**A CIPHER MODEL FOR TEXT AND IMAGE DATA USING VIGENÈRE WITH ADVANCED ENCRYPTION STANDARD (AES)**

**BY**

**ERONDU UDOCHUKWU IHEANACHO**

**(20PGCD000155)**

**SEPTEMBER, 2022**

**A CIPHER MODEL FOR TEXT AND IMAGE DATA USING VIGENÈRE WITH ADVANCED ENCRYPTION STANDARD (AES)**

**M.Sc. DISSERTATION**

**BY**

**ERONDU UDOCHUKWU IHEANACHO**

**(20PGCD000155)**

**SUPERVISOR**

**DR. (MRS) M.O. ADEBIYI**

**CO-SUPERVISOR**

**DR. E.O. ASANI**

**DEPARTMENT OF COMPUTER SCIENCE,**

**LANDMARK UNIVERSITY, OMU-ARAN**

**SEPTEMBER 2022**

**A CIPHER MODEL FOR TEXT AND IMAGE DATA USING VIGENÈRE WITH ADVANCED ENCRYPTION STANDARD (AES)**

**ERONDU UDOCHUKWU IHEANACHO**

**(20PGCD000155)**

**A Dissertation submitted to the Department of Computer Science, College of Pure and Applied Sciences, Landmark University, Omu-Aran. Nigeria.**

**In Partial Fulfilment of the Requirements for the Award of the Degree of Masters of Science (MSc) in Computer Science.**

**SEPTEMBER 2022**

# DECLARATION

I, **ERONDU UDOCHUKWU IHEANACHO (20PGCD000155),** a MSc student in the Department of Computer Science, Landmark University, Omu-Aran, hereby declare that this thesis entitled “**A CIPHER MODEL FOR TEXT AND IMAGE DATA USING VIGENÈRE WITH ADVANCED ENCRYPTION STANDARD (AES)**”, submitted by me is based on my original work. Any material(s) obtained from other sources or work done by any other persons or institutions have been duly acknowledged.

ERONDU UDOCHUKWU IHEANACHO (20PGCD000155)

----------------------

Signature & Date

# CERTIFICATION

This is to certify that this thesis has been read and approved as meeting the requirements of the Department of Computer Science, Landmark University, Omu-Aran, Nigeria, for the Award of MSc Degree.

Dr. (Mrs) M.O. Adebiyi Date

Supervisor

Dr. E.O. Asani Date

(Co- Supervisor)

Dr. Mrs. M.O. Adebiyi Date

(Head of Department)

Prof. R.O. Oladele Date

(External Examiner)

# ABSTRACT

In recent times, data exchange and transfer have increased in exponential dimensions, thus making the task of securing information and information infrastructure a critical one. Cryptography has been used extensively as to a viable technique for securing information assets and infrastructure. Cryptography protects data and communications channels to ensure that only the intended audience may access, read, and process it. Cryptosystems deploy mathematical models and computational transpositions to change plain messages to unintelligible ones known as ciphers. The Vigenère cipher is a widely used cryptographic method for encrypting and decrypting messages. The recurring nature of the Vigenère cipher's key is its biggest drawback. The ciphertext can be viewed as a series of interconnected Caesar ciphers, each of which is straightforward to decipher if a cryptanalyst accurately predicts the length of the key. The primary goal of this research is to improve upon existing encryption and decryption methods by combining a variant of the Vigenère cipher and the Advanced Encryption Standard (AES) algorithm to create a system with increased attack resistance. In this research, a Vigenère cipher variant using AES to protect data from cryptanalysis and pattern prediction was developed. The created model was put through its paces on both text and image datasets, with testing done with an eye toward both speed and memory consumption. When compared to previous methods, the improved Vigenère-AES fared better in terms of encryption/decryption time, computational resources utilized, and key generation methodologies. The model can be used to protect sensitive data from hackers.

# DEDICATION

This research is dedicated to the almighty God in heaven for His endless and undying love and to my *gods* on earth Prof. E.S. Erondu and Dr.(Mrs.) C.I. Erondu for all the support and for always believing in me.

# ACKNOWLEDGMENT

I wish to thank my supervisor Dr. Marion Adebiyi for being a mother and a mentor. She never failed in her duty to guide and tutor me as well as ensuring I gave my best. I like to also thank my co-supervisor Dr. E.O. Asani for his support toward the successful completion of this project. To the Dean College of Pure and Applied Sciences (DCPAS), Prof. A.A. Adebiyi for his constant drive. Special gratitude goes to all members of the Department of Computer Science, Landmark University Omu-Aran. Notable mention goes to Dr. Michael Arowolo for all the support, encouragement and above all for letting me tap from his wealth of knowledge and experience, Dr. Moses Abiodun, Dr. Amos Bajeh, and Dr. Babatunde Gbadamosi, thank you all for the push.

I am eternally grateful for the wonderful people I crossed paths with at Landmark University; Otu-Ashi, Abiodun, Cyril, Alex and Nnaemeka for all the memorable moments we shared. To my supportive colleagues; Nehemiah (Prof), Mr. Suraj and Mary, thank you all for everything. The ever-cheerful Bukky and Wutem, thank you for making my stay in Landmark fascinating. To the supportive Abigail, thank you for always looking out for ‘the boys’.

I will not fail to mention the words of encouragement, support and drive given to me by Oluchi Okereke (Lulum), I really appreciate you for demonstrating and exhibiting the true definition of friendship. Thank you, Bala and Nonsue for everything.

It will be criminal to forget the impact of my amiable grandma, Lady J.N. Elele, she always highlighted the importance of hard work, diligence and being a good ambassador, thank you mama for everything. Special thanks to my sweet aunt Mrs. Gloria Ezeronye for all the love and support. My wonderful and exceptional siblings can’t be left out; Oluchi, Dika (Agu), Uche and Nanya, you all have been nothing short of amazing. Thank you Nedu (Uncle Mijey) for all the role you played, I can never forget.

As the popular cliché goes; “always save the best for the last”, I wish to render my unreserved appreciation to the ones I owe everything, my greatest support system, my lovely parents Prof & Dr.(Mrs.) E.S. Erondu for the sacrifice, love, support, words of encouragement and prayers given to me which made my journey smooth and seamless. I could never ask for better parents!

# TABLE OF CONTENT

DECLARATION…………………………………………………………………… ii

CERTIFICATION……………………………………………………………………iii

ABSTRACT………….……………………………………………………………… iv

DEDICATION……………………………………………………………………… v

ACKNOWLEDGEMENT………………………………………………………….. vi

TABLE OF CONTENTS…………….…………………………………………… viii

LIST OF TABLES………………………………………………………………….. x

LIST OF FIGURES………………………………………………………………… xi

LIST OF ALGORITHMS……………………………………………………………xii

INTRODUCTION 13

1.1 Background to the Study 14

1.2 Statement of the Problem 14

1.3 Justification for the Study 15

1.4 Aim and Objectives of the Study 15

1.5 Research Questions 16

1.6 Scope of the Study 16

1.7 Significance of the Study 17

1.8 Arrangement of the Dissertation…………………………………………17

CHAPTER TWO 18

REVIEW OF RELATED LITERATURE 18

2.1 Conceptual Issues 18

2.1.1 Cryptography 18

2.1.2 Terminologies Used in Cryptography…………………………………. 18

2.1.3 Classification of Cryptographic Algorithms……………………………19

2.1.3.1 Symmetric Algorithms………………………………………………… 20

2.1.3.2 Asymmetric Algorithms……………………………………………….. 25

2.1.4 Homomorphic Encryptions…………………………………………… 28

2.1.5 Deception-based Techniques……………………………………………29

2.1.6 Evaluation metrics for Encryption-Decryption Algorithms…………….33

2.1.7 Key Management: The Diffie Hellman algorithm…………………… 35

2.2 Review of Methodological Approaches………………………………...37

2.2.1 Advanced Encryption Standard (AES)…………………………………37

2.2.2 Hybrid Cubes Encryption Algorithm (HiSea)…………………………39

2.2.3 Vigenère Algorithm………………………………………………….. 39

2.3 Gap Identified in the Literature………………………………………….40

CHAPTER THREE 44

METHODOLOGY 44

3.1 Research Design 44

3.2 Research Instruments 47

3.2.1 System Requirements…………………………………………………...47

3.2.2 Hardware Requirements………………………………………………...47

3.2.3 Software Requirements…………………………………………………48

CHAPTER FOUR 49

RESULTS AND DISCUSSIONS 49

4.1 Results……………………………………………………………………..49

4.2 Evaluation and Discussion……………………………………………… 62

CHAPTER FIVE 65

CONCLUSION AND RECOMMENDATION 65

5.1 Conclusion……………………………………………………………...65

5.2 Contribution to Knowledge…………………………………………….66

5.3 Recommendation……………………………………………………….66

REFERENCES 67

# LIST OF TABLES

**Table Page**

**Table 4.1** Vigenère and Vigenère-AES Results ………………………………… 63

**Table 4.2** Comparison Results with Existing Models …………………………. 64

# LIST OF FIGURES

**Figure Page**

Figure 3.1 Research Design …………………………………………………………. 44

Figure 3.2(a) Vigenère-AES Encryption ..…………………………………………...45

Figure 3.2(b) Vigenère-AES Decryption ………………………............................... 45

Figure 4.1 The Environment for the Developed Approach………………………… 50

Figure 4.2 Loaded Text Data with 32 Bytes………………………………………... 51

Figure 4.3 Randomly Generated Simulated Key…………………………………… 52

Figure 4.4 Generated Ciphertext…………………………………………………….52

Figure 4.5 Decrypted Text…………………………………………………………. 53

Figure 4.6 Display of Plain text to be Encrypted using 64 Byte Key Size…………..54

Figure 4.7 Generated Simulated Key for the 64 Bytes Key Size……………………54

Figure 4.8 Ciphertext for 64 Bytes Key Size Message………………………………55

Figure 4.9 Decrypted Message Using 64 Bytes Key Size…………………………...55

Figure 4.10 Vigenère-AES Encryption using 32-Bytes Key-size…………………... 56

Figure 4.11 Results of Vigenère-AES using 32 Byte Key-Size…………………….. 56

Figure 4.12 Results of Vigenère-AES using 64 Bytes Key-size……………………. 57

Figure 4.13 Loaded Image Data Using 32 Byte Key Size………………………….. 58

Figure 4.14 Generated Simulated Key for the Loaded Image……………………… 58

Figure 4.15 Scrambled Encrypted Image using Vigenère with 32 bytes key size…. 59

Figure 4.16 Decrypted Image Data…………………………………………………. 60

Figure 4.17 Results of Vigenère-AES for 32 Bytes………………………………… 60

Figure 4.18 Results of Vigenère for 64 Bytes……………………………................. 61

Figure 4.19 Results of Vigenère-AES for 64 Bytes……………………….................. 61

# LIST OF ALGORITHMS

**Algorithms Page**

Algorithm 3.1: Vigenère Encryption Algorithm…………………………………………. 46

Algotihm 3.2: AES Algorithm..................................................................................... 47

# CHAPTER ONE

# INTRODUCTION

* 1. Background to the Study

The volume of data files exchanged over the internet continues to grow in tandem with the rapid advancement of computer technology. As a result, both academic and research disciplines are interested in the secure transmission of data across public channels (Jang-Jaccard & Nepal, 2014). The increased use of digital media for data transmission across diverse channels can expose messages conveyed over networks to intruders or third parties. Information security attempts to address security breaches and the abuse of confidential information intercepted by unauthorized parties as major issues (Pourbabak et al., 2019). Information Security is concerned with several issues, including privacy. For instance, a third party can be prohibited from deciphering information sent over an insecure connection all through signal broadcast using cryptographic encryption approaches. Cryptographic technologies for ensuring the safety of digital content have become increasingly important in modern-day data transmission.

Encryption of messages has become vital in today's technological age to ensure that data transferred via communications channels is safeguarded and difficult to decode. An enormous amount of data is sent through the internet, which is considered to be the most efficient medium despite being a public access medium (Al-Janabi et al., 2017). As a result, numerous researchers have devised efficient algorithms to encrypt this data from simple text into ciphers to counteract this flaw (Al-Shaaby & AlKharobi, 2017). Encryption is the process of transforming data into intelligible forms, often referred to as ciphers, which may not be decoded without the use of special keys, while decryption is the process of converting the ciphers back to plain, readable form by authorized.

It is possible that the keys are interchangeable, or that there's a subtle break in the middle. For the sake of maintaining a secure channel for exchanging secret evidence, the keys denote shared but unpublished text among users. (Rao & Nayak, 2014). Cryptographic techniques may be classified by the use of keys, as symmetric or asymmetric cryptographies. Symmetric Encryption uses a single key to encrypt and decrypt the information and is thus referred to as private key encryption. Advanced Encryption Standard (AES), Triple Data Encryption Standard (3DES), Blow-fish, and Serpent are examples of symmetric algorithms. Asymmetric or public key encoding is a technique of encoding in which encrypted information with a receiver's free key may individually be decoded by the possessor of the conforming private key, who is presumably the possessor of the key and the individual linked with the public key utilized for the sake of privacy (Harmening, 2017). Rivest Shamir Adleman (RSA), the Diffie–Hellman key, the Digital Signature Standard (DSS) and the Digital Signature Algorithm (DSA), are examples of asymmetric encryption methods. Modern cryptography uses complex and advanced mathematical algorithms to encrypt text, as well as image encryption techniques based on Red, Green, and Blue (RGB) pixel displacement, in which the pixels of images are scrambled to produce a cipher image.

* 1. **Statement of the Problem**

Securing information on the internet has become a critical task in recent times. Encryption is a key component of information security to protect such data. Research has shown that the strength of the cryptologic method and the confidentiality of the key regulates the safety of encoded data (Yassein et al., 2017).

The Vigenère cipher is well-known for its simplicity and efficiency. However, key repetition has become a known vulnerability of this technique. Thus, a cryptanalyst who has guessed the key length correctly may be able to break its cipher by simply treating it as a series of separate Caesar ciphers. (Bhateja *et al.*, 2015). To mitigate this, the Vigenère-AES cipher is used in this study to perform multi-level encryption, to improve security against cryptanalysis. With Vigenère Cipher and AES, it is possible to get a more secure ciphertext result, which will reduce the likelihood that the key is known by third parties.

* 1. **Justification for the Study**

To protect data, encryption and decryption have been widely employed; one of the simplest encryption methods is known as the Vigenère Cipher, but it has its drawbacks. This study shows an improved Vigenère-AES cipher that is more secure against attacks such as Friedman, Kasiski, and others. To create an extremely difficult-to-break cipher a Vigenère Cipher with AES has been developed.

Using multiple encryption models makes it more difficult to perform cryptanalysis, frequency analysis, pattern prediction, and brute-force attacks. To present an encoding method for secure data communication that is based on existing cryptological methods, cryptography researchers are increasingly turning to the Vigenère cipher. An improved substitution cipher key is used to increase complexity by scrambling texts and delivering a predetermined imbalance in the assembled encoded texts, which is achieved using the Vigenère cipher and AES.

* 1. **Aim and Objectives**

This study aims to develop a cipher model for the encryption and decryption of text and image data using Vigenère cipher and AES.

