Towards Secure Encryption in Cloud Environment

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Information Security

Submitted By
Neha
(Roll No. 801633005)

Under the supervision of:

Dr. Anil Kumar Verma
Professor, CSED

Ms. Tarunpreet Bhatia
Lecturer, CSED

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY
PATIALA – 147004

July 2018

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "*Towards Secure Encryption in Cloud Environment*", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Information Security* submitted in Computer Science and Engineering Department of Thapar Institute of Engineering and Technology, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Anil Kumar Verma & Ms. Tarunpreet Bhatia* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

(Neha)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. Anil Kumar Verma)

Professor, CSED

(Ms. Tarunpreet Bhatia)

Lecturer, CSED

ACKNOWLEDGEMENTS

First of all, I would like to thank almighty God, who gave me opportunity and strength to carry out this work. The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible.

With immense privilege, I express my heartfelt sense of gratitude and respect to my supervisor **Dr. Anil Kumar Verma, Professor, CSED & Ms. Tarunpreet Bhatia, Lecturer, CSED** for their valuable guidance and constant encouragement during the course of my work. There extensive knowledge, vision and motivating approach helped me in the successful completion of the work. I thank them for their continuous criticism, useful suggestions apart from valuable guidance and moral support. It was a great experience working with them.

I am also thankful to **Dr. Maninder Singh, Professor, Head, CSED** for providing the needed support and motivational approach. I am grateful to **Dr. Shreelekha Pandey, Assistant Professor and PG Coordinator** for constantly encouraging each and every student to put their best foot forward in their field of work and helping at every stage of the course.My heartfelt thanks to the faculty members of CSED for extending their cooperation and being panel members of various presentations related to dissertation and providing the needful guidance.

It's my fortune to gratefully acknowledge the support of my friends for making the course enjoyable, interactive and full of discussion such memories will remain as a lifelong remembrance will be cherished in the years to come.

Finally, I dedicate this dissertation to my dearest family members for their blessings, understanding, patience and continuous encouragement. This work would have not been possible without their support.

(Neha)

ABSTRACT

In the advent of information technology, the dramatic rise of computational and storage limits alongside the improvement of overall correspondences guarantee exceptionally customized, very much planned and advantageous administrations. In the cloud storage condition, it ends up being particularly real in light of the way that the data is arranged in different spots. Data security and privacy assurance are the two essential standards of customer's worry. Numerous strategies in cloud storage have been researched in academics as well as enterprises. Information security and protection assurance are winding up more vital for the fate of developing cloud technology. So, this dissertation work concentrates on the security of data in information systems. Homomorphic cryptography is a possible and promising nominee for this reason. This innovation permits controlling encoded information and performing consistent operations on it without really getting to the decoded data. This study presents implementation of Advanced Encryption Standard algorithm and proposed technique giving a concise view on how the encoding/decoding of both techniques works for the file encryption and decryption. And examines how homomorphic cryptography can empower security, by first giving a profound understanding of the field of protection in computer science.

Keywords: Cryptography, Cloud, Data Security, Encryption, Decryption, Homomorphic Encryption, Advanced Encryption Standard.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AES	Advance Encryption Standard
AHEE	Algebra Homomorphic Encryption scheme based on updated ElGamal
API	Application Program Interface
API	Application Program Interface
BGV	Barkerski-Gentry-Vaikuntanathan
CSPs	Cloud Storage Providers
C_T	Ciphertext
DES	Data Encryption Standard
EHC	Enhanced Homomorphic Cryptosystem
HE	Homomorphic Encryption
IT	Information Technology
JCA	Java Cryptography Architecture
NEHE	Non-interactive Exponential Homomorphic Encryption
OS	Operating System
P_K	Public Key
RSA	Rivest–Shamir–Adleman
S_K	Secret Key

CHAPTER 1

INTRODUCTION

These days our life depends too much on technology and it is quite noticeable because of availability of ample of technologies for instance e-commerce, e-voting and so on. Hence, Homomorphic encryption (HE) has stimulated in the direction of becoming a requirement intended for up-to-date security modernizations to confirm remarkable information that is carried on the internet as of slightly unapproved discovery and amendments. Inimitable of the approaches on the way to secure data on the way to communicate the aforementioned in encoded form. In any case, the individuals who want the data are vital to accomplishing deciphering which intermittently drives problems alike secrecy defilement or else insider menace. In order to retain these problems away, it is desirable that data porters slog on customers encrypted data lacking knowledge about the original data or plain text.

1.1 Cloud Computing

In information technology (IT) archetype, cloud computing is defined as a technique that aids users with ubiquitous and convenient access on-demand IT resources and higher-level services with least management efforts. It uses well-established distribution network called the internet to communicate with the users and high-end Data Centers to accomplish their tasks [36]. It helps the organization to use third-party cloud services, in order to create a prime focus on the core business aspects rather than expending their valuable resources on maintenance and IT infrastructure.

The term “Cloud Computing" came into boom with the release of Elastic Compute Cloud (EC2) product by Amazon.Inc in 2006. But the concept dates back to 1996 when well-known computer scientist John McCarthy stated: "computation may one day be used and organized as an essential part of public service utility system" [8]. John McCarthy said:

> "*If computers of the kind I have advocated become the computers of the future, then computing may someday be organized as a public utility just as the telephone system is a public utility. The computer utility could become the basis of a new and important industry."*

The development of services across the Web & the complications in dealing with the colossal and complex data led to the emergence and success of cloud computing archetype where service providers provide the access to the collective computing assets such as computing hardware, storage units & networking to the remote users via internet technology. These computer assets are charged using "Pay-as-You-Go" utility, which means registered user has to pay for only the selected assets that are provided during the period of use, and the same constraints apply for the billing also like the use of other utilities such as electricity, water, gas, etc.

Many authors define cloud computing differently and a little consensus is always appreciated in having a universally applied definition. But, this multitude of cloud computing reflects the variety and applicability of the technological prosperity that cloud computing technology has.

However, definition proposed by "National Institute of Standards & Technology" (NIST) [24] became most referred and widely accepted definition by the public.

> *"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."*

The NIST definition concentrates on the prime features that cloud computing offers from other existing IT services.

1.1.1 Cloud Deployment Models

The deployment model initially indicates the manner in which Cloud infrastructure is established and managed. As per NIST, the four main deployment models [24] are:

1. **Public Cloud:** These clouds are made accessible to general public. The model where everyone has the provision and subscription to cloud services free and sometimes even pay for certain segments.

2. **Private Cloud:** Unlike the public cloud, the private clouds are operated and established in a single private organization. The dedicated infrastructure can be either accommodated by the primary organization itself or can authorize some third-party organizations. A private cloud offers a limited number of amenities for the set of defined clients, via the firewall technology that guarantees better control and management.

3. **Community Cloud:** This model has the ability to share services between several organizations, so as to meet their specific demands such as collaborative mission, security, political etc. these clouds are designed to meet the general and high computational needs of a community and other existing individual companies.

4. **Hybrid Cloud:** This is the advanced combination of distinct cloud archetype. An organization has the option of creating its own private cloud computing archetype, which exploits some existing public cloud services.

1.2 Data Security in Cloud

In IT security of data become a remarkable matter. Security of data turns out to be particularly genuine in the cloud computing environment since information is distributed in various types of machinery and storage devices such as servers, PCs, and numerous mobile devices, for example, wireless sensor networks and smartphones. Achieving security of data in the cloud computing is considered extra complex as compared to conventional data frameworks. In order to make cloud computing accepted by means of clients and endeavor, clients concern about security is needed to be fixed initially to make cloud environment reliable. The trustworthy environment is the fundamental requirement in the direction of getting assurance from clients towards acceptance of this innovation [38].

Various methods for data storage security and protection of privacy in cloud computing are illustrated in figure 1.1.

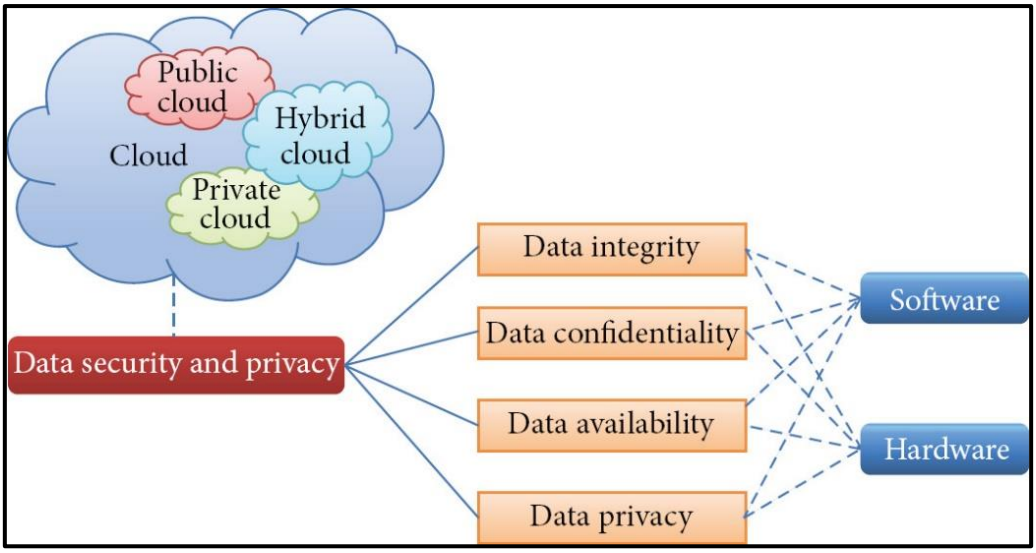


Figure 1.1: Organization of Data Security and Privacy in Cloud Computing [38]

Various characteristics of data security in cloud are confidentiality, integrity, and availability. As data privacy goes hand in hand with data security so issues related to data privacy are also studied.

1.3 Cryptography

Cryptography is defined as the process of accomplishing the security to make the information secure by encoding to make it non-understandable form so that no third party can identify it. Various goals of cryptography are as shown in figure 1.2:

- Authentication
- Confidentiality
- Integrity

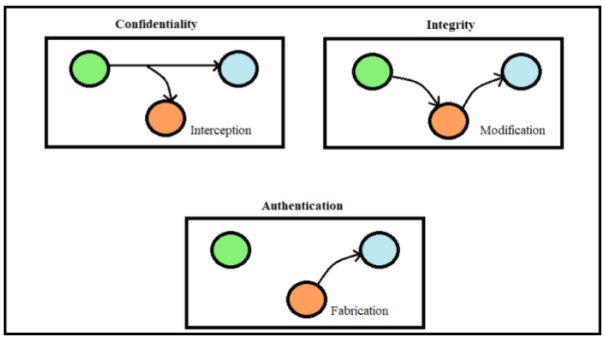


Figure 1.2: Goals of Cryptography

1.4 Security Goals

The use of information has been revolutionized by computer networks during past two decades. Computer networks are used by users to send & receive information.

Various tasks can be performed over a computer network such as shopping, banking, paying bills etc. which means it carries both personal & financial data which needs to be protected from unauthorizes people who can mishandle the data.

There are some fundamental objectives of system security that ought to be accomplished for secure communications. These are as follows:

- **Confidentiality:** To maintain the secrecy of the message that is being transmitted over a network is called as confidentiality. The message will be understood by the sender and concerned receiver only. The eavesdroppers should not be having access to read or modify the message. To maintain the security of the message, it should be sent in the encoded form.

- **Integrity:** Any message sent over the network must reach the intended receiver without any modification made to it. If any changes were made, the receiver must be able to detect that some alteration happened. Integrity can be achieved by attaching a checksum to the message. This checksum ensures that an attacker cannot alter the message and hence, that integrity is preserved.

- **Availability:** Information created and stored by an organization should be available all the time to authorized users, failing which the information ceases to be useful. Availability is also equally important for organizations because unavailability of information can adversely effect on organizations day to day operations. For example, imagine the status/service of a bank if its customers are unable to meet the transactions using their accounts.

