risks and vulnerabilities created by poorly constructed systems, protocols, and processes. To tackle these issues, it is necessary to develop and deploy a defensive architecture in accordance with best practices. There is a cost connected with cryptography use.

This is a difficult job that will need a substantial commitment of time and money. The processing time of information rises when cryptographic methods are used. Establishing and maintaining a public key infrastructure needs substantial financial resources, which must be committed in order to use public key cryptography. The computational complexity of mathematical problems is the foundation for the security that may be provided by cryptographic methods. It is feasible for a cryptographic system to become susceptible to attack if processing power is enhanced or if a solution to certain mathematical problems is discovered.

1.3 CONTRIBUTIONS AND OBJECTIVES

Users of Cloud Data Storage Services will be able to continually check the integrity of their data thanks to this work's overarching goal, which is to make it possible for third parties to provide a privacy preservation system to customers of Cloud Data Storage Services. The following are some of the specific goals that this work aims to achieve:

a. To develop a mechanism that, while processing data, maintains the data's confidentiality while preventing the user from having to access the entire file's contents at each stage of processing; this is one of the specific goals of this work.
b. prevent cloud administrators from gaining access to the contents of stored files through the use of cryptographic algorithms and passwords strong enough to resist brute force attacks during the anticipated period of storage;
c. guarantee the recoverability of a file that is stored in the cloud according to the criticality of the file, allowing the client to not have to keep a copy of the original file;
d. allow cloud storage providers to do a security check on files without access to any of the files themselves.

1.4 SIGNIFICANCE OF THE STUDY

The Security of Homomorphic Encryption in Cloud Computing is a new method for safeguarding data that may transmit the results of computations conducted on encrypted data without revealing the raw data. This safeguards the data's secrecy. Using a Cloud Computing platform, this dissertation investigates a variety of various Homomorphic Encryption cryptosystems, including RSA, Paillier, and El Gamal. Taking into account the following criteria: Homomorphic Encryption is a kind of encryption that protects the privacy of data. Security precautions made with reference to the keys used.

1.5 RESEARCH QUESTIONS

a. How will cloud-stored text documents be homomorphically encrypted for searches that do not need the documents or the query to be decrypted on any device other than the one doing the search?
b. Can you describe the data type of the encryption key?
c. How many iterations are necessary to do homomorphic encryption and a search for phrases inside a text document?
d. The index's contents and structure will decide whether it is kept in plaintext or ciphertext. At which location will the textual records be encrypted and decrypted?

1.6 THESIS ORGANIZATION

For a better understanding of this work, the thesis organization is described below:

In the second chapter, we will look at some of the previous research that has been done on this topic.

Cloud computing, trust, security, and homomorphic encryption are the primary topics that are reviewed in Chapter 3, which gives an overview of the most important ideas discussed. In addition, works that are linked to this topic are discussed, as are some of the outstanding issues.

The idea for a system that permits the execution of the proposed architecture as well as the protection of individual privacy is presented in Chapter 4.

The simulations that were run and the findings that were obtained are presented in Chapter 5.

This work comes to a close with the discussion of Chapter 6, which summarizes the findings, draws conclusions about the findings, and outlines potential future directions that might be pursued as a direct consequence of this study?

2. LITRITAURE REVIEW

2.1 INTRODUCTION

The classic technique of encryption necessitates the exchange of a key, which may be public or private, between the parties participating in a secure transmission. On the other hand, this method may lead to privacy breaches. Users or service providers who possess the key are the only individuals who have ownership of the data. When employing popular cloud services, customers often surrender control over the amount of confidentiality offered to sensitive data. Even if the encryption keys are kept confidential, it is common practice to pass encrypted data to a third party that does not need access to the content. Users may be susceptible to monitoring even after they have ceased using the services of untrustworthy servers, providers, or cloud operators. Homomorphic encryption, often known as HE, is a kind of encryption that does not need the data to be decrypted beforehand.

This may make it feasible to avoid the aforementioned issues. Craig Gentry is credited with presenting the first viable and workable fully homomorphic encryption (FHE) solution in 2009. This component of the HE approaches has been understood for over three decades. Significant progress has been made in this regard, but there is still a considerable way to go before FHE can be reliably implemented on any platform. As a direct result, the HE and FHE models are the major focus of this investigation. In the beginning, we will explore the concepts of higher education as well as the specifics of the well-known PHE and SWHE, both of which are crucial building blocks for finishing FHE effectively.

In the next phase, the essential FHE families that have served as the foundation for subsequent FHE schemes are detailed. In addition, we examine recent advancements in the use and enhancement of FHE systems based on the Gentry model. In this section, we shall ultimately explore the future possibilities of research. This chapter gives a summary of the findings and conclusions of each study, as well as a review of the previous research conducted on this issue.

2.2 PREVIOUS WORKS

2018, Joseph Selvanayagam and his colleagues published research with the title Secure File Storage on Cloud Using Cryptography. This was a study that was authored by Selvanayagam. [1]

Using a wide range of cryptographic strategies, the goal of this effort is to conduct research on the possible security issues that are linked with data that are kept in the cloud. In this paper, both asymmetric and symmetric techniques, both of which are well-known for their use in encryption and decryption, are reviewed. In this work, extensive information on AES and DES, including coverage of each step of the process, has been provided. This article goes through a few different encryption techniques, one of which is the RC-2 Encryption Algorithm.

In 2018 Bin-hwaang Lee [2] conducted research on the problem of ensuring the safety of data stored in cloud computing environments by using AES. In this research, concerns about cloud computing security are investigated, and the appropriate security measures that may be taken to address such concerns are uncovered. In this post, the authors established a website for data security and employed AES as a data security algorithm. This was a follow-up to their previous discussion, which was centered on the HEROKU Cloud and its usage of AES for cloud data storage.

In 2019 S. Lei His paper, titled "Planning and Studies of Cryptography Cloud framework,"[3] discussed many cryptographic frameworks applicable to cloud computing and was named after the paper's title. In addition to that, they have gone into great detail discussing the several techniques of protecting cloud computing by using both public and private keys, as well as a virtualized cryptography machine (VCM) and how it operates. It is one of the research papers on cloud cryptography that has the most information on virtual cryptography machines, and it is one of the most thorough articles ever written on the topic (VCM). Which of the several companies that offer cryptography services are you referring to? In order to demonstrate that customers will be able to access cryptographic services via a cloud computing architecture, they presented the CC framework as a component of this.

In 2021 Ahmad.S.A. The paper that he wrote on "Hybrid Cryptography Automated system in Cloud Computing"[4] discussed the hybrid approach, which combines two separate encryption methods to provide more reliability to data compared to using just one encryption method. This is because one encryption scheme is relatively easy to crack, whereas two encryption algorithms are more difficult for third parties to decrypt. He wrote the paper to discuss the hybrid approach, which combines two separate encryption methods to provide more reliability to data compared to using just one encryption method. In light of the fact that there have been more instances of data breaches

recently, this is an innovative approach to ensuring the safety of our information. Due to the fact that he covered a wide range of approaches used by a number of researchers, his review study is a good resource for expanding our knowledge of cryptography algorithms. The comparison in this research illustrates the numerous hybrid approaches that are available.

The research paper authored by Pandey and titled "Data Security in Cloud-Based Applications" was handed in in the year 2019. [5] He highlighted the security problems that we face now in this piece, and he did it by focusing on the word "today." Because of this, he recommended making use of a method that is known as AES. The Advanced Encryption Standard (AES) is a block encryption method that uses a private key for further security. In this piece of work, he provided a comprehensive explanation of the AES method. In addition to that, he went through the three security patterns—namely, filtering, encryption, and authorization—that may be used to guarantee that one's data is kept safe. Filtering is the first security pattern, followed by encryption, and then authorization.

In the year 2020, the researchers Sarojini et al. proposed a technique that they named the Enhanced Fully Trusted Access Optimization Approach (EMTACA). [6] Establishing a trusting relationship between cloud services and cloud service providers is one way to circumvent any security risks associated with cloud computing. Through the use of the EMTACA algorithm, which is a component of this article's system, the experimentation described in this research was successful in maintaining the data's secrecy, integrity, and availability, which are the three most important components of data security.

Using methods such as watermarking and fingerprinting, it is possible to include extra information in digital data: With the assistance of the homomorphic property, a unique identifier is constructed for previously encrypted material. Typically, watermarks are required to safeguard the copyright of digital objects by identifying the owner or seller of each particular item. To guarantee that fingerprint information is not unlawfully disseminated via fingerprinting programs, it is vital to be able to identify the purchaser. This will prevent the abuse of the information. The concept of oblivious transmission is intriguing in the world of cryptography. In two-party 1-of-2 oblivious transfer protocols, the first party typically transmits a bit to the second party with a probability of 1.2 and subsequently informs the second party of the bit's reception. There are several basic cryptographic primitives known together as commitment schemes. A player makes a commitment

in a system that employs obligations. As a result, she is able to choose a value from a set and become so devoted to it that she can no longer alter her mind. Even if she decides to do so at this time, she is not compelled to do so. Using the homomorphic property, the functioning of some commitment systems may be made more efficient. It is customary for a cryptographic lottery to require each participant to choose a number from a preset range that will be used to decide the winning ticket.

The following is an example of a technique that may be used when employing a homomorphic encryption scheme to accomplish this objective. Each participant participates by selecting a random integer and encrypting it. Using the homomorphic characteristic, it will be feasible to effectively calculate the encryption of the sum of the random values. The needed functionality may be achieved by combining this approach with a threshold decryption algorithm. Mix-nets, which gather ciphertexts and then emit randomly the plaintexts that correspond to those ciphertexts, are an example that everyone may examine.

One technique for ensuring privacy inside this system is to hide the permutation that decides which outputs correlate to which inputs from all users other than the mix-net. Specifically, if an input/output pair must be identified, there should be no input/output pair that is more efficient than a random pair. In such mix-nets, it is important to be able to do encryption, and homomorphic encryption provides this capacity. The gathering of information using wireless sensor networks by leveraging in-network data aggregation, intermediary nodes in WSNs may aggregate incomplete findings. This may assist in reducing communication overhead and optimizing bandwidth utilization in wireless networks. If the sensor nodes are compelled to share the data they gather with the aggregator node, however, this strategy poses privacy and security concerns.

In extremely sensitive applications, such as healthcare and military surveillance, where the data acquired by the sensor is highly secret, aggregation that protects privacy is a critical component. [35] V.FHE Gentry's FHE approach employs ideal lattices in order to provide an asymmetric encryption solution. After the production of secret keys, a large number of public keys containing "noise" are generated. This guarantees that an attacker cannot determine the secret key based only on the public keys. The fault in the original approach caused the "noise" in the ciphertext to become more apparent as more computations were performed. Because the ciphertext has an overwhelming quantity of "noise," it can no longer be decoded to disclose the original message. Data encryption

is achieved via the use of technologies known as partially homomorphic encryption (SHE). This concept suggests that there is a limit to the amount of homomorphic processes an individual can do.

[5] In addition, Gentry describes the fundamental process for converting the SHE to the FHE. This is referred to as "bootstrapping." As the quantity of "noise" increases, it becomes more difficult to correctly decipher information. Using the bootstrapping approach, the issue may be handled by first conducting homomorphic decryption on the ciphertext, then executing a single calculation on it, and then re-encrypting it using a new public key. As a result, the bootstrapping procedure is exceedingly theoretical and, hence, inefficient. For bootstrapping, the nonstandard assumption of circular security is required in addition. It is believed that encrypting the secret key with its own public key offers the required degree of security [36, 5, 37]. In 2009, Gentry released its first FHE scheme, and in the years that followed, the business developed a substantial number more

2010 saw the first application of FHE, which was conducted by Smart and Vercauteren [28]. Since then, much effort has been invested in the establishment of more realistic policies. The FHE technique can only be used when there is just one person doing the calculations. Therefore, the input received from users must be encrypted with the same key. Imagine instead a situation in which users who have uploaded encrypted data to the cloud are trying to calculate a joint function utilizing their encrypted data while using distinct keys. In 2012, Lopéz-Alt et al. [44] developed a multi-key FHE approach based on the NTRU cryptosystem [45]. This strategy is used when many parties are engaged. Despite the fact that this scenario is more difficult than the case of a single user, considerable progress has been made in this field. (36-37-47-8) According to the author's best knowledge, there is no multi-key FHE implementation available in the single-user setting. The choice to adopt nonrival systems that have not yet been implemented has the unexpected effect of making difficulties in general unanticipated. To finish this project effectively, you will need to overcome a number of obstacles, including analyzing the plan and turning it into code, choosing parameters, constructing algorithms, selecting benchmarks, and conducting testing. In addition, Perlman and Brakerski argue that their system cannot be implemented in its present state due to a number of unorthodox assumptions.

As a direct result, putting it into effect will be an extraordinarily difficult undertaking. It is vital to remember that our proposal is not a viable solution to the issue on its own. Due to the manner in

which we use the bootstrapping equipment, the assessment procedure requires much more effort. Instead, we intend to illustrate that multi-key FHE has theoretical constraints and to open the door to further enhancements that will bring solutions closer to a reality that can be implemented. Paragraph 5.1 of the FHE Plans Let's take a closer look at the fundamentals of FHE in a single-key and multi-key environment, as well as the many ways it may be implemented. 100 percent homomorphic single-key encryption in both directions: Even while the construction of homomorphic encryption may seem straightforward, this is not always the case. On the other hand, there are situations that need the usage of many types of operations. It is common practice in the field of cloud computing to outsource calculations that are impossible to do on one's own owing to their scale or complexity. In order to do this, the homomorphic encryption techniques used must enable arbitrary activities. GSW: Let's suppose that the value of s in C1 and C2 is identical to the value of s in C2. Is it feasible to build an encryption system in which the secret key s is represented by C and C's eigenvalues are the message that m wishes to transmit as an eigenvector? If the eigenvector is to be kept hidden, it is advised that the message be buried; nevertheless, if the eigenvector is known, it is recommended that the message be retrievable.

It is feasible to find an eigenvalue in polynomial time using Gaussian elimination; hence, the answer seems to be no. Gentry, Sahai, and Waters [49] went one step further by employing approximation eigenvectors instead of eigenvectors as secret keys in order to further this concept. This action was taken to advance this notion further. In order to decrypt the ciphertext, the noise vector must have a norm value that is less than the modulus. When the requirement is no longer enforced, the LWE issue becomes very difficult to remedy. Gentry, Sahai, and Waters were able to create a successful encryption system by beginning with a basic principle that has served as the basis for several other encryption approaches. Homomorphic and fully homomorphic multi-key encryption: The possibility of fully homomorphic encryption (FHE) to encrypt data locally while outsourcing computation of the encrypted data without disclosing the actual data was one of the motivations for the development of this technology. FHE can solve this issue when ciphertexts are encrypted with the same key for a single user and only that user.

The multi-key FHE that was developed by Lopéz-Alt and colleagues [44] enables the computation of functions on ciphertexts encrypted with multiple public keys. After the calculation is complete, the ciphertext must be decrypted using the private keys of all parties involved. To put it another

way, all parties must cooperate in order to decode the ciphertext and get the output. If you currently have a GSW FHE and want to switch to a multi-key FHE, you may convert the GSW FHE to the multi-key FHE format. Clear and McGoldrick developed an extension of the GSW method for multikey FHE in 2014. As a consequence, the very first LWE-based multi-key FHE was produced. In 2016, Mukherjee and Wichs[47] published an updated and simplified version of their prior findings. This research led to the creation of SHMK FHE. If you intend to join ciphertexts encrypted using several ciphers, you must first complete a process known as bootstrapping. Before beginning the homomorphic computation, this phase stipulates that all necessary keys must be known. Before commencing the calculation, each of these investigations required comprehensive and correct understanding of all input data.