The specific objectives are:

1. to design a multi-level encryption model;
2. to implement the designed model; and
3. to evaluate the implemented model and compare it with other existing models.
   1. **Research Questions**

In the present world, where the volume of data exchanged has increased, information security has become a critical task. As a result, the following research questions are addressed in this study:

1. How does cryptography convert plain text messages into unreadable messages, and how do encoding procedures play a role in information security systems?
2. How has the Vigenère method, as an instance of a polyalphabetic stream cipher, been used to determine the length of the encryption key?
3. How can a multi-level model based on Vigenère cipher and AES be designed to provide better security against cryptanalysis and pattern prediction?
4. What is the performance of the Vigenère-AES model vis-à-vis existing techniques?
   1. **Scope of the Study**

This study suggests a Vigenère-AES approach that uses a key to generate a new ciphertext from an equivalent fixed-length plaintext, which then serves as a new key. This study is limited to the adoption of two cryptographic algorithms; Vigenère cipher and Advanced Encryption Standard (AES). The datasets to be tested by the modified model are text and image data.

* 1. **Significance of the Study**

The Vigenère cipher and AES were combined in this study to create a more complex algorithm. As a result, this approach aims to strengthen the Vigenère cipher's security flaws. In other words, by doing this research, we are making it harder to decipher the ciphertext from the plain text and vice versa. The aim of the revised version of the proposed method is to supply the recipient with statements that make sense to them and lack ambiguity. The idea of incorporating more characters via a tailored method into cryptographic algorithms is not without merit. Given the availability of sophisticated computers, the complexity of such processes must be raised. Improve the safety of the standard cipher algorithm while simultaneously expanding the number of characters it can encrypt. It is anticipated that the study will have significant implications for researchers and practitioners in the field of data security.

* 1. **Arrangement of the Dissertation**

This dissertation is organized into five chapters. Chapter one includes an introduction to the study carried out, a statement of the problem, justification for the study, aim and objectives, research questions, overview, significance, and scope of the study. The second chapter covers a review of fundamental concepts as well as a detailed review of the methodological approach that describes the methodological approach of this study to tackling information security. Chapter three covers the description of the conceptual design, materials, and method, as well as a dataset, performance model and metrics. Chapter four focuses on testing, discussion of the results obtained, and evaluation of the techniques. The thesis is finally concluded in chapter five with a summarized discussion of results, contributions to knowledge, recommendations, and suggestions for further work.

**CHAPTER TWO**

1. **REVIEW OF LITERATURE**
   1. Conceptual Issues
      1. Cryptography

Contingent on the security needs and potential risks engaged, different cryptographic processes, including symmetrical key or unrestricted key cryptography, may be utilized to protect data transmission and handling. Moreover, the encryption algorithm allows multiple mathematical calculations to be executed on encoded data. (Gaire et al., 2019). Cryptographic techniques have three important goals:

* **Authentication:** Validating a user's or processer's identity.
* **Privacy:** Trust the data's contents are concealed.
* **Integrity:** Guarantees that the message received is the same as the message that was sent.

Depending on the circumstances, one or more of these objectives may be prioritized.

#### **Terminologies Used in Cryptography**

Various cryptographic terminologies exist among these are:

**Authentication:** At the receiver end, the individuality of the sender is verified. A user or system can demonstrate their authenticity to somebody else they do not know.

**Confidentiality:** The most frequently addressed target. It denotes that location and distribution are only received by the authorized user.

**Integrity**: means ensuring the message received is the same as the message sent. Solitary-approved operators can change the information.

**Access control:** this is the process of ensuring that one approved user has access to the specified information. Non-repudiation is a technique of promising note broadcast between revelries using a digital signature or encryption. It aids in the prevention of authentication attempt denial.

**Plain text:** Before encrypting ciphertext, plain text is a simple readable text in cryptography. *plaintext* usually means unencrypted information pending input into cryptographic algorithms, usually encryption algorithms. User A might direct communication to User B asking how he or she is. In this case, the plain text message will be, "How are you?"

**Ciphertext:** Text that has been encrypted using a cryptographic algorithm is called ciphertext. To understand ciphertext, it must first be decrypted to reveal the original, plaintext message. The method used to reverse the encryption process is called a decryption cipher.

**Encryption:** Encryption is the method used to secure data in cryptography. During this procedure, the plaintext is transformed into ciphertext, another representation of the data.

**Decryption:** The conversion of encrypted data into its original form is called *Decryption*. It is generally a reverse process of encryption.

#### **Classification of Cryptographic Algorithms**

#### Cryptographic algorithms are generally classified in the literature as symmetric and asymmetric algorithms.

#### **Symmetric Algorithms**

To encode and decode a message using a symmetric (secret key) encryption method, the same key must be used. The encryption/decryption public key is kept secret and accessible only by the authorized sender and receiver. The confidentiality and safety of the keys used in the encoding and decoding processes are what give symmetric key encoding its power. Based on message bit grouping, symmetrical encoding procedures are classed as block or stream ciphers. Sets of message types of a static magnitude (a block) encode all on one occasion and are directed to the receiver in a block cipher. Furthermore, the block code can be classified as binary or non-binary, depending on the ultimate outcomes of the keys, messages, and ciphertexts. Message bit sizes for the binary block code are 256, 192, 128, and 64 whereas the non-binary block code has no standard dependent on code execution (Johnson, 2020).

3DES, DES, AES, HiSea, BLOWFISH, RC4, and others are examples of symmetric key encryption techniques. In a symmetric block cipher, the encryption method transforms plaintext to ciphertext using the undisclosed key produced by the key schedule procedure. Similarly, the ciphertext is decrypted with the identical key formerly directed to the intended recipient. Since this dimension of the key is like the message length, the stream cipher's block magnitude is one character, which is insufficient for a package dealing out. The stream cipher's process is illustrated in the resulting stages (Faheem et al., 2017)

1. To generate the sole code of ciphertext, a sole code of plaintext is combined with a sole code from the key torrent.
2. The receiver obtains the ciphertext code from Step 1.
3. Steps 1 and 2 are repeated till the whole message is sent.

#### **Data Encryption Standard (DES)**

Data Encryption Standard (DES) was the first synchronous encoding technique established by IBM and embraced in 1972 by the National Bureau of Standards (NBS), and it is responsible for evaluating and implementing the standard block cipher. It has a 64-bit key with 56 bits derived at random that are explicitly utilized key bits by the algorithm. The remainder 8 bits are established to brand a similarity of an individual 8-bytes so they are used for fault discovery and are not used by the procedure. For encoding and decoding, DES employs a single secret key. The key size is 56 bits, and the message is encoded with a block size of 64 bits. Correspondingly, the decryption of a 64-bit ciphertext employs the same 56-bit key that was used to generate the initial 64-bit message block. To process the 64 bits input, the DES algorithm employs a different combination, 16 rounds of the key, and the completed pseudorandom. (Chaloop et al., 2017).

The DES symmetric key block cipher was created by IBM. There are 16 rounds of the Feistel structure used in this technique, each with a distinct key, and each cycle has a 64-bit block size. Here is a breakdown of the main DES stages (Wilson, 2016):

* + 1. The 56-bit key is split into two sets of 28 bits. i. Each of these sections has been streamlined into this one. Depending on the iteration, one or two bits of the key are swapped around. A compressed key of 48 bits is used to encrypt the plain text block this time around.
    2. The 64-bit information is split into two pieces, each of which is 32 bits in length. The block size is raised from 24 bits to 48 bits by enlarging half of it. Afterward, the result is XOR'd with the 48-bit compression key created in the first stage.
    3. The output is then passed to an S-box, which alters the key bits and reduces the 48-bit block to 32 bits. The bits are permuted in the P-box, which receives its input from the S-box. Both blocks are then exchanged and used as input for the next iteration. The rearranged key halves are used in the first stage.

#### **Blowfish algorithm**

It encrypts 64-bit data with a variable-length key of 32–448 bits. For modifying packet size, it performs better than any other algorithm. A successful attack on this method is yet to be discovered. Text-to-image translation, for example, takes a long-time using Blowfish. Packet switching, for example, is incompatible with Blowfish because of the frequency of key changes. (Minaam et al., 2010).

This algorithm has a higher encryption and decryption throughput than others. The Blowfish technique is utilized to encode and decode the data in the scheme since it is so quick. The blowfish technique may also be used to encrypt and decode massive volumes of data quickly, which will increase the speed and security of big amounts of data (Manku & Vasanth, 2015).

Feistel-based Blowfish is an encryption and decryption algorithm based on symmetric block ciphers. In 1993, Bruce Schneier introduced it. Since most encryption techniques are proprietary, they can't be shared with the public. If you're looking for a fast, open-source encryption algorithm that doesn't require a license or patent, Blowfish is the best option. Up to 448 characters can be encoded in a single 64-bit block. There were 16 rounds in the Blowfish algorithm used to encrypt the data during transmission. Blowfish is a 16-round Feistel structure. This method has undergone substantial research and is now widely accepted as a reliable block cipher. VLSI hardware and software can both benefit from this algorithm's ability to encrypt data. The input is 64-bit data (Chaloop, Abdullah, et al., 2017).

EP1...EP18 are utilized in the opposite order in the Blowfish algorithm decryption process, which is analogous to the encryption process. Using four S-boxes instead of one S-box, Blowfish was able to stop resemblance amongst unlike bytes after bytewise permutation of 32-bit input to the function Fn was utilized. Since just one S-box was utilized in each step, there were four separate outputs, and each of those outputs could not be permuted easily. As far as I can tell, the four S-box architecture appears to be more secure, faster, and easier to develop. To the output of four S-boxes, a simple XOR of the four values is all that is needed. Each round of XOR processes results in an addition to the final result, and all XOR processes finish with an addition (Mahindrakar, 2014).

Key size is a factor in the Blowfish algorithm's processing time. To prevent brute force attacks and provide higher security than currently available encryption solutions, the generation of subkeys is an important step in the encryption process. Blowfish's dependability will be compromised if a high number of weak keys are used. Even though it made use of a 64-bit unit, a larger unit is still preferred (Quilala et al., 2018).

#### **The Two-fish algorithm**

As a symmetric block cipher, Twofish encrypts and decrypts using the same key. The maximum key length that may be used with Twofish is 256 bits, and its block size is 128 bits. (The National Institute of Standards and Technology mandated that the algorithm support key sizes of 128, 192, and 256 bits.) Twofish is fast in hardware and on both 32-bit and 8-bit CPUs (smart cards, embedded circuits, and the like). To be useful, an encryption technique must be both secure and fast. The security and performance of encryption methods need to be stable. Block ciphering with a single 256-bit key is employed by two-fish. Key-dependent S-boxes can be balanced by implementing an encoding and decoding craft, as well as a set-up time and code size. It can withstand both known and unknown threats. (Thirupalu et al., 2020).

Using 256-bit keys, the AES algorithm provides a high level of protection. It can withstand numerous forms of attack. Encryption with AES is faster than with DES. As a result, AES beats DES in terms of speed. This algorithm's crackability has yet to be demonstrated. It's a pretty safe way to encrypt data. Data can be secured against impending dangers, and liesmash assaults, by encrypting it. In comparison to other symmetric algorithms, AES encryption consumes less storage space and functions well without any defects or limits. The system cannot become unresponsive while a large amount of data is being received because the AES encryption method is being used for data transfer. Intruders are less likely to pose as the third party and infiltrate the network if they cannot access the third party. Advanced Encryption Standard (AES) is a faster and more efficient algorithm for encrypting data than existing algorithms (Musliyana et al., 2015).

It is a symmetric algorithm devised by Bruce Schneier. With a key length of 56 bits, it can encrypt blocks of up to 128 bits. The Twofish approach has a sophisticated encryption strategy that involves pre-calculated S-blocks that are also key-dependent. Half of the n-bit keys are used as encryption keys, while the other half is used to alter the algorithm. There are a total of 16 rounds in the Twofish algorithm for any given key size. The following are the features of the Twofish algorithm's design: Keys K0 to K7 are used for XORing the input and output values. These XOR methods are also known as input and output whitening. To construct the F-function, there are five distinct sorts of procedures to choose from: a static left variation by 8 bits, and S-boxes that rely on the input key (Ahmad et al., 2021).

#### **Asymmetric Algorithms**

Asymmetric key encoding is also known as public key encryption because diverse keys are used for message encryption and decryption. Encryption keys, also identified as public keys, could be utilized to encrypt messages. The decryption key, also identified as the secret or private key, is utilized to decrypt the message. Users can detect message authentication when the strength of asymmetric key encryption is merged with a numerical signature. RSA, Diffie-Hellman, and other asymmetric encryption algorithms are examples. (Virtue & Rainey, 2015).

Since it is utilized to produce undisclosed keys, the key schedule algorithm is critical to the progress of encoding and decoding keys because cryptanalysts use this attack to try all conceivable groupings to obtain the unique text, the irrelevant key cohort procedure produces frail keys that are easily brute-forced. For cryptographical procedures, the AES requires key lengths of 128, 192, and 256 bits. The number of rings is 10, 12, and 14 for that key length, and the round keys are extracted from the code key and used in the block cipher construction using the key schedule algorithm. The use of a huge number of rounds in the advance of a fully secure block cipher ensures high diffusion, as does the use of an invertible transformation (Mushtaq et al., 2017).

Claude Shannon projected the following five principles for better ciphers:

1. The degree of secrecy required to cipher a message determines the amount of work. The rate of data tends to deteriorate with time, more computation work is required to guard message confidentiality for years, which may not be protected in the context of the data model.
2. Cryptographic keys and encoding procedures ought to be simple. The encoding procedure should be able to encode any communication with any key and be easy to comprehend.
3. A cipher's operation ought to be as simple as conceivable.
4. Error cohort should be restricted.
5. The ciphertext message's the magnitude and storing space should be limited. Under no settings should the magnitude of the ciphertext exceed the magnitude of the plaintext, regardless of where it was executed.

It is worth noting, from a historical standpoint, that these five principles for a good cipher were projected before the supercomputer age and are still impeccably valid today. The Shannons also present the two values of misperception and dispersal, both of which are critical to the growth of safe encoding and decoding procedures. The code of misperception denotes concealing and thwarting the connection between ciphertext and keys as much as possible (encryption or decryption key). It will help cryptanalysts avoid ciphertext-based secret key prediction. The diffusion principle seeks to hide and complicate the relationship between plaintext and ciphertext. It ensures that even minor changes in the plaintext have an avalanche effect on the ciphertext (Osaghae, 2018). The cipher component, as well as the connection between misperception and dispersal Evaluative restrictions to assess capability and safety, the execution test must pass. Each encryption algorithm has advantages and disadvantages.

#### **The Rivest Shamir Adleman (RSA) algorithm**

Cloud data security has been achieved using a free key cryptosystem. Public key cryptography is the most rapidly evolving, and it has the potential to revolutionize the field. There are two main components to the algorithm: the public key and the private key There is simply one person who has access to both the Public Key and the Private Key, which is the person who originally owned the data (Segar & Vijayaragavan, 2013).

Decryption is performed by the Cloud user and encryption by the Cloud provider. The information can be decrypted using the equivalent Private Key after it has been encoded with the Public Key. The Asymmetric algorithm gets its name from the point that it makes use of two keys in addition to the secret one. The number of inferences determines the framework structure of the RSA algorithm. It's the safest framework out of the big three. The private key cannot be deciphered by an outsider because of the factorization of large numbers. So the RSA algorithm is a viable option for safe data transmission in cloud computing (Johnson, 2020a).