1.5 Motivation

The basic motivational idea behind this work is a secure mechanism for sharing files which are stored on the internet. User's data is not completely protected by the cloud service providers (CSPs). A large amount of investment is done by these providers to

ensure that their files are stored in a secure way which means that encryption of such files is done as per their recommendations and standards. The cryptographic algorithms used by CSPs are symmetric and well controlled so that they have possession of all the keys and able to completely control the user's files (when required for any lawful purpose can be provided by them to the government). Although CSPs always talks in terms of user's data security as only a few of the top people and highly efficient trusted employees of the organisation have access to cryptographic keys but recent cases of leaking out the private data for some wrong purpose; monetary and political benefit has proved that their secure process is not efficient enough to make the user's trust them.

So, the main motive of carrying out this research work is to develop a framework for securing the privacy of stored data in the cloud. The traditional encryption techniques in combination with homomorphic encryption are used in this dissertation work. The proposed scheme protects the user's privacy while satisfying basic security requirement in cloud data storage.

This area was chosen because of diverse allure in the security field. Security of data is taken as the base focus. The greater part of the scientists selects one base issue and condense their research into different sub-issues. It is possible to achieve integrity of information only if traditional encryption algorithms are combined with the newly developed homomorphic encryption.

1.6 Organization of Dissertation

Rest of the dissertation is systematized as follows. Chapter 2 incorporates ongoing improvements and foundation work around there. Chapter 3 provides details on the cryptographic techniques. Chapter 4 gives a short portrayal of the problem statement. Definite notes on proposed work alongside all parts are depicted in Chapter 5. Chapter 6 gives notes on the execution of segments presented in Chapter 5 and it additionally portrays execution of the proposed algorithm. Chapter 7 gives final observations and notes for probable improvements.

CHAPTER 2

LITERATURE SURVEY

Work done by different researchers in the area of cryptography is included in this section. From the writing overview different perceptions have been drawn and recorded toward the finish of this part. From the explanations, different objectives have additionally been drawn in order to achieve data security.

Diffie & Hellman [1] exhibited a portrayal about a hike to the necessity for new sorts of cryptographic frameworks due to growing utilization of teleprocessing, they additionally discuss starting of providing different tools to tackle the cryptographic issues of long standing by theories of communication and computation.

Rivest et al. [2] exhibited a portrayal of using a factorial of a large number in the direction of implementing a public key cryptosystem. Concluded that the security of a system is must to be more enhanced. It uses "trap-door one-way permutation". There are so many examples to discover and implement in the public key cryptosystems.

Goldwasser & Micali [3] they proposed the first usage of another probabilistic model of information encryption, additionally, implementations security is demonstrated under the intractability supposition for selecting Quadratic Residuosity modulo composite numbers whose factorization is ambiguous. Concluded that confirmations hold for any message space by means of probability distribution and withdrawing information regarding plaintext from ciphertext is challenging over a period of time for a foe.

El Gamal [4] portrayed about a public key cryptosystem and a signature scheme depending upon the struggle of calculating discrete logarithms over finite fields along with the execution of Diffie-Hellman key distribution scheme. Concluded that for a similar security level, assessments for the running time of processing discrete logarithms and calculating whole numbers are the best-identified scheme till then, resulting into larger sizes of public file and ciphertext than Rivest-Shamir-Adleman (RSA) scheme.

Naccache & Stern [6] in this paper based upon the difficulty of computing higher residue modulo they designated public key cryptosystem, also proposed two new schemes: deterministic and probabilistic. They concluded that deterministic version is practically oriented encryption and probabilistic version exhibits a homomorphic encryption scheme having expansion rate ample improved than formerly anticipated such schemes, and performances of both deterministic and probabilistic schemes is measured during encryption and decryption.

Paillier [7] introduced public key cryptosystem build on Composite Degree Residuosity classes. Three encryption techniques comparable to RSA are introduced in this paper based on the trapdoor mechanism. No proof of security was provided against chosen ciphertext attacks, it is believed that schemes 1 & 3 can become resistant against these attacks just by performing little modifications. His assumptions are more secured in standard model under appropriate conditions.

During several years number of homomorphic schemes have been introduced to the world either additively homomorphic (Goldwasser-Micali [3] 1984, Naccache-Stern [6] 1998, Paillier [7] 1999) or multiplicatively homomorphic (RSA [2] 1978, El Gamal [4] 1984). Fully homomorphic encryption's need emblem over in 1991 once Feigenbaum et al. [5] queried: "If there exists an encryption function Enc () such that Enc(x+y) and Enc(x*y) are easy to compute from Enc(x) and Enc(y)?" and in 2009 it's responded Craig gentry [19] executed first fully homomorphic scheme starting with somewhat homomorphic encryption and it became an open issue in the field of cryptography. A long search for the existence of privacy homomorphisms emerged in 1978 ended, even though it's not that useful for practical applications.

Ferguson et al. [11] displayed that Advanced Encryption Standard (AES) has an incredibly essential and direct fastened mathematical equation. Further, this procedure is exceptionally straightforward in contrast with other block ciphers and its security only depends upon the infeasibility of finding the procedure of this particular kind. Concluded that any new kind of arithmetical attacks can grow in the near future which can trade off the security of AES and it isn't protected to use AES in security general frameworks.

Elbirt et al. [13] in this paper assessment of various AES finalists algorithm with field-programmable gate arrays (FPGAs) based kit were performed on both software as well as hardware. And concluded that for hardware implementations of encryption algorithms, reprogrammable devices, for example, FPGAs are very appealing alternatives in order to accomplish higher performances and security. They exhibited a short portrayal regarding AES algorithms FPGAs implementations on the basis of different archetypes. In this paper, they have also compared various algorithms resulting in getting the most suitable algorithm for FPGAs hardware implementation.

Su et al. [14] they exhibited short portrayal regarding effective hardware implementation of AES algorithm by means of key expansion ability. In this paper, they concluded that high-throughput and low cost consuming AES processors were designed. By using transformation techniques, the hardware overhead of the S-Box was reduced to 64%.

Mangard et al. [15] they depicted a brief portrayal about an extremely consistent and ascendable AES hardware architecture which was appropriate for full-custom in addition for semi-custom design flows. They concluded that design was versatile as far as throughput and the used key size are concerned. Resemblances of encoding & decoding were used to give an abnormal state of execution by utilizing just a generally little region. This execution was come to by adjusting the combinational ways of the plan.

Al-qdah & Hui [18] in this paper they give us the Encryption and Decryption applications which will support all types of file format. Their method was simple but effective, they have used simple key generation method of random number generation. And lastly, XOR operation was done to perform rotations of bits. For decryption of the message, same encryption key was used. Concluded that simple encryption tool was effective enough for all types of files.

Gentry [19] executed first fully homomorphic encryption plot by beginning from somewhat homomorphic encryption usage which came about into taking care of an open issue in the field of cryptography, this idea, initially called a privacy homomorphism, was presented by Rivest, Adleman and Dertouzos not long after the creation of RSA by Rivest, Shamir, and Adleman.

Zhou & Tang [22] they have implemented one of the cryptographic technique i.e. RSA algorithm. This paper gives the complete idea of how the RSA encryption and decryption works using the information regarding RSA public key algorithm. Also, this paper provides encryption procedure and the code for encryption in details.

Gadanayak & Pradhan [25] here they have proposed an idea of encryption for various audio files. They have used AES technique to encrypt audio files before Huffman's entropy coding. Proposed a new very complex encryption technique for the audio files which is much difficult to break. Here AES technique enhances the cryptographic security of audio files.

Pavithra & Ramadevi [27] in this paper comparison of various encryption algorithms for audio & video files on the basis of processing time and throughput have been performed. They have compared encoding & decoding time for various formats of audio and video files having different sizes and concluded that overall time is directly proportional to the size of the file. AES is concluded as a better solution by them.

Sharma & Pateriya [28] discussed various approaches for selection of encryption method to extend the speed of encryption. Selecting the encryption technique is one of the important features of encryption. They have selected various encryption techniques and compared them on different attributes. Selective Encryption technique is faster than all the full encryption technique.

Sen [29] exhibited a short portrayal about the issue of computing over scrambled information, i.e. homomorphic encryption, its applications, properties and different homomorphic encryption methods, for example, additive homomorphic encryption to be specific the Paillier and Goldwasser-Micalli cryptosystems, and the multiplicative homomorphic encryption RSA and El Gamal cryptosystems, and so on., with suitable cases and evidences.

Mahajan & Sachdeva [30] they have analyzed the effectiveness of the following encryption algorithm that is AES, Data Encryption Standard (DES), and RSA. They have simulated these algorithms on the basis of two parameters that are Encryption Time and Decryption Time and concluded the result that AES Decryption performs better than all three algorithms.

Ryan [31] discuss the security issues and challenges in cloud computing and possible solutions in it. Here he explored four methodologies said vary significantly in their pertinence and the idea of the security ensures they offer. And concluded that CryptDB supports various applications but still it is limited to database searches, its security is a bit weaker than the key translation approach. The main reason behind this is that CryptDB uses weaker cryptography. Also, fully homomorphic encryption also compromises security. Excalibur is unmistakably the most flexible of the four methodologies, yet its security ensures are of a significant distinctive nature (and naturally somewhat weaker) than the other three.

Asok et al. [32] depicted a short portrayal about the idea of Bio Cryptography in order to remove the drawback of authentication of authorized user's. Iris image was utilized for generating a 128-bit secret key. They made use of the secret key for encoding and decoding of audio signals. MATLAB was used for getting a binary form of encoded & decoded audio signals.

Tebaa & Hajii [33] displayed a short depiction about the issue of secure computing over cloud information by giving a concise presentation about cloud computing, virtualization systems, qualities of cloud computing administrations, issues looked by cloud computing and applying the different sorts of homomorphic encryption strategies to accomplish the objective of secure cloud computing.

Parmar et al. [34] in this paper case study on various ideologies and possessions of homomorphic encryption are given and then several homomorphic algorithms (RSA, ElGamal, Paillier, Barkerski-Gentry-Vaikuntanathan (BGV), Enhanced Homomorphic Cryptosystem (EHC), Algebra Homomorphic Encryption scheme based on updated ElGamal (AHEE), Non-interactive Exponential Homomorphic Encryption (NEHE)) are explored and a short description is portrayed providing a contrast of above-mentioned algorithms. And worth of homomorphic encryption in preserving privacy and processing information placed on the remote server was concluded, as operations on the ciphertext were allowed in it, providing the same results after computations as the working directly on the raw data.

Kangavalli & S. [37] proposed a technique by means of byte level homomorphism after similar study of outcomes attained on or after various homomorphic encryption

calculations, for example, Paillier cryptosystem, RSA algorithm, EHC, NEHE, Algebra Homomorphic Encryption scheme in light of refreshed Elgamal with BGV conspire in view of the properties of homomorphic encryption, it's applications, protection of information, their worth, and illustration about cloud data storage.

Bensitel & Romadi [38] they have discussed cloud computing and their security-related problem. In this paper, they evaluated the partial homomorphic encryption and fully homomorphic encryption. Explained various applications of homomorphic encryption on cloud computing.

Sridokmai & Prakancharoen [39] they discussed the Paillier's encryption and other homomorphic encryption cryptosystems. Also, they concluded that Paillier encryption was better for high-end security than all the other encryption cryptosystems, and the fastest probabilistic homomorphic system is better for decryption than RSA.

Biksham & Vasumathi [41] they proposed the idea for encrypted data in cloud computing security. Using cryptographic encryption security is provided to a user by CSPs. Query method is used for decryption but using the decryption method again and again on the ciphertext it will exploit the integrity and authenticity of the message. Here the new technique is used i.e. homomorphic encryption which provides encryption without decrypting the ciphertext again and again and provides the more boosted results.

Mathur & Bansode [42] they joined the AES and Elliptic Curve Cryptography (ECC) to support the private key cryptosystems through public key cryptosystems. In this the key length of AES was expanded to 192 bits and number of iterations were increased to 12. Hybrid encryption was used to improve the competency and lower drawbacks. ECC was taken as a factor for encoding and AES key was used for transferring in order to encrypt the communication data. They concluded that series of table lookup should be used to escalate the AES performance.

Gong et al. [44] they have projected an equal value retrieval algorithm provided by homomorphic encryption. This algorithm is implemented on character data in the database and it gives the better retrieval efficiency. They have also used fuzzy retrieve technique in index generation algorithm for character data which will help to boost the

working of server's filtering of unwanted data and decrease the wastage of time and space.