In 2016, Brakerski and Perlman [36] shown how to improve prior work to eliminate this restriction and allow an infinite number of homomorphic processes with an endless number of participants. This was made feasible via the expansion of previous work. It is possible to dynamically include additional parties in a computation throughout the course of an activity. This is permitted at any moment. In addition, they increased the length of the ciphertext and the complexity of the atomic operation space. This specific design is referred to as the FDMK (Fully Dynamic Multi-Key) FHE system. The approach's dynamic nature might be ascribed to the ease with which more players can be included into the computation. Gentry [5] offered Bootstrapping as a method for completing this job because of its seeming simplicity. [2] is one instance of this.

Table 2.1: Summary of the literature review

Author	Year	Method	Results or aims
Joseph Selvanayagam et.al	2018	AES, DES and RC-2	investigate the potential security risks associated with cloud-stored files using a variety of cryptographic approaches
Ben Huwang et.al	2018	AES and HEROKU Cloud	created a website for data security and used AES as a data security algorithm
H. Lei et.al	2019	virtualization cryptoraphy machine (VCM)	introduced the CC framework, demonstrating that users would be able to access cryptographic services through a cloud computing architecture
Ahmad.S.A et.al	2021	a novel technique to securing data	he addressed a variety of techniques used by various scholars
A.F. Pandey et.al	2019	AES algorithm	he explained the three security patterns, namely, filtering, encryption, and authorization
S. sarojini et al	2020	EMTACA algorithm	security challenges may be avoided by establishing a mutual trust between cloud services and cloud service providers

3. MATERIALS AND METHODS

This chapter describes the necessary and essential concepts in the development of this work, in order to make a classification of the existing cloud computer architecture for a better understanding of the different CC behaviors.

3.1 CLOUD COMPUTING

The term "cloud computing" (CC) refers to an emerging model for the delivery of computing infrastructure; many definitions of this concept are available online. One of the causes for this condition is due to the fact that computer security (CC) is a relatively new field, and the authors propose various strategies or directly tied to the qualities of the services that are provided. This is one of the reasons why this situation has arisen. Only a few of these definitions will be discussed in this book; nevertheless, those that are discussed include the primary elements that may be found in the other definitions. According to [12], cloud computing (CC) is a model that enables convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction. CC can be defined as a model that allows for this. Regarding [14], network computing (NC) can be conceptualized as a collection of virtualized computing services that are made available to users via the internet. These services are characterized by a number of important qualities, including low cost, location, reliability, high scalability, security, and sustainability.

According to [11], the CC is a large group of virtualized computing resources that are easily accessible and usable, such as hardware, development platforms, and services. These resources can be dynamically configured and adjusted to the workload in order to allow for the maximum use of these resources, which are economically exploited through the pay-as-you-go model. Not only does cloud computing deal with a computational paradigm, but it also deals with a business model. Cloud computing brings together a number of technologies, including virtualization, distributed computing, grid computing, and service-oriented architecture, amongst others. The model is a representation of the converging trends of efficiency and agility in information technology. [4] In order to attain efficiency, it is necessary to make optimal use of the processing capacity of current computers by relying on resources that are both highly scalable in terms of both hardware and software.

In addition to this, it incorporates the concepts of environmentally responsible computing by enabling computers to be physically situated in areas of the world where the cost of energy is lower and to be accessible remotely through the internet from any geographic location. The capability of fast deployment, processing in parallel, the use of intense computing for analysis, and the utilization of interactive mobile apps that reply in real time to user demands are all factors that contribute to agility. In the long run, agility can also be achieved by having the ability to make use of computational tools that can be rapidly deployed and scaled while simultaneously reducing the need for the enormous upfront investments that are typical of today's IT setups.

This is how long-term agility can be achieved. According to [2], the fundamental aspect for CC was the development and operation of extremely large-scale datacenters in regions with minimal operating expenses. These datacenters had to be in places where they could be run efficiently. In these settings, a decrease in expenditures associated with energy consumption, network bandwidth, operations, hardware, and software was found to be between five and seven times lower.

3.1.1 Key Features of Cloud Computing

Features such as on-demand service delivery, access flexibility, resource sharing, elasticity and measurement of consumed services are essential in CC. Table 3.1 presents the description of each of these characteristics.

Table 3.1: Characteristics of cloud computing [2]

Characteristic	Overall question	Capability
1. Trust, respect, and responsibility	To what extent are trust, respect, and responsibility core values?	• Openness and respect of opinions/ideas • Coaching and delegating management • Involvement and commitment to decisions
2. Fact-based decision making	Is fact-based decisions part of culture at all levels in the organization?	• Openness and respect of opinions/ideas • Coaching and delegating management • Involvement and commitment to decisions
3. Creativity and entrepreneurship	Is creativity encouraged, valued and part of product and technology strategy?	• Strategic role of NPD • System/process for capturing ideas • Separated research from development
4. Digital tools in product D&E	What is the perceived role of digital tools (relative to others) in achieving goals?	• People and process over tool and technology • Stabilization before automation • Implementation, rather than tool, as a competitive factor
5. Simple and visual communication	To what extent is visual communication anchored in the culture?	• A3s used for visual communication • Visual management practices/environment • Visual communication for learning and problem-solving

3.1.2 Cloud Computing Architecture

A significant number of the technologies that underpin CC, like as virtualization and grid computing, are not necessarily brand new. The convergence of these technologies, on the other hand, in an environment in which information can be accessed independently of the device used by the consumer or the geographic location of the customer, indicates a significant departure from the method of computing known as conventional computing. [16] The infrastructure of the cloud is a collection of hardware and software that is organized in two layers: the physical layer, and an

abstraction layer. The hardware resources that are necessary to deliver cloud services are contained inside the physical layer of the cloud. On the other hand, the abstraction layer is made up of the software that is installed on top of the physical layer, and it is the medium through which the cloud reveals its qualities.

[17] The physical layer's hardware resources, including as servers, storage, and network devices, are often housed in datacenters, which itself may be set up in a variety of geographic locations. The physical layer is one of the three layers that make up the OSI model. The virtualization and management layers are combined to form what is known as the abstraction layer. [18] Because it enables location independence, resource sharing, and the quick provisioning or release of these components, the virtualization layer is the necessary component for the cloud computing implementation. The management layer is responsible for monitoring traffic and reacting to spikes or dips in resource use by either generating new servers or eliminating ones that are no longer required.

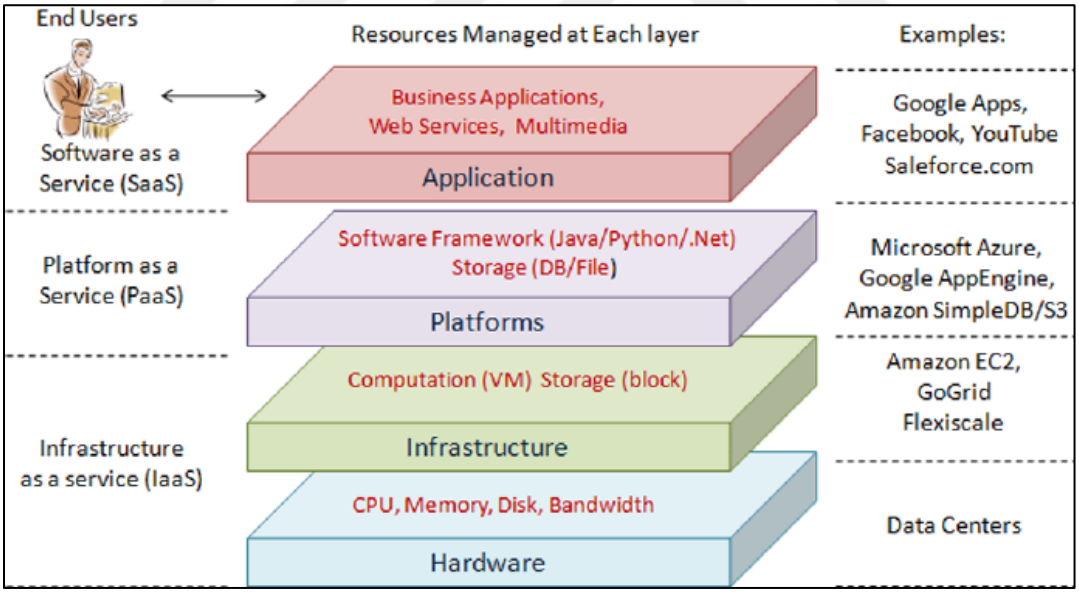

Figure 3.1: Cloud Computer architecture [16]

It is vital to be familiar with the most popular service models in order to have an understanding of the many forms of assets that may be housed in the cloud as well as the various forms of attacks that can be launched against them. It is essential to have an understanding of the ways in which various players, such as end users and external agents, may interact with the individual components that make up any service model. The different players' degrees of engagement in the process of

maintaining the safety of cloud assets are directly proportional to the types of interactions that take place and the associated levels of control [12]. The usual design of a cloud is shown in a clear and straightforward manner in Figure 3.2. This architecture highlights the fundamental aspects of cloud computing, including the many service levels that are used most often.

The many deployment models, including public, communal, hybrid, and private, are listed at the top. The following properties of cloud computing services are outlined in the table that can be seen directly below: big pool of available computer resources, provided on demand, with high levels of quick flexibility, in the form of a metered service, and delivered across a network with a substantial capacity.

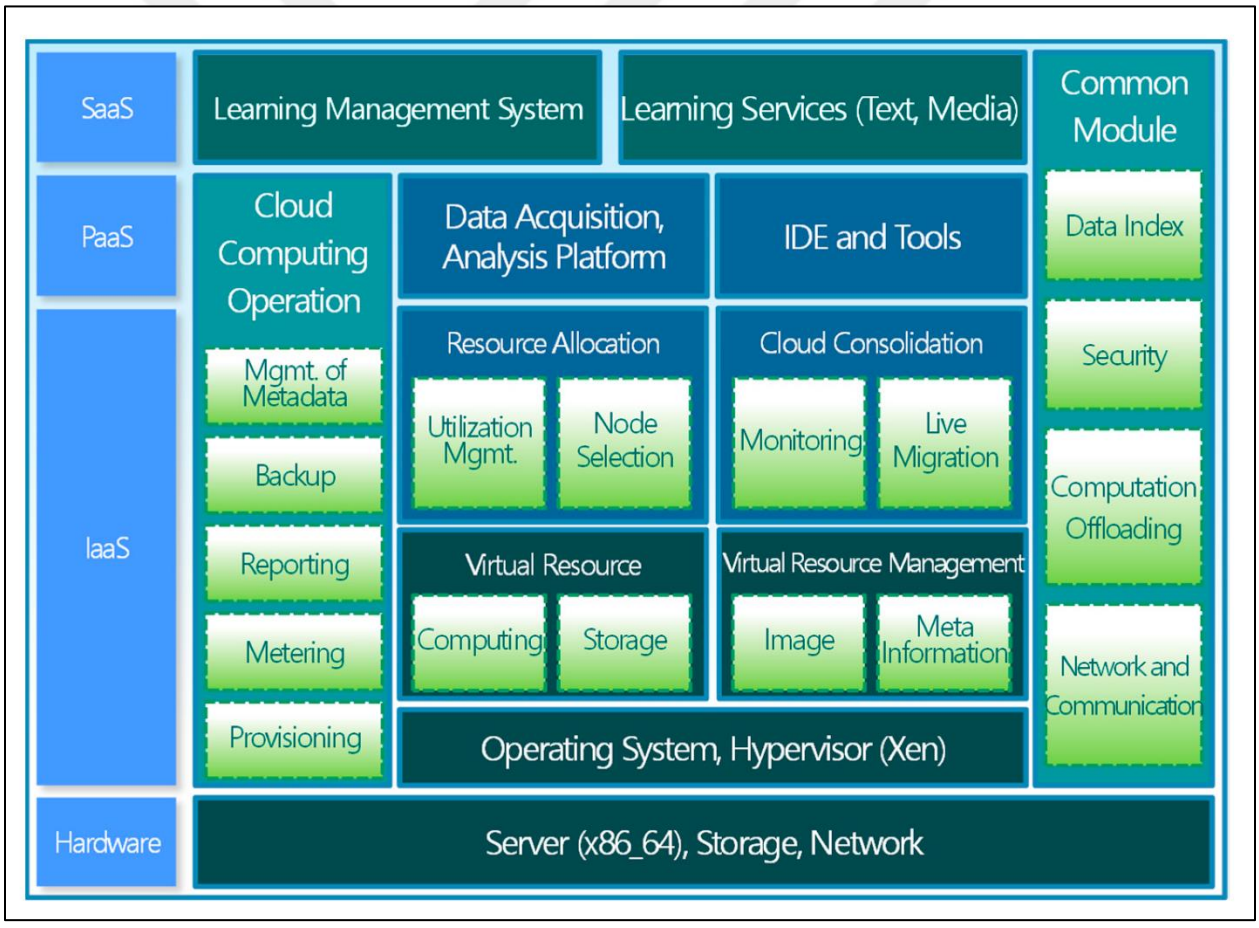

Figure 3.2: Elements of Cloud computing structure [12]

The public cloud is the most typical deployment strategy. In this approach, an organization makes its computing capacity available for purchase on an open market. A community deployment is a situation in which a consortium of organizations share the costs of deploying and maintaining data

centers, leveraging the resources through a metered service provisioning interface that is typical of cloud computing.

In general, this is referred to as a "community deployment." In a private deployment, a single organization is responsible for covering the expenses of the data center, while the organic units of that company make use of the resources available in the form of cloud services. And lastly, where there is extra capacity in a community or private deployment, and that capacity is supplied to the general public, a hybrid approach is being examined. The standard service layers are shown in the third frame of this animation. The services that are provided at the lowest layer are often referred to as IaaS. These services usually reflect more fundamental resources, which are direct representations of physical equipment. Some examples of these services include processor time, disk space, and network traffic. These solutions are used by way of a collection of virtual computers, each of which is administered by way of a bespoke user interface developed by the service provider. This interface need to provide capabilities such as generating virtual machines, deleting them, turning them on and off, copying and customizing them. There are more complicated services, such as application servers, database management systems, and other tools directly tied to the requirements of an application, which are located in the second layer [15].

The services provided in this layer are known as Platform-as-a-Service (PaaS), and they cater to consumers who have less stringent requirements. These consumers need only an environment in which to run an application, and they do not need to be concerned with the specifics of the environment, such as the settings of the operating system or the network interfaces. A management interface is used to facilitate communication between the consumer and the provider. This interface typically offers functionality such as the ability to create, make available, pause, or remove an application, as well as the ability to send and update application sources and configure resource usage limits. Finally, the highest-level services, which are categorized as Software-as-a-Service, are located on the very top layer (SaaS). Under this style of service delivery, the customer interacts directly with apps that were built by the service provider, and the customer pays a fee to utilize particular functions.