As an asymmetric cryptography method, Ron Rivest, Adi Shamir, and Leonard Adleman established the RSA encryption algorithm in 1977. RSA, like every other asymmetric encryption method, uses two key pairs to encode and decrypt data. When one person encrypts data using a publicly shared key, the other user can decode it using their private key. the practice of encoding and decoding data using the RSA method (Na et al., 2013):

1. Key creation: Choose two distinct figures, U and V.
2. Determine the value of T, which is equal to the product of U and V.
3. Determine the function of Euler's totient by (T) = (U 1) (V 1)
4. Determine the value of e under the following conditions: 1 e (T) and (e, (T))) = 1
5. Determine p, where p = e1 (mod (T)).
6. For encryption, use the following equation: Ke = C (mod T)
7. For decryption, use the following equation: C p = K (mod T) Where C represents the cipher data, K represents the initial data, (e, T) represents the public key, and (p, T) represents the private key.

**ElGamal Encryption**

Improved user authentication and performance in terms of security against attacks are provided by the ElGamal cryptosystem. ElGamal's encryption and decryption algorithms are separated into three distinct components. It takes RSA 2655 milliseconds to encrypt a piece of data, but ElGamal takes 60235 milliseconds. In comparison to ElGamal, RSA decryption takes an average of 72671 milliseconds. Due to this, RSA encrypts faster than ElGamal but takes longer to decrypt. Since key generation is time-consuming, it is tolerated because it will be done regularly. Based on randomization and discrete logarithm problems, ElGamal delivers data secrecy. Keys previously used for symmetric text encryption can now be encrypted using this algorithm, which adds an additional degree of protection (Kim et al., 2012).

#### **Homomorphic Encryptions**

Homomorphic encryption makes it feasible to analyze or change encrypted data without disclosing the data to anybody else while still maintaining the security of the data. As long as the ciphertext is completely homomorphic, any number of addition and multiplication operations can be applied. PHE techniques like RSA and Paillier are not sufficient for cloud computing. PHE can only conduct a finite number of operations. Additive or multiplicative effects are possible with partial HE. Attacks against the RSA cryptosystem can be made by using common modules. Data cannot be adequately protected during processing using conventional encryption techniques (Will & Ko, 2015).

Data security concerns in cloud computing appear to be alleviated by the use of homomorphic Encryption. However, it is currently impossible and difficult to implement both fully and partially homomorphic methods in cloud computing. On encrypted data, the complete homomorphic method took a long time to finish, and partial homomorphic methods are also inefficient because they can only perform calculations on homomorphic encryption-based data security resolutions are unfeasible in the real-life context (Biksham & Vasumathi, 2017).

* + 1. **Deception-based Techniques**

Salting and differential masking are two techniques used to improve password security while reducing the time it takes to hash passwords. Although this strategy is better suited for passwords that take less time to process, brute-force attacks are still possible. However, brute force assaults are not completely protected by the hashing algorithm mentioned above that is employed to increase computation time and produce fake plaintext messages. Countermeasures have been designed to delay, detect, and confound an attacker trying to redirect data from one network to another. Encrypting messages with low-entropy keys like passwords is made easier with the introduction of Honey Encryption (HE). Honey messages are created by decrypting a ciphertext using HE and then decrypting it with any of a slew of erroneous keys. HE has the benefit of providing security even when brute-force attackers are unable to test every possible combination due to a lack of entropy. It provides a level of protection that goes much beyond brute-force limits. Keys with high min-entropy can be protected against partial disclosure using HE as well. Standard HTTPS certificate keys cannot be protected by this new PBE system, despite its higher level of security (Johnson, 2020).

Methods for creating honeywords have been developed to improve hashed passwords. Each user's true password is kept up-to-date in sync with the honeywords they create. An inversion file of hashed or honeyword passwords cannot be used to determine the genuine password. Honeywords can be detected by the auxiliary server or honey checker and an alert will be raised when they cannot be distinguished from the true password during the login procedure because honeyword generation methods generate honeywords that are very similar to the true password, this method may induce users to make typing errors while entering their password (Gadgil, 2016). It works with all user accounts; it has a significant advantage over a comparable technique known as honeypot accounts. Using honeywords as an additional layer of security for existing password scheme is simple and effective.

Honey encryption is also connected with Format Preserving Encryption (FPE) and Format Transforming Encryption (FTE), which preserve the format of the encrypted data. It is possible to encrypt data while maintaining the original format of the input data by using a technique known as "format-preserving encryption," or "FPE." Popularity is growing for honeyword-based approaches due to the benefits they provide over typical password-based ones. Since different communication formats and probability characteristics necessitate different message space outlines, honey encryption has been used in many different contexts. In FPE, the message space for plain text and the message space for ciphertext is the same. FTE message space and ciphertext message space are two different things. Using honey as an encryption key, a plaintext communication is converted to a seed array and stored in the seed space. This means that the ciphertext message space is separate from the cipher message space. For this reason, it is better suited to smaller message spaces rather than larger ones, due to the high processing costs associated with big message spaces. Depending on the application, HE's capacity to secure sensitive private data differs It has been used to protect MANETs from brute-force attacks, and the honey encryption approach has been used to save credit card details and a reduced type of text messaging. There is a large amount of data coming in that is consistent across message spaces. Genetic data is often highly non-uniform in its probability distributions. Honey Encryption (HE), a new theoretical foundation for encryption, is integrated into the GenoGuard system. Confidentiality can be guaranteed in terms of information theory for encrypted data. An unsolved challenge in applying HE approaches to genetic data sets has been addressed by GenoGuard, which focuses on genetic data. A brute-force attack can be used in GenoGuard to try to guess passwords and decrypt them. (Kapil et al., 2020a).

The enlargement of the Graphics Processing Unit (GPU) allows the attacker to crack hashed password data. As a result, researchers are looking into a novel way to store both fake and actual passwords in the database (Manjula et al., 2019).

An attacker either chooses between risking log-in, the potential detection of compromised password-hash files and not attempting to log in at all when utilizing a honey checker. A honey checker is required to gain access to the system, even after having the password file and the hash code. This technique can withstand attacks while simultaneously addressing the storage overhead problem. The honey circular list methodology was examined to ameliorate the current difficulty of honeyword production because of this; a hashing passwords database has been established. The attacker can’t estimate the distance of this password, hence he is unable to log into the system. An algorithm that uses a honey circle list can solve the storage problem that has hampered prior techniques (Juels & Rivest, 2013).

A partial solution to the problem of honeyword production methods can be found with this method. Distribution Transformation Encoder (DTE) is a new encoding and decoding system for Honey Encryption, but it has some limits for assigning plaintext messages

Plaintext messages cannot be inserted into the seed space by the DTE in that system because of this constraint. Distribution transform encoder (DTE) is used in honey encryption to transform the seed space into the message space M. In a wide range of practical situations, HE automatically enhances security. An attacker can decrypt encrypted data using brute force to make false meaningful messages. The use of the DTE process's cumulative distribution function (CDF) solves the problem of message space constraint. Due to the known weaknesses, the security of DES cannot be assured (Juels & Ristenpart, 2014).

The brute force computation of hash values is one of the most recent challenges to password-based authentication. Honeyword-based authentication systems can effectively prevent this danger by making password breaking noticeable. Present systems have a lot of drawbacks, such as several system vulnerabilities and weak DDoS resistance as well as a large storage requirement. Paired Distance Protocol (PDP) was proposed as a novel method for generating honeywords that addresses virtually every flaw in the existing methods. PDP not only has a detection level of 97.23 percent, but it similarly greatly saves loading expenses, according to the full analysis (Chakraborty & Mondal, 2015). Based on the unpredictability of the RS, PDP provides the best possible level of security. Co-relational hazard, DoS resistance, and a Typo safety risk are some of its drawbacks.

Doubts concerning the safety of cloud data have been raised in light of the recent security issues associated with the use of public cloud data storage. Traditional password-based cloud data protection solutions are ineffective in defending against password guessing and password cracking assaults. When it comes to protecting encrypted data, extended honey encryption (XHE) is preferred. Any attempt to decrypt these files by using a wrong password is met with a flood of indistinguishable false data that the attacker will never be able to distinguish from the original data. As a result, password guessing and cracking attempts become increasingly difficult. Homomorphic Encryption (HE) and Attribute-Based Encryption (ABE) are two recent breakthroughs in cryptography that can be incorporated into the XHE scheme (Tang et al., 2016).

* + 1. **Evaluation metrics for Encryption-Decryption Algorithms**

Firstly, there must be an evaluation of the performance parameters of a secure encryption scheme before applying it for practical real-life use. Al-janabi, (2007), identified some performance metrics for encryption-decryption algorithms:

* 1. Encryption period: The time required to change plaintext to ciphertext is discussed as the encoding period. The time to encode is limited to milliseconds and is governed by the message block and key sizes. It has a straight influence on the encryption algorithm's performance. Making the encoding system receptive and profligate, each cryptologic procedure required the least encoding period.
  2. Decoding period: To make a cryptologic procedure profligate and receptive, the decoding period ought to be shorter than the encoding period, and it ought to also be restrained in milliseconds.
  3. Memory utilized: Various techniques have an impact on how much memory is consumed. The key size, loading vectors, and type of operation all factor into how much memory is used. Modest memory size is ideal since it cuts the whole cost of the system.
  4. Throughput: This is considered by distributing the entire block size (MegaByte) encoded by the entire encoding period. When the throughput value is increased, the algorithm's power consumption decreases.
  5. Avalanche effect: It determines that if the plaintext changes, the ciphertext changes suggestively. It calculates the difference between plaintext and ciphertext variations. The hamming distance calculates the avalanche effect. If a high grade of diffusion is vital, a high avalanche outcome is preferable. It is evaluated by distributing the humming distance by the file size and replicates the performance of cryptologic procedures. Avalanche = x100 (Overall amount of bits – the number of flip bits)
  6. Entropy: The random matrix method is used to estimate the overall strength of the algorithm's implementation. Entropy is a statistical measure that is utilized to assess the unpredictability and indecision of data. With increasing randomness, the association between the ciphertext and key turns out to be more multifaceted. Encryption algorithms necessitate a high level of randomness to encrypt the plaintext, resulting in little or no dependence between the ciphertext and key. This is known as the misperception property. A high level of ambiguity is required because it makes it demanding for an assailant to guess the total set of information. This valuation restriction specifies the transmission bandwidth. Encrypted characters or bits encoded with fewer bits require less storage and bandwidth. It likewise affects the system's cost.

A substantial number of investigations are available on the use of encryption-based procedures for safeguarding private information on the cloud. The majority of contexts with encoding employ password-based encryption (PBE). This type of system is vulnerable to brute force guess outbreaks. PBE is a renowned technique for generating strong cryptographic keys (Kapil et al., 2020b). PBE methods are vulnerable to brute-force occurrences. It is not a secure data storage system and is vulnerable to attacks (Dworak & Boryczka, 2021).

An investigation has been conducted on the usage of a secret key in typical encryption and decryption methods. In the world of standard-issue symmetric encryption, the DES algorithm is the go-to block cipher. The ciphertext is subject to brute force attacks since it is made up of eight blocks of the cipher. The complexity of executing cryptanalysis on the ciphertext is increased by using multiple rounds of distributiTor to provide maximum security, a tiny key is used together to encode and decode. When it comes to proteusers’user's data, Triple DES outperforms the DES algorithm, which encrypts and decrypts files and documents. The discovery of triple-DES came at a time when DES was proving too weak for most users. For this reason, they needed a quick and simple method of gaining strength. If a system relies on DES, combining several DES into a composite function is more likely to be accessible and eliminates the conflict that would arise if a new cipher was introduced that was superior to DES. Triple DES is far safer than any other cryptographic technique, including single DES. Uses include cloud security and information security (Abdul et al., 2009).

#### **Key Management: The Diffie Hellman algorithm**

The Diffie-Hellman key exchange is still widely used in a wide variety of modern security protocols. It enables an insecure key to be securely established between two parties who have never met before so that their conversations can be encrypted (Sen, 2015).

The information exchanged is stored in a hash function. Authenticated key exchange (AKE) protocols are explained to create a group of participants with the same secret key, which is then utilized to ensure the integrity of multicast message transmissions." To safely exchange cryptographic keys over a public network, this technique is employed. To ensure that only the data owner has access to the information, the Diffie Hellman protocol is used. A triple-encryption system can be recommended for data security. It seems that homomorphic encryption is a fairly effective method, although additional study and attention are needed. It aims to alleviate the data owner's concerns about the cloud provider's ownership of his data (Bresson et al., 2001).

The Diffie Hellman-DSA Key Exchange technique has been suggested to reduce the vulnerability to password-guessing attacks. A method known as Diffie-Hellman permits two people to safely create a secret key without exchanging the secret key. However, a man-in-the-middle attack can compromise this protocol. The Diffie-Hellman key exchange can be authenticated using the Digital Signature Algorithm (DSA). Initially, a set of encryption parameters is chosen for the system as a whole, and then a single user's private and public keys are generated (Vollala et al., 2021).

The private key is utilized by the source to discover the signature, while the public key is used by the recipient to verify and authenticate it. In the DSA algorithm, the key size is 3072 bits. Only one key can be used in the original DSA algorithm, which has security constraints of its own. As a result, to increase the key's difficulty in decoding, more than one key is used. Biometric information provides a non-repudiation approach that overcomes the constraints of password-based authentication, like the inclination of users to use an easy password. The cloud system's security can be protected by this system's safeguards (Zhang et al., 2019).

* 1. Review of Methodological Approaches

#### **Advanced Encryption Standard (AES)**

AES is a cipher that has been around since 1988. There are three distinct key sizes for the Advanced Encryption System: 128-bit, 256-bit, and 128-bit (128,192 or 256 bits). It is possible to partition a 128-bit data block into four operational blocks. A 44-by-44 matrix known as the state is used to store these pieces of data (Rahimunnisa et al., 2013) The four operations are required for each cycle of encryption. The following are the steps to take:

1. Substitute Bytes: A Substitution Box is used in this operation to complete a byte-by-byte replacement of the block. To substitute a byte, the bytes are interpreted as two hexadecimal digits.
2. Shift Rows: There is no change in the bytes in the first row of the state during this operation. The 2nd, 3rd, and 4th rows cyclically move to the left by 1, 2, and 3 bytes, respectively.
3. Mix Columns: The bytes in each column are mixed in this operation by multiplying the state with a fixed polynomial matrix, and the new value of the columns is placed.
4. Add Round Key: By performing various operations on the cipher key, a round key is formed, and each byte of the state is merged with the round key via bitwise XOR.

Inverse functions like Inverse Substitution Bytes, Inverse Shift Rows, and Inverse Mix Columns are used in decryption. Both encryption and decryption rely on the Add Round Key stage. There are Nr-1 rounds of Add Round Key stage output before getting to the final round, which is determined by the key length. During each round, all four phases are executed (Abikoye et al., 2019).