Harerimana et al. [45] in this paper talk about usage of Paillier homomorphic encryption scheme in Java as an Application Program Interface (API) is exhibited which brings about the formation of HE Java library and this library is additionally utilized as a part of an electronic voting system. Apache JMeter is utilized as a part of a request to assess the execution of electronic voting system and demonstrated that Paillier library functions admirably.

CHAPTER 3

CRYPTOGRAPHIC TECHNIQUES

3.1 Basic Process

Encryption is one of the utmost crucial stratagems aimed at providing security to data, principally meant for an end-to-end guarantee of data communicated over systems. Generally employed on the internet in order to guarantee that user information directed amongst a program and a server, together with PINs, payment facts above and beyond the further discrete information that must be regarded as private [48]. Organizations and folks as well commonly employ encryption towards safeguarding sensitive data laid on Computers, servers, and mobile phones.

In computer science, encryption is defined as the procedure by means of which plaintext or else other kinds of data is transformed from a human-readable form into an unreadable form such that for getting information back into a readable format it must be decoded by using certain methods.

In cryptography, encryption is defined as the method of encrypting a note or info to such an extent that it becomes unreadable by peoples other than the authorized personnel's only. It doesn't itself forestall impedance, then rather refutes the vibrant material towards a subsequent interceptor [16].

In cryptography, decryption is defined as the reverse of the encryption process. In this process, the ciphertext is converted back to the plaintext so that peoples or computer can read and understand. The encryption and decryption algorithms are together known as ciphers. Ciphers need not necessarily be unique for each communicating pair, rather a single cipher can be used for communication between multiple pairs of senders & receivers.

Figure 3.1 is a simple illustration showing the process of encryption, the envisioned data or message, denoted as plain text, then is encoded with an encryption algorithm – a secret message – engendering ciphertext, is decoded in order to read.

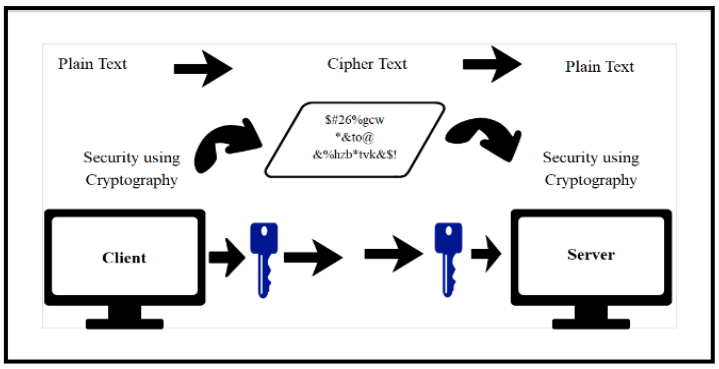


Figure 3.1: Traditional Encryption Process

Figure 3.2 is an illustration of the basic problem with traditional encryption technique. Following are the main issues describing the same-

- What if Party A does not trust Party B, with its confidential data.
- Data needs to be decrypted whenever we have to perform any computation on the data.

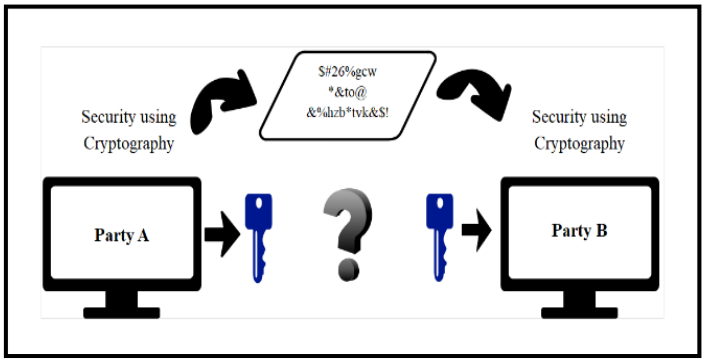


Figure 3.2: Problem with Traditional Encryption

A new technique has been taken into use to remove the problems with traditional encryption techniques, which is **homomorphic encryption**.

3.2 Types of Encryption Algorithms

There are two types of encryption algorithms: symmetric and asymmetric algorithm as described below:

3.2.1 Symmetric Encryption Algorithm: The aforementioned utilizes an equivalent key for encoding and decoding. Thus, it's symmetric. It is used for a long time and

bothers people how to share a key before they want to communicate with each other. For example – DES, AES.

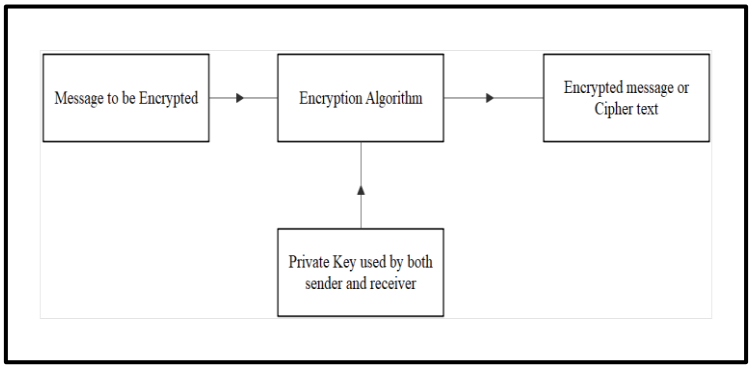


Figure 3.3: Symmetric Encryption Algorithm

3.2.2 Asymmetric Encryption Algorithm: It uses dissimilar keys for encoding and decoding. And it's a remarkable progress in cryptography. Ron Rivest, Adi Shamir and Leonard Adleman in 1978, invented first asymmetric encryption algorithm. Thus, named after their initials from each inventor, and becomes RSA.

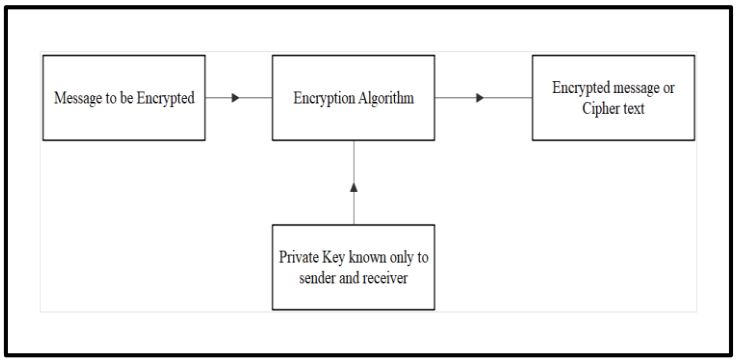


Figure 3.4: Asymmetric Encryption Algorithm

3.3 Homomorphic Encryption

Rivest, Adlamen, and Dertouzos proposed Homomorphic Encryption in 1978. Homomorphic encryption is defined as a process in which all the operations are carried out upon the ciphertext without knowledge of the original information in order to keep plaintext secret.

Homomorphic encryptions enable complicated scientific measures while in transit to be performed on encoded data without eliminating encoding [29]. According to

mathematics, homomorphic demonstrates the alteration of an individual informational index into another while keeping connections amid items in the two groups.

3.3.1 Need for Homomorphic Encryption

The purpose of homomorphic cryptosystems is to guarantee the secrecy of information in transmission and storage procedures, for example, the capability to employ calculations towards malicious revelries. In the event that a client may perhaps receipts a fact categorized in single mathematical context and encrypt the aforementioned into a fact in an alternative mathematical context in a manner that translating posterior in the direction of initial mathematical context is tough, at that point the client possibly will encrypt expensive calculations and refer back on the way to the malicious revelries. These malicious revelries at that point play out the comparing calculation in the second mathematical context, restoring the outcome to the client. After getting the outcome, the client can translate it into an answer in the initial mathematical context, despite the fact the malicious revelry adapts nonentity of which calculation was really executed.

3.3.2 Working of Homomorphic Encryption

Homomorphic encryption is trusted to have a critical influence in distributed computing, enabling administrations to stockpile encrypted data in an open cloud and exploit the cloud supplier's diagnostic administrations.

Different components of homomorphic encryption are represented in Figure 3.5. These components work in an accompanying way;

1. enc (m): - using secret key S_K the plain text will be encoded by the user then corresponding ciphertext is generated and sent to the server along with public key P_K.

2. eval (enc (m)): - in order to compute the ciphertext $C_{T,}$ the server applies a function and as per the required function this is performed using public key P_K.

3. dec (eval (enc (m))): - the user achieves the final result by decoding the evaluated plain text using his secret key.

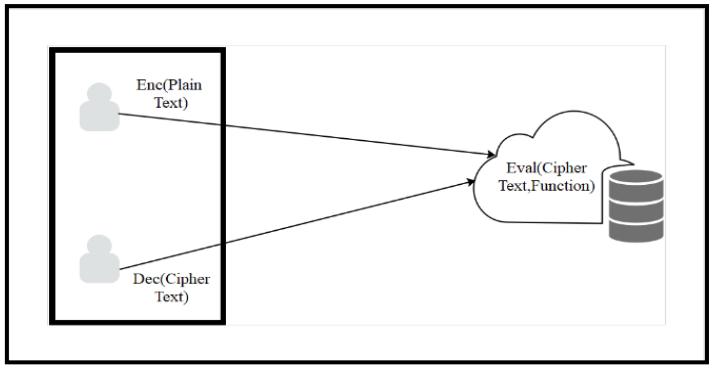


Figure 3.5: Homomorphic Encryption Components

Figure 3.6 is representing an extremely basic example showing the working of homomorphic encryption:

- Organization PQR has necessary data that comprises the numbers 2 and 20. To encrypt this necessary data, Organization PQR increases every component by 2, making another set comprising of individuals 4 and 40.

- Organization PQR directs the encoded necessary data to the cloud for safe storage. A couple of days after the fact, the administration communicates to Organization PQR and requests aggregate of necessary data components.

- Organization PQR is exceptionally occupied, hence requests cloud provider to carry out the task. The cloud provider, who just approaches the encrypted necessary data, finds the aggregate of 4 + 40 and yields the appropriate response 44.

- Organization PQR decodes the cloud provider's answer and furnishes the administration with the decoded reply, 22.

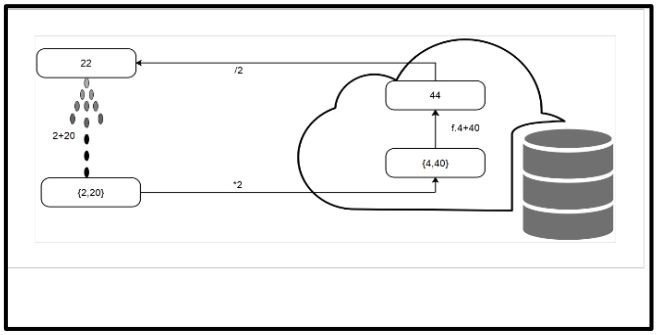


Figure 3.6: Working of Homomorphic Encryption

3.3.3 Existing Homomorphic Encryption Cryptosystems

On the basis of several parameters, comparison of different Homomorphic Encryption cryptosystems is given in the table below:

Table 3.1: Features of Prevailing Homomorphic Encryption

Various Homomorphic Encryption Cryptosystems					
Parameters / **Type**	**Tenant**	**Type of Homomorphic Encryption**	**Secrecy of Information**	**Security practiced to**	**Keys Used by**
RSA [2]	Cloud Computing	Multiplicative Homomorphic Encryption	Is assured in communication and excavation procedures	Cloud Provider Server	The user (For encoding and decoding distinct keys are used)
Paillier [7]	Cloud Computing	Additive Homomorphic Encryption	Is assured in communication and excavation procedures	Cloud Provider Server	The user (For encoding and decoding distinct keys are used)

El Gamal [4]	Cloud Computing	Multiplicative Homomorphic Encryption	Is assured in communication and excavation procedures	Cloud Provider Server	The user (For encoding and decoding distinct keys are used)
Goldwasser-Micali [3]	Cloud Computing	Additively Homomorphic Encryption, only encrypt a single bit	Is assured in communication and excavation procedures	Cloud Provider Server	The user (For encoding and decoding distinct keys are used)
Boneh-Goh-Nissim [29,34]	Cloud Computing	Unlimited number of additions but only one multiplication	Is assured in communication and excavation procedures	Cloud Provider Server	The user (For encoding and decoding distinct keys are used)

Gentry [19]	Cloud Computing	Fully	Is assured in communication and excavation procedures	Cloud Provider Server	The user (For encoding and decoding distinct keys are used)

3.4 Advanced Encryption Standard

This standard was basically introduced to oust Data Encryption Standard, National Institute of Standards and Technology (NIST) of the United States of America (USA) [9] [10] [13] started Advanced Encryption Standard (AES) selection strategy in January 2, 1997 when world's best people working in this field came forward to come up with their thoughts regarding this new encryption algorithm called as Advanced Encryption Standard. NIST anticipated every entry to be open for appropriate remarks. There were some prerequisites for the applicants of the AES which are listed below:

- Each block of AES should encrypt 128 bits of plain text.
- AES ought to have the capacity to encode the plaintext utilizing any of the three key lengths (i.e. 128-bit, 192-bit or 256-bit).
- The efficiency in AES should be equally good for software as well as in hardware.