The ones that are the most expressive (in transaction volume and commercial value) include consumer relationship management (CRM) or sales platforms, particularly those that are associated with e-commerce, enterprise content management (ECM), enterprise resource planning

(ERP), and enterprise platforms. Risk Management (ERM). This industry is responsible for billions of dollars in revenue for organizations like as Salesforce and SAP. This cloud services stack demonstrates that the degree of control that is left to the customer decreases proportionally with the complexity of the services being provided. When you relinquish authority over your business's operations, you also relinquish control over the safety of its data. This does not imply that it will be more vulnerable because, on the other hand, it delegates the responsibility to the provider, who will probably have greater investment capacity and more qualified personnel, to know and respond adequately to the various threats. This does not mean that it will be more exposed.

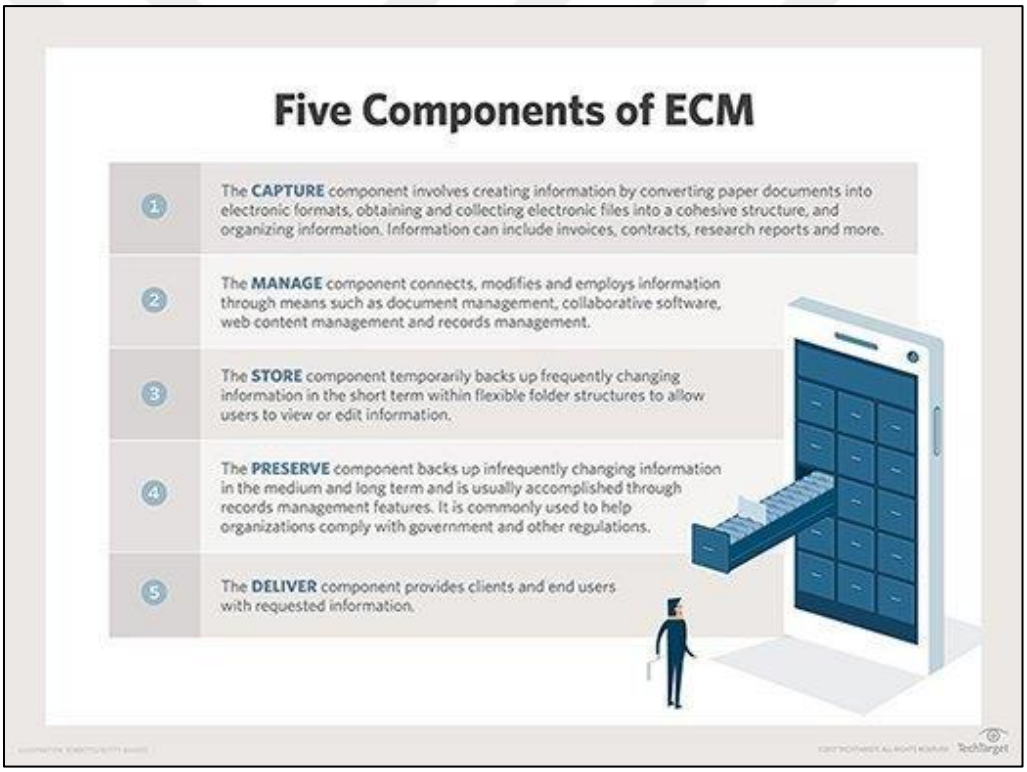

Figure 3.3: Enterprise Content Management (ECM) in cloud computing (CC) [15]

3.1.3 Service Delivery Models

When using CC, it is not required for the client to have knowledge about the number of servers utilized, the hardware and software configurations, how the processes are planned, or even the geographic location of the datacenter. CC abstracts all of this information for them. What is important is that users may access the service at any time, on any device, and in any location. It is

possible to install a cloud solution using a private, community, public, hybrid, or federated paradigm. The architecture of a cloud system will differ depending on which model is used. The features of the computing burden, and not necessarily the size of the organization, are the primary considerations that guide the selection of the model to be used. CC is now regarded as one of the most promising technologies in the information technology industry. This is due to the fact that, as a result of its distinctive qualities, it is fundamentally able to solve a number of shortcomings that have been recognized in conventional designs.

3.2 CLOUD NETWORK SECURITY

Cloud computing is a model of doing business in which a pool of computing power is made available over a high-capacity network, on demand and with elastic provisioning, in the form of a metered service [2]. [Citation needed] [Citation needed] [Citation needed] [Citation needed] [Citation needed] [Citation needed] [Citation The fact that cloud consumers can convert their capital expenditures into operational costs is the primary advantage of this model. This allows cloud providers to take full responsibility for deploying and maintaining the infrastructure that is necessary for the operation of their customers' systems, relieving cloud consumers of this burden entirely. As a result, they steer clear of the dangers and expenses associated with under- or over-provisioning [3]. The idea of delivering computers as a service is not a new one; in fact, discussions on the topic date back to the 1960s [9, 7].

However, the appropriate economic backdrop has only lately been identified, in which computers and communication are part of the fundamental business of the majority of firms. This realization came very recently. Because of this new economic setting and the maturing of technology that facilitates the remote delivery of computer resources, the market for cloud computing was able to emerge. The creation of virtualizes, hypervisors, and statistical techniques for multiplexing and load distribution are some of the causes that made this possible. But the expansion of capacity and the cheapening of internet services were unquestionably the deciding elements.

These characteristics made it possible for data centers to be placed distant from significant metropolitan areas, which reduced the cost of installation as well as the cost of power. The combination of these elements made it possible for an unprecedented expansion in the size of data

centers, which in turn produced the surplus capacity that is currently sold in the market for computing resources and is known as cloud computing [8].

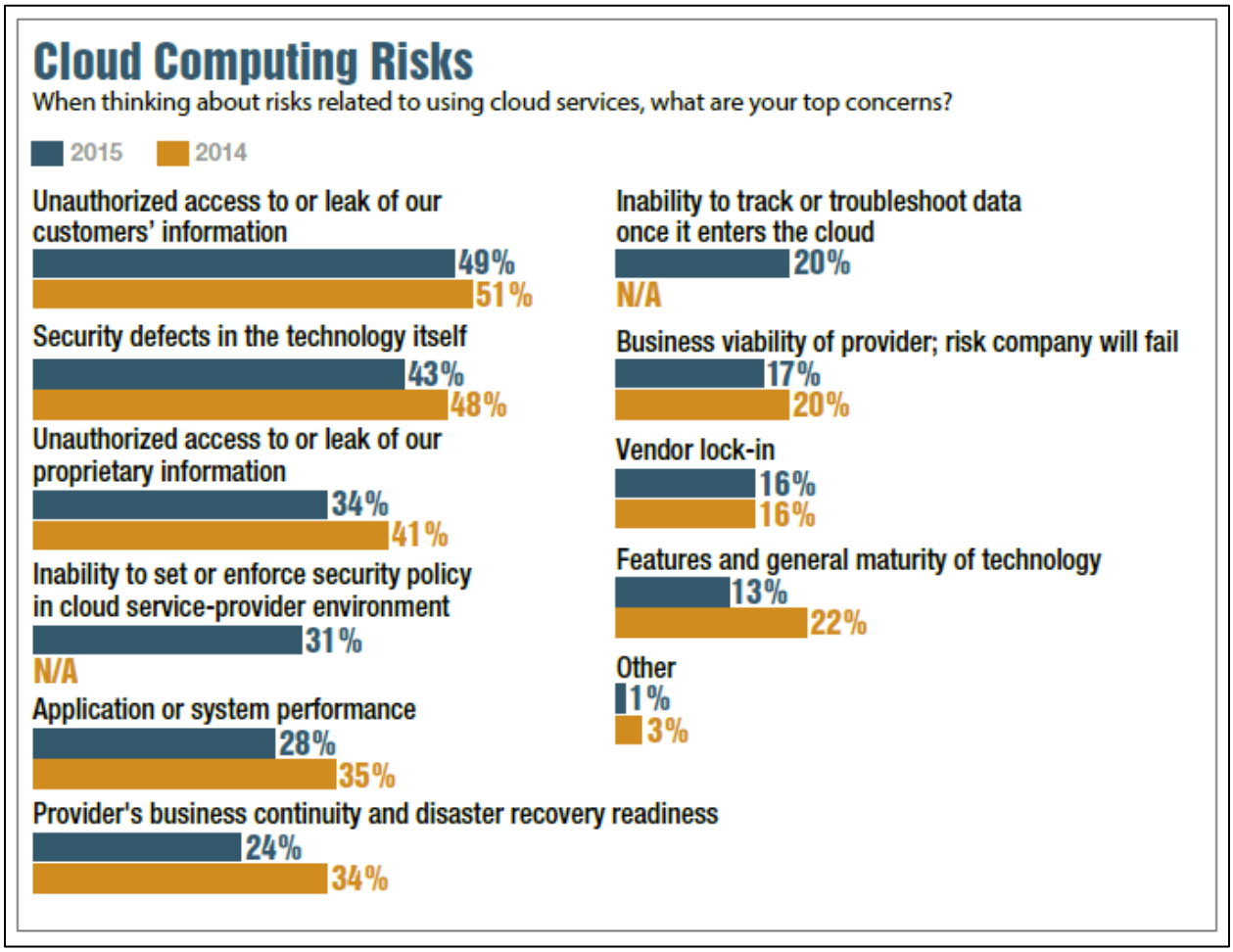

Figure 3.4: Risk factors in CC according to [12]

Naturally, the many parties (see Figure 3.4) in this market have their own unique interests, which results in a variety of perspectives on cloud security [12]. In the context of the economy as a whole, security refers to the measures taken to guarantee the company's ongoing operations and to safeguard its core principles (facilities, products, services, knowledge, corporate image, etc.).

This protection is achieved by a process that involves identifying, evaluating, and managing the risks of loss or disruption. Professor of computer security at the University of Cambridge Ross Anderson offers this definition of computer security: "Computer security is defined as a series of measures taken over the course of the entire life cycle of the software or hardware apparatus (design, development/manufacturing, deployment, maintenance, and improvement) in order to

prevent exceptions to the desirable operation of the software or hardware apparatus." It claims that any exceptions, in this context, may be described in terms of unlawful access to or modification of data, whether as a consequence of malice, error, or eventuality [5].

This definition applies regardless of the reason for the unauthorized access or manipulation. In other words, the availability, integrity, and secrecy of information are the fundamental factors that govern the proper functioning of computer systems. In a broad sense, cloud security can be viewed as a series of functional requirements that need to be clearly defined and implemented in the various layers of the service infrastructure. This is necessary in order to guarantee that the business expectations of the actors that are involved will be met. As a result, the interfaces, application programming interfaces (APIs), environment settings, operational and management procedures (such as maintenance and auditing), and service delivery models are only some of the parts of the cloud that are impacted by these needs.

3.2.1 Service Providers

Cloud service providers essentially want to ensure that they will be able to continue charging customers for the usage of their resources. In order to do this, they need to accurately measure as well as identify and eliminate any abnormalities in the usage of the resource. Having a trustworthy setting that customers are prepared to spend more for is obviously essential for them. Providers are also subject to the legal framework of the region in which they operate. This legal framework may require providers to comply with a variety of standards, certifications, and audit processes in order to maintain their ability to do business there. Therefore, the provision of security for providers is inextricably tied to the management of access to their assets, in addition to the maintenance of certain standards and levels of service. In turn, customers have a responsibility to continue exercising stewardship over their holdings, even if the technology infrastructure that underpins your business is being outsourced. Because customers do not have control over the continuity of the services they get from the cloud, they must further contend with the possibility that their cloud service provider or carriers would be unavailable (ISPs that give access to the high-capacity network to the provider, consumer or users finals).

This is significant because customers are ultimately responsible for the protection of the end-personal, user's private, and personally identifiable data, even if they may share control with

providers and carriers over the availability of their systems. Information security and end-user privacy are consequently more closely tied to cloud users' sense of safety while using cloud services. When carrying out the traditional tasks of the state, government institutions make it a priority to encourage the establishment of stable markets, which in turn contribute to the growth of the economy and the well-being of society. On the other hand, because they operate highly important systems, such as those used for monitoring and regulating the banking system, for fiscal and tax administration, and for controlling the justice process, they use an enormous amount of computing resources.

This is one of the reasons why they are so resource-intensive. For this precise reason, these types of institutions are entrusted with the most sensitive information that pertains to persons and businesses in a country, particularly that which is associated with matters of national security. Therefore, governmental bodies have a twofold interest in encouraging the evolution of cloud computing market technologies, agents, standards, and best practices. This is because they have a dual role to play in the market. For the sake of the market as a whole, or so that they may take use of cloud computing services that can be relied upon. Other players, such as auditors and brokers, have their market share virtually dictated by the availability of security standards and processes. This is the case because of the importance of the industry. Brokers or brokers

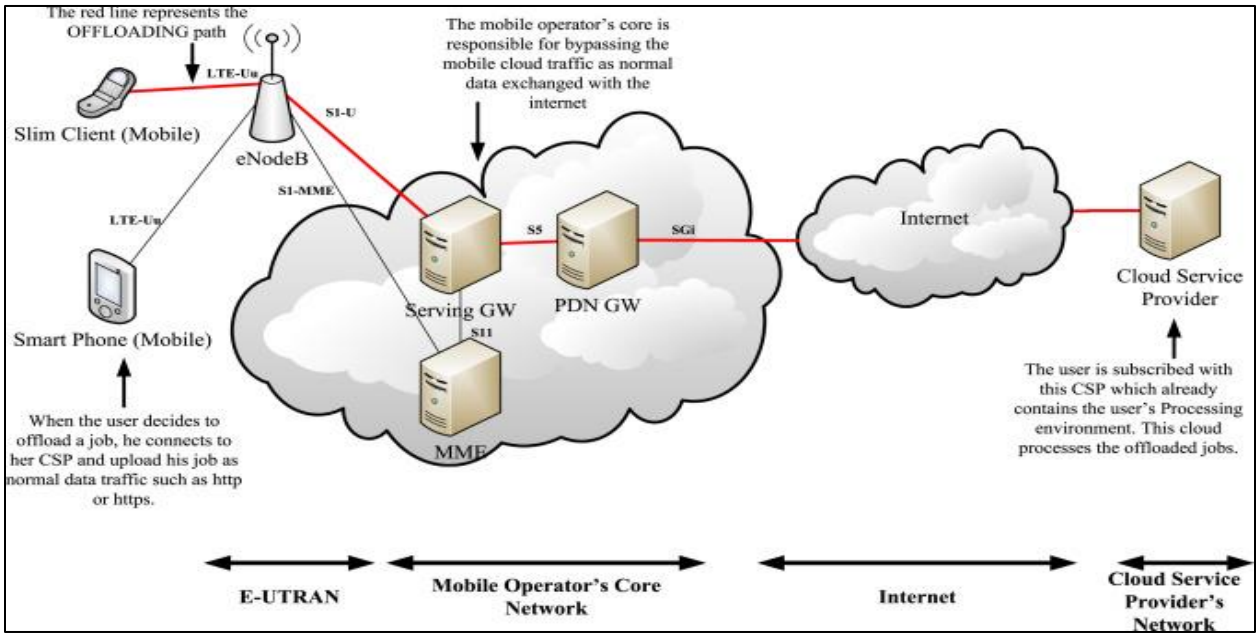

Figure 3.5: Combined ISP in CC [15]

As shown in Figure 3.5, may mix services offered by many suppliers or only add value to the offering of a single source. In any scenario, service level guarantees and security measures are an important component of your company's operations. Auditors, on the other hand, simply would not exist in this setting if there were not preexisting standards and norms of behavior that were relevant to cloud computing services. It is also essential to keep in mind that, in the vast majority of instances, the cloud consumer is not the final user of the computational inputs. The entity that pays the provider for the usage of resources, which are then consumed by the end user via the use of apps or services made accessible to them, is referred to as the consumer.