Using a secret key created at random, a message is encrypted and then decoded using a one-time pad and key that match. This is a cryptographic technique. Because the key is double the length of the plaintext, this stream encryption extends the Vigenère cipher. One-time pad key encryption and decryption include an XOR operation using the following formulae.

The letters P, K, and C stand for plaintext, key text, and ciphertext, respectively. The advantage of using a random key to encrypt messages is that the algorithm cannot be broken by examining a series of messages. One-time pad cryptography has a major benefit because it generates random keys. Ciphertext is generated when a non-random message is paired with a random time pad key, creating a completely random sequence. One-time pad cryptography cannot be used if the plaintext length exceeds the key length, because the key for this cryptography must be the same length as the plaintext in order for it to function (Jantan et al., 2018).

For AES, a substitution–permutation network is frequently employed. Some of the procedures entail replacing defined outputs for specified inputs (substitution), while others just move bits around (permutations). Bytes, not bits, are used in all AES calculations. In other words, the 128bits of plaintext are broken down to 16 bytes, which are then arranged in a matrix. The length of the key influences how many bands are used in AES. 10 bands for 128-bit keys, 12 bands for 192-bit keys, and 14 bands for 256-bit keys are used in this algorithm (Sousi et al., 2020).

#### **Hybrid Cubes Encryption Algorithm (HiSea)**

Since the encryption / decryption passwords, data to be encrypted, data to be decrypted, and internal operations are all integer values, the cipher is classified as a symmetrical non-binary block cipher. The HiSea encryption technique was developed by Sapiee Jamel in 2011. The plaintext size is 64 bytes of decimal ASCII characters for encryption purposes. Inner matrix multiplication of layers between Magic Cubes (MC) results in the Hybrid Cube (HC) (MC). Algorithm designers can utilize any grouping of HC layer items to make their algorithms more complicated. This is the overall concept of the HiSea, which uses an order 4-matrix to organize the plaintext, keys, and ciphertext during encryption (Chitrakar et al., 2020).

* + 1. **Vigenère Algorithm**

This encryption employs a sequence of distinct Caesar ciphers for alphabetic texts, which is then decrypted. Polyalphabetic substitution is an easy way to replace each alphabet with a different cipher alphabet. For centuries, the Vigenère cipher was thought to be secure, but its flaw was later discovered (Kadry & Smaili, 2011). With Friedrich Kasiski's invention, the period, key, and plaintext could all be determined with relative ease. As the name suggests, the Vigenère encryption is designed to hide the frequency of letters in the plaintext. However, the recursive nature of the Vigenère cipher's key is a serious problem. An expert cryptanalyst can quickly decipher any encrypted text by guessing the key length. The key length can be found using several tests, such as the Kasiski and Friedman ones. As a result of the advent of computers, the Vigenère cipher has become even more straightforward to decrypt. Even with long keys, most ciphertexts can be deciphered in a few seconds. This encryption, according to modern standards, is easily deciphered and provides little security. In addition to the Advanced Encryption Standard, it is employed in a wide variety of more strong encryption methods (AES). This approach can be seen as an algebraic formula, where letters A–Z represent 0–25, and the Vigenère encryption formula is;

### This equation is equivalent to (Pi+Ki). modem = Ci in this context.

### The ciphertext character C is represented by the letter C in this example.

### P = Plain text character

### There are 26 alphabet characters in the Vigenère cipher, and each of these characters is known as a key phrase character (K m) (Zaid & Mohd, 2007).

* 1. Gap Identified in the Literature

An encryption algorithm was created by (AbdElminaam et al., 2014) for the purpose of comparing the power consumption of the new protocols with that of four present hybrid protocols. Different parameters, such as data block sizes and encryption/decryption speeds, were used to compare the various methods. Experimental results indicate the utility of each strategy.

Kumar and Rana, (2016), data Security with a Modified AES Algorithm As a result, text protection in digital media is required for online banking, account passwords, email passwords, and other sensitive information. This research examines data security and compression using advanced encryption standards (AES). The number of rounds (Nr) for the encryption and decryption processes of the AES algorithm was increased to 16 in the study, which improved the system's security. Theoretical research and experimental results showed that this AES strategy provides great speed as well as less data transfer over unprotected connections.

According to the AES-Elliptic Curve Cryptograpphy (ECC) Model for Data Security in Cloud Storage Computing proposed by Rehman et al., (2021), users' private information is vulnerable to being revealed, leaked, or lost in a number of different scenarios. Several techniques have been created to uphold the safekeeping and integrity of data using cryptographic methods such as Elliptic Curve Cryptography. Using this technology, data can be securely transferred over the cloud while maintaining its integrity and security. The system uses the ECC and AES algorithms to ensure data integrity and authentication. The results show that the proposed strategy is more efficient and gives superior outcomes when equated to existing methods.

Chaloop and Abdullah, (2021), A hybrid security paradigm based on AES and RSA algorithms was presented. Every minute, the enhanced internet, networking companies, health data, and cloud apps have considerably expanded data. The current research focuses on using cryptography to protect sensitive data that is transmitted between individuals, businesses, organizations, cloud applications, and others across networks. Data must first be encrypted using a cryptographic technique before being delivered from the sender to the network recipient. Second, the recipient uses the decryption process to show the original data. They provided AES, RSA, and mix algorithms, compared their effectiveness using time analysis. The hybrid algorithm performs better in terms of security, according to the results of the testing.

Ahamed and Krishnamoorthy, (2020), proposed encryption and decryption of SMS Since the former method utilized 26 10 matrices and the key consisted of letters, the Vigenère cipher can be broken effortlessly using the Modified Vigenère Cipher Algorithm. To improve security, this study developed a modified Vigenère cipher algorithm. Existing Vigenère cipher attacks are insecure, thus, a method that uses Rivest-Shamir-Adleman was devised to improve Vigenère cipher security (RSA). The RSA algorithm is asymmetric, with public and private keys. The advantage of employing this approach is that it is harder to discover factors for huge composite numbers. This procedure is safe, but it takes time. RSA was merged with the Vigenère encryption in the proposed technique such that the proposed algorithm is secure and takes less time than the RSA algorithm. This method is extensively used to encrypt SMS messages.

For the purpose of protecting data transmission over public and private networks, (He et al., 2021) published a security study of data encryption techniques. This study found that the security index for networked communications could be raised by 25% by employing a link encryption method, by 35% by employing a node encryption technique, and by 40% by employing an end-to-end encryption algorithm. The RSA and DES algorithms are two of the more well-known ones, and they are both employed in many different kinds of encryption. Data links in a network typically employ an encryption algorithm, either a link encryption algorithm, a node encryption algorithm, or an end-to-end encryption algorithm.

Al-Shambri and Al-Alassery (2021). Securing Fog Computing for E-Learning System: Integration of Two Encryption Algorithms provides a thorough analysis of the strengths and weaknesses of the RSA, AES, and ECC algorithms proposed for fog-enabled cybersecurity systems. These algorithms are compared in terms of their encryption and decryption speeds, key generation techniques, and other characteristics offered by software written in Python. The communication needs between the fog and the e-learning system are met by a hybrid cryptography system that combines RSA and AES encryption to optimize safety, data size, and response time. By creating a testbed for an e-learning website scenario with ASP.net and C#, it was possible to show the benefits and drawbacks of Integrated Encryption Schemes.

It was suggested that a remark be written about the RSA and ElGamal encryption algorithms' time and space complexity by Emmanuel et al., (2021). In the design and development of high-speed computing devices, understanding algorithm computational complexity is crucial. The efficiency of algorithms influences the entire essence of computation; this is especially true for algorithms with exponentially growing solution spaces. Cryptographic algorithms are a good example of this type of algorithm. Using past research, the purpose of this study is to compare the computing speeds of the RSA and ElGamal cryptographic algorithms.

The purpose of this research was to compare the results of prior experiments to see if there was a clear winner between the RSA and ElGamal algorithms. It is hoped that this study will motivate additional research into the dynamics of cryptographic systems, with the ultimate goal of quantifying their complexity and assessing their influence on theoretical computer science. Experimental results from many of the reviewed papers favor RSA for encrypting text, images, and audio data due to its lower energy consumption, shorter encryption time, and smaller key size. However, ElGamal has been shown to be superior for decrypting the same data due to its faster decryption time.

# CHAPTER THREE

## METHODOLOGY

* 1. **Research Design**

The process flow for this study is depicted in figure 3.1.

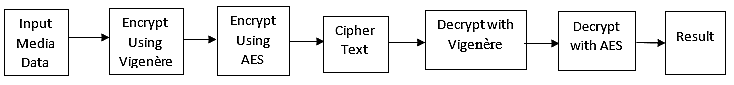


Figure 3.1. Research Design

The data is taken as an input in the form of a Numerical text or image with any key size of 32 or 64 bytes in this study. The dataset used in this study were acquired from publicly available text and image data from Kaggle (url) and UCI repositories (url). The key is sent through the sender in two phases for system execution and operation, with the first phase using the Vigenère Cipher and then the AES cipher to form the Vigenère-AES cipher. After encryption with a scrambled Numerical text or image, the prescribed encrypted cipher is used. The Vigenère-AES cipher is then decrypted in and inverse form, which is AES-Vigenère, with the results as an output in the form of the original Numerical text or image. This prevents intruders and detectors from committing any assaults or attacks on the system, as well as stealing information. Figure 3.2 (a) and 3.2 (b) presents the flowchart of the encryption and decryption phase of the Vigenère-AES model.

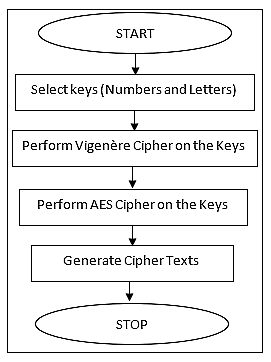
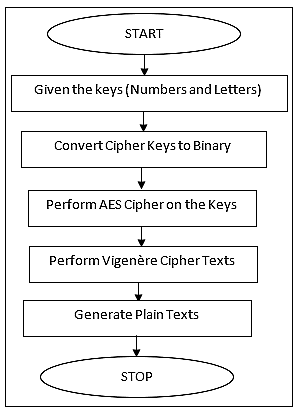
 

Fig3.2 (a) Vigenère-AES Encryption Process

Fig3.2 (b) Vigenère-AES Decryption Process

The Vigenère-AES model is a milti-level model comprising the Vigenère and the AES ciphers. The model encrypts data first through the Vigenère and then passes the enciphered file to the AES for further encryption. The decryption process is a reverse of the process involving AES first, then Vigenère. Algorithm 3.1 and algorithm 3.2 present the Vigenère and AES algorithms respectively.

|  |
| --- |
| Algorithm 3.1: Vigenère Encryption Algorithm |
| Input: plainText = raw\_input  Image = Img |
| Output: Cipher |
| # Creates the base Alphabet which is used for finding preceeding characters from the ciphertext.  baseAlphabet = ('a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z')  plainText = raw\_input("call from file")  key = raw\_input("Please enter the key word")  keyList = []  keyLength = 0  while keyLength < len(plainText):  Adds the users entered key into a list character by character.  key = len(plaintext)  for char in key:  if keyLength < len(plainText):  keyList.append(str(char))  keyLength = keyLength + 1  cipherCharIndexValue = 0  keyIncrement = 0  for plainTextChar in plainText:  cipherCharIndexValue = baseAlphabet.index(keyList[keyIncrement]) + baseAlphabet.index(plainTextChar)  while cipherCharIndexValue > 25:  cipherCharIndexValue = cipherCharIndexValue - 26    completeCipherText.append(baseAlphabet[cipherCharIndexValue])  keyIncrement = keyIncrement + 1  print ''.join(completeCipherText) # Makes the result a strings for printing to the console |

|  |
| --- |
| Algorithm 3.2: AES Algorithm |
| Input: Vigenère Cipher |
| Output: Cipher |
| # AES 256 encryption/decryption using pycrypto library  import base64  import hashlib  from Crypto.Cipher import AES  from Crypto import Random  BLOCK\_SIZE = 16  pad = lambda s: s + (BLOCK\_SIZE - len(s) % BLOCK\_SIZE) \* chr(BLOCK\_SIZE - len(s) % BLOCK\_SIZE)  unpad = lambda s: s[:-ord(s[len(s) - 1:])]  password = input("Enter encryption password: ")  def encrypt(raw, password):  private\_key = hashlib.sha256(password.encode("utf-8")).digest()  raw = pad(raw)  iv = Random.new().read(AES.block\_size)  cipher = AES.new(private\_key, AES.MODE\_CBC, iv)  return base64.b64encode(iv + cipher.encrypt(raw))  def decrypt(enc, password):  private\_key = hashlib.sha256(password.encode("utf-8")).digest()  enc = base64.b64decode(enc)  iv = enc[:16]  cipher = AES.new(private\_key, AES.MODE\_CBC, iv)  return unpad(cipher.decrypt(enc[16:]))  encrypted = encrypt("This is a secret message", password)  print(encrypted)  decrypted = decrypt(encrypted, password)  print(bytes.decode(decrypted)) |

### **Research Instruments**

The developed Vigenère-AES model was implemented on Windows platform using MATLAB 9.7 R2019b version.