In 2000, USA government choose the Rijndael algorithm as AES, and was proposed by Joan Daemen and Vincent Rijmen [13]. The ability to encode 192 or 256 bits of plain text is present in AES algorithm in the same way as original Rijndael algorithm.

3.4.1 Basic AES

AES is a non-feistel cipher that operates on a data block of 128-bits and comprises several rounds for encoding and decoding [25]. It is available in three versions, depending on the key size and number of rounds (N_r) used.

- AES-128 with key size 128-bits & 10 rounds
- AES-192 with key size 192-bits & 12 rounds
- AES-256 with key size 256-bits & 14 rounds

16 bytes of plaintext is encrypted by AES into 16 bytes of ciphertext and as per the requirement of the user AES can use variable key sizes of 16 bytes, 24 bytes or 32 bytes respectively. Figure 3.7 describes general working of AES where plain text of 128-bits is given as input yields an output of 128-bits called ciphertext.

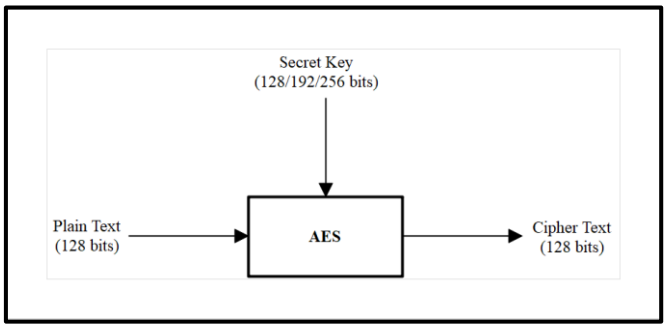


Figure 3.7: Working of AES

The round keys used in each version are always 128-bit long, which is the same size as that of plaintext or ciphertext block. In AES, the round keys are generated using the key-expansion algorithm and the number of round keys generated is always equal to the number of rounds plus one.

AES is a byte-oriented encryption algorithm comprises of 10/12/14 rounds and there are 4 different types of transformations for each round of AES. Each transformation in AES is invertible in nature and during decryption, the inverse of these transformations is used.

3.5 RSA Algorithm

RSA is the primary public-key encryption system by means of homomorphic possessions. The RSA cryptosystem termed afterward its designers R. Rivest, A. Shamir, and L. Adleman and it was portrayed in a paper in 1978. Algorithm's title contains primary letters of surnames of all the three inventors. RSA was initially licensed by M.I.T. be that as it may, was out to the public realm in September 2000. It comprises of three steps (1) Creating key (2) Encryption (3) Decryption.

RSA algorithm possibly will be used to offer together privacy and digital signatures and algorithms security depend upon the infeasibility of the integer factorization problem.

3.5.1 Algorithm

Key generation

RSA algorithm generates keys as follows:

1. Assume s & t be two large prime numbers.
2. Calculate m = st.
3. Calculate ϕ (m) = ϕ (s). ϕ (t) = (s-1). (t-1)
4. Select an integer f such that 1 < f < ϕ (m), i.e. gcd (f, ϕ (m)) = 1 and f is coprime with ϕ (m).
5. Determine d as f^{-1}(mod ϕ (m)); such that, d is a modular multiplicative inverse of f(mod ϕ (m)).
    - The public key exponent is liberated as f.
    - The private key exponent is reserved as d.

Encryption and Decryption Process

If a message is encrypted by D for E, which is then decrypted by E

For Encrypting following steps are followed by D:

a) Acquire public key (m; f) of E.

b) Let q be the message.

c) Determine ciphertext as $C_T = q^f$ mod (m)

d) Ciphertext C_T is sent from D to E.

For Decrypting i.e., obtaining plaintext q from ciphertext C_T, the following step is followed by E:

Using private key d plain text is generated as $q = C_T^d$ mod (m)

3.5.2 Security

The computational issue of calculating extensive whole numbers prompts the security of RSA. The capability to factor bigger and bigger numbers increments with the additions in figuring force and disclosure of more capable considering calculations. The quality of encoding is will undoubtedly size of the key, and an exponential augmentation in quality is conveyed by multiplying the length of the key, in spite of the fact that this damages execution. For the most part, keys of RSA are 1024 or 2048-bits in length, yet it is accepted by the specialists that sooner rather than later 1024-piece keys could be broken, that is the reason govt. furthermore, associations are moving to at least 2048-bits key length.

3.6 Paillier Cryptosystem

A probabilistic public-key algorithm, in 1999, Paillier encryption scheme [7] is, labelled after and developed by Pascal Paillier. It is assumed that the issue of calculating n^{th} residue classes is computationally hard. Paillier cryptosystem stand founded upon decisional composite residuosity assumption, that's an intractability supposition.

It is generally additive homomorphic scheme which implies sum of plaintexts m_1 and m_2 as $(m_1 + m_2)$ can be computed when encryption of m_1 and m_2 along with secret key is provided. It is a probabilistic asymmetric encryption scheme having homomorphic properties for both addition and multiplication. This scheme is efficient both in encoding and decoding as it requires numerous bits at a time for one operation with a steady expansion factor. The generalized Paillier scheme when compared to the original scheme, reduces the expansion factor from 2 to nearly 1. Paillier cryptosystem can be utilized to fathom certain discrete logarithms and furthermore gives semantic security in contrast to chosen-plaintext attacks. The Paillier scheme is a partially homomorphic encryption scheme which initially could perform only additions on ciphertext but later improvements in the scheme propose that multiplication can also be performed on the ciphertexts [44].

3.6.1 Algorithm

The Paillier encryption scheme is composed of key generation, encryption, and decryption algorithms as follows:

Key Generation

1. Select two independent and random large prime numbers p and q, such that

   gcd (pq, (p-1) (q-1)) = 1

   This property is assured if both primes are of equal length.
2. Calculate n = pq, λ = lcm (p-1, q-1)
3. Select random integer g where g ε $Z^*_{n^2}$
4. Ensure n divides the order of g by inspecting the presence of the subsequent modular multiplicative inverse:

   $\mu = (L (g^{\lambda} \pmod{n^2}))^{-1} \pmod{n}$; where function L is defined as $L = (u - 1) / n$
   Note: that the notation a=b does not denote the modular multiplication of times the modular multiplicative inverse of b, but rather the quotient of a divided by b.
5. Finally, the public (encryption) key is (n; g) and the private (decryption) key is (λ, μ).
   If using p; q of equivalent length, a simpler variant of the above key generation steps would be to set

   $g = n + 1$, $\lambda = \varphi(n)$, $\mu = \varphi(n)^{-1} \pmod{n}$; where $\varphi(n) = (p-1)(q-1)$.

Encryption

Let m be a message to be encrypted where m ε Z_n.

Select random r where r ε Z^*_n

Compute ciphertext as $c = g^m * r^n \pmod{n^2}$

Decryption

Let c be the ciphertext to decrypt, where c ε $Z^*_{n^2}$

Compute the plaintext message as $m = L (c^{\lambda} \pmod{n^2}) \mu \pmod{n}$

The various functions available in the library to perform Paillier homomorphic encryption operations are:

- Paillier(int bitLengthVal, int certainity): This function takes bitLengthVal and certainity as integer parameters where bitLengthVal is the number of bits of modulus and certainity is the probability that the prime number represented by new bigInteger will exceed (1-2^(-certainity)).

- Paillier(): A public method that is used for constructing an instance with 512 bits of modulus and at least 1-2^(-64) certainty of primes generation.

- KeyGeneration(int bitLengthVal, int certainity): This is a public method used for setting the public and private key. It takes bitlengthVal and certainity as integer parameters.

- Encryption(BigInteger m, BigInteger r): a public method that takes BigInteger m and r as plaintext and random plaintext respectively. The encryption of plaintext m returns a ciphertext $c= g^m*r^n \bmod n^2$ where random input r is automatically generated to help with encryption.

- Decryption(BigInteger c): This public method takes ciphertext c as a parameter to decrypt c so as to return plaintext m by using the formula $m = L(c^\lambda \bmod n^2)$ .u mod n.

- main(): Here the operations on encrypted texts em_1, em_2 of plaintexts m_1, m_2 respectively are done. The input is the plaintext as integers. To test the homomorphic properties, addition and multiplication are performed on em_1 and em_2. And if decryption of the sum and multiplication of encrypted texts should return the same sum and multiplication of plaintexts. Decryption of sum can be obtained by $D(E(m_1)*E(m_2) \bmod n^2) = (m_1 + m_2) \bmod n$ and decryption of product can be obtained by $D(E(m_1)\text{^}m_2 \bmod n^2) = (m_1*m_2) \bmod n$.

CHAPTER 4

PROBLEM STATEMENT

4.1 Research Gaps

Although many studies have been carried out and different kinds of literature have been published that describe the importance and working of various HE schemes, there is always a constant battle going on between code breakers and code makers which lead to the development of strong encrypting as well as decrypting algorithms. In the concept of cryptography, a fundamental goal is to categorize relative complexity of different cryptographic tasks [9]. According to the results of the latest Google Trends search for HE, the interest on HE over a period of 9 years i.e. since Gentry proposed the first fully working homomorphic encryption is ~ 45%. This means that research in this field is still being continued and many advancements in this subject area can be anticipated in the coming future.

It is also perceived that in the recent years, when compared to other topics in the field of cryptography, there is only a little enhancement done in HE and there exist only a few traces of flexibility for applying HE in real time. This gap that is identified out indicates that there is a need to understand the various perceptions of HE and expand the current state of the art that can turn limitations of HE to effective cryptographic developments.

4.2 Problem Statement

The development of the Internet and e-commerce has conveyed towards the matter of secrecy in electronic communication. Huge sizes of personal and sensitive data are electronically transmitted and put away consistently. What assurances do one have that a message sent to someone else isn't intercepted and read without their knowledge or consent?

Unlike now, cryptography in ancient times was more linked with encryption which was mainly for message confidentiality. Conventional methods of encryption would protect

their data while it is in transit, but not while the computation is in progress. This technique involves all logical and mathematical operations required in the computation, which may be denoted by circuits or gates, be applied to the encoded form of the information [1]. Homomorphic encryption is one such type of encryption where random calculations are performed on the encoded data without disclosing the original plaintext or information.

4.3 Research Objectives

The fundamental objectives of the research are:

1. To gain an insight into the importance of encryption techniques both public & private key in the current state of cryptographic studies.

2. To identify the advantages and limitations of HE & importance of integrating with traditional encryption schemes for enhanced security.

3. To implement AES algorithm and proposed technique for comparing & analyzing their working.

4. This proposed work is reasonable for text & audio samples. Confidentiality of information is maintained using proposed technique.

CHAPTER 5

PROPOSED WORK

In the proposed work, we have applied evaluation function using encryption on the various kinds of data with the help of Java and Java Cryptography Architecture (JCA). Figure 5.1 demonstrates stages followed in the proposed technique. Encryption process to transform the files into unreadable form is illustrated in section 5.2. The process of proposed technique is shown in section 5.3, and section 5.4 depicts the process of decryption with a specific end goal to get the original files.

5.1 Overall System Architecture

The proposed system works in an optimized way without giving any problem. Figure 5.1 represents both receiver and sender side of the algorithm used. The secret information to be transmitted is encrypted using the proposed technique.