This might be a member of the cloud-consuming organization's employees, or it could be an agent working independently of the company. In the case of a virtual shop, for instance, the consumer would be the organization that is responsible for the store's upkeep, while the end user would be the client that engages in business transactions with that organization. In scenarios such as the one depicted in this illustration, it is very common for the end user to be unaware that they are interacting with more than one organization and that the data that they have entrusted to the store is, in a sense, also being delivered to the cloud provider. This is because the end user is dealing with more than one organization. Therefore, guaranteeing that only those companies that are permitted to have access to your data is an essential part of data security from the standpoint of end users.

This conversation is helpful for elaborating on the point that the security of cloud computing cannot be considered in a reductionist manner. It is not feasible to reduce it to a universal mathematical model since it is not a computing paradigm and it is not a particular class of computational systems. With that kind of model, it would be able to present security proofs and the limitations that they impose. Because of this, the literature in this field do not give any universally applicable answers. On the other hand, the vast majority of them are centered on particular attacks, particularly those that are associated with the invasion of virtual machines and the breaking of cryptographic keys via common side-channels in shared settings. Therefore, in order to evaluate the safety of a resource that is stored in the cloud, the first step is to decide which point of view to adopt: whose interests will be safeguarded? From the manufacturer, the purchaser, or the ultimate recipient? In addition, the security of an application or the data that it manages is not determined by the simple accumulation of mechanisms and technologies; rather, it is

determined by an in-depth comprehension of the connections that exist between the assets that need to be protected and the threats that are technically feasible and economically relevant.

3.3 SECURITY THREATS AND COUNTER MEASURES

There is a one-to-one correlation between the kind or degree of cloud service and the kinds of security risks that the customer is exposed to as a direct result of using that service. Customers of IaaS should be worried about the vulnerabilities that are characteristic of shared environments, as well as assaults on virtualization tools, denial of service attacks, and other types of threats to data processing. PaaS customers, on the other hand, are free to concentrate more on the safety of their data since the PaaS provider is responsible for ensuring that their processes are secure. It is abundantly obvious that customers of SaaS have a far greater degree of confidence in the provider, and their primary concerns are the consistency of the service levels and business circumstances (availability, performance, pricing, etc.) given by the supplier [59].

3.3.1 Homomorphic Encryption and DNS Hijacking

Take note that there is a feature that is shared by all of them, and that is the fact that any cloud service can be administered and used over the Internet. The user interface that an end user interacts with is almost often an application that was developed using web technology and may either be executed in a web browser or on a mobile device. As a result, everyone has to be prepared to defend themselves against attacks such as session injection, session fixation, DNS hijacking, CSRF, and XSS. The most fundamental step is to enhance the authentication and authorization processes by making use of cryptographic primitives. One example of this would be to implement the SRP (Secure Remote Password) protocol, which is described in RFC 5054 [33]. Increasing the degree of security of cloud apps is also possible via the use of methods such as two-factor identification [30].

Other minimum measures include the correct use of the HTTPS protocol, combined with the standardized security features present on the end user's machine. These standardized security features include HSTS (Strict-Transport-Security), CSP (Content-Security-Policy), and SRI (Sub resource Integrity) headers, which alter the interaction between the application server and the user device agent. This prevents the downgrade of TLS settings and mitigates the risk of injecting

malicious code into the When the objectives of an attack on a web application are exactly the customer account management interfaces with the cloud provider, the possibility for web application vulnerabilities is amplified. There have been many incidents of attacks in which control of the service was taken by taking advantage of these vulnerabilities.

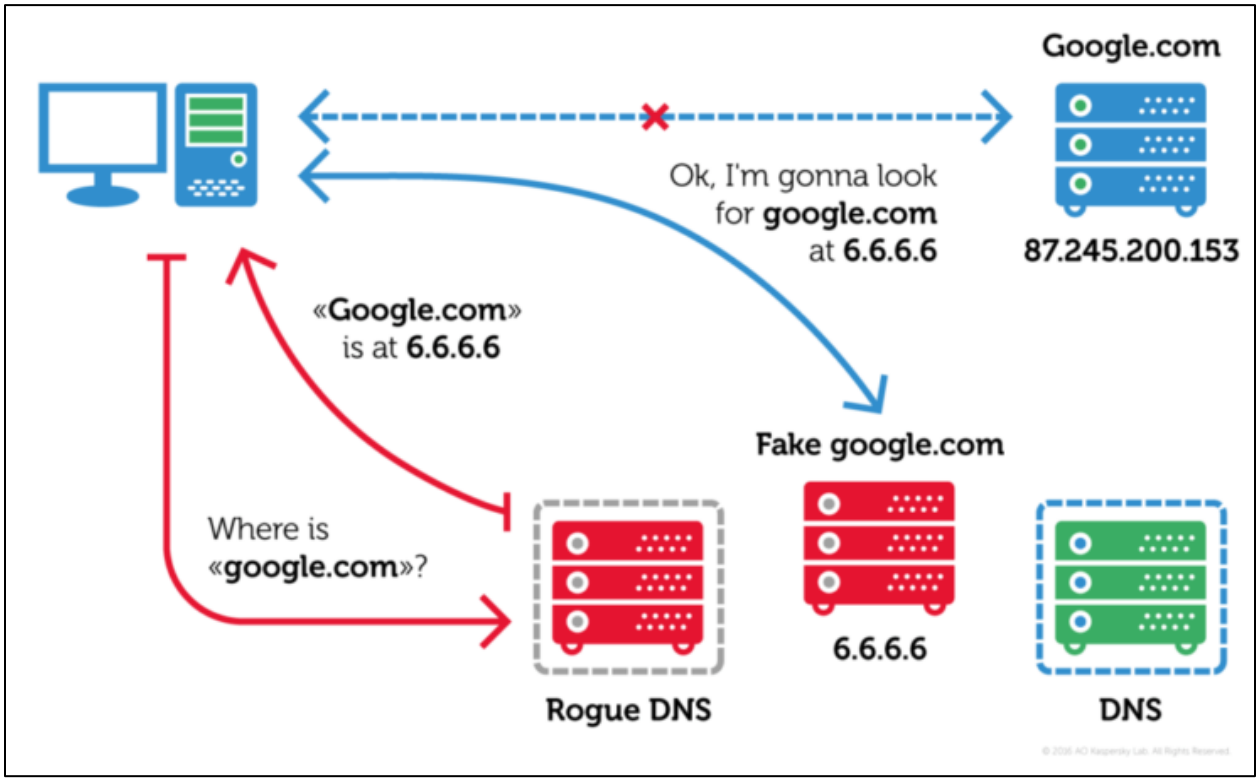

Figure 3.6: DNS hijacking through HSTS traffic in CC [33]

Another security vulnerability associated with administrative interfaces, particularly for IaaS and PaaS services, is that they are not very intuitive. There is also very little uniformity in terms of the ideas and capabilities that they provide. This results in the customer making setup mistakes, which ultimately creates vulnerabilities that may be exploited by attackers to get access to their service account or to their application [14]. Transport the vulnerabilities that are related to user agents (such browsers, email readers, and RSS feeds, for example) are related to the vulnerabilities that are related to the transport channel. The security of the HTTPS protocol, which is the primary response to threats to communication between providers, customers, and end users, has severely decreased as a result of widespread oversights in the installation and usage of the protocol.

The formation of a secure TCP channel by the use of another protocol, such as SSL or TLS, is necessary for the security of HTTPS on the one hand. Additionally, the most popular application servers and browsers on the market place a greater emphasis on the ease of use than they do on the requirements for security. As a result, during the initial steps of protocol negotiation, they continue to permit downgrading to older implementations of SSL that have been shown to be demonstrably insecure. On the other hand, even when greater care is taken in configuring the protocol that underpins HTTPS, even when the most recent version of TLS is used, and even when a stronger combination of cryptosystem and hash function is used, there are still vulnerabilities introduced by the HTTP protocol itself, or by the way that HTTPS uses the TLS channel. These vulnerabilities can be mitigated by using a stronger combination of cryptosystem and hash function. The reliance of HTTPS on the 'web-of-trust,' which is the trust model constructed around the public key and certificate infrastructures that provide meaning to the RSA keys that are used to authenticate hosts on the web [28], is the most significant flaw in the protocol. The X.509 certificate that is supplied by a web host is validated with the help of a pre-installed list of public keys that originate from reputable certificate authority.

This list is included with commercial browsers and is used to verify the certificate. This is how the browser determines whether or not the server is legitimate and also how it exchanges cryptographic keys that will be used to safeguard communications sent during a session. Impersonation of the certifying authority, faults in certificate revocation lists, and difficulties in the execution of validation processes are some of the several types of attacks that may be launched against this system. On many mobile devices, carelessness is so bad that just the certificate is tested for validity, but not its relationship to the host (domain) that provided it [40].

This is due to the fact that the only thing that is examined is the certificate's digital signature. Attacks against DNS and NTP protocols, which are rarely configured to travel over a secure channel, can also cause a device to accept an expired certificate or accept a valid certificate issued to a different domain. Both of these scenarios are possible due to the fact that DNS and NTP protocols are rarely configured to travel over a secure channel [18].

Table 3.2: DNS, NTP and DHCP purposes in clouds [18]

Protocol	Purpose
CDP/LLDP	To obtain VLAN/Voice VLAN, negotiate PoE
EAPOL	To authenticate endpoints to the access port
DHCP	To obtain IP address and initial configuration information
ARP	To discover the network's default gateway
TFTP/HTTPS	To obtain device configuration files, firmware, Certificate Trust List, etc.
NTP	Synchronization to the network clock, ensure accurate CDRs
DNS	To resolve hostnames to IP addresses

In addition, there are security flaws that are associated with certain implementations. These flaws may be fixed by simply upgrading the cryptographic libraries, but they continue to be a problem since consumers and providers are mostly stationary. An illustration of this would be the implementation of OpenSSL's TLS and DTLS (TLS variant for UDP channels) protocols, which are distributed on millions of Linux hosts. Even after the publication of concrete attacks (such as Heartbleed and FREAK), these protocols have not yet been updated on a significant number of servers [12]. It is possible to draw the conclusion that the security of communication channels is dependent on a series of measures that require total control of the configuration of application servers, cryptographic libraries, network interfaces, and other components of the operating systems.

This is something that is not feasible for the majority of consumers who use IaaS services, and it is simply unthinkable in the context of PaaS and SaaS services. As a consequence of this, it is essential to keep in mind that any information that is being transmitted will be made public, unless it has been previously and dependably encrypted on the user's device, and the keys that are required for this cipher are never transmitted along with the data. In this case, the information will be safe. Discretion and personal identity New technologies that are used in the construction of rich-interface web applications, such as the WebSQL, Web Storage, Web Sockets, Web Workers, and Web Messaging standards of HTML5, as well as the new message format and binary frame format of the HTTP/2 protocol, all introduce specific vulnerabilities. Although it makes programs

incredibly agile and responsive, the potential of collecting a significant amount of data on the client and opening many binary streams from a single HTML page might be harmful to users' privacy. Both of these features can be accessed from a single HTML page. The many traces that are left behind by the program may be exploited by utilizing sophisticated device fingerprinting methods in order to connect a device with an application user (fora, web mails, social networks, etc.).

It is the goal of various initiatives, such as the DNT (do not track) extension of the HTTP protocol and the standards proposed in the context of the PRISM (privacy-aware secure monitoring) project of the European Union, to lessen the negative impact that traffic analysis tools have on the privacy of users. The majority of projects aim to protect user privacy by altering one end of the connection, specifically the configuration and functionality of user devices. This may be observed in technologies that are widely used, such as the incognito mode found in browsers and proxy networks like the Thor project. All of these measures, however, are inadequate since the outcome is equally dependent on the care given by service providers and, more importantly, on a more proactive attitude on the side of the user [16]. The consumer of cloud services often has very little or no influence on the configuration or behavior of end-user devices. As a consequence of this, it does not have any solutions that can regulate the quantity of information that leaks to network traffic analyzers. However, it does have the potential to lessen the amount of user identification that is exposed via apps. It was for this reason that the German court ordered Facebook to block services such as public profiles, the search for friends, and the listing of persons who clicked on the "like" button linked with publications.

3.3.2 Threat Path Processing

The dangers that have been discussed up to this point are associated with the journey that the information taken from the device used by the user to the cloud server. Even if the information is sent to the cloud in a safe manner, while it is being processed, it will be vulnerable to a number of dangers that are characteristic of distributed applications and shared settings. The most fundamental kind is associated with the negligence of the programmer, who often forgets to encrypt the communications that are sent back and forth between the application server and the storage or database server, which, in a public cloud, might even be located on a different continent. Assuming there isn't such a fundamental error, the problem of building secure communication routes between the many components of cloud applications still remains.

3.3.3 Virtual Machines VM

The issue of obtaining a powerful and isolated source of randomness for each virtual machine (VM) is one of the largest obstacles that we may highlight among the most challenging tasks [72]. These different sources of randomness are an essential component of the cryptographic primitives as well as the protocols used for safe processing and communication. The most significant use is found specifically in the realm of algorithms that produce or derive cryptographic keys. Desktop computers are dependent on sources such as the electrical status of peripherals and user activity, which may be seen on the display as the number and address of running processes, visual components, or mouse pointer locations. These sources are not present on a server since a server does not have any peripherals and does not have direct interface with the user. As a result, it has the potential to produce keys for each virtual machine (VM) with a low level of entropy or keys that are quite similar to one another, if not precisely the same, in numerous of them. Another very common type of vulnerability involves flaws in the software used for virtualization, monitoring, and resource orchestration. These vulnerabilities, which can be exploited by an attacker so that the attacker obtains data from a VM, can also involve flaws in the communication between the components of this fundamental software stack. The major focus in the academic community has been the detection and mitigation of side-channel attacks, which leverage information spilled by these components to break the cryptographic keys of target virtual machines (VMs).