### **System Requirements**

The developed model was implemented using the MATLAB programming language running on Intel Pentium Core i5 2.4GHz, Windows 10 (64bit). The software and hardware requirements for the implementation of the model are presented as follows:

### **Hardware Requirements**

The hardware configuration needed for the implementation of the system is as follows:

* An Intel Pentium Core i3 CPU or its equivalence
* A minimum of 2.0 GHz processor speed
* A minimum of 4 GB of Random Access Memory

### **Software Requirements**

The hardware configuration needed for the implementation of the system is as follows:

* Windows 8 Operating system or its equivalence
* MATLAB 8.5 R2015aSP1 and above

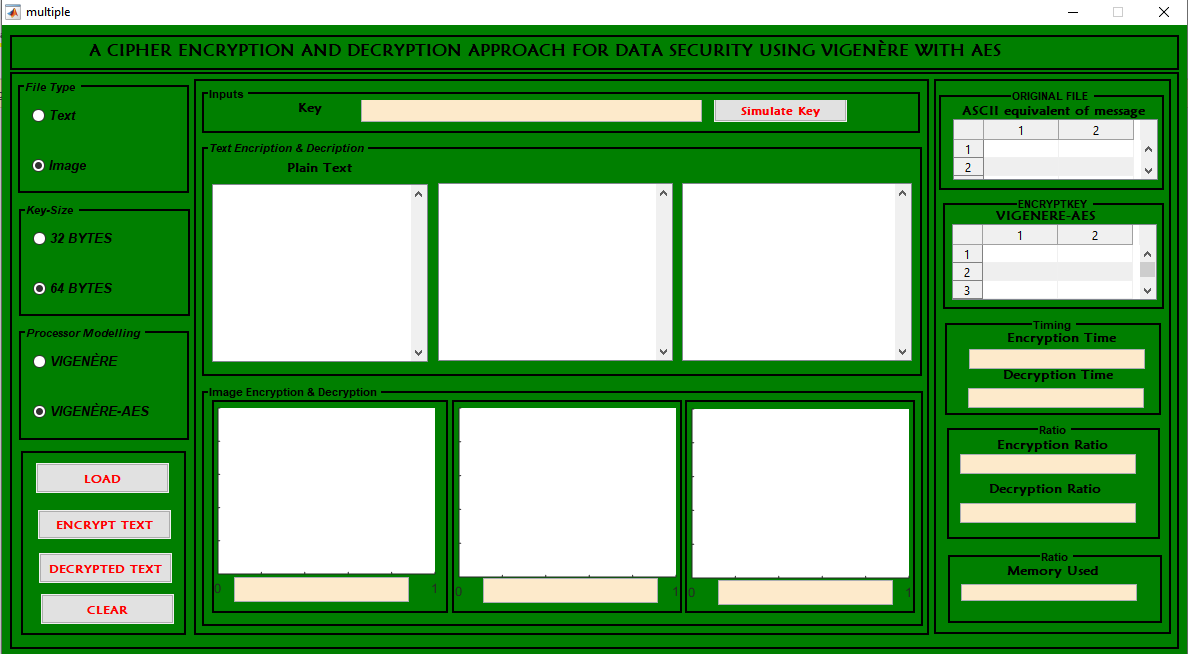
# CHAPTER FOUR

## RESULTS AND DISCUSSIONS

* 1. **Results**

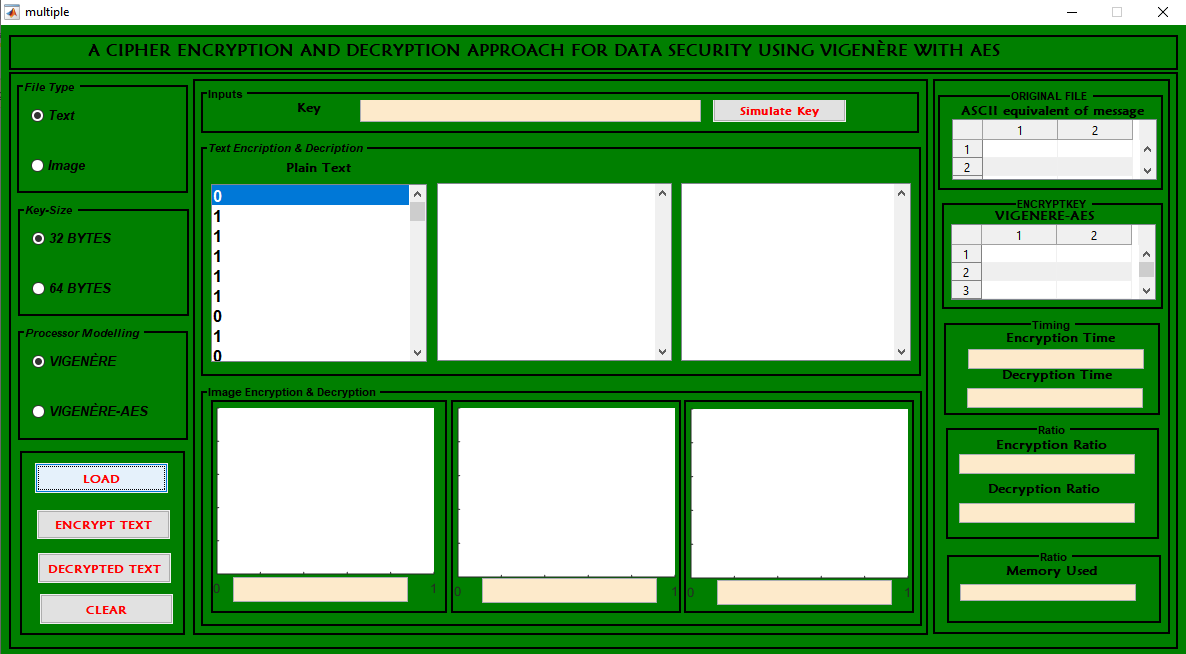
This study shows a technique that combines plain text and key phrase characters and replaces them with several cipher characters due to multiple tables, whereas, in the traditional Vigenère technique, each combination has exactly one value. To define the length of the key, the approach shows a technique that is much more resistant to attack. The outcome used these algorithms to find factors for large composite numbers, as well as a method that is more secure and less time consuming. To accomplish this, the MATLAB application environment was used to develop the model and compare the performance evaluation of the developed approach action on the basis of time and memory using the same key (Private key). This chapter demonstrates the step-by-step processes used to get the results and compares them to other existing models and approaches. Figure 4.1 depicts the graphical user interface of the developed model. in the environment, it gives users the privilege to select and load texts or images, generate simulated keys, select a key size of either 32 or 64 bytes, select either of Vigenère or Vigenère-AES approaches, encrypting and decrypting messages.

To generate ciphertext, the bytes of text and images are taken and then transposed and encrypted. Data for this project will be ciphered solely utilizing byte values. This technique does not alter the image's values. It's necessary to move the numerical values around and encrypt them to get the ciphertext. This implies that the total size of the image and text does not vary during the encryption and decryption procedure. In the course of encoding and decrypting, the image's properties are kept untouched.



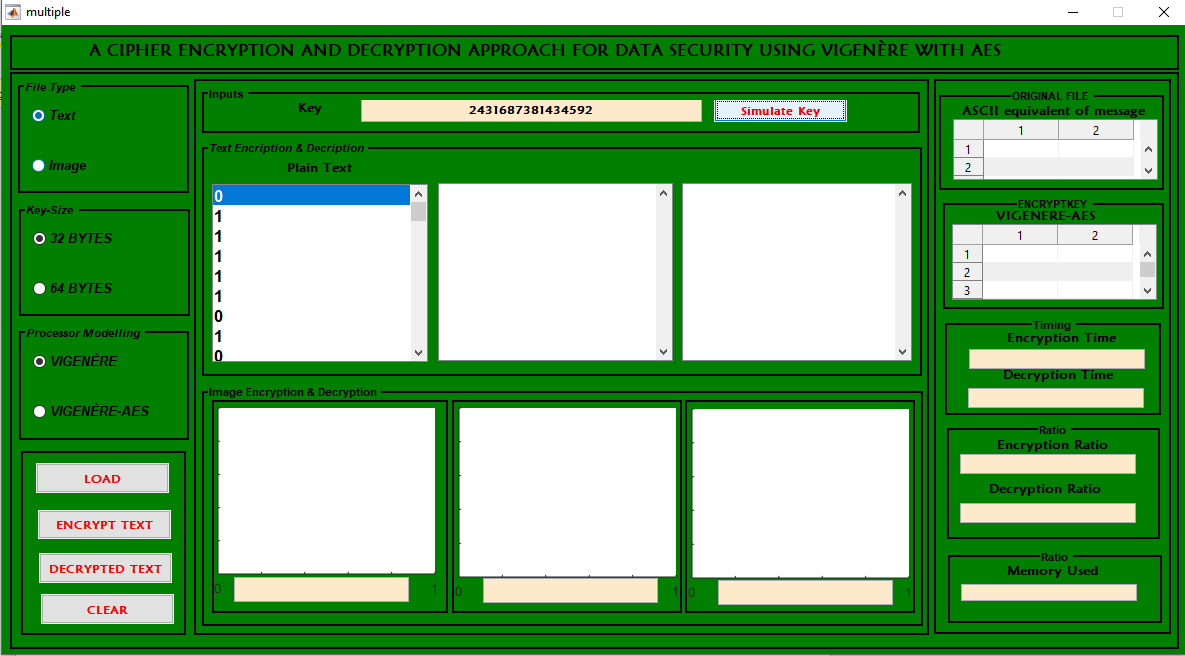
**Figure 4.1: The Environment for the Developed Approach**

In this model, users can load numerical text data, figure 4.2 shows loaded numerical text data, using 32 bytes key size, the loaded data is displayed in plain text. The numerical data is used to depict messages required to be encrypted by a sender and scrambled using either Vigenère or Vigenère-AES cipher and decrypted by the receiver that has the key.



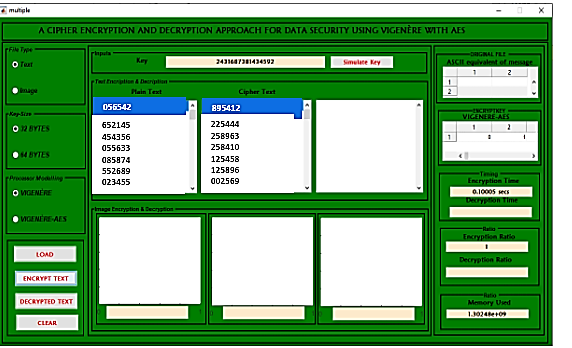
**Figure 4.2: Loaded Text Data with 32 Bytes**

This model generates random simulated keys for senders to encrypt information available to be sent, such as numerical texts and images. The keys are generated in form of numbers and tokens comprising 16 digits, making it difficult for intruders to guess, unless it is obtained from the sender. Figure 4.3 shows the simulated key that was randomly generated as a secret key for the sender and the receiver.

****

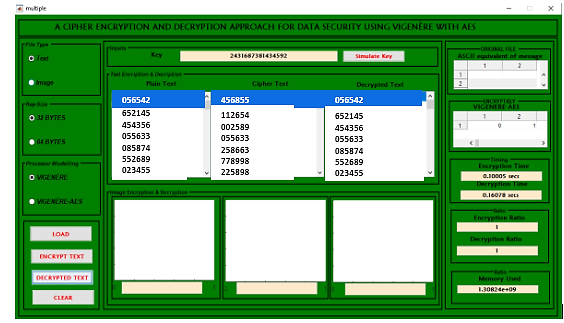
**Figure 4.3: Randomly Generated Simulated Key**

This model generates a scrambled ciphertext, having generated a simulated key for the plain message available by the sender to the receiver. Figure 4.4 shows the ciphertext available for intruders as a message, rather than the original text.



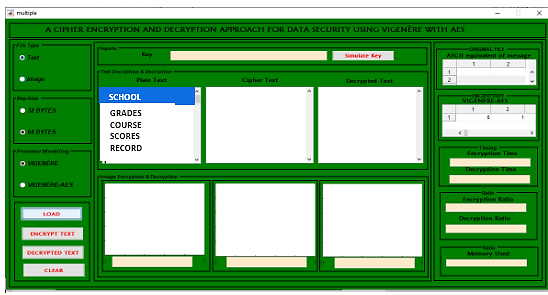
**Figure 4.4: Generated Ciphertext**

The model decrypts the ciphertext using the sender’s simulated key, which displays the encrypted message. Figure 4.5 shows the decrypted message using 32 bytes key size. It takes 0.10005 Secs as an encrypted time, 0.16078 Secs decrypted time and 1.30824 x 109 KB Memory used.



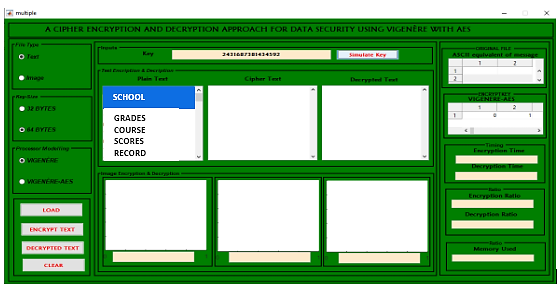
**Figure 4.5: Decrypted Text**

This model uses a 64 Byte key size to also send encrypted information, figure 4.6 shows the encrypted plain text.

****

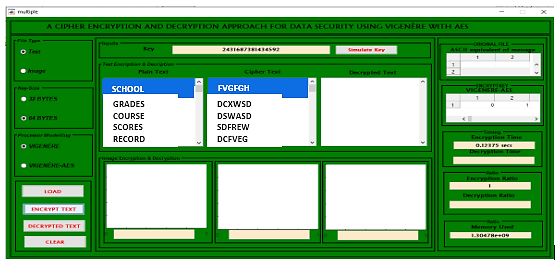
**Figure 4.6: Display of Plain text to be Encrypted using 64 Byte Key Size**

The 64 Byte key size also requires the sender to generate a simulated key as a token and a secret 16-digit key for the message ready to be encrypted by the sender to the receiver. Figure 4.7 shows the generated simulated key for the message.



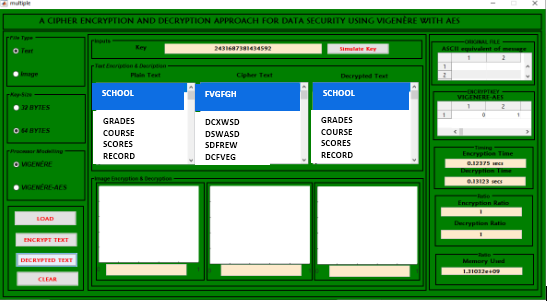
**Figure 4.7: Generated Simulated Key for the 64 Bytes Key Size**

In this model the message with its simulated generated key is encrypted and the message is being scrambled as a ciphertext to intruders. Figure 4.8 shows the ciphertext.



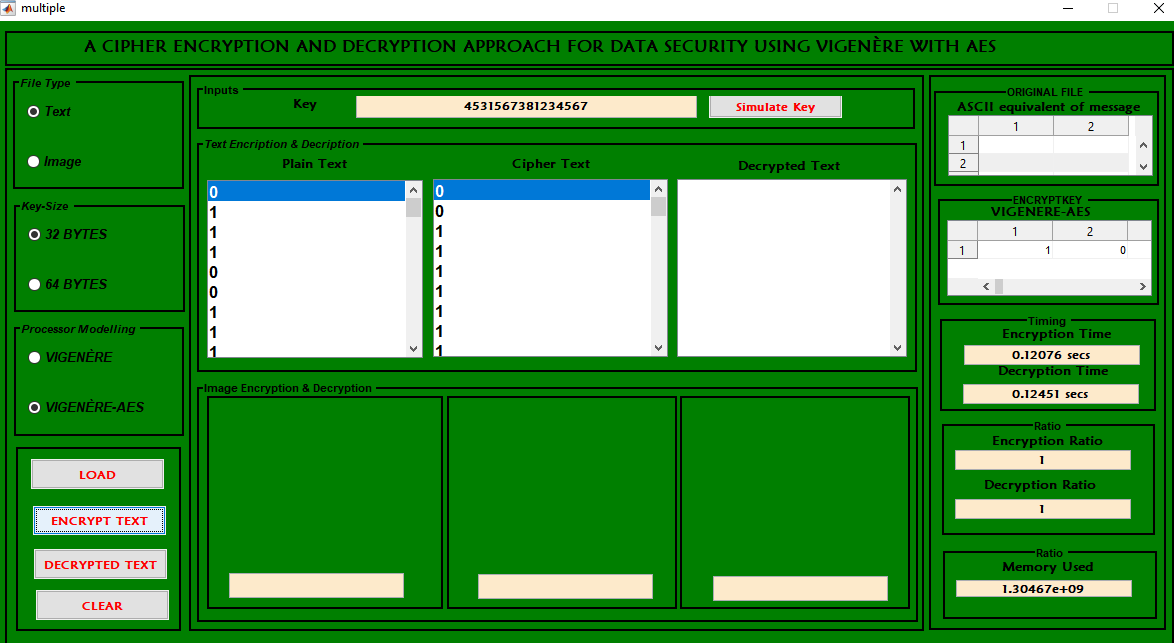
**Figure 4.8: Ciphertext for 64 Bytes Key Size Message**

Ciphered text is decrypted by receivers using the simulated generated key. Figure 4.9 shows the decrypted message for the 64 Bytes key size message. The obtained results for the 64 Bytes key size encrypted text show 0.12375 Secs Encryption Time, 0.13123 Secs Decryption Time and 1.31032 x 109 Memory Used using Vigenère.