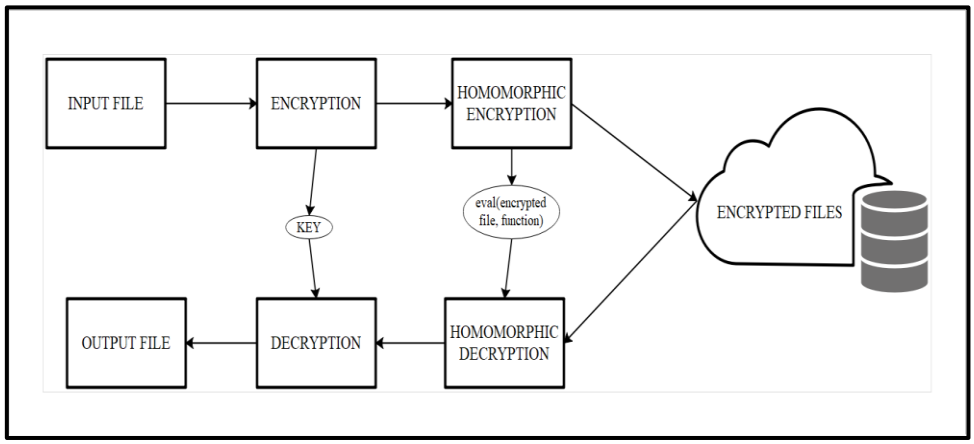


Figure 5.1: Proposed System Design

The encrypted information is then further encoded by using HE method and stored in the cloud. At the receiver side, the encrypted file containing the secret information is been received. From, this secret information is retrieved using the decryption algorithm. For decryption same cipher or key that was used while encrypting the text is used.

Java & Java Cryptography Architecture

Reasons for selecting java as the language of development are as follows:

- Java has certain properties which are required for the development of large projects. It also enables the feature of easy and quick development.
- Java is not bound to any operating system (OS) or platform. Any program that is written in Java in one operating system can be easily used in other OS too without going through major changes.
- Java has many inbuilt cryptographic libraries which can be used in any project. A framework is used in Java programming language for working with cryptography is Java Cryptography Architecture (JCA). It forms part of the Java security API. These APIs are designed to provide an easy way for developers to integrate security into application code.

5.2 Encryption Process

Encryption is defined as the procedure by means of which plaintext or else another kind of data is transformed from a human-readable form into an unreadable form such that for getting information back into a readable format it must be decoded by using certain methods.

Figure 5.2 is a simple illustration showing the process of encryption in which the envisioned data or message denoted as plain text, is encoded with an encryption algorithm into a secret message – engendering ciphertext.

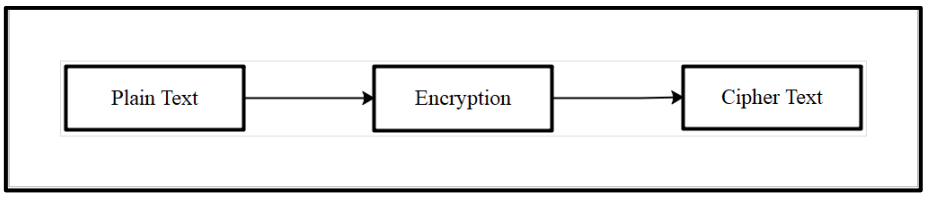


Figure 5.2: Encryption Process

Here, we confine to the explanation of a typical round of AES encryption [10]. Each round comprises four sub-processes: Substitution, Permutation, Mixing and Key Adding. The round process is represented in figure 5.3.

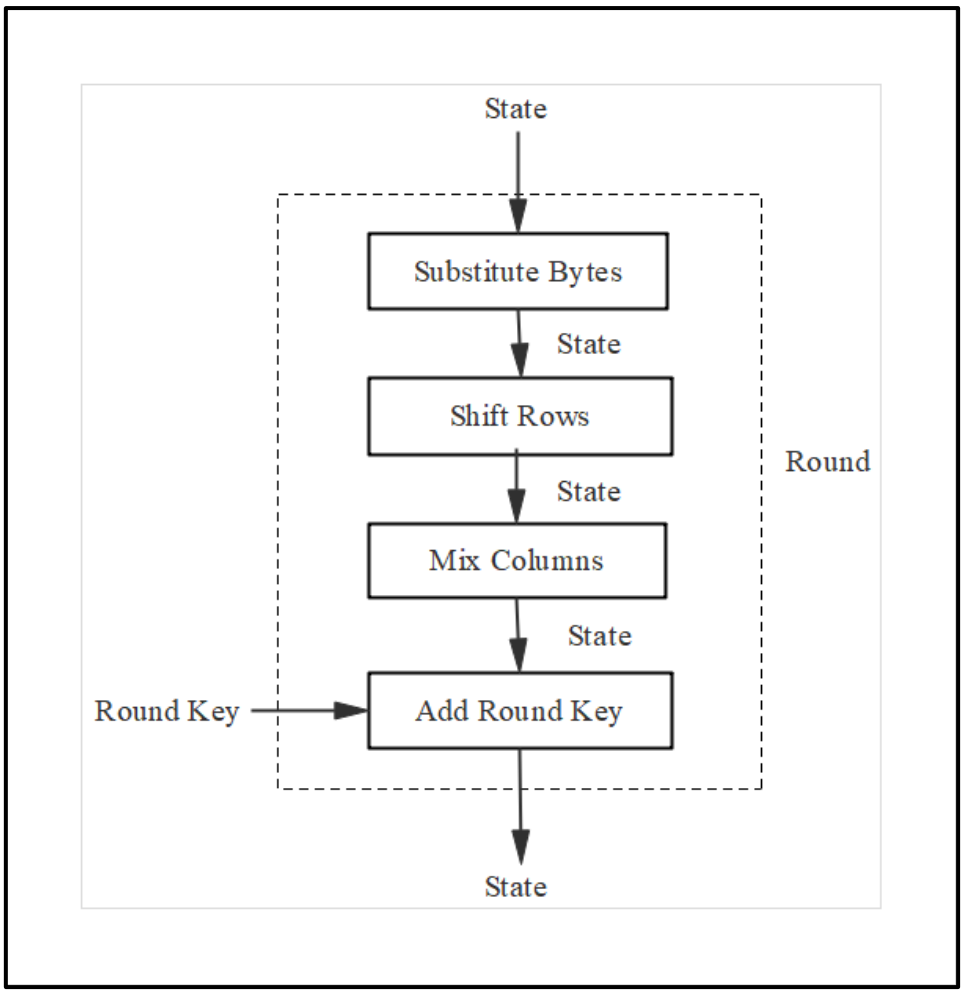


Figure 5.3: Structure of Encryption Round of AES

Substitution Round

In AES, substitution is performed for all bytes and that too, using only one table. This implies that if 2 bytes are same then their transformations are also same.

- **Substitute bytes:** Input to this transformation is a state organized as a 4*4 matrix of bytes. The bytes in the matrix are substituted one at a time, thus there are 16 distinct byte-to –byte transformations. Each byte is treated as two hexadecimal digits, where the first digit specifies the row and the second digit specifies a column of substitution table. The value at the intersection of the row and column in the transformation table is the new byte with which given byte is to be replaced.

Permutation Round

It performs a byte-level permutation, such that the order of bits in each byte does not change in the resultant bytes.

- **Shift Rows:** Bytes in rows of input state matrix are shifted to left and number of bytes to be shifted depends on the row number.

Mixing Round

Bits inside bytes are changed on the basis of bits in neighbouring bytes.

- **Mix Columns:** A constant square matrix is used. A square matrix is multiplied by each column of state matrix resulting into a column.

Key Adding Round

It makes use of round key.128-bit round key is considered as 4 32-bit words and each 32-bit word is treated as a column matrix.

- **Add round key:** Self-invertible. Each column of state matrix is revoked with corresponding column matrix to produce a new column.

5.3 Process of Proposed Technique

In proposed technique all the operations are carried out upon the ciphertext without knowledge of the original information in order to keep plaintext secret.

Figure 5.4 illustrates about the application of evaluation function over the encrypted data obtained from the process of encryption.

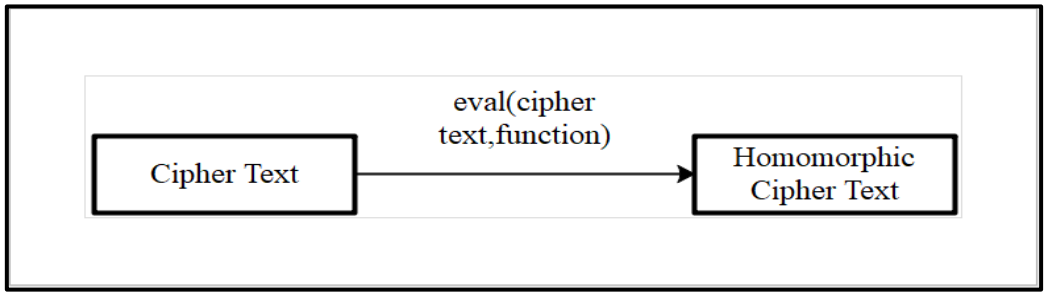


Figure 5.4: Evaluation Process

Different components of proposed technique are as follows:

1. enc (m): In this phase, the plain text will be encoded by the user using secret key S_K then corresponding ciphertext is generated and sent to the server along with public key P_K.

2. eval (enc (m)): In order to compute the ciphertext $C_{T,\text{ the}}$ server applies a function and as per the required function this is performed using public key P_K.

3. dec (eval (enc (m))): In this phase the user achieves the final result by decoding the evaluated plain text using his secret key.

5.4 Decryption Process

Decryption is defined as the reverse of the encryption process. In this process, the ciphertext is converted back to the plaintext so that people or computer can read and understand.

Figure 5.5 is a simple illustration showing the process of decryption in which encoded data or secret message denoted by ciphertext, is decoded with a decryption algorithm into an envisioned data or message, denoted as plain text.

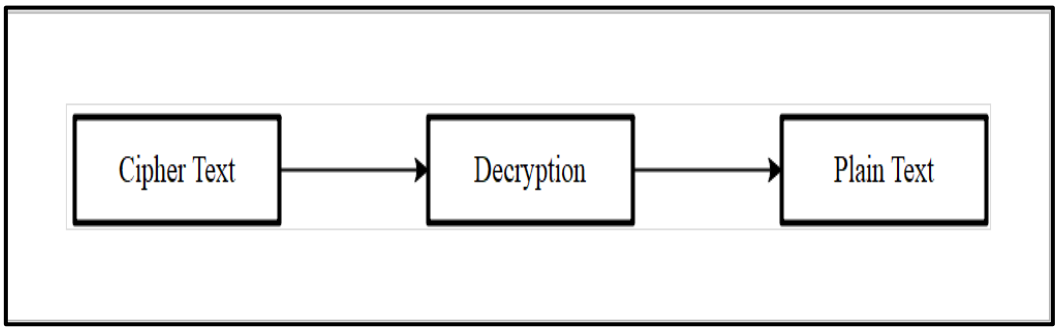


Figure 5.5: Decryption Process

Here, we confine to the explanation of a typical round of AES decryption [10]. Each round comprises four sub-processes conducted in the reverse order:

- Add round key
- Mix columns
- Shift rows

- Byte substitution

The round process is represented in figure 5.6.

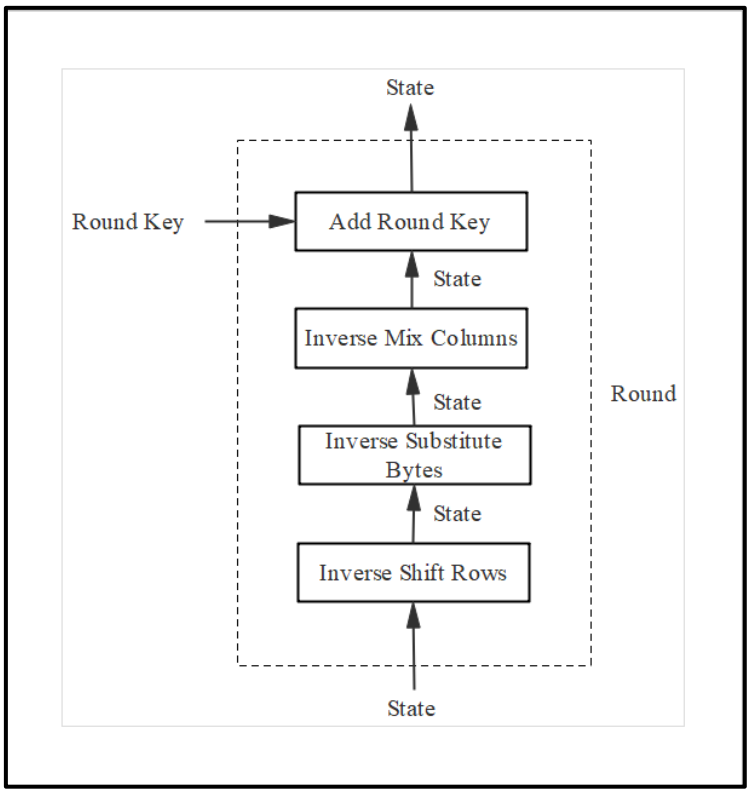


Figure 5.6: Structure of Decryption Process

Substitution Round

In AES, Substitution is performed for all bytes and that too, using only one table. This implies that if 2 bytes are same then their transformations are also same.