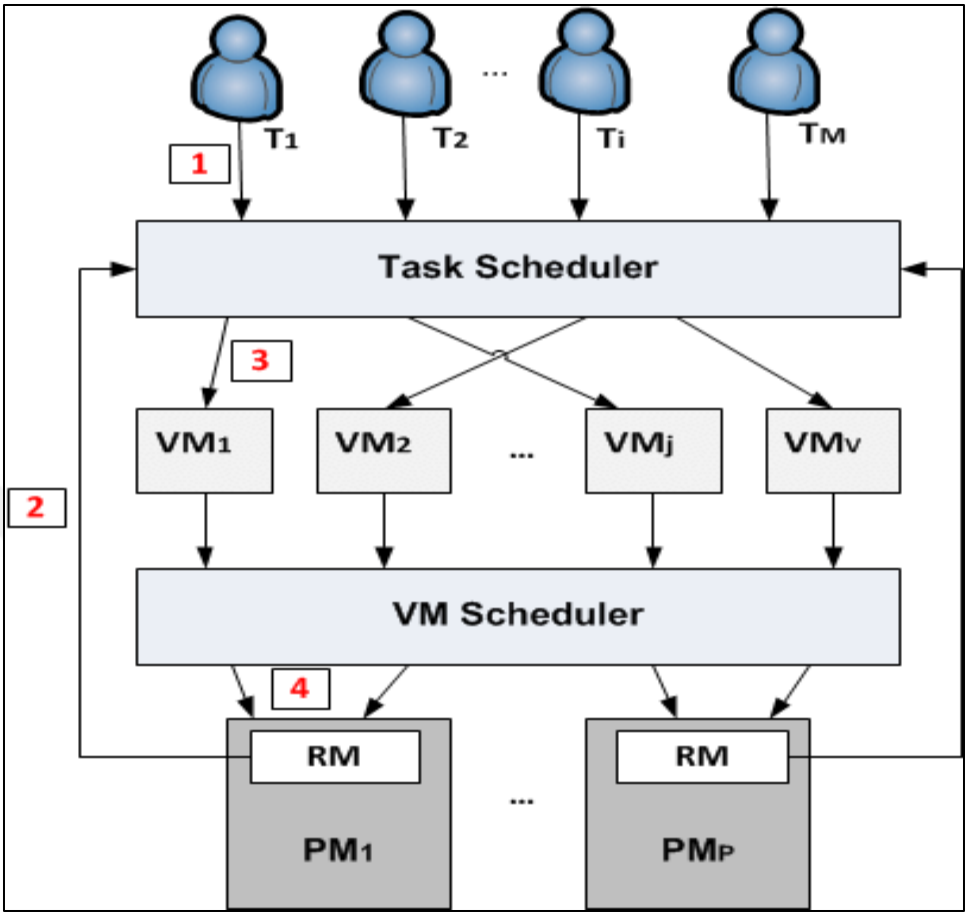

Figure 3.7: Role of VM ware in clouds [12]

In addition, the provider has the ability to engage in unethical behavior by exploiting the standard functionality of the software that maintains the virtual machines (VMs). For example, it has access to VM snapshots, which are comprehensive views of the state of a machine at a particular point in time. These views include the data that is persistently stored on virtual disks, as well as the contents of memory and network buffers. Because there aren't any auditing or abnormal usage detection tools integrated into the main software stack itself, the risk of assaults coming from inside the organization, known as the "insider threat," is significantly increased. Note that even if the attack is not carried out by an agent that is inside to the provider, the mere leak of a collaborator's credentials may expose all customers [12].

This is something that should be taken into consideration. Changing the structure of this software so that general administration privileges (such as monitoring the state of physical resources, enabling users, and assigning VMs to users, etc.) are kept separate from other administrative tasks

(such as taking snapshots, copying VMs, and other administrative tasks) is one solution to this issue. [24]. another approach would be to include more stringent auditing procedures, which would ensure that any effort at misuse made by an administrative agent would not be able to be covered up or removed [16]. The administration of virtual machines that are operating inside of virtual machines provided by the provider is an example of a different method that is referred to as "VM nesting." The consumer is the one who is responsible for managing and coordinating the operation of these machines and has the ability to use features such as encryption that is based on a password.

This ensures that an attacker cannot access the content of the second level even if they have complete access to the first level because they do not have adequate passwords and keys [74]. Compiling and configuring application servers and other components of this service architecture that are managed by the customer may help increase security, and even make it possible for processing to take place on an encrypted version of the data [13].

3.3.4 Consumer Platform

Customers that utilize services provided by a Platform-as-a-Service provider do not have a great deal of control over the service, and as a result, they are unable to use such intricate defensive strategies. Despite this, businesses may improve the degree of information security by reducing the surface area of their apps that are exposed to potential threats. The creation of barriers – preferably with the help of powerful cryptographic primitives – so that the information can only make sense within the context of a particular component of the application is known as the virtual partitioning of the application. This is a technique that is used quite frequently and is considered to be one of the most effective. Therefore, even if a vulnerability in the program is exploited at one point, the information created or processed in other partitions will still be safe [47, 36]. Encrypting all of the data using unique keys that are assigned to each user or data item represented is yet another method of logical partitioning that may be used. All of the processing requirements of less complex applications may be satisfied by using a mix of partly homomorphic cryptosystems as the application of choice. After being encrypted, the data is subsequently stored and processed utilizing application servers and database systems that are typical within the industry [35, 19]. Throughout the duration of this investigation, this tactic was used as the chosen approach.

The selection of cryptosystems that are appropriate for the semantic domain of the application, which means that they are appropriate for the format and nature of the data, as well as the essential operations that are to be performed on them while they are stored in the cloud, is the primary concern when implementing this kind of procedure. The user never gives up control of the data because the data are encrypted in such a way that it is possible to perform basic operations. This allows the user to use the computing power of the cloud without disclosing any relevant information to the provider or any other party. The rationale behind this decision comes from the intuition that the user never gives up control of the data attacker. There are works that concentrate on applications that deal with textual content, enabling keyword searches and even the establishment of chat rooms [11, 6, 5]. Other works concentrate on applications that deal with structured data, which necessitate the utilization of relational or object-oriented databases. [7].

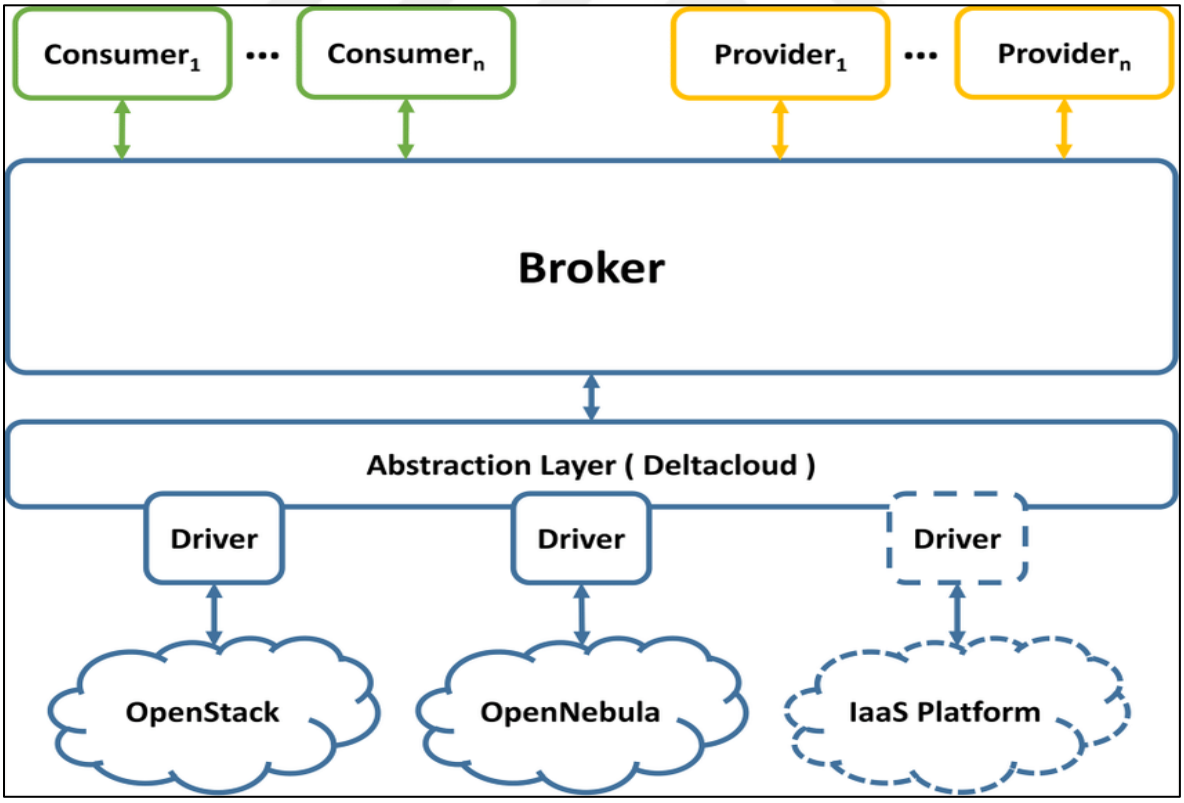

Figure 3.8: General view of the consumer and provider interaction platform in clouds [7]

3.3.5 Storage Space

Users and prospective consumers of cloud services have the most significant worries around data loss and leakage. Data leakage is without a doubt the most difficult challenge to solve, since it is dependent on security from beginning to finish, beginning with the user's device and continuing via transport channels, cloud processing, and storage. During this whole trip, there does not exist a single method or product that is capable of handling data security on its own. However, there are a variety of methods that may protect against data loss and enable the customer to make an accurate assessment of the quality and dependability of the services provided by the supplier. Techniques like as Proof-of-Ownership, Proof-of-Possession, and Proof-of-Retrievability make it possible for a customer with minimal computational power to design non-forgeable challenges for the provider to meet in order to demonstrate accurate data storage [55, 17, 15]. Oblivious Storage is a method that not only enables the user to conceal the information that is persistently stored in the cloud, but also the access patterns and the correlation that exists between the various blocks of data [17]. It is possible to say that this is the sector of cloud security activities that has produced the most mature results, with a number of commercial solutions already having been consolidated and being made readily accessible. The possibility of lock-in is another hazard; this one, however, is given far less consideration, despite the fact that it is maybe more pertinent. That is, how difficult it is to move an application along with all of its data to a different service provider. In the contractual connection that exists between the customer and the provider, the consumer will always have a hypo sufficient condition. That is to say, the customer is nothing more than a price taker and will never be in a position to ensure the upkeep of the contractual conditions on their own unilaterally. As a result, you are required to accept the possibility that the service provider may suddenly alter the terms or the quality of the service.

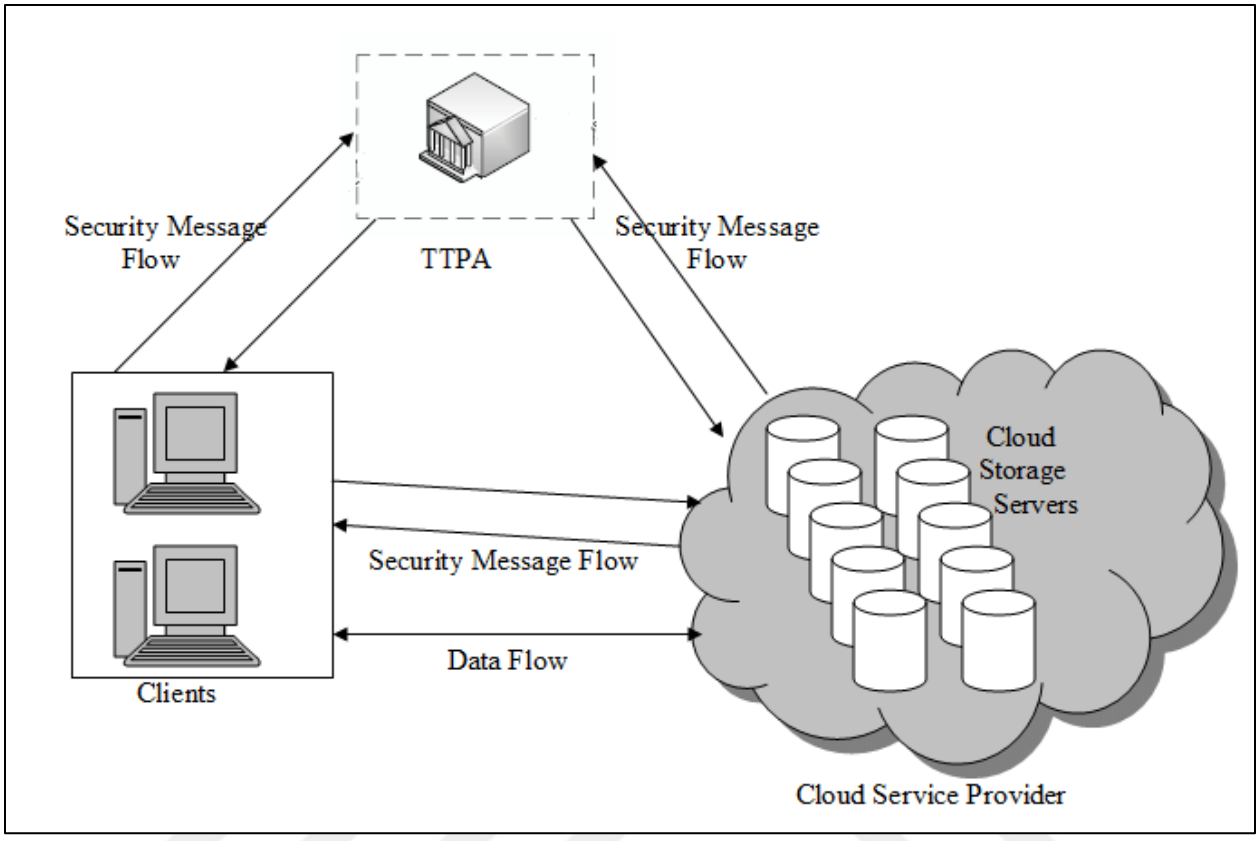

Figure 3.9: Storage mechanism in CC [17]

The service provider also has the option of withdrawing entirely from the market. As a result of all of these factors, customers may discover that they lack the resources necessary to ensure that they retain ownership of their data and can continue to access it. Lock-in occurs often as a result of the usual practice of charging more significantly for network traffic than the storage capacity itself. In addition, each service provider use a distinct file system, database, and even virtual machines and disk format for its customers. The full stack of services and technology that a supplier offers might vary quite a bit from one another. These diverse implementations not only serve as a tool for compelled customer loyalty but also attempt to simplify and expedite the process of integrating the several components of the provider's infrastructure in order to make it simpler and more cost-effective. For instance, Amazon divides its services up into highly specialized "zones" that each have their own software stacks that are suited for either storing, processing, or distributing data (CDN). Microsoft, the second biggest IaaS provider, arranges its solutions in a manner that is closer to that of ordinary operating systems, with all of the functionality in a single package, in a single core software stack. This is accomplished by using a single core software

stack. Each of the major providers of PaaS, such as Google and Heroku, manages its own application container format, as well as their own application programming interfaces (APIs), libraries, and even their own language to define and grade services. When a consumer deploys an application in the cloud, they must thus be prepared with tools that ensure the ability to effectively pick a new provider and move their data at any moment. This must be done before they install the application. The use of numerous clouds (or many cloud services) from the perspective of the application design is one of the methods described in the existing body of research [106]. The usage of frameworks that provide the decoupling of the application from the application server and other aspects of the environment [52] is another factor that should not be overlooked since it is a significant consideration. In addition, it is feasible to establish a general application programming interface (API) and then construct adapters that translate this generic API for each unique environment offered by the various providers.

4. SIMULATION AND RESULTS

In this chapter, we have proposed a method for generating a partly homomorphic encryption. This method ensures the preservation of the order between the encrypted data that is stored on the Cloud, which enables the execution of the various operations that can be performed on the data that is being protected. Because it is based on linear and modular expressions, this method is straightforward and easy to implement. We determined how difficult it would be to adapt a program so that it would function in accordance with our methodology. A set amount of performance and security effect may be supported by query and application types when employing linear computation. Python3 was also discussed while using a simple example in which it was believed that the cloud database is hosted by an unreliable service provider. During the implementation phase, we conducted an investigation into the suggested method and verified that it is applicable to a variety of cloud-based services. A significant portion of the testing is focused on validating its functioning within the context of a cloud database. Our suggestion is an approach that is more formal than architectural, and as a result, it is flexible to the database that is developed. This allows it to circumvent the limits that are inherent to relational databases. It is capable of meeting the requirements for high-frequency read and write operations, high-efficiency access and storage, high-availability, and scalability. Because a database requires a fixed schema and is typically organized in a distributed structure, and because cloud database data, in contrast to SQL database data, does not impose any data structure, there is no need to declare a schema of table before inserting it because there is no need to do so.