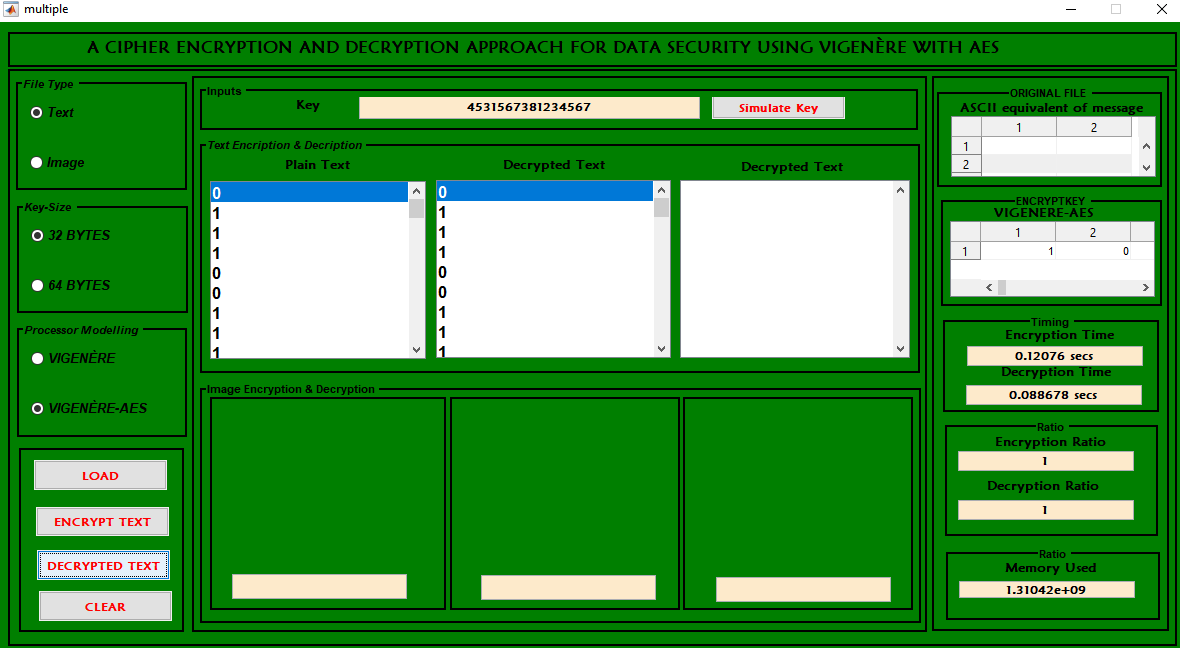
**Figure 4.9: Decrypted Message Using 64 Bytes Key Size**

Using Vigenère-AES this model shows the obtained results for 32 bytes and 64 bytes key sizes in figures 4.10,4.11, and 4.12 for encrypting and decrypting plain text data.

****

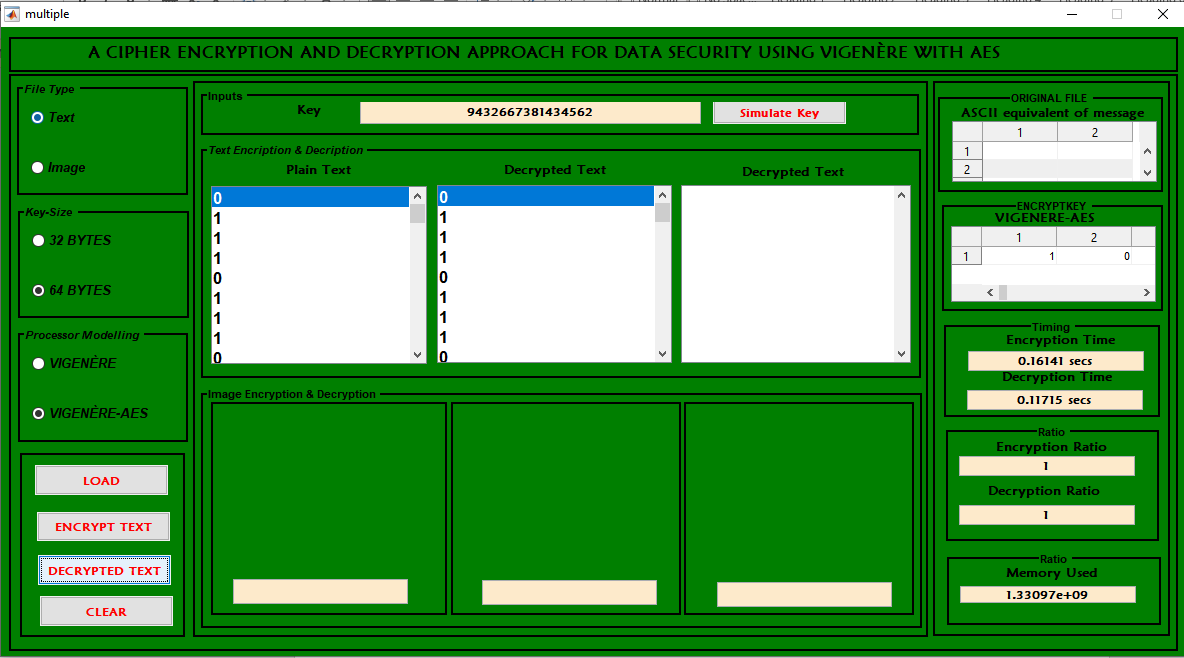
**Figure 4.10: Vigenère-AES Encryption using 32-Bytes Key-size**

In this model, with Vigenère-AES using 32-byte key-size obtained 0.12076 Secs encryption time, 0.088678 Secs decryption time and 1.31042 x 109 memory used as shown in Figure 4.10.

****

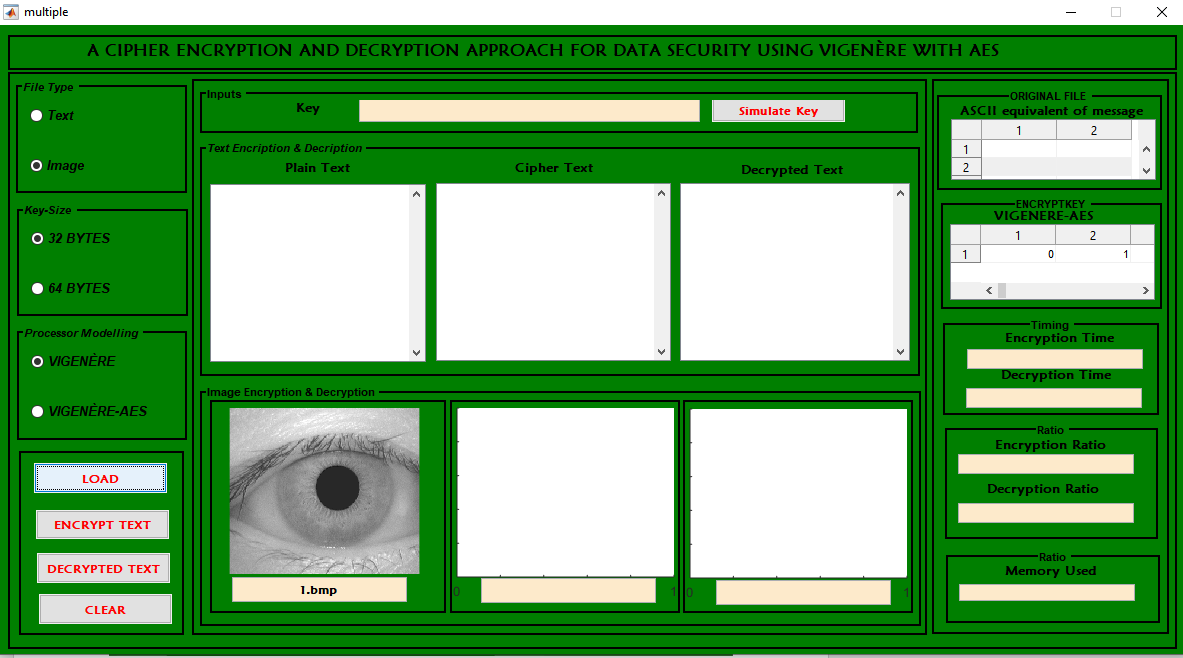
**Figure 4.11: Results of Vigenère-AES using 32 Byte Key-Size**

Figure 4.12 shows the result obtained using Vigenère-AES with 64 Bytes key-size. The results show that Vigenère-AES with 64 bytes key sized obtained 0.16141 Secs Encryption time, 0.11715 Secs decryption time and 1.33097 x 109 memory used.

****

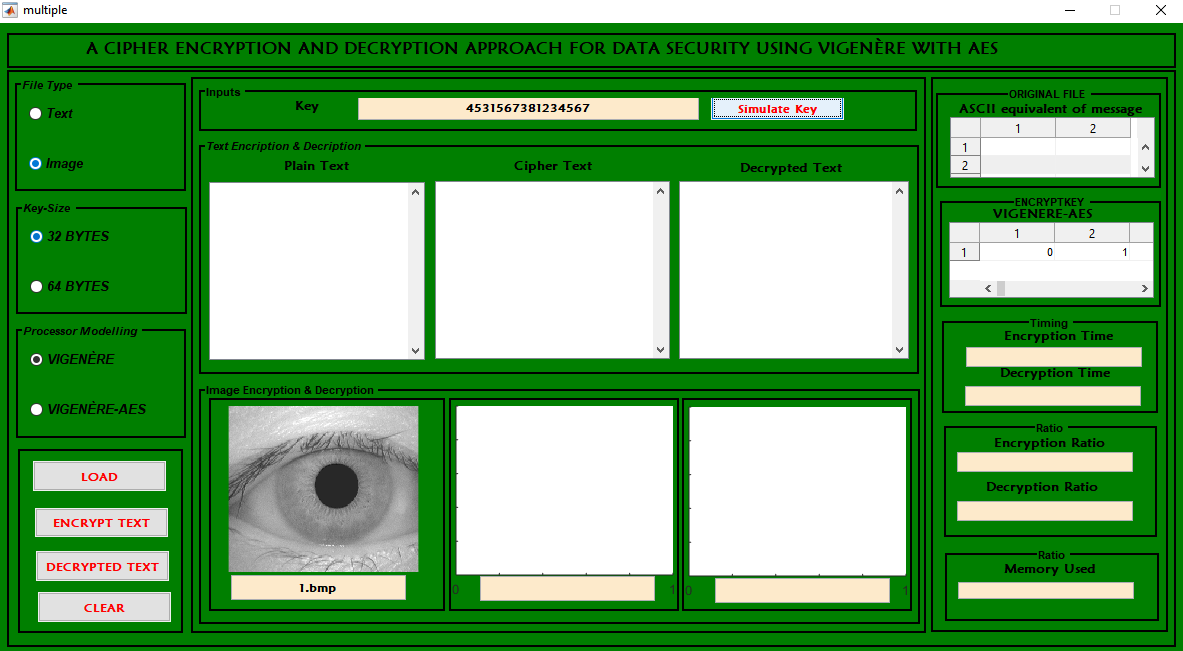
**Figure 4.12: Results of Vigenère-AES using 64 Bytes Key-size**

In this model, Images are also encrypted using iris as a dataset, they are considered as secured information for an organization, that needs to be secured and not to be shared or obtained by a third party. The images are loaded using 32 Bytes key size. Figure 4.13 shows the loaded image using the 32 bytes key size.



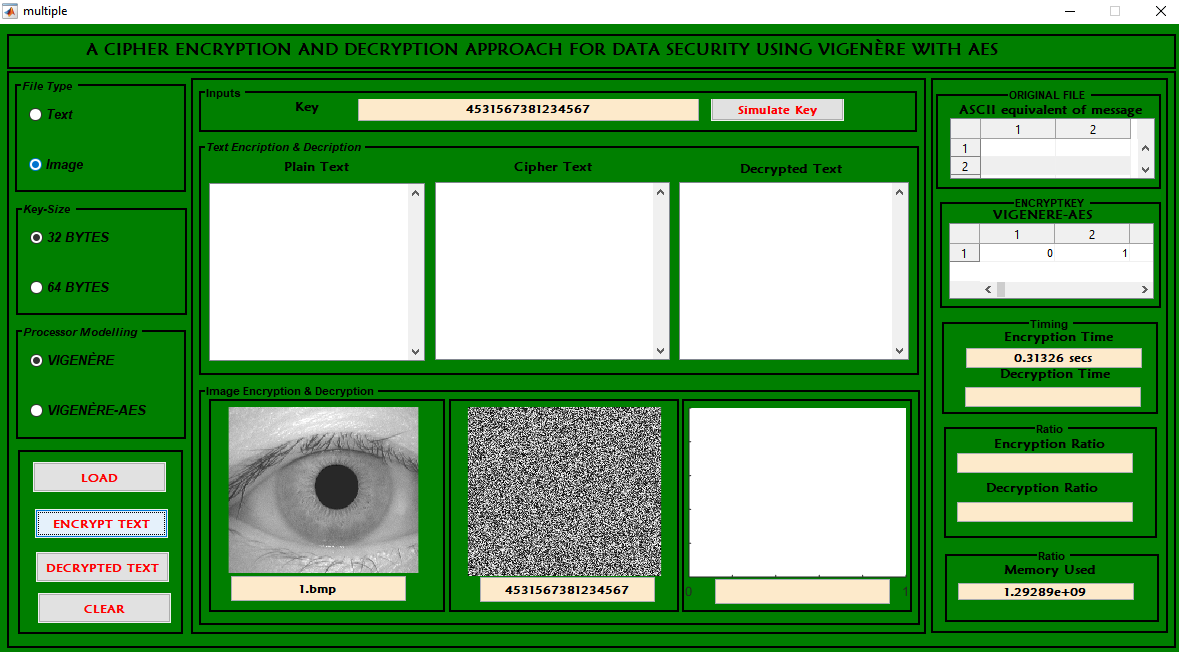
**Figure 4.13: Loaded Image Data Using 32 Byte Key Size**

The loaded image data also requires a generated simulated key, figure 4.14 shows the generated token comprising 16 digits.

****

**Figure 4.14: Generated Simulated Key for the Loaded Image**

The loaded image is encrypted by the sender that has the generated simulated key, the image data is secured from an intruder by having the image being scrambled, and can only be viewed by the receiver. Figure 4.15 shows the scrambled image. The results obtained using Vigenère for images obtained 0.31326 Secs for encryption time with 1.29289 x 109 memory used.