- **Inverse Substitute byte:** It is used at the decryption side and is the inverse of substitute bytes transformations. The bytes in the matrix are substituted one at a time, thus there are 16 distinct byte-to –byte transformations. Each byte is treated as two hexadecimal digits, where the first digit specifies the row and the second digit specifies the column of substitution table. The value at the intersection of the row and column in the transformation table is the new byte with which given byte is to be replaced.

Permutation Round

It performs a byte-level permutation, such that the order of bits in each byte does not change in the resultant bytes.

- **Inverse Shift Rows:** Bytes in rows of input state matrix are shifted to right and number of bytes to be shifted depends on the row number.

Mixing Round

Bits inside bytes are changed on the basis of bits in neighbouring bytes.

- **Mix Columns:** A constant square matrix is used. A square matrix is multiplied by each column of state matrix resulting into a column.
- **Inverse Mix Columns:** Uses inverse of constant square matrix.

Key Adding Round

Makes use of round key.128-bit round key is considered as 4 32-bit words and each 32-bit word is treated as a column matrix.

- **Add round key:** Self-invertible. Each column of state matrix is revoked with corresponding column matrix to produce a new column.

CHAPTER 6

EXPERIMENTAL RESULTS

As discussed in Chapter 5, we have used traditional encryption technique AES with the HE for encoding various files. The present study is focused on improving the security of data so that data cannot be tampered by the attackers while transmitting it over the public network. Thus, improving the security of the system. We have applied AES technique on the original data and then applied HE technique on the encrypted data. After applying this proposed technique, we get homomorphically encoded data as output which is then stored over the cloud. We have further implemented this work on the basis of time taken to encrypt and decrypt the data using both techniques. We have illustrated the results obtained using various file samples and made the comparison between both the techniques using above-mentioned factors.

This technique helps in improving the security of the cryptosystem. We have performed various experiments for implementing the above approach. Here, section 1 describes how data is encrypted using AES algorithm and proposed scheme and section 2 discuss the results obtained using above techniques. Experiments conducted in this work are discussed further in this chapter.

6.1 Screenshots

This section displays screenshots of experiments performed for encoding and decoding the data using AES algorithm and proposed technique on the selected files by the user displaying the time taken to encrypt and decrypt the data using this technique.

Presently in this section, we are working on encryption and decryption process using both AES and proposed technique.

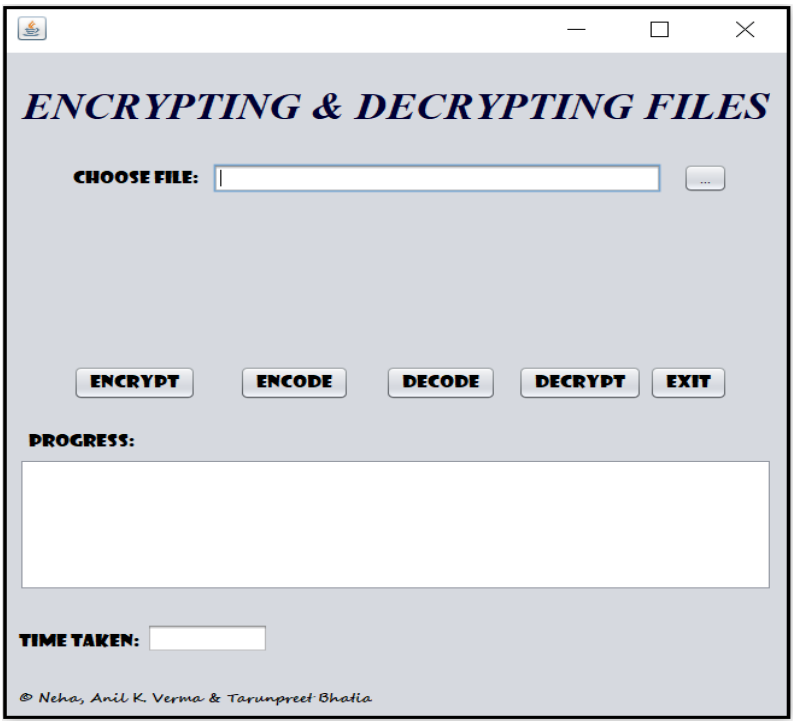


Figure 6.1: GUI of proposed work

First, we will select the file on which task is to be performed from the choose file panel. Now, we will be selecting the original file to be used for encrypting the data in it using the selected method shown in figure 6.2.

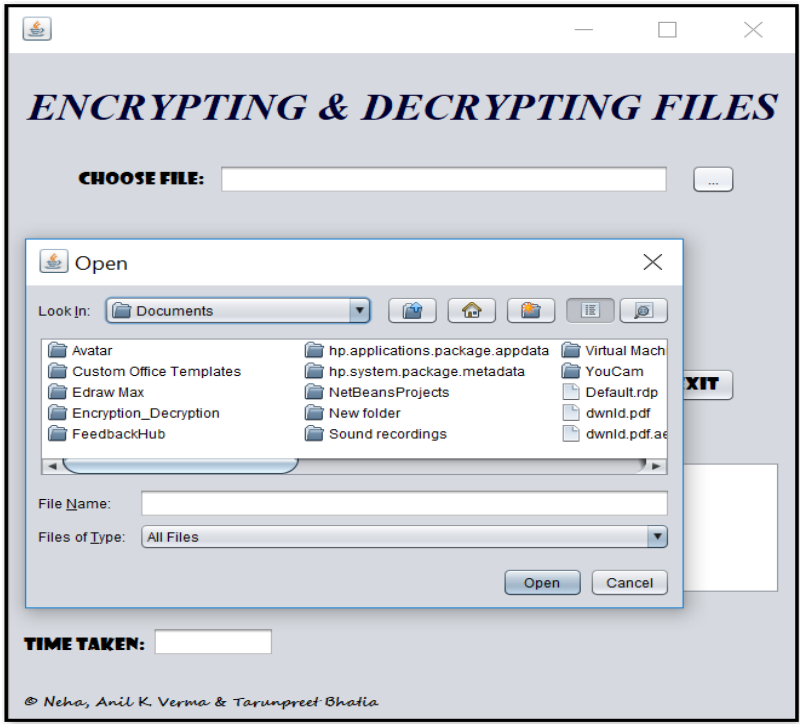


Figure 6.2: Selecting the file for the encryption process

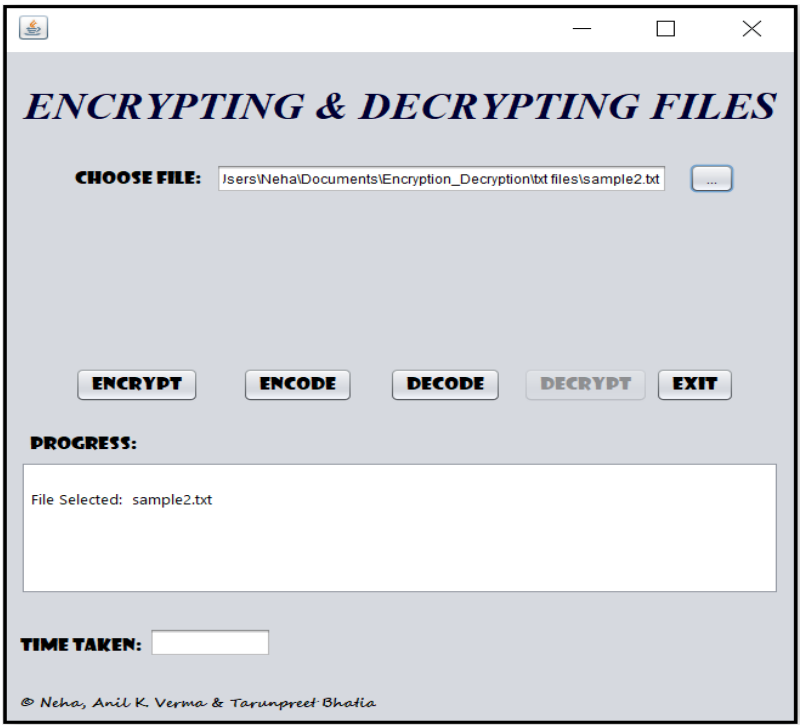


Figure 6.3: File selected

Then, we will choose the method as encrypt, encode (encryption using proposed technique), decode (decryption using proposed technique) or decrypt out of the methods defined as shown in figure 6.4.

Figure 6.4: Selecting the encryption method

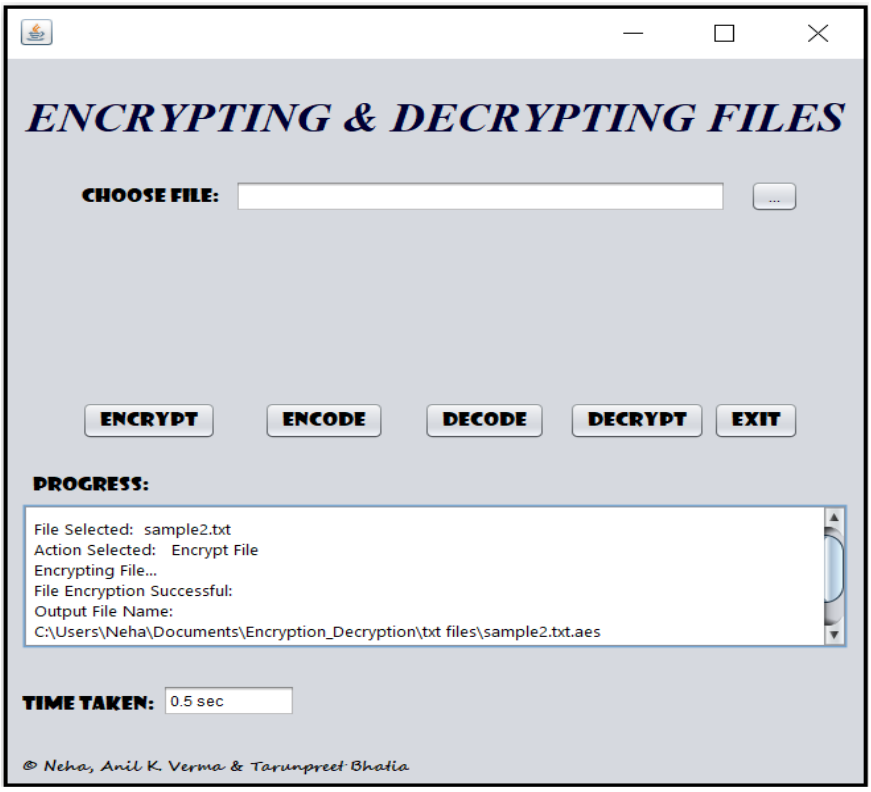


Figure 6.5: File is encrypted using AES

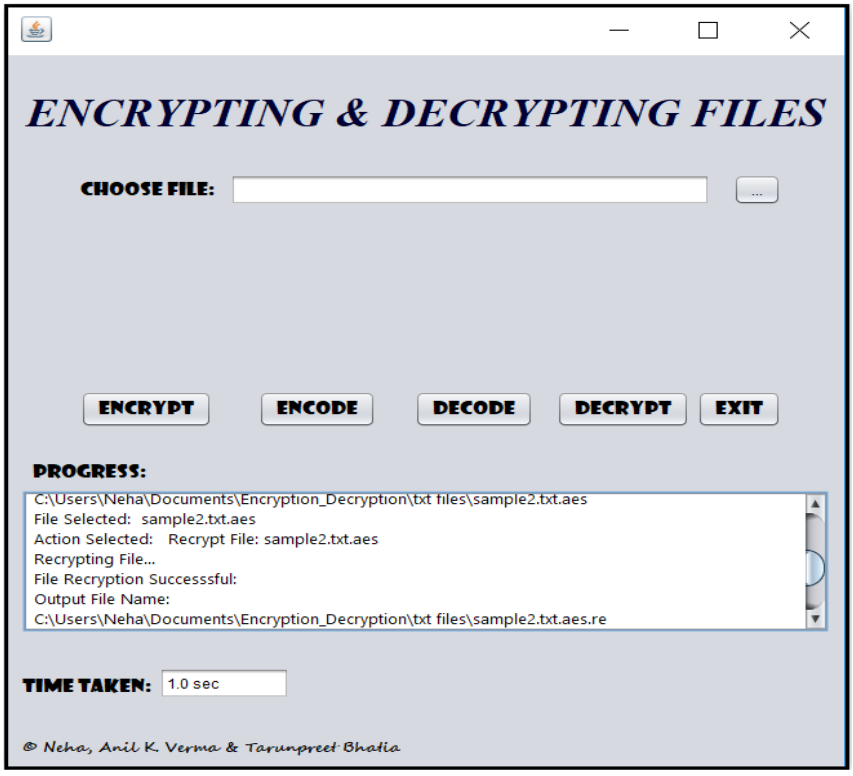


Figure 6.6: File is encrypted using proposed technique

6.2 Comparative Analysis of AES with Proposed Technique

In this section experiments have been performed for making the comparison between the AES and proposed technique on the basis of time taken to encrypt and decrypt the data in files. This proposed technique has been implemented in various text and audio samples.