4.1 SYSTEM SETUP

In this work, we provide a description of a high-level security architecture of cloud database storage and communication services. Specifically, we focus on how these services interact with one another. A depiction in schematic form of the suggested architecture may be seen in figure 4.1. The framework was built with two levels of construction. The first layer is the database service provider, and it resides on a public cloud that cannot be trusted. The second layer of the architecture is known as the client tier, and it is deployed into the client environment via a client proxy. A proxy is used by the client in order to query the encrypted database. This proxy also maintains the communication that occurs between the encrypted database and the client applications. A request

that is carried out by the client is transformed by the proxy into an encrypted request that is processed directly in the cloud. The query result is decrypted by the proxy before being sent to the client once it has been obtained from the cloud after being processed. A metadata module, which includes database schemas as well as encryption and decryption keys, is required for the proxy to function properly.

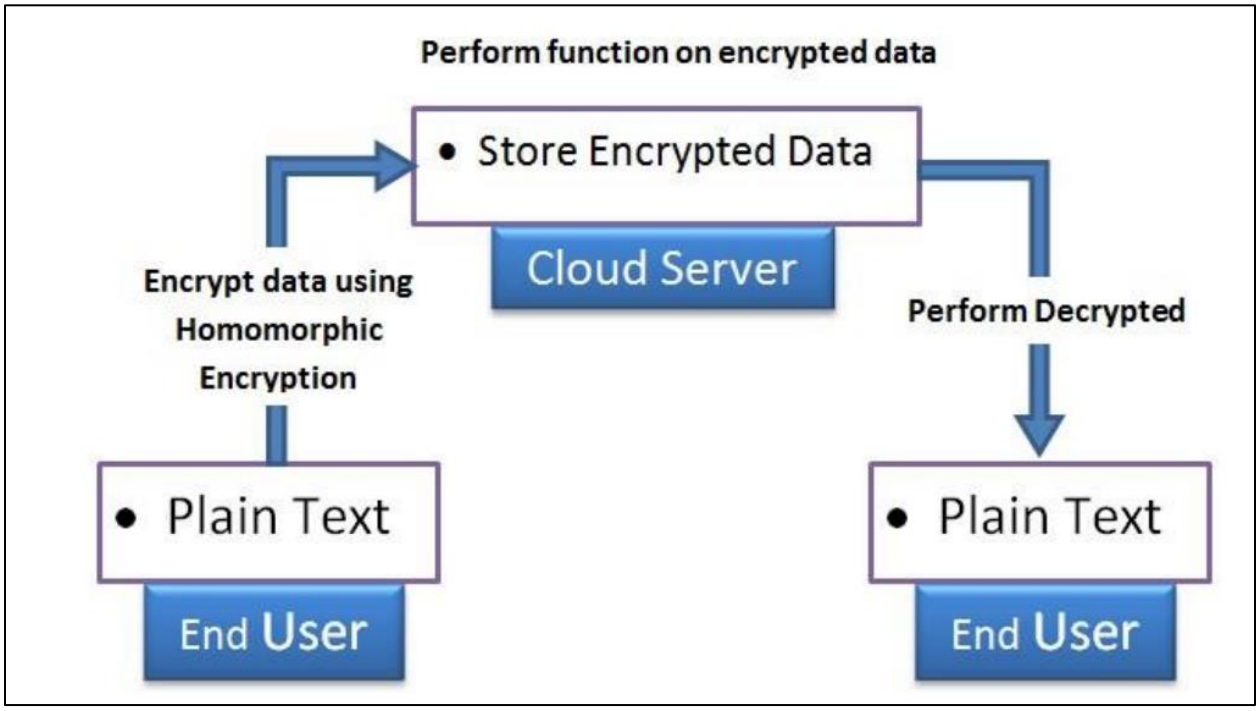

Figure 4.1: General view of the proposed system [9]

To be more specific, the proxy makes advantage of the PHE encryption and indexing method described earlier in order to construct an index and ciphertext (IA, CA) for each attribute A of the tuple before it is added to the database. The tuple is then saved into the encrypted database once it has been processed. The proxy does an examination of the request's syntax whenever it is received from the client. When it comes to the parameters of a ranking operation, the proxy determines the index that corresponds to the value in the query condition. On the other hand, when it comes to the parameters of a homomorphic operation, the proxy determines its encryption. After that, the proxy will deliver the query that was just passed to the cloud so that it may be run against the encrypted database. The proxy will decrypt the result after it has been received from the Cloud and is ready to be used.

During this stage, the proxy sends queries to the client in its capacity as the administrator of the database management system in order to establish the client's database (DBMS). Metadata is produced as a direct consequence of this activity; the structure of the metadata is determined by the proxy encryption procedure. The schema that was developed, which was then encrypted, is not the same as the one that is utilized in the client layer. All of the client's database schemas are included in the metadata, and an additional encrypted database schema is produced for each of the client's database schemas. Algorithm 1 outlines the processes that are involved in the transformation architecture that has been defined.

Algorithm 1: Paillier probabilistic encryption

Paillier's encryption scheme

Paillier's probabilistic encryption scheme PA_g is given by $(\mathcal{K}_\mathrm{P}, \mathcal{E}_{\mathrm{P}g}, \mathcal{D}_{\mathrm{P}g})$ as follows:

Key generation algorithm $\mathcal{K}_\mathbf{P}$:

(1) Choose two large primes p, q, with $\gcd(pq, \varphi(pq)) = 1$.

(2) Compute $n = pq$.

(3) Compute $d = \lambda(n)$.

(4) Choose $g \in \mathbb{Z}^*_{n^2}$ s.t. the order of g in $\mathbb{Z}^*_{n^2}$ is an nonzero multiple of n .

Output: $ek = (n, g)$ and $dk = d$.

Encryption algorithm $\mathcal{E}_{\mathbf{P}g}$:

Input: $ek = (n, g)$ and $m \in \mathbb{Z}_n$.

1. Choose random $r \in \mathbb{Z}^*_n$.
2. Compute $c = \gamma_g(m, r) = g^m r^n \bmod n^2$.

Output: $c \in \mathbb{Z}^*_{n^2}$.

Decryption algorithm $\mathcal{D}_{\mathbf{P}g}$:

Input: $dk = d$ and $c \in \mathbb{Z}^*_{n^2}$.

1. Compute $m = \dfrac{L(c^{\lambda(n)} \bmod n^2)}{L(g^{\lambda(n)} \bmod n^2)} \bmod n$.

Output: $m \in \mathbb{Z}_n$.

Because modular arithmetic uses both positive and negative integers to express quantities, negative numbers may also be represented by positive numbers. Every integer will be represented as a positive value that is smaller than n 2 in the cipher domain. In the notation Z n2, the multiplicative inverses of negative values are used to denote such values. When it comes time to decode the result that was produced for the sum (1), there will be a chance of two different values: one positive and one negative. This is one of the drawbacks that was discovered through carrying out this experiment. However, the numbers that were obtained are quite variable, which validates the conclusion that was made based on the issue modeling. The next part will provide a description of some of the experimental findings that were achieved in relation to the processing time.

Setup	Set $n = pq$, $\lambda = \text{lcm}(p-1, q-1)$. Select $g \in \mathbb{Z}^*_{n^2}$ such that $n \mid \text{ord}_{n^2}(g)$. Compute $\rho = \left(\frac{g^\lambda - 1 \bmod n^2}{n}\right)^{-1} \bmod n$, and s which is the integer with the least absolute value such that $\lambda \mid ns - 1$. Publish n, g and keep λ, ρ, s in secret.
Enc.	Given $m \in \mathbb{Z}_{n^2}$, set $m = m_1 + nm_2$. The ciphertext is $c \leftarrow g^{m_1} m_2^n \bmod n^2$.
Dec.	$m_1 \leftarrow \rho \left(\frac{c^\lambda - 1 \bmod n^2}{n}\right) \bmod n$, $m_2 \leftarrow (cg^{-m_1})^s \bmod n$. $m \leftarrow m_1 + nm_2$.

As was pointed out before, the Paillier cryptosystem is additive homomorphic, which gives rise to two distinct features. The first one says that the product of two ciphertexts will be decoded as the same as the total of the plaintexts that correspond to those ciphertexts, which means that:

$$D(E(m_1, r_1) \cdot E(m_2, r_2) \mod n^2) = m_1 + m_2 \mod n.$$

The second says that a ciphertext raised to a constant k will be deciphered as the product of the plaintext by the constant, that is:

$$D(E(m_1, r_1)^k \mod n^2) = km_1 \mod n.$$

In the next subsection, a way to use properties to homomorphically process the Gaussian quadrature calculation will be demonstrated. Homomorphic processing is done in the right part of equation (1). Values f(xi) are encrypted, while weights wi are kept flat. Therefore, property (5) is used to perform the multiplication wi · E(f(xi)) and property (4) is used to calculate the sum, as follows:

$$D(E(f(x_i), r_1)^{w_i} \mod n^2) = w_i \cdot f(x_i) \mod n;$$

Paillier calculations can only be performed with positive integers, due to modular arithmetic, however, Gaussian quadrature uses decimal numbers. To make these systems compatible, a fixed-point representation was used. Choosing a fixed precision of decimal places j, a decimal number α can be represented by the integer a, whose first digits will be the integer part of α and the last j digits will be the j fractional digits of α.

4.2 IMPLEMENTATION OF PAILLIER ALGORITHM

In the beginning, the programming language Python 3 was used in order to carry out an implementation of Paillier's method. This programming language has built-in support for modular exponentiation as well as arbitrary-precision integers (making it unnecessary to implement an algorithm such as double-and-add). In addition to the functions that have previously been specified by the theory, an implementation of the Euclid algorithm that includes integers was also carried out. The program finds the inverse of the integer a, which is a negative one, if it exists (mod n). During the process of key generation, it is used to do the calculation of L (g (mod n 2)) 1 (mod n). After then, the technique was implemented in the json programming language by making use of the PHE library (partially homomorphic encryption Library), which is a library that has been enhanced for the arithmetic of arbitrary-precision integers. This was done for performance reasons. Not only does the library enable modular exponentiation, but it also has a native implementation for the modular inverse computation and the Miller-Rabin primality test, both of which are required in order to get k-bit random primes during the process of key creation.

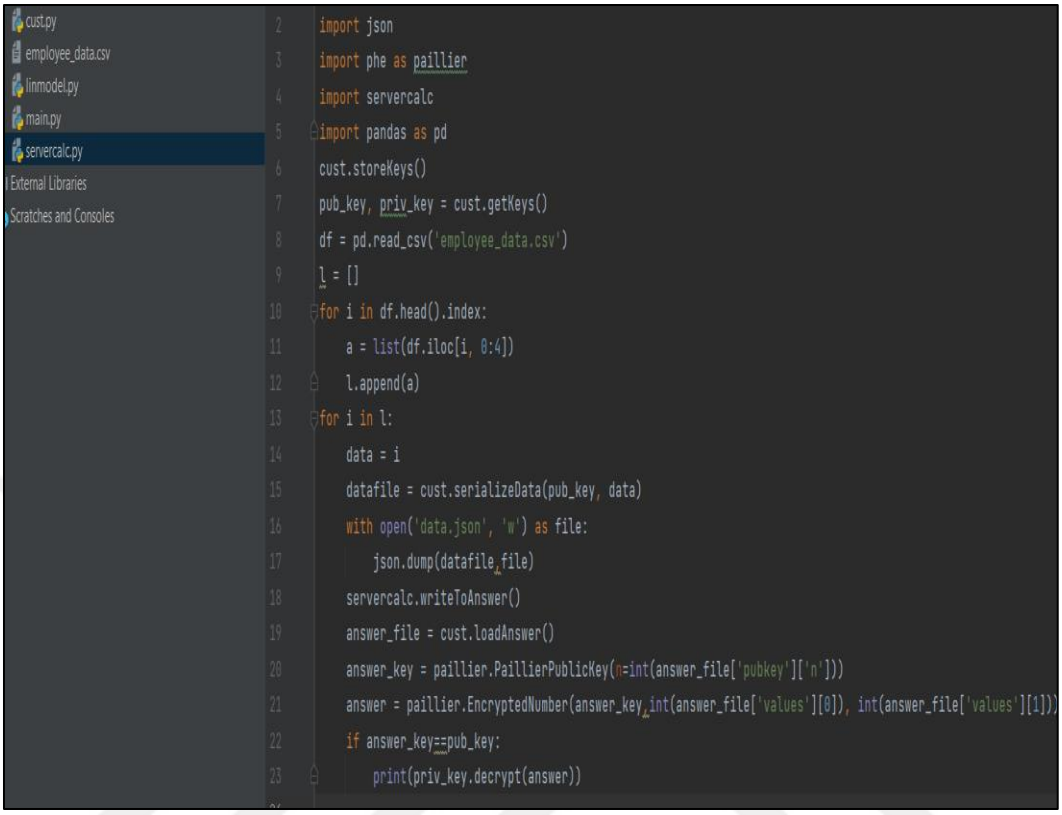

Figure 4.2: Python 3 libraries needed for the execution

The following procedures need to be carried out in order to produce random k-bit primes: A random k-bit string may be generated using either the 'dev'urandom output of Windows or the 'rdrand' instruction, provided that the latter is enabled by the CPU (it has better performance because it is a simple machine-level instruction). After that, it performs a straightforward AND operation with 1 in order to validate that the value of the string is an odd integer. The next step is to perform the Miller-Rabin primality test, which will reveal if the number is prime with an error probability that is inversely proportional to the number of iterations of the method. The Miller-Rabin primality test was developed by Miller and Rabin. If there is a low chance of it becoming a prime, the procedure will be carried out once again.

```
TEST (n)
1. Find integers k, q, with k > 0, q odd, so that
   (n − 1 = 2^k q);
2. Select a random integer a, 1 < a < n − 1;
3. if a^q mod n = 1 then return("inconclusive");
4. for j = 0 to k − 1 do
5. if a^(2^j q) mod n = n − 1 then return("inconclusive");
6. return("composite");
```

Note that at each iteration it is necessary to obtain the inverse of the highest degree coefficient of v, however this must be the inverse modulus q, obtained through the Euclid's algorithm extended to integers.