****

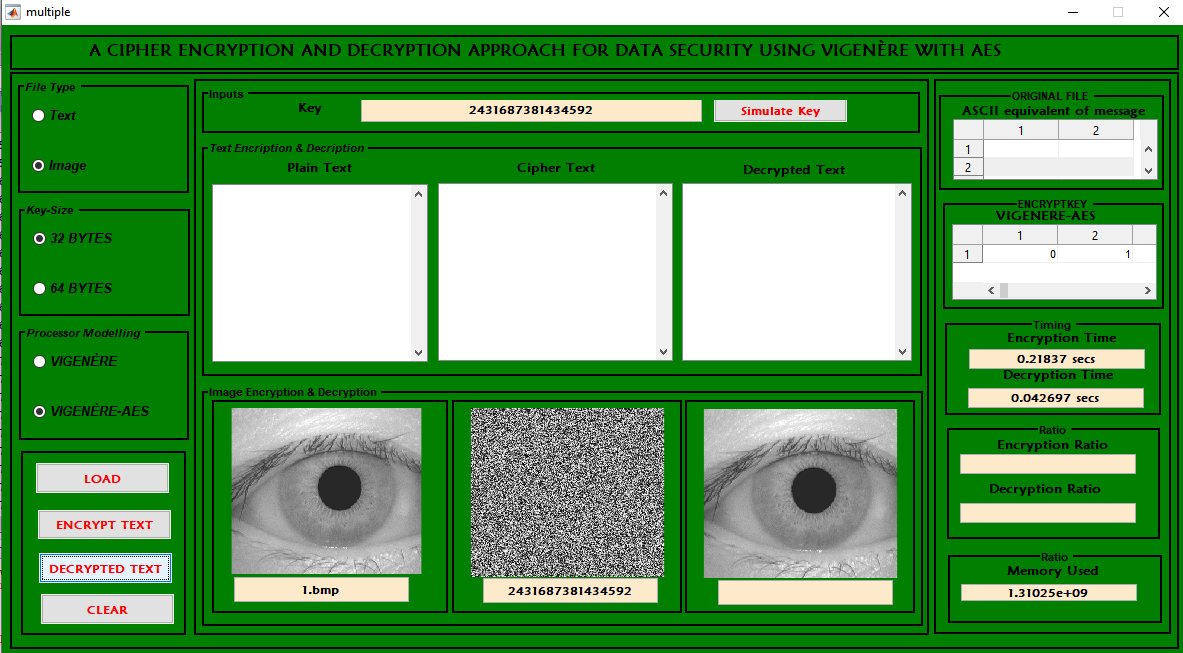
**Figure 4.15: Scrambled Encrypted Image using Vigenère with 32 bytes key size.**

The encrypted image data is decrypted by a receiver with the generated simulated key for 32-byte key size. Figure 4.16 shows the decrypted data with 0.043485 Secs decryption time.

****

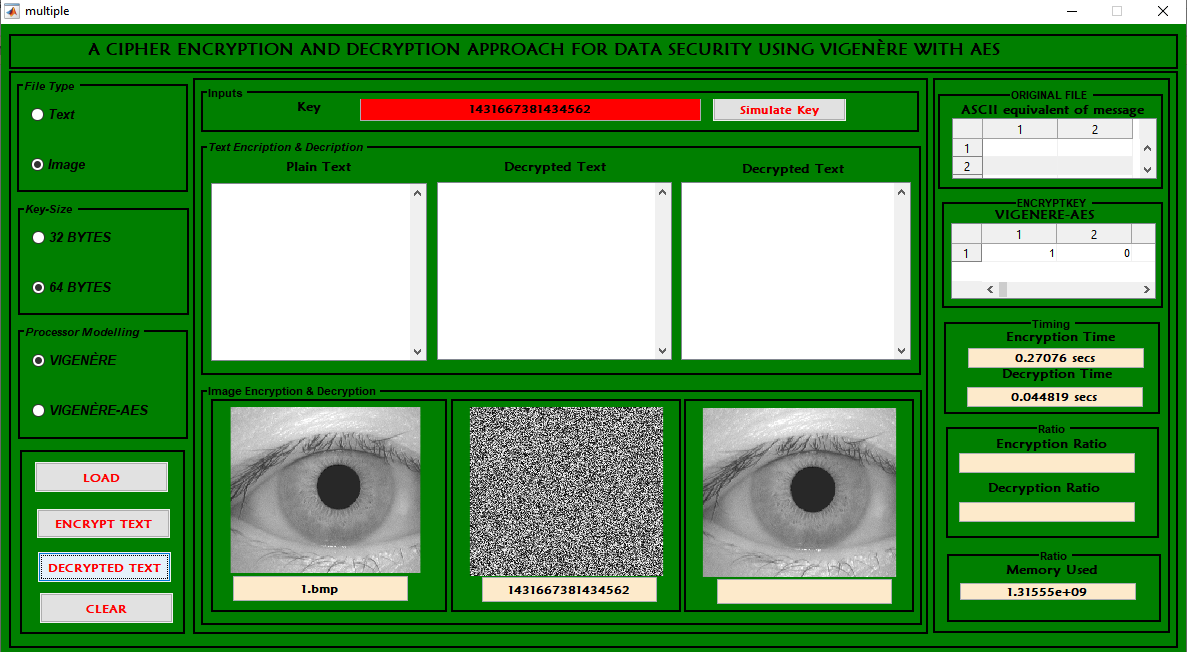
**Figure 4.16: Decrypted Image Data**

In this model Vigenère-AES was used on the image, the results are displayed in figure 4.17.

****

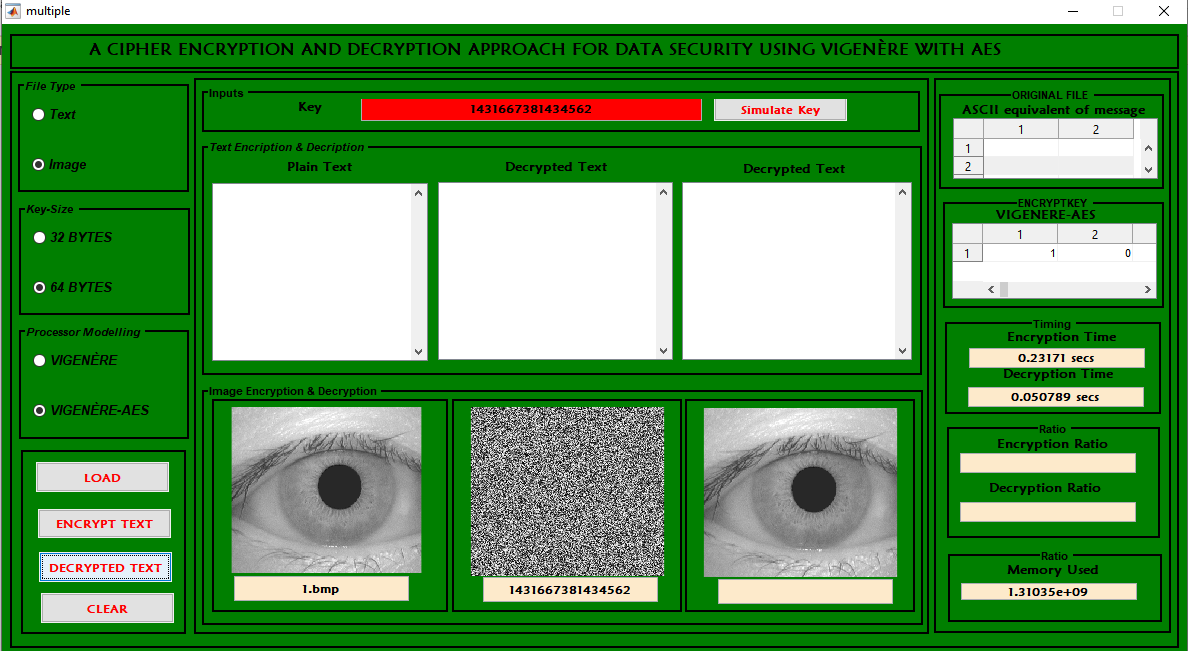
**Figure 4.17: Results of Vigenère-AES for 32 Bytes**

In this model Vigenère using 64 bytes key size was used on the image, the results are displayed in figure 4.18.



**Figure 4.18: Results of Vigenère for 64 Bytes**

In this model Vigenère-AES using 64 bytes key size was used on the image, the results are displayed in figure 4.19.



**Figure 4.19: Results of Vigenère-AES for 64 Bytes**

* 1. **Evaluation and discussion**

In this study, the Vigenère cipher and the Vigenère-AES algorithm were used to test data security on picture and text data messages, and they were found to be more efficient when compared to existing models, as indicated in table 4.1 and table 4.2. One of the attempts to manage data security while maintaining confidentiality is the proposed Vigenère-AES encryption. It is also worthy of note that the Vigenere cipher has a lot of advantages such as: The Vigenere cipher is much more difficult to break than other ciphers, making it more secure, it is also more resistant to attack than other ciphers, it can be used to encrypt any type of data, making it more versatile, it is relatively simple to implement and use, making it user-friendly. The vigenere cipher has a long history and has been used by some of the most famous code-breakers in history, making it a trusted and reliable cipher. However, there are also limitations associated with the Vigenere cipher algorithm and they include; it is not very secure because it can be broken with frequency analysis, the Vigenere cipher is also vulnerable to known-plaintext attacks, Vigenere cipher is not very flexible and can only be used to encrypt short messages, another disadvantage of Vigenere cipher is that it is relatively easy to implement, which means that more people can use it and potentially find ways to break it.

AES on its own also has its strengths as well and they include; it is a very strong encryption algorithm that is difficult to break, it is fast and efficient, making it appropriate for use in present applications, owing to its robust nature, it is flexible and can be used in a number of different ways; including hardware and software implementations and it is available free of charge for anyone to use. AES is a great algorithm, but it has a few disadvantages. First, AES is a block cipher, implying that it encodes data in blocks rather than streams. This can make AES slower and more resource-intensive than other algorithms. Second, AES is a relatively new algorithm, so it may not be as well-vetted or supported as other algorithms. Finally, AES key sizes can be 128, 192, or 256 bits long. The larger the key size, the better secure the encryption, but also the more processing power and memory required.

Considering the strengths and weaknesses of both algorithms, a combination of both algorithms ensured a better, stronger and more secure cipher as each other’s strengths complimented the other’s weaknesses. The combined cipher entailed the features of both the Vigenere cipher and the AES algorithm to provide a more secure method of encryption. The Vigenere cipher adds an element of unpredictability to the AES algorithm, making it more difficult for attackers to break the code. This makes the combined cipher more resistant to brute force attacks, the combined cipher is a strong encryption method that can be used to protect sensitive data. When used in conjunction with other security measures, it can provide a high level of protection for information.

**Table 4.1: Vigenère and Vigenère-AES Results**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S/N** | **Data Type** | **Key Size** | **Algorithm** | **Encryption Time (Secs)** | **Decryption Time (Secs)** | **Memory Used (KB)** |
| 1 | Text | 32 | Vigenère | 0.10005 | 0.16078 | 1.30824 |
| 2 | Text | 32 | Vigenère-AES | 0.12076 | 0.12451 | 1.30467 |
| 3 | Text | 64 | Vigenère | 0.12375 | 0.13123 | 1.31032 |
| 4 | Text | 64 | Vigenère-AES | 0.16141 | 0.11715 | 1.33097 |
| 5 | Image | 32 | Vigenère | 0.31326 | 0.043485 | 1.29123 |
| 6 | Image | 32 | Vigenère-AES | 0.21837 | 0.042697 | 1.31025 |
| 7 | Image | 64 | Vigenère | 0.27076 | 0.44819 | 1.31555 |
| 8 | Image | 64 | Vigenère-AES | 0.23171 | 0.050789 | 1.31035 |

**Table 4.2: Comparison Results with three Existing Models**

|  |  |  |  |
| --- | --- | --- | --- |
| **S/N** | **Paper** | **Algorithm/Method** | **Results (Decryption Time Secs)** |
| 1 | Abdul Hussien et al.,( 2021) | AES | 0.800000 |
| 2 | Zhou & Bi, (2022) | Chaotic Algorithm | 0.185100 |
| 3 | Panda, (2016) | Blowfish | 0.2185 |
| 4 | This Dissertation | Vigenère-AES | 0.050789 |

**CHAPTER FIVE**

* 1. **CONCLUSION AND RECOMMENDATION**
  2. **Conclusion**

A Vigenère-AES cipher is proposed as one of the attempts to manage data security while maintaining confidentiality. It is critical to keep an eye on network security. Data security, privacy, secrecy, and trustworthiness may all be achieved with the use of cryptography. Single classic ciphers are the least complex and most susceptible cryptographic algorithms because of multiple barriers, constraints, and a smooth system. Vigenère Cipher is a well-known cipher, but it does have a few flaws. A new method that is an updated variation of the Vigenère cipher was developed as a mix of AES and Vigenère. This method is significantly more secure against attacks such as Active, Passive, and Friedman assaults. This was done to overcome the limitations of the Vigenère encryption (attacks), but it only takes text as inputs. Information such as text and graphics are used as input transmitted messages in the proposed system, which are encrypted and decrypted to original content using the MATLAB programming language. Intelligence agencies and militaries can use this system to send classified information. Many different cryptographic methods exist, but they still need to be examined by researchers in order to increase data security and privacy. In future, to achieve the aim of getting the suggested approach approved by conducting a performance and security study on communications. Audio and video files are used as input. Creating a web application.

* 1. **Contribution to Knowledge**

This study's main contribution to knowledge is:

1. Implementation of a generalized encryption-decryption matrix method that is applicable to any multimedia data.
2. a requirement for perfect secrecy by integrating Vigenère with the AES cipher to achieve complete secrecy was achieved.
3. Using symmetric nature of keys to eliminate the difficulty of sharing and storing a long key.
   1. **Recommendation**

The most common means of protecting data is cryptography. Several limitations make the Vigenère cipher one of the cryptographic systems regarded weakest and most basic. AES, a more secure cipher than Vigenère, was proposed as a replacement for Vigenère. Although there are various cryptographic approaches yet this sector still demands substantial attention of the research community for the development of data security. By using this model for audio and video data in the imminent future, we hope to validate the recommended approach, and do security and performance research. The level of protection provided by a single encryption technique is insufficient. According to this study, to meet the needs of today's security requirements, a modified Vigenere cipher with an AES algorithm has been presented. It is possible, however, to include more well-known and state of the art encryption techniques. OTP (One Time Pad) can be used to generate a random key of the same length as the plain text message, overcoming the limitation of replication of the key in the Vigenère encryption.

**REFERENCES**

AbdElminaam, D., Abdul Kader, H. M., Hadhoud, M. M., & El-Sayed, S. M. (2014). Developing and Evaluation of New Hybrid Encryption Algorithms. *INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY*, *13*(1), 4038–4052. https://doi.org/10.24297/ijct.v13i1.2926

Abdul, D. S., Abdul Kader, H. M., & Hadhoud, M. M. (2009). Performance evaluation of symmetric encryption algorithms. *Innovation and Knowledge Management in Twin Track Economies Challenges and Solutions - Proceedings of the 11th International Business Information Management Association Conference, IBIMA 2009*, *1*–*3*, 1001–1006.