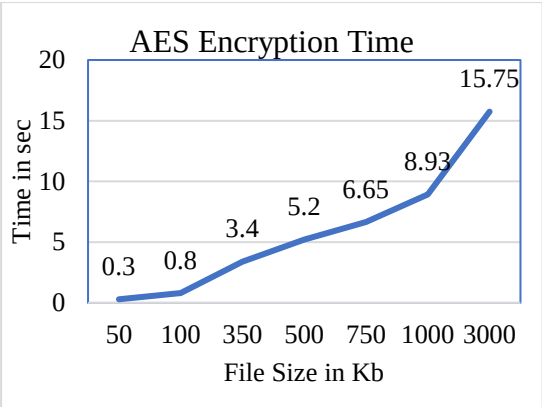
The three most important aspects considered while information hidings are security, capacity and the robustness. In proposed technique, we have implemented data encryption with cryptographic techniques. In this way, more security can be provided to the system. By comparing the two techniques, we find out

1. Time taken for encoding in proposed technique is less than the time taken for encoding in AES algorithm.

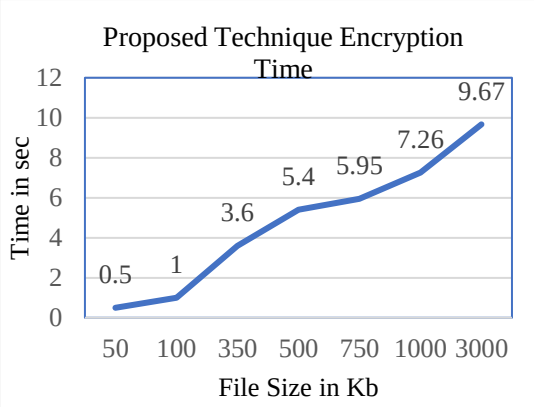
2. Change in the encryption & decryption time of proposed technique and AES is shown in table 6.1 & table 6.2 which fulfills the prerequisite of secret communication.

Table 6.1: Comparison between Proposed technique & AES for text samples

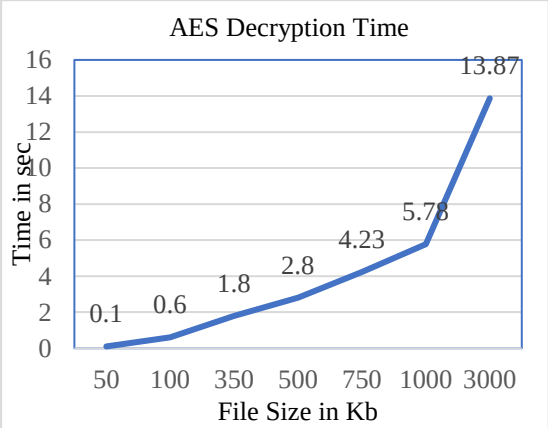
S.No.	**Input file size (Kb)**	**AES Technique Time (sec)**		**Proposed Technique Time (sec)**	
		Encryption Time	Decryption Time	Encryption Time	Decryption Time
1.	50	0.3	0.1	0.5	0.2
2.	100	0.8	0.6	1.0	0.5
3.	350	3.4	1.8	3.6	1.2
4.	500	5.2	2.8	5.4	1.8
5.	750	6.65	4.23	5.95	2.34
6.	1000	8.93	5.78	7.26	3.53
7.	3000	15.75	13.87	9.67	5.74



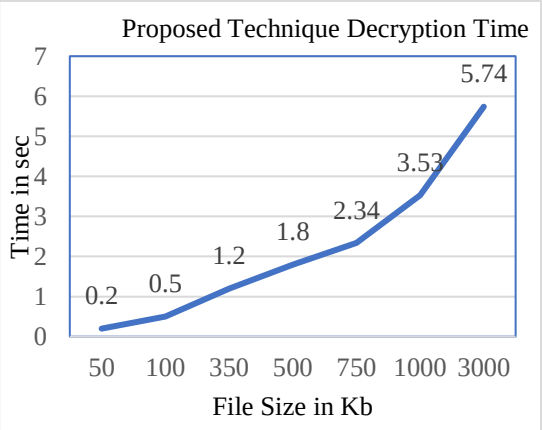
(i)



(iii)



(ii)

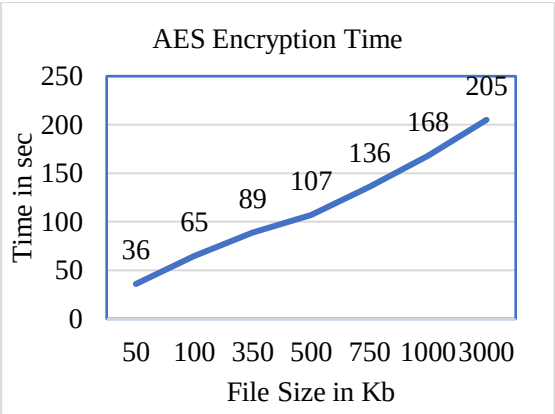


(iv)

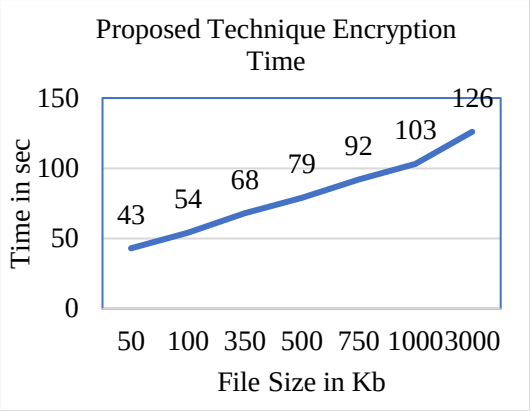
Figure 6.7: (i) AES Encryption Time, (ii) AES Decryption Time, (iii) Proposed Technique Encryption Time & (iv) Proposed Technique Decryption Time

Table 6.2: Comparison between Proposed technique & AES for audio samples

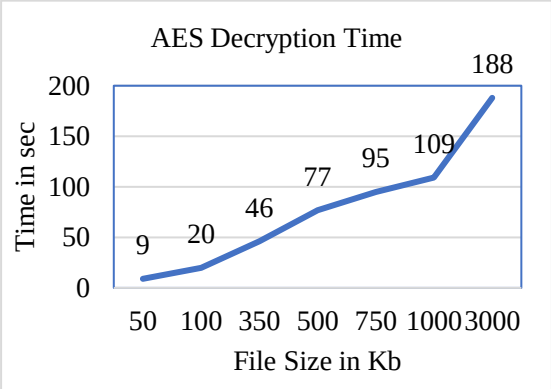
S.No.	**Input file size (Kb)**	**AES Technique Time (sec)**		**Proposed Technique Time (sec)**	
		Encryption Time	Decryption Time	Encryption Time	Decryption Time
1.	50	36	9	43	17
2.	100	65	20	54	23
3.	350	89	46	68	39
4.	500	107	77	79	47
5.	750	136	95	92	56
6.	1000	168	109	103	71
7.	3000	205	188	126	89



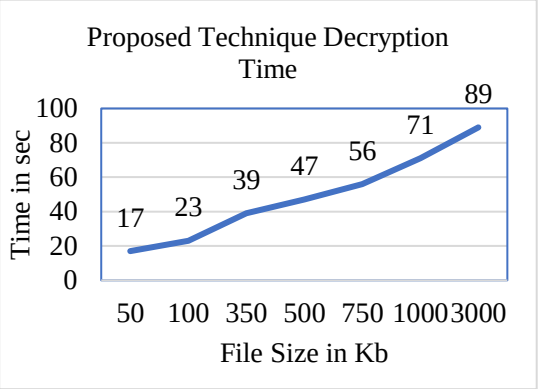
(i)



(iii)



(ii)



(iv)

Figure 6.8: (i) AES Encryption Time, (ii) AES Decryption Time, (iii) Proposed Technique Encryption Time & (iv) Proposed Technique Decryption Time

We can unmistakably observe from the tables and graphs that AES begins with a short encryption time then as the size of files increases, the time increments quickly. We would thus be able to infer that with little files as input AES works best. For proposed method, it begins with a higher timing than AES for little files, yet then at larger files, it gives the outcomes speedier than AES which improves it for vast files as far as performance is concerned.

The samples used for implementing this research work have been used from an online source and then these samples were used for making a comparison between both the techniques on the basis of various parameters defined in this proposed work [47].

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

The proposed work grants an enactment of an encryption algorithm using traditional encryption and homomorphic encryption for enhancing the security of different kind of data so that they can be transmitted and stored on the public cloud. We have proposed an approach for enhancing the security of data from the attackers. The approach used in proposed work and results obtained can be summarized as follows:

- The integrity of information is maintained using traditional encryption along with homomorphic encryption.
- The proposed work will not be changing the size of the files and is suitable for .txt, .doc, .pdf, .mp3 etc. samples.
- This combination of traditional encryption and homomorphic encryption make sure that even if the attacker interprets the files and will not be able to discover the secret information as it has been encoded twice using evaluation function.
- It provides better security to the system.

7.2 Future Scope

An interested researcher may implement a few more things to this work:

- It is difficult to obtain a system that satisfies both criteria of high security and robustness, therefore, to find a new mechanism to satisfy our needs is yet to be investigated.
- Future work is focused on increasing the capacity of secret information and confidentiality of the system.

REFERENCES

[1] W. Diffie and M. Hellman, “New directions in cryptography”, *IEEE Transactions on Information Theory,* 22(6):644-654, 1976.

[2] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems*”, Communications of the ACM,* 21(2): 120-126, 1978.

[3] S. Goldwasser and S. Micali, “Probabilistic encryption”, *Journal of Computer and System Sciences,* 28:270-297, 1984.

[4] T. El Gamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", *IEEE transactions on information theory* 31.4: 469-472, 1985.

[5] J. Feigenbaum and M. Merritt, “DIMACS Series in Discrete Mathematics and Theoretical Computer Science*”,* volume 2, chapter Open Questions, *Talk Abstracts, and Summary of Discussions,* pages 1-45, ACM, 1991.

[6] D. Naccache and J. Stern, “A new public key cryptosystem based on higher residues*”, In ACM Conference on Computer and Communications Security,* pages 59-66, 1998.

[7] P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes”, *Advances in Cryptology Eurocrypt,* 1592:223-238, 1999.

[8] S. Garfinkel, *Architects of the information society: 35 years of the Laboratory for Computer Science at MIT,* MIT Press, 1999.

[9] J. Daemen and V. Rijmen, “The Block cipher Rijndael”, *In the proceedings of Third International Conference on smart card Research and Applications,* CA, Lecture Notes in computer science, Vol.1820, pp. 227-284, 2000.

