SCHOOL OF GRADUATE STUDIES

ENHANCING THE SECURITY OF USER DATA STORED ON CLOUD

MSC THESIS

HANA MOTI

JANUARY, 2024

WOLKITE, ETHIOPIA

Wolkite University

School Of Postgraduate Studies

Enhancing the Security of User Data Stored on Cloud

A Thesis Submitted to the School of Graduate Studies in Partial Fulfillment for the Degree of Master of Computer Science and Engineering

Hana Moti

Major Adviser: Negalign Wake Hundera (Ph.D.)

Co-Adviser: Teshome Yihune Birle (MSc.)

January, 2024

Wolkite, Ethiopia

APPROVAL SHEET

SCHOOL OF GRADUATE STUDIES

WOLKITE UNIVERSITY

Enhancing the Security of User Data Stored onCloud

Submitted by:

Hana Moti Adugna		10/01/2024
Name of Student	Signature	Date

Approved by:

Negalign Wake Hundera (PhD)		11/01/2024
Name of Major Advisor	Signature	Date
Tashome Yihune (MSc)		
Name of CO-Advisor	Signature	Date
		9/1/2024
Name of Chairman, DGC	Signature	Date
Name of Dean, SGS	Signature	Date

WOLKITE UNIVERSITY

SCHOOL OF GRADUATE STUDIES

We hereby certify that we have read and evaluated this Thesis titled "**Enhancing the Security of User Data Stored on Cloud**" prepared under our guidance by Hana Moti. We recommend that the Thesis shall be submitted as fulfilling the requirements for the award of a MSc. Degree in Computer Science and Engineering.

Negalign Wake Hundera (PhD)		11/01/2024
Name of Major Advisor	Signature	Date
Tashome Yihune (MSc)		
Name of CO-Advisor	Signature	Date

As members of the Board of Examiners of the Master of Science Thesis open defense examination, we have read and evaluated this Thesis prepared by Hana Moti and examined the candidate. We hereby certify that, the thesis is accepted for fulfilling the requirements for the award of the degree of Masters of Science (MSc) in Computer Science and Engineering.

Name of External Examiner Kindle Biredagn (Ph.D.)	Signature	Date Jan-09-2024
Name of Internal Examiner Barmura Yibleta	Signature	Date 09/01/2024
Name of Chairperson Kuma Yadi	Signature	Date 09/01/2024

Final approval and acceptance of the thesis are contingent upon the submission of the final copy of the thesis to the School of Graduate Studies (SGS) through the Department/School Graduate Committee (DGC/SGC).

TABLE OF CONTENTS

DECLARATION

By my signature below, I declare and affirm that this thesis is my own work. I have followed all ethical principles of scholarship in the preparation, data collection, data analysis and completion of this thesis. All scholarly matter that is included in the thesis has been given recognition through citation. I affirm that I have cited and referenced all sources used in this document. Every serious effort has been made to avoid any plagiarism in the preparation of this thesis.

This thesis is submitted in partial fulfillment of the requirement for a degree from the school of Graduate Studies at Wolkite University. The thesis is deposited in the Wolkite University library and is made available to borrowers under the rules of the library. I solemnly declare that this thesis has not been submitted to any other institution anywhere for the award of any academic degree, diploma or certificate.

Brief quotations from this Thesis may be used without special permission provide that accurate and complete acknowledgment of the source is made. Requests or permission for extended quotation from, or reproduction of this thesis in whole or in part may be granted by the head of school or department or dean of the school of graduate Studies when in his/her judgment the proposed use of the material is in the interest of scholarship. In all other instance, however, permission must be obtain from the author of the thesis.

Name: Hana Moti Adugna

Date: 10/01/2024

School/Department: Computer Science and Engineering

Signature: 

ACKNOWLEDGEMENT

My gratitude is directed toward my GOD, who has supported me in every situation and is the foremost creator of heaven and earth, the beginning and the end.

In order for my thesis to be successful, I would like to thank my advisor Negalign Wake Hundera Ph.D. for his suggestions and diligent work with me.

I appreciate the guidance and supervision of my co-advisor, Mr. Tashome Yihune Birle(Msc). Also my family especially my husband Mr. Chala Boka (Msc.) my son Wabi Chala, my daughter Keisan Chala and my brothers Dani'el Moti (BSc), Samu'el Moti and Johanis Moti(BSc). thanks for them for giving me enough time, understanding that I should give priority to my Research work, who stood me and made me to successful this Research work, you have a place for me whenever.

Finally, I'd want to express my gratitude to everyone who helped me with these study Research in whatever form.

By: Hana Moti

Signature: 

Date: 10/01/2024

Big thanks for all!!!!

ABBREVIATIONS AND ACRONYMS

AES	Advanced Encryption Standard
CSE	Computer Science and Engineering
CCKS	Encrypted text combines with encrypted key
GCD	Greatest common divisor
ERSA PrT	Private Key of Sender
CK, ERSAPuR	Public key of receivers
RSA	Rivest, Shamir and Adleman
VPN	Virtual Private Network
IDEA	International Data Encryption Algorithm
DES	Data Encryption Standard
3DES	Triple Data Encryption Standard
LZW	Lempel-Ziv-Welch
VPN	virtual private network
PMC	Poor Man's Compression
PMC-MR	Poor Man's Compression–Mid-range

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES IN THE APPENDICES

ABSTRACT

Data protection is now more important than ever to protect against hacking due to the Internet's explosive development in text transfer. Many encryption and decryption algorithms are used to offer a high level of security, including DH (Diffie Hellman), RSA (Rivest Shamir-Adleman), and AES (Advanced Encryption Standard).However, these algorithms frequently call for large key sizes, which can make implementation difficult. In this article, a hybrid technique for data encryption with RSA and AES with LZW (Lempel-Ziv-Welch) compression technique is proposed. This thesis mainly concentrates on evaluating the effectiveness of text data encryption and decryption methods utilizing the AES and RSA algorithms, LZW compression, MEGA cloud storage and Cyber Ghost VPN for safe storage and internet access. The paper highlights how these methods increase algorithm strength, key generation, and decryption speed to guarantee the security and privacy of sensitive user data. The goal of this research is to provide a secure data transport solution that overcomes the shortcomings of