Abdul Hussien, F. T., Rahma, A. M. S., & Abdul Wahab, H. B. (2021). A Secure Environment Using a New Lightweight AES Encryption Algorithm for E-Commerce Websites. *Security and Communication Networks*, *2021*, 1–15. https://doi.org/10.1155/2021/9961172

Abikoye, O. C., Haruna, A. D., Abubakar, A., Akande, N. O., & Asani, E. O. (2019). Modified advanced encryption standard algorithm for information security. *Symmetry*, *11*(12). https://doi.org/10.3390/SYM11121484

Ahamed, B. B., & Krishnamoorthy, M. (2020). SMS Encryption and Decryption Using Modified Vigenere Cipher Algorithm. *Journal of the Operations Research Society of China*. https://doi.org/10.1007/s40305-020-00320-x

Ahmad, G. I., Singla, J., & Giri, K. J. (2021). *Security and Privacy of E-health Data*. 199–214. https://doi.org/10.1007/978-981-15-8711-5\_10

Al-Janabi, S., Al-Shourbaji, I., Shojafar, M., & Shamshirband, S. (2017). Survey of main challenges (security and privacy) in wireless body area networks for healthcare applications. *Egyptian Informatics Journal*, *18*(2), 113–122. https://doi.org/10.1016/j.eij.2016.11.001

Al-janabi, S. T. F. (2007). Block Cipher Design after AES and the New Cipher Nahrainfish. *I-Manager’s Journal on Software Engineering*, *1*(4), 46–61. https://doi.org/10.26634/jse.1.4.744

AL-Shaaby, A. A., & AlKharobi, T. (2017). Cryptography and Steganography: New Approach. *Transactions on Networks and Communications*, *5*(6). https://doi.org/10.14738/tnc.56.3914

Alshambri, A. H., & Alassery, F. (2021). Securing Fog Computing For E-Learning System Using Integration of Two Encryption Algorithms. *Journal of Cyber Security*, *3*(3), 149–166. https://doi.org/10.32604/jcs.2021.022112

Bermejo Higuera, J., Bermejo Higuer, J. R., Sicilia Montalvo, J. A., & González Crespo, R. (2022). *Introduction to Cryptography in Blockchain* (pp. 1–34). https://doi.org/10.1007/978-981-16-3412-3\_1

Bhateja, A. K., Bhateja, A., Chaudhury, S., & Saxena, P. K. (2015). Cryptanalysis of Vigenere cipher using Cuckoo Search. *Applied Soft Computing*, *26*, 315–324. https://doi.org/10.1016/j.asoc.2014.10.004

Biksham, V., & Vasumathi, D. (2017). Homomorphic Encryption Techniques for securing Data in Cloud Computing: A Survey. *International Journal of Computer Applications*, *160*(6), 1–5. https://doi.org/10.5120/ijca2017913063

Bresson, E., Chevassut, O., & Pointcheval, D. (2001). Provably authenticated group diffie-hellman key exchange – the dynamic case. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *2248*, 290–309. https://doi.org/10.1007/3-540-45682-1\_18

Chakraborty, N., & Mondal, S. (2015). *A New Storage Optimized Honeyword Generation Approach for Enhancing Security and Usability*. http://arxiv.org/abs/1509.06094

Chaloop, G. S., & Abdullah, Z. M. (2021). ENHANCING HYBRID SECURITY APPROACH USING AES AND RSA ALGORITHMS. *Journal of Engineering and Sustainable Development*, *25*(4), 58–66. https://doi.org/10.31272/jeasd.25.4.6

Chaloop, G. S., Abdullah, Z. M., Harba, E. S. I., Uttar, I., Section, P., Conference, I., Engineering, C., Janeiro, D. O. R. I. O. D. E., Maqsood, F., Ahmed, M., Mumtaz, M., Ali, M., Thirumalai, C. S., Senthil Kumar, M., Faheem, M., Jamel, S., Hassan, A., A., Z., Shafinaz, N., & Mat, M. (2017). A Survey on the Cryptographic Encryption Algorithms. *International Journal of Advanced Computer Science and Applications*, *8*(4), 442–448.

Chaloop, G. S., Z. Abdullah, M., Harba, E. S. I., Uttar, I., Section, P., Conference, I., Engineering, C., Janeiro, D. O. R. I. O. D. E., Maqsood, F., Ahmed, M., Mumtaz, M., Ali, M., Thirumalai, C. S., Senthil Kumar, M., Faheem, M., Jamel, S., Hassan, A., A., Z., Shafinaz, N., & Mat, M. (2017). A Survey on the Cryptographic Encryption Algorithms. *International Journal of Advanced Computer Science and Applications*, *8*(4), 442–448.

Chitrakar, R., Bhusal, R., & Maharjan, P. (2020). Hybrid Approaches to Block Cipher. In *Computer and Network Security*. IntechOpen. https://doi.org/10.5772/intechopen.82272

Dworak, K., & Boryczka, U. (2021). Breaking Data Encryption Standard with a Reduced Number of Rounds Using Metaheuristics Differential Cryptanalysis. *Entropy*, *23*(12). https://doi.org/10.3390/e23121697

Faheem, M., Jamel, S., Hassan, A., A., Z., Shafinaz, N., & Mat, M. (2017). A Survey on the Cryptographic Encryption Algorithms. *International Journal of Advanced Computer Science and Applications*, *8*(11). https://doi.org/10.14569/IJACSA.2017.081141

Gadgil, M. A. A. (2016). Enhancing Security in user authentication through honeyword. *International Journal of Scientific Research and Management*. https://doi.org/10.18535/ijsrm/v4i6.17

Gaire, R., Ghosh, R. K., Kim, J., Krumpholz, A., Ranjan, R., Shyamasundar, R. K., & Nepal, S. (2019). Crowdsensing and Privacy in Smart City Applications. In *Smart Cities Cybersecurity and Privacy* (pp. 57–73). Elsevier. https://doi.org/10.1016/B978-0-12-815032-0.00005-6

Harmening, J. T. (2017). Virtual Private Networks. In *Computer and Information Security Handbook* (pp. 843–856). Elsevier. https://doi.org/10.1016/B978-0-12-803843-7.00058-2

He, Y., Ye, N., & Zhang, R. (2021). Analysis of Data Encryption Algorithms for Telecommunication Network-Computer Network Communication Security. *Wireless Communications and Mobile Computing*, *2021*, 1–19. https://doi.org/10.1155/2021/2295130

Jang-Jaccard, J., & Nepal, S. (2014). A survey of emerging threats in cybersecurity. *Journal of Computer and System Sciences*, *80*(5), 973–993. https://doi.org/10.1016/j.jcss.2014.02.005

Jantan, A., Abiodun, O. I., & Arshad, H. (2018). *An Enhanced Practical Difficulty of One-Time Pad Algorithm for Resolving the Key Management and Distribution Problem*. https://doi.org/10.1.1.252.9299

Johnson, L. (2020a). Security component fundamentals for assessment. In *Security Controls Evaluation, Testing, and Assessment Handbook* (pp. 471–536). Elsevier. https://doi.org/10.1016/B978-0-12-818427-1.00011-2

Johnson, L. (2020b). System and network assessments. In *Security Controls Evaluation, Testing, and Assessment Handbook* (pp. 447–469). Elsevier. https://doi.org/10.1016/B978-0-12-818427-1.00010-0

Juels, A., & Ristenpart, T. (20142). Honey encryption: Encryption beyond the brute-force barrier. *IEEE Security and Privacy*, *12*(4), 59–62. https://doi.org/10.1109/MSP.2014.67

Juels, A., & Rivest, R. L. (2013). Honeywords. *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security - CCS ’13*, 145–160. https://doi.org/10.1145/2508859.2516671

Kadry, S., & Smaili, M. (2011). An Improvement of RC4 Cipher Using Vigenere Cipher. *Arxiv Preprint ArXiv:1111.5641*.

Kapil, G., Agrawal, A., Attaallah, A., Algarni, A., Kumar, R., & Khan, R. A. (2020a). Attribute based honey encryption algorithm for securing big data: Hadoop distributed file system perspective. *PeerJ Computer Science*, *2020*(2), 1–31. https://doi.org/10.7717/peerj-cs.259

Kapil, G., Agrawal, A., Attaallah, A., Algarni, A., Kumar, R., & Khan, R. A. (2020b). Attribute based honey encryption algorithm for securing big data: Hadoop distributed file system perspective. *PeerJ Computer Science*, *6*, e259. https://doi.org/10.7717/peerj-cs.259

Kim, H., Kim, C. H., & Chung, J. M. (2012). A novel elliptical curve ID cryptography protocol for multi-hop ZigBee sensor networks. *Wireless Communications and Mobile Computing*, *12*(2), 145–157. https://doi.org/10.1002/wcm.947

Kumar, P., & Rana, S. B. (2016). Development of modified AES algorithm for data security. *Optik*, *127*(4), 2341–2345. https://doi.org/10.1016/j.ijleo.2015.11.188

Mahindrakar, M. S. (2014). Evaluation of Blowfish Algorithm based on Avalanche Effect. *International Journal of Innovations in Engineering and Technology*, *4*(1), 99–103. http://ijiet.com/wp-content/uploads/2014/06/15.pdf

Manjula, J., Ruth, R., Jaswanth, G., Lokesh, N., & Harshita, P. (2019). A Modified Honey Keyword Generation to Prevent Bruteforce Attack. *International Journal of Recent Technology and Engineering*, *8*(4), 11278–11282. https://doi.org/10.35940/ijrte.d9534.118419

Manku, S., & Vasanth, K. (2015). Blowfish encryption algorithm for information security. *ARPN Journal of Engineering and Applied Sciences*, *10*(10), 4717–4719. https://doi.org/Blowfish encryption algorithm for information security

Minaam, D. S. A., Abdual-Kader, H. M., & Hadhoud, M. M. (2010). Evaluating the effects of symmetric cryptography algorithms on power consumption for different data types. *International Journal of Network Security*, *11*(2), 78–87.

Mushtaq, M. F., Jamel, S., Mohamad, K. M., Khalid, S. K. A., & Deris, M. M. (2017). Key Generation Technique based on Triangular Coordinate Extraction for Hybrid Cubes. *Journal of Telecommunication, Electronic and Computer Engineering*, *9*(3-4 Special Issue), 195–200.

Musliyana, Z., Arif, T. Y., & Munadi, R. (2015). Security enhancement of Advanced Encryption Standard (AES) using time-based dynamic key generation. *ARPN Journal of Engineering and Applied Sciences*, *10*(18), 8347–8350. https://doi.org/Security enhancement of Advanced Encryption Standard (AES) using time-based dynamic key generation

Na, Q., Wei, W., Zhang, J., Wang, W., Zhao, J., Li, J., Shen, P., Yin, X., Xiao, X., & Hu, J. (2013). Analysis and Research of the RSA Algorithm. *Information Technology Journal*, *12*(9), 1818–1824. https://doi.org/10.3923/itj.2013.1818.1824

Osaghae, E. O. (2018). Replication of ciphertext in cryptographic system. Journal of Applied Sciences and Environmental Management, 22(8), 1193-1197.

Pourbabak, H., Chen, T., & Su, W. (2019). Emerging data encryption methods applicable to Energy Internet. In *The Energy Internet* (pp. 181–199). Elsevier. https://doi.org/10.1016/B978-0-08-102207-8.00008-4

Quilala, R. L., Sison, A. M., & Medina, R. P. (2018). Modified SHA-1 Algorithm. *Indonesian Journal of Electrical Engineering and Computer Science*, *11*(3), 1027. https://doi.org/10.11591/ijeecs.v11.i3.pp1027-1034

Rabah, K. (2005). Theory and Implementation of Data Encryption Standard: A Review. *Information Technology Journal*, *4*(4), 307–325. https://doi.org/10.3923/itj.2005.307.325

Rahimunnisa, K., Karthigaikumar, P., Christy, N. A., Kumar, S. S., & Jayakumar, J. (2013). PSP: Parallel sub-pipelined architecture for high throughput AES on FPGA and ASIC. *Open Computer Science*, *3*(4), 173–186. https://doi.org/10.2478/s13537-013-0112-2

Rao, U. H., & Nayak, U. (2014). Cryptography. In *The InfoSec Handbook* (pp. 163–181). Apress. https://doi.org/10.1007/978-1-4302-6383-8\_8

Rehman, S., Talat Bajwa, N., Shah, M. A., Aseeri, A. O., & Anjum, A. (2021). Hybrid AES-ECC Model for the Security of Data over Cloud Storage. *Electronics*, *10*(21), 2673. https://doi.org/10.3390/electronics10212673

Segar, T. C., & Vijayaragavan, R. (2013). Pell’s RSA key generation and its security analysis. *2013 4th International Conference on Computing, Communications and Networking Technologies, ICCCNT 2013*. https://doi.org/10.1109/ICCCNT.2013.6726659

Sen, J. (2015). Security and privacy issues in cloud computing. *Standards and Standardization: Concepts, Methodologies, Tools, and Applications*, 1370–1414. https://doi.org/10.4018/978-1-4666-8111-8.ch065

Smart, N. P. (2016). *Historical Ciphers* (pp. 119–132). https://doi.org/10.1007/978-3-319-21936-3\_7

Sousi, A.-L., Yehya, D., & Joudi, M. (2020). AES Encryption: Study & Evaluation. *CCEE552: Cryptography & Network Security*.

Tang, J., Cui, Y., Li, Q., Ren, K., Liu, J., & Buyya, R. (2016). Ensuring security and privacy preservation for cloud data services. *ACM Computing Surveys*, *49*(1). https://doi.org/10.1145/2906153

Thirupalu, U., Scholar, R., & Reddy, E. K. (2020). Performance Analysis of Cryptographic Algorithms in the Information Security; Performance Analysis of Cryptographic Algorithms in the Information Security. *International Journal of Engineering Research and Technology (IJERT)*, *8*(2). www.ijert.org

Virtue, T., & Rainey, J. (2015). Privacy and Security in Healthcare. In *HCISPP Study Guide* (pp. 61–89). Elsevier. https://doi.org/10.1016/B978-0-12-802043-2.00004-5

Vollala, S., Ramasubramanian, N., & Tiwari, U. (2021). Cryptographic Techniques. In *Energy-Efficient Modular Exponential Techniques for Public-Key Cryptography* (pp. 3–30). Springer International Publishing. https://doi.org/10.1007/978-3-030-74524-0\_1

Will, M. A., & Ko, R. K. L. (2015). A guide to homomorphic encryption. In *The Cloud Security Ecosystem* (pp. 101–127). Elsevier. https://doi.org/10.1016/B978-0-12-801595-7.00005-7

Wilson, P. (2016). Design Optimization Example. In *Design Recipes for FPGAs* (pp. 265–281). Elsevier. https://doi.org/10.1016/B978-0-08-097129-2.00019-2

Yassein, M. B., Aljawarneh, S., Qawasmeh, E., Mardini, W., & Khamayseh, Y. (2017). Comprehensive study of symmetric key and asymmetric key encryption algorithms. *2017 International Conference on Engineering and Technology (ICET)*, 1–7. https://doi.org/10.1109/ICEngTechnol.2017.8308215

Zaid, M., & Mohd, W. (2007). Evolution of Cryptography. *Evolution of Cryptography*, *17*(January), 1–6. https://scholar.google.com/scholar?q=evolution+of+cryptography&hl=en&as\_sdt=0&as\_vis=1&oi=scholart

Zhang, P., Schmidt, D. C., White, J., & Dubey, A. (2019). *Consensus mechanisms and information security technologies* (pp. 181–209). https://doi.org/10.1016/bs.adcom.2019.05.001

Zhou, H., & Bi, H. (2022). Wireless Sensor Network Security Based on Improved Identity Encryption. *Scientific Programming*, *2022*, 1–10. https://doi.org/10.1155/2022/2308825