Table 4.1: Paillier encryption and some variants that will later be used for comparison

	$n = pq$ is an RSA modulus, $\lambda = \text{lcm}(p-1, q-1)$.
Variant 1 (Paillier)	$g \in \mathbb{Z}^*_{n^2}$, $\text{ord}_{n^2}(g) = \alpha n$. PK: n, g; SK: α.
	$m \in \mathbb{Z}_n$, $r \in \mathbb{Z}_n$, $c = g^{m+rn} \bmod n^2$.
	$m = \left(\frac{c^\alpha - 1 \bmod n^2}{n}\right) / \left(\frac{g^\alpha - 1 \bmod n^2}{n}\right) \bmod n$
Variant 2 (Damgård-Jurik)	$\kappa = \tau\lambda$, $\tau = \lambda^{-1} \bmod n$. PK: n; SK: κ.
	$m \in \mathbb{Z}_n$, $r \in \mathbb{Z}_n$, $c = (1+mn)r^n \bmod n^2$.
	$m = \frac{c^\kappa - 1 \bmod n^2}{n}$
Variant 3 (Choi-Choi-Won)	$g^\lambda = 1 + n \bmod n^2$. PK: n, g; SK: λ.
	$m \in \mathbb{Z}_n$, $r \in \mathbb{Z}_n$, $c = g^m r^n \bmod n^2$.
	$m = \frac{c^\lambda - 1 \bmod n^2}{n}$
Variant 4 (Catalano-Gennaro -Howgrave-Nguyen)	$e < n$, $d = e^{-1} \bmod \phi(n)$. PK: n, e; SK: d.
	$m \in \mathbb{Z}_n$, $r \in \mathbb{Z}_n$, $c = (1+mn)r^e \bmod n^2$.
	$m = \frac{\frac{c}{(c^d \bmod n)^e} - 1 \bmod n^2}{n}$

Both calculating the polynomial inverse during key creation and performing modulo q reductions are examples of activities that are very resource-intensive. Since it is not required for q to be a

prime, we are free to select it as a power of two in order to maximize the efficiency of modular reductions. To do this, simply carry out an AND operation between the number and a mask in which the log q 1 most significant bits are set to 1 and the remaining bits are set to 0. Note that since polynomials in C are represented as pointers to allocated regions, the exchange of vectors u v and r s in the expanded Euclid method may be done in constant time rather than O time. This allows the process to run much more quickly (N). Despite the fact that q values are used in the literature that may be represented by 32-bit integers, we decided to utilize the Phe library since each coefficient of the polynomial is represented by a multiple-precision integer. Because we anticipated that the numbers would be somewhat huge, we decided it would be best to tackle an early implementation in C for 64-bit integers.

In order to perform operations such as multiplication and modular reduction between 64-bit numbers, it was essential to develop a library for 128-bit integers. The library that was obtained, in addition to having a restricted amount of bits, also had a performance that was inferior to that of PHE, which had been optimized by the use of vectorization and assembly instructions. For this reason, we would rather use PHE in the process of putting Paillier into effect.

4.3 TEST RESULTS

All of the codes that were used for the experimental assembly that is described in this section were written in the PYTHON3 programming language. They made use of the PHE library, which not only offers a native implementation for the modular inverse calculation and primality tests, but also offers an optimized representation of arbitrary precision integers. The OpenMP Application Programming Interface (API) for the C programming language was used to measure the time by calling the pub key, priv key = cust.getKeys() function. Experimentation was carried out on a computer equipped with an Intel i7-6500U 2.50 GHz quad-core processor, 16 gigabytes of random access memory (RAM), and the Windows 11 operating system in order to evaluate the amount of time required to perform the Gaussian quadrature calculation in plane mode and homomorphically when compared with Paillier. In order to provide an approximation of the definite integral R 5 5 and x cos(x)dx, the Gauss-Legendre quadrature was randomly selected as the method to be used for the testing. Experiments were carried out for approximations ranging from the order of five to one hundred, and the accuracy of the decimal places used ranged from three to nine. The results

of the computation using flat values are shown in Figure 4.3, along with the average processing times that were achieved.

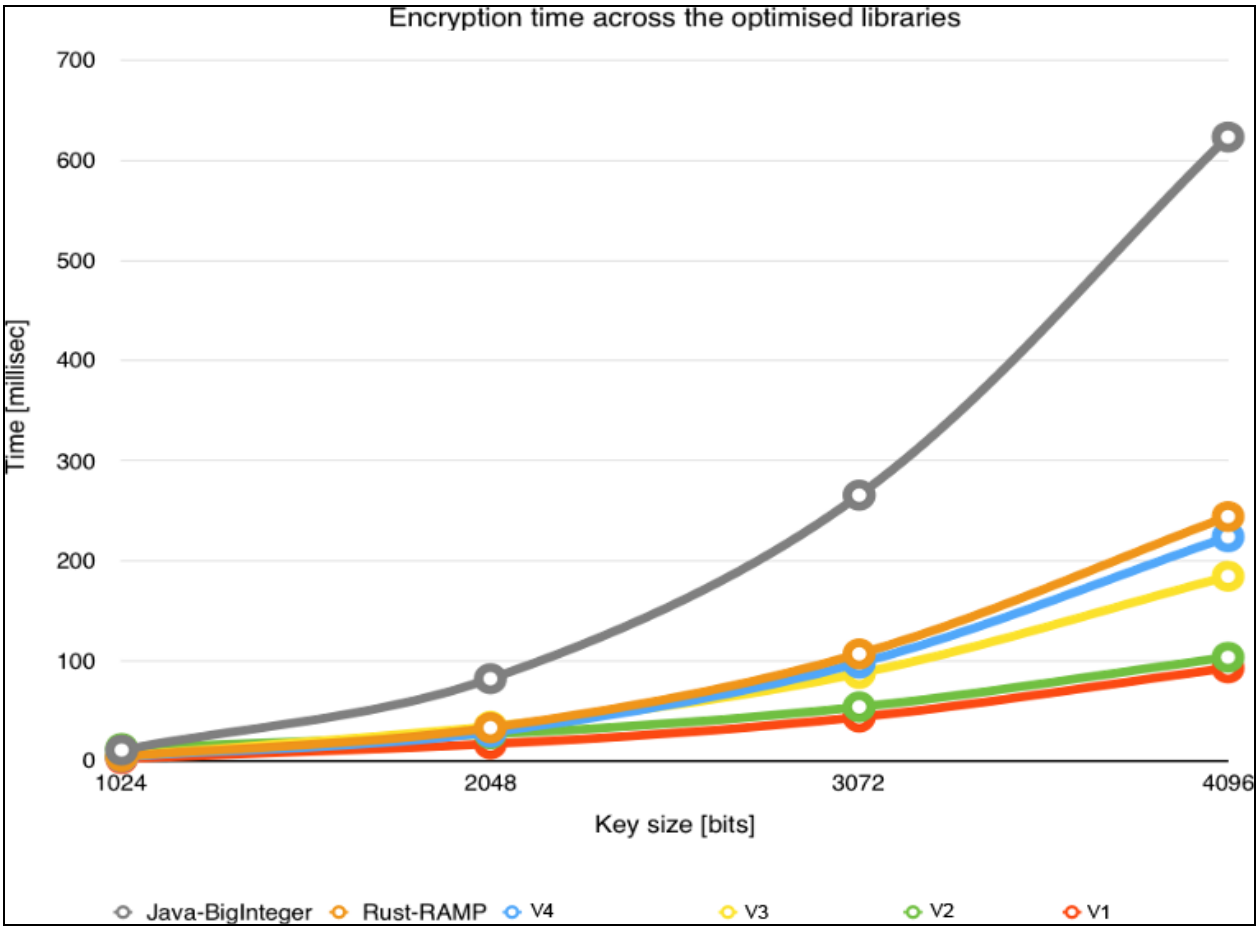

Figure 4.3: Encryption time comparison of the proposed V1 algorithm and other variations.

Based on our examination of the data and the graph, we are able to draw the conclusion that the degree of accuracy used contributes positively to the amount of efficiency that is lost when homomorphic processing is utilized. When precision is applied to three decimal places, the homomorphic processing time is roughly 1153 times slower than the plaintext processing time. When precision is applied to nine decimal places, the average value of the homomorphic processing time rises to 3234. Utilizing parallelization strategies is one of the possible alternatives that may be used to increase the performance of cryptography using Paillier.

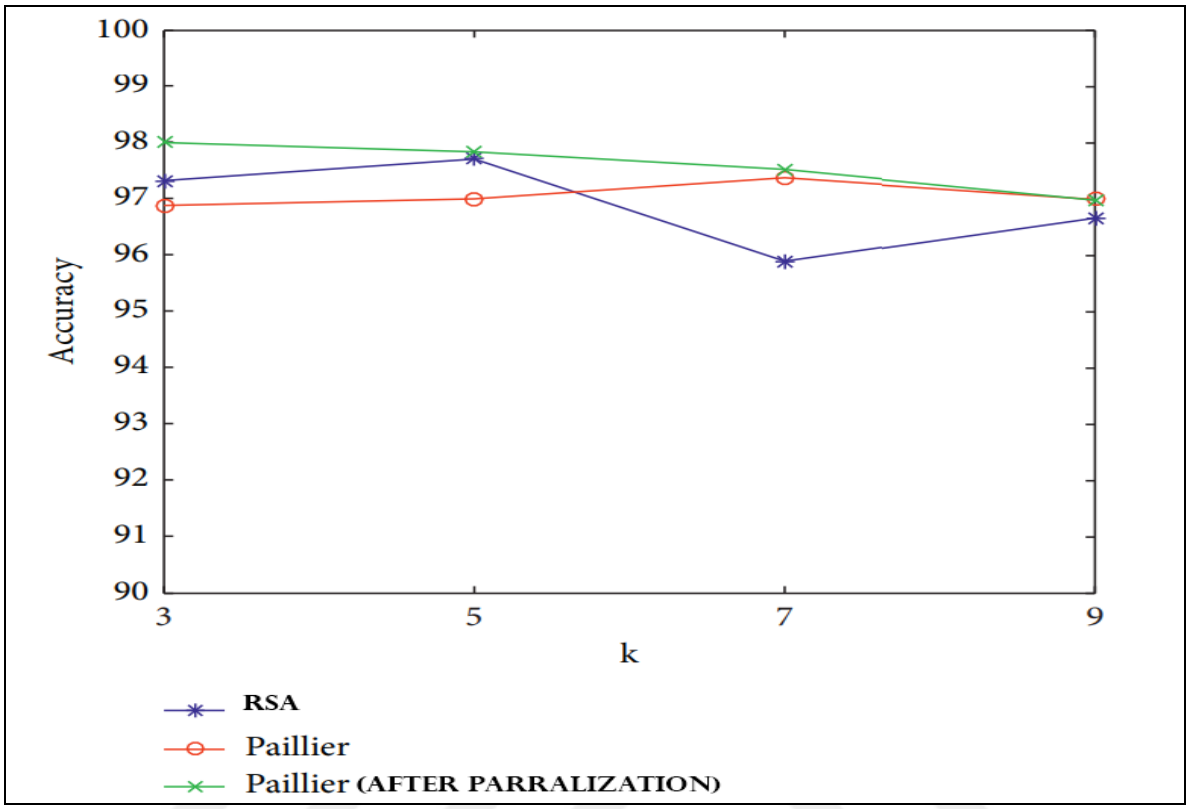

Figure 4.4: Performance of the proposed method after parallelization. [14]

Devised an effective parallelization technique for modular exponentiation that may be found in their paper. The concluding thoughts of this work will be addressed in the part that comes after this one. It was anticipated that the use of homomorphic encryption would lengthen the processing time.

Table 4.2 Evaluation of different dataset for the same encryption scheme

Dataset Name	Dataset Size	Functions and Processing Time		
		Encryption Time	Decryption Time	Indexing Time
DataSet 1	781 KB	163,094 MS	147,634 MS	36,221 MS
DataSet 2	26 KB	446,63 MS	15,427 MS	38,641 MS
DataSet 3	170 KB	62,272 MS	39,447 MS	23,246 MS
DataSet 4	55 KB	42,020 MS	26,858 MS	20,487 MS
DataSet 5	588 KB	116,069 MS	87,067 MS	19,553 MS
Data Set 6	594 KB	98,270 MS	80,102 MS	21,466 MS
DataSet 7	1.2 MB	191,919 MS	103,678 MS	31,051 MS
DataSet 8	50 KB	41,209 MS	16,991 MS	25,801 MS
DataSet 9	440 KB	69,467 MS	42,287 MS	31,779 MS
DataSet 10	139 KB	58,358 MS	61,839 MS	30,218 MS

The findings of the experiments support the hypothesis that using homomorphic processing of data results in a significant loss of efficiency. According to these findings, using homomorphic processing of data is on average 3234 times slower than using flat processing of data, with a precision of nine decimal places. This provides an excellent contrast to the increased safety and discretion that are realized via the use of homomorphic cryptography. When it comes to making decisions, the idea of reasonableness should always be used. This includes deciding whether or not to utilize one approach instead of another. It is essential to keep in mind that the results were achieved for the same system and did not make use of any optimization techniques.

The ability of homomorphic encryption to handle sensitive data on devices belonging to third parties is one of its primary advantages. The use of cloud processing or machines with high computational power can enable homomorphic computation of data without compromising its security or privacy. This makes this strategy a potential option for use with low-computing-power devices, such as sensors and smart cards, which are examples of this type of device.

5. CONCLUSIONS AND FUTURE WORK

In this thesis, contributions to the fields of HE cryptography and analytical query processing on HE encrypted and kept in the cloud were presented. These contributions include: (i) the specification of a cryptography scheme known as the homomorphic cryptography methodology for records; (ii) the specification of an OLAP system known as COOL, which processes analytical queries over encrypted records maintained in the cloud by utilizing the scalability provided by the cloud; and (iii) an extensible query processing system for encrypted records.

5.1 CONCLUSIONS

Cloud computing has the potential to save organizations money, but it still has to overcome a number of challenges before it can realize this promise. It is necessary to find solutions to the problems of data security, privacy, and confidentiality that surface as a direct consequence of extensive network access. At the present time, homomorphic encryption is one of the most efficient technologies available for preserving the confidentiality of data stored in the cloud. It is generally accepted knowledge that all homomorphic encryption methods allow for the processing of part or all encrypted data, which in turn increases the data's level of security. In this article, we take a look at homomorphic encryption algorithms and schemes, both of which are important areas of discussion within the realm of cryptography. This study tackles these problems in further depth by making use of a cloud computing encryption technology known as Homomorphic Encryption (HE), which is detailed in length throughout the paper. In addition, an in-depth discussion is held on the most frequent methods that may be used in order to achieve the appropriate degree of security. The use of cloud computing comes with a number of benefits, some of the most notable of which being flexibility, accessibility, and scalability. Nevertheless, this presents a number of potential security risks, which calls for severe precautions to be taken. Understanding the possible dangers and issues with security in today's culture is very necessary in order to keep oneself secure in this world. Working in the cloud requires a greater amount of effort from all involved parties (customers, providers, and the network) in comparison to the implementation of conventional security solutions. This is necessary to keep up with the dynamic and ever-evolving nature of the cloud computing environment. The following are some final recommendations for how to act when

faced with dangers and aggressors. Ensure the application's safety by putting in place a number of different levels of protection. Not duplicate (e.g. factor authentication).

The management of authentication and identity procedures. Data that is sent to a location that is not part of your local network should be encrypted; cloud servers should have this capability. It is the most secure means of storing data. Organizations should safeguard their data by using cloud-based security measures. Organizations must choose the cloud computing model that most effectively fulfills their requirements (e.g., group can use a hybrid model if they need to employ personal information on the private model, and they can use a public model to manage application).

When establishing security settings, you should avoid utilizing the defaults that the manufacturer has given. Make use of application interfaces that have a significant amount of security. Isolation and division of multi-tenant systems are both possible via the use of methods such as segmentation and isolation. Customers who are aware of the relevance of safety are more likely to have a positive connection with the service provider they use.

5.2 FUTURE WORK

Researchers have longed for a cloud-based operating environment for a very long time. Data privacy and security have emerged as two of the most critical factors to consider while designing a wide range of applications, particularly those that will be hosted in the cloud, in the modern world. We want to focus more in the future on the following fields of study: — - Gentry's creation of completely homomorphic encryption is recognized as a key achievement in the world of cryptography as of 2010. This encryption approach permits the linking of varied computations in order to homomorphically process encrypted data. Fully homomorphic encryption enables an entity to do computations on behalf of a user and provide just the encrypted result. On the other hand, this is not a viable solution in terms of the performance itself. In the sections that follow, we will evaluate the totally homomorphic encryption system that we have built and compare it to other current cryptographic methods that use partial homomorphic properties. FHE has the additional constraint that several users cannot access the system simultaneously. Homomorphically vast and sophisticated algorithmic calculations are required; however, this is not feasible for real-world applications such as fully homomorphic encryption (FHE), which requires a substantial amount of

computer labor. In order to discover a solution to this issue, we will study a number of potential solutions.