[10] V. Rijmen, and J. Daemen. "Advanced encryption standard", *Proceedings of Federal Information Processing Standards Publications, National Institute of Standards and Technology*: 19-22, 2001.

[11] Niels Ferguson, Richard Schroeppel, and Doug Whiting, "A simple algebraic representation of AES*", International Workshop on Selected Areas in Cryptography, Springer*, Berlin, Heidelberg, 2001.

[12] A. J. Elbirt, W. Yip, B. Chetwynd and C. Paar, "An FPGA Based Performance Evaluation of the AES Block Cipher Candidate Algorithm Finalists", *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 9, No. 4, pp. 545-557, 2001.

[13] J. Daemen and V. Rijmen, "The Design of Rijndael: AES-The Advanced Encryption Standard", *Springer Publications*, 2002.

[14] C. P. Su, T. F. Lin, C. T. Huang and C. W. Wu, "A High-Throughput Low-Cost AES Processor", *IEEE Communications Magazine*, Vol. 41, No. 12, pp. 86-91, 2003.

[15] S. Mangard, M. Aigner, and S. Dominikus, "A Highly Regular and Scalable AES Hardware Architecture", *IEEE Transactions on Computers*, Vol. 52, No. 4, pp. 483-491, 2003.

[16] Goldreich, Oded, "Foundations of Cryptography: Volume 2, Basic Applications", Vol. 2. *Cambridge university press*, 2004.

[17] Kun Peng, Colin Boyd, and Ed Dawson, "A Multiplicative Homomorphic Sealed-Bid Auction Based on Goldwasser-Micali Encryption*", In International Conference on Information Security, Springer, Berlin, Heidelberg*, Volume 3650 (pp. 374-388), 2005.

[18] Majdi Al-qdah, Lin Yi Hui, "Simple Encryption/Decryption Application", *International Journal of Computer Science and Security* 1.1, 33, 2007.

[19] C. Gentry, "Fully homomorphic encryption using ideal lattices", *In Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 169–178. ACM, 2009.

[20] R. L. Krutz & R.D. Vines, "Cloud security: A comprehensive guide to secure cloud computing", *Wiley Publishing*, 2010.

[21] Xin Zhou, and Xiaofei Tang, "Research and implementation of RSA algorithm for encryption and decryption", *Strategic Technology (IFOST), 2011 6th International Forum on,* Vol. 2. IEEE, 2011.

[22] M. Naehrig, K. Lauter, and V. Vaikuntanathan, “Can homomorphic encryption be practical?,” *In Proceedings of the 3rd ACM workshop on Cloud computing security workshop,* (pp. 113-124). ACM., 2011.

[23] Bismita Gadanayak, Chittaranjan Pradhan, "Encryption on MP3 compression", *MES Journal of Technology and Management* 2 (1), 86-89, 2011.

[24] P. Mell, & T. Grance, “The NIST definition of cloud computing”, Communications of the ACM 53, no. 6:50, 2011.

[25] Christof Paar and Jan Pelzl, “Understanding Cryptography”, *Springer Publications*, 2012.

[26] M. Brenner, H. Perl, and M. Smith, “Practical Applications of Homomorphic Encryption,” *Int. Conf. Secur. Cryptogr. (SECRYPT 2012),* pp. 1–10, 2012.

[27] S. Pavithra, E.Ramadevi, “Throughput Analysis of Symmetric Algorithms”, *International Journal of Advanced Networking and Applications*, Volume-4, Issue-2, Pages:1574-1577, 2012.

[28] Saurabh Sharma, Pushpendra Kumar Pateriya, “A Study on Different Approaches of Selective Encryption Technique”, *International Journal of Computer Science and Communication Networks*, Vol. 2.6, 658-662, 2012.

[29] Jaydip Sen, “Homomorphic Encryption: Theory & Application”, *arXiv preprint* arXiv:1305.5886, July 2013.

[30] Prerna Mahajan, and Abhishek Sachdeva, "A study of encryption algorithms AES, DES and RSA for security", *Global Journal of Computer Science and Technology,* 2013.

[31] Ryan, Mark D., "Cloud computing security: The scientific challenge, and a survey of solutions", *Journal of Systems and Software,* 86.9, 2263-2268, 2013.

[32] Sruthi B. Asok, P.Karthigaikumar, Sandhya R, Naveen Jarold K, N.M Siva Mangai, “A Secure cryptographic scheme for Audio Signals”, *International Conference on communication and Signal Processing,* April 3-5, 2013, India.

[33] Maha TEBAA, Said EL HAJII, “Secure Cloud Computing through Homomorphic Encryption”, *arXiv preprint*, arXiv:1409.0829, 2014.

[34] Parmar, P. V., Padhar, S. B., Patel, S. N., Bhatt, N. I., & Jhaveri, R. H., "Survey of various homomorphic encryption algorithms and schemes", *International Journal of Computer Applications,* 91.8,2014.

[35] Shine P James, Sudhish N. George, Deepthi P P, “An Audio Encryption Technique based on LFSR based Alternating Step Generator”, In *Electronics, Computing and Communication Technologies (IEEE CONNECT), 2014 IEEE International Conference on*, pp. 1-6. IEEE, 2014.

[36] Y. Sun, J. Zhang, Y. Xiong, & G. Zhu, “Data security and privacy in cloud computing”, *International Journal of Distributed Sensor Networks*, *10*(7), 190903, 2014.

[37] R.Kangavalli, Dr.Vagdevi S, “A Mixed Homomorphic Encryption Scheme for Secure Data Storage in Cloud”, In *Advance Computing Conference (IACC), 2015 IEEE International*, pp. 1062-1066. IEEE, 2015.

[38] Yasmina Bensitel, and Rahal Ramadi, "Secure data in cloud computing using homomorphic encryption", *Journal of Theoretical and Applied Information Technology,* 82.2, 206, 2015.

[39] T. Sridokmai and S. Prakancharoen, “The homomorphic other property of Paillier cryptosystem,” In *Science and Technology (TICST), 2015 International Conference on*, pp. 356-359. IEEE, 2015.

[40] Kunerd, “*jPaillier*,” 2015. [Online]. Available: https://github.com/kunerd/jpaillier.

[41] V. Biksham, and D. Vasumathi, "Query-based computations on encrypted data through homomorphic encryption in cloud computing security", *Electrical,*

Electronics, and Optimization Techniques (ICEEOT), International Conference on. IEEE, 2016.

[42] Nishtha Mathur, and Rajesh Bansode, "AES Based Text Encryption Using 12 Rounds With Dynamic Key Selection", *Procedia Computer Science*, 79, 1036-1043, 2016.

[43] Garry Kranz, Stephen J. Bigelow, and Jeff Hawkins, "*Essential Guide: Windows Server 2016 release broadens reach across IT spectrum*", Posted by Margaret Rouse, this was last updated in January 2017.

[44] Zhibin Gong, Xiao Youan, Long Yihong, and Yang Yanli, "Research on database ciphertext retrieval based on homomorphic encryption", In *Electronics Information and Emergency Communication (ICEIEC), 2017 7th IEEE International Conference on*, pp. 149-152. IEEE, 2017.

[45] Radjab Harerimana, Syh-Yuan Tan, and Wei-Chuen Yau, "A Java implementation of paillier homomorphic encryption scheme", *Information and Communication Technology (ICoIC7), 2017 5th International Conference on*, pp. 1-6, IEEE, 2017.

[46] M Babenko, N Chervyakov, A Tchernykh, N Kucherov, M Deryabin, G Radchenko, PO Navaux, V Svyatkin, "Security analysis of homomorphic encryption scheme for cloud computing: Known-plaintext attack", *InYoung Researchers in Electrical and Electronic Engineering (EIConRus), 2018 IEEE Conference of Russian,* pp. 270-274, 2018 Jan 29, IEEE.

[47] https://www.freesoundeffects.com/free-sounds/explosion-10070/

APPENDIX A

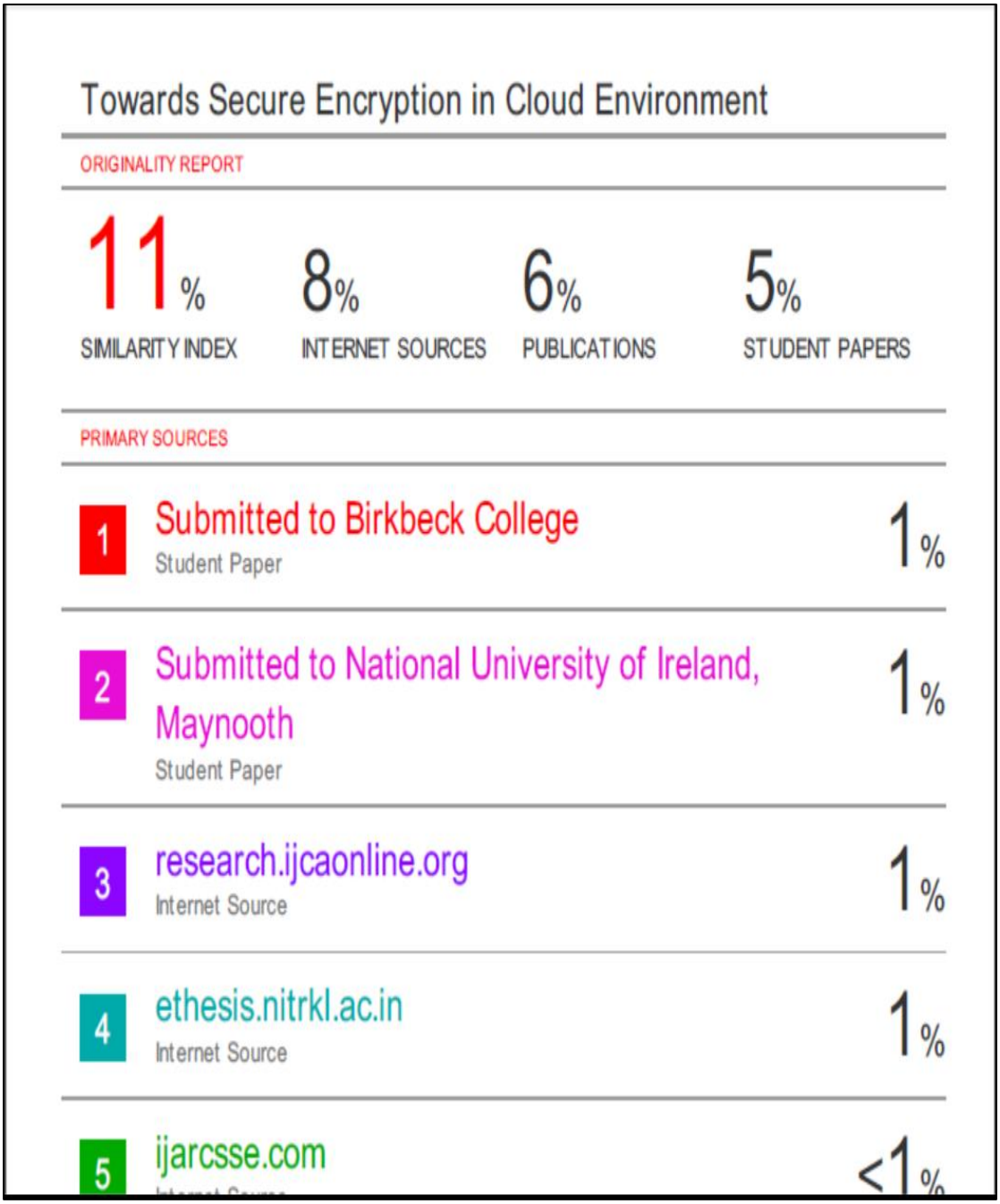
PUBLICATION

---

[1] Neha, Tarunpreet Bhatia, Anil K. Verma, "Data Security Using Homomorphic Encryption in Cloud", *In Proc. of International Conference on Materials, Applied Physics and Engineering 2018.*

APPENDIX B

PLAGIARISM REPORT



Towards Secure Encryption in Cloud Environment

ORIGINALITY REPORT

11%	8%	6%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Birkbeck College Student Paper	1%
2	Submitted to National University of Ireland, Maynooth Student Paper	1%
3	research.ijcaonline.org Internet Source	1%
4	ethesis.nitrkl.ac.in Internet Source	1%
5	ijarcsse.com	<1%