REFRENCES

[1] Joseph Selvanayagam1, Akash Singh2, Joans Michael, Jaya Jeswani,Secure File Storage on cloud using cryptography: (IRJET),2018

[2] Bih-Hwang Lee, Ervin Kusuma Dewi, Muhammad Farid Wajdi Data Security in Cloud Computing using AES under HEROKU cloud: IEEE 2018

[3] S. Lei, Wang Ze-wu, "Research and Design of Cryptography Cloud Framework," IEEE. 2018.

[4] S. A. Ahmad and A. B. Garko, "Hybrid Cryptography Algorithms in Cloud Computing: A Review," 2019 15th International Conference on Electronics, Computer and Computation (ICECCO), Abuja, Nigeria, 2019, pp. 1-6, doi: 10.1109/ICECCO48375.2019.9043254.

[5] Pandey S., Purohit G.N., Munshi U.M. (2018) Data Security in Cloud-Based Applications. In: Munshi U., Verma N. (eds) Data Science Landscape. Studies in Big Data, vol 38. Springer, Singapore.

[6] Sarojini, G. & A, VIJAYAKUMAR & Selvamani, K.. (2017). Trusted and Reputed Services Using Enhanced Mutual Trusted and Reputed Access Control Algorithm in Cloud. Procedia Computer Science. 92. 506-512. Mezzovico, Switzerland.

[7] B. Bindu, K. Lovejeet & L. Pawan, "Secure File Storage In Cloud Computing Using Hybrid Cryptography Algorithm", International Journal of Advanced Research in Computer Science 9(2), 2017.

[8] C. Biswas, U. D. Gupta and M. M. Haque, "An Efficient Algorithm for Confidentiality, Integrity and Authentication Using Hybrid Cryptography and Steganography", International Conference on Electrical, Computer and Communication Engineering, pp. 1-5, 2019.

[9] N Jirwan, A Singh & S Vijay, "Review and Analysis of Cryptography Techniques", Inter. J.Sci. Engineer. Res. 4(3): 1-6, 2019

[10] Y. Sharma, H. Gupta & S.K Khatri, "A Security Model for the Enhancement of Data Privacy in Cloud Computing", Amity International Conference on Artificial Intelligence pp.898-902. doi: 10.1109/AICAI.2019.8701398, 2019.

[11] Lundervold AS, Lundervold A. An overview of deep learning in medical imaging focusing on MRI[J]. Z Med Phys, 2019, 29 (2): 102-127.

[12] Suzuki K. Overview of deep learning in medical imaging[J]. Radiol Phys Technol, 2017, 10(3): 257-273.

[13] Ma XY, Hadjiiski LM, Wei J, et al. U-Net based deep learning bladder segmentation in CT urography[J]. Med Phys, 2019, 46 (4): 1752-1765.

[14] Chen LC, Papandreou G, Kokkinos I, et al. DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs[J]. IEEE Trans Pattern Anal Mach Intell, 2018, 40(4): 834-848.

[15] Chandra S, Kokkinos I. Fast exact and multi-scale inference for semantic image segmentation with deep Gaussian CRFs[J/OL]. Siddhartha Chandra lasonas Kokkinou, 2016. (2019-01-11) [2020-01-11]. http://github. com/siddharthachandralgcrf. DOI: 10.1007/978-3-319-46478-7_25.

[16] Zheng S, Jayasumana S, Romera-Paredes B, et al. Conditional random fields as recurrent neural networks[J]. IEEE Int Conf Comput Vis, 2015: 1529-1537.

[17] Shelhamer E, Long J, Darrell T. Fully convolutional networks for semantic segmentation[J]. IEEE Trans Pattern Anal Mach Intell, 2017, 39(4): 640-651.

[18] Liu Z, Li X, Ping L, et al. Semantic image segmentation via deep parsing network [EB/OL]. (2016-04-15)[2018-07-14]. http: //www. reserchgate. net/publication/281670742. DOI: 10.1109/ CVPR.2017.549.

[19] Lin G, Milan A, Shen C, et al. Refinenet: Multi-path refinement networks with identity mappings for high-resolution semantic segmentation[J/OL]. arXiv, (2016-11-25) [2018-07-14]. 1611.06612v3, 2016. https://ieeexplore. ieee. org/document/ 8100032/citations# citations. DOI: 10.1109/CVPR.2017.660.

[20] Zhao HS, Shi JP, Qi XJ, et al. Pyramid scene parsing network [C/OL]. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). arXiv, (2017-04-27) [2018-07- 17]. 1612.01105v2, 2017. https://www.computer.org/csdl/proceedings-articlecvpr/2017/0457g230. /12OmNvrMUeP

[21] Chen LC, Papandreou G, Schroff F, et al. Rethinking atrous convolution for semantic image segmentation[J/OL]. arXiv, (2017-06-17) [2018-06-17]. 2017, 1706.05587v1. https://ui. adsabs.harvard.edu/abs/2017arXiv170605587C.

[22] D. Boneh, E.-J. Goh, and K. Nissim, Evaluating 2-DNF formulas on ciphertexts, in Theory of Cryptography - TCC'05, ser. Lecture Notes in Computer Science, vol. 3378. Springer, 2005, pp. 325-341.

[23] C. Gentry, S. Halevi, and V. Vaikuntanathan, A simple BGN-type cryptosystem from LWE, in EUROCRYPT, 2010, pp. 506–522

[24] M. Fellows and N. Koblitz, Combinatorial cryptosystems galore!, Finite Fields: Theory, Applications and Algorithms, 1993, pp. 51–61.

[25] E. Kushilevitz and R. Ostrovsky, Replication is not needed: Single database, computationally-private information retrieval, in FOCS, 1997, pp. 364–373.

[26] M. Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, Fully homomorphic encryption over the integers, in EUROCRYPT, 2010, pp. 24–43, http://eprint.iacr.org/2009/616.pdf.

[27] N. P. Smart and F. Vercauteren, Fully homomorphic encryption with relatively small key and ciphertext sizes, Public Key Cryptography, ser. Lecture Notes in Computer Science, P. Q. Nguyen and D. Pointcheval, Eds., vol. 6056. Springer, 2010, pp. 420–443.

[28] D. Stehlé and R. Steinfeld, Faster fully homomorphic encryption, in ASIACRYPT, 2010, pp. 377-394.

[29] Z. Brakerski and V. Vaikuntanathan, Fully homomorphic encryption from ring-LWE and security for key dependent messages, in CRYPTO, vol. 6841, 2011.

[30] Paulo Martins, Leonel Sousa and Artur Mariano, A Survey on Fully Homomorphic Encryption: An Engineering Perspective, ACM Comput. Surv. 50, 6, Article 83, 2017, https://doi.org/10.1145/3124441.

[31] Abbas Acar, Hidayet Aksu, A. Selcuk Uluagac, Mauro Conti, Survey on Homomorphic Encryption Schemes: Theory and Implementation, 2018, ACM 0360-0300/2018/07-ART79, https://doi.org/10.1145/3214303.

[32] Samiha Jlilab, Hassan Satori, Khalid Satori, Computing on Encrypted Data into the Cloud through Fully Homomorphic Encryption, TMLAI-Transactions on Machine Learning and Artificial Intelligence, Volume 5, No 4, 2017.

[33] M. Seetha and A. K. Koundinya, Comparative Study and Performance Analysis of Encryption in RSA , ECC and GoldwasserMicali Cryptosystems, vol. 3, no. 1, 2014, pp. 111–118.

[34] Jaydip Sen , Homomorphic Encryption: Theory & Applications, 2013.

[35] Z. Brakerski and R. Perlman, Lattice-based fully dynamic multi-key fhe with short ciphertexts, in Annual Cryptology Conference, Springer 2016, pp. 190–213.

[36] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, Fully homomorphic encryption without bootstrapping. Cryptology ePrint Archive, Report 2011/277, 2011, http://eprint.iacr.org/2011/277.

[37] N. P. Smart and F. Vercauteren, Fully homomorphic simd operations, Des. Codes Cryptography, vol. 71, 2014, pp. 57–81.

[38] Z. Brakerski and V. Vaikuntanathan, Efficient fully homomorphic encryption from (standard) lwe, SIAM Journal on Computing, vol. 43, no. 2, 2014, pp. 831–871.

[39] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, (leveled) fully homomorphic encryption without bootstrapping, in Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, ITCS '12, ACM, 2012, pp. 309–325.

[40] L. Ducas and D. Micciancio, Fhew: Bootstrapping homomorphic encryption in less than a second, in Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2015, pp. 617–640.

[41] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. Cryptology ePrint Archive, Report 2016/870, 2016, http://eprint.iacr.org/2016/870.

[42] S. Halevi and V. Shoup, Helib. https://github.com/shaih/HElib. 2017.

[43] A. López-Alt, E. Tromer, and V. Vaikuntanathan, On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption, in Proceedings of the forty-fourth annual ACM symposium on Theory of computing, ACM, 2012, pp. 1219–1234.

[44] J. Hoffstein, J. Pipher, and J. H. Silverman, Ntru: A ring-based public key cryptosystem, in International Algorithmic Number Theory Symposium, Springer, 1998, pp. 267–288

[45] Alexander Wood, Kayvan Najarian, Delaram Kahrobaei, Homomorphic Encryption for Machine Learning in Medicine and Bioinformatics, ACM Comput. Surv. 0, 0, Article 0 (0000), https://doi.org/10.1145/1122445.1122456.

[46] P. Mukherjee and D. Wichs, Two round multiparty computation via multi-key fhe, in Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2016, pp. 735-763.

[47] C. Peikert and S. Shiehian, Multi-key fhe from lwe, revisited, in Theory of Cryptography Conference, Springer, 2016, pp. 217–238.

[48] C. Gentry, A. Sahai, and B. Waters, Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attributebased, in Advances in Cryptology–CRYPTO 2013, Springer, 2013, pp. 75–92.

[49] M. Clear and C. McGoldrick, Multi-identity and multi-key leveled fhe from learning with errors, Cryptology ePrint Archive, Report 2014/798, 2014. http: /eprint.iacr.org/2014/798.

[50] Goldreich, O., Goldwasser, S., Halevi, S.: Public-key cryptosystems from lattice reduction problems. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, Springer, Heidelberg 1997, pp. 112– 131.

[51] Micciancio D, Improving Lattice Based Cryptosystems Using the Hermite Normal Form. In Silverman, J.H. (ed.) CaLC 2001. LNCS, vol. 2146, Springer, Heidelberg, 2001, pp. 126–145.

[52] Gentry, C., Halevi, S. Implementing Gentry's Fully-Homomorphic Encryption Scheme. In Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, Springer, Heidelberg 2011, pp. 129–148.

[53] Jean-Sébastien Coron, Avradip Mandal, David Naccache, and Mehdi Tibouchi, Fully Homomorphic Encryption over the Integers with Shorter Public Keys, P. Rogaway (Ed.) CRYPTO 2011, LNCS 6841, pp. 487–504.

[54] Shai Halevi and Victor Shoup. Algorithms in HElib. In: CRYPTO 2014, Part I. Ed. by Juan A. Garay and Rosario Gennaro. Vol. 8616. LNCS. Springer, Heidelberg, 2014, pp. 554-571, doi: 10.1007/978-3- 662-44371-2_31.

[55] Shai Halevi and Victor Shoup. HElib - An implementation of homomorphic encryption, 2014, https://github.com/shaih/HElib.

[56] Shai Halevi and Victor Shoup. Bootstrapping for HElib. In: EUROCRYPT 2015, Part I. Ed. by Elisabeth Oswald and Marc Fischlin. Vol. 9056. LNCS. Springer, Heidelberg, 2015, pp. 641-670, doi: 10.1007/978-3-662-46800-5_25.

[57] Hao Chen, Kyoohyung Han, Zhicong Huang, Amir Jalali, and Kim Laine, Simple Encrypted Arithmetic Library (SEAL) v2.3.0. https://www.microsoft.com/enus/research/project/simple-encrypted-arithmeticlibrary/, 2016.

[58] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. TFHE: Fast FullyHomomorphic Encryption Library over the Torus, 2016, https : / / github . com /tfhe/tfhe.

[59] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. Faster Fully Homomorphic Encryption: Bootstrapping in Less Than 0.1 Seconds. In: ASIACRYPT 2016, Part I. Ed. by Jung Hee Cheon and Tsuyoshi Takagi. Vol. 10031. LNCS. Springer, Heidelberg, 2016, pp. 3–33. doi: 10.1007/978-3-662-53887-6_1.

[60] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène, Faster Packed Homomorphic Operations and Efficient Circuit Bootstrapping for TFHE, In: ASIACRYPT 2017, Part I. Ed. by Tsuyoshi Takagi and Thomas Peyrin. Vol. 10624. LNCS. Springer, Heidelberg, 2017, pp. 377– 408.

[61] Jean-Claude Bajard, Julien Eynard, Anwar Hasan, and Vincent Zucca. A Full RNS Variant of FV like Somewhat Homomorphic Encryption Schemes. Cryptology ePrint Archive, Report 2016/510, http://eprint.iacr.org/2016/510, 2016.

[62] T. Lepoint, FV-NFLlib, GitHub repository. https : / / github . com /CryptoExperts/FV-NFLlib, 2016.

[63] Carlos Aguilar Melchor, Joris Barrier, Serge Guelton, Adrien Guinet, Marc-Olivier Killijian and Tancrède Lepoint. NFLlib: NTT-Based Fast Lattice Library. In: CT-RSA 2016. Ed. by Kazue Sako. Vol. 9610. LNCS. Springer, Heidelberg, 2016, pp. 341– 356. doi: 10.1007/978-3-319-29485-8_20.

[64] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based. In: CRYPTO 2013, Part I. Ed. by Ran Canetti and Juan A. Garay. Vol. 8042. LNCS. Springer, Heidelberg, 2013, pp. 75–92. doi: 10.1007/978-3-642-40041-4_5.

[65] Jacob Alperin-Sheriff and Chris Peikert. Faster Bootstrapping with Polynomial Error. In CRYPTO2014, Part I. Ed. by Juan A. Garay and Rosario Gennaro. Vol. 8616. LNCS. Springer, Heidelberg, 2014, pp. 297–314. doi: 10.1007/978-3- 662-44371-2_17.

[66] Martin R Albrecht, Melissa Chase, Hao Chen, Jintai Ding, Shafi Goldwasser, Sergey Gorbunov, Shai Halevi, Jeffrey Hoffstein, Kim Laine, Kristin E Lauter, Homomorphic encryption standard. IACR Cryptol. ePrint Arch., 2019:939, 2019.

[67] Martin R Albrecht, Melissa Chase, Hao Chen, Jintai Ding, Shafi Goldwasser, Sergey Gorbunov, Shai Halevi, Jeffrey Hoffstein, Kim Laine, Kristin E Lauter, 2018.

[68] Z. S. Lafta and M. Ilyas, "Privacy Preserving Homomorphic Encryption In Cloud Storage," in 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 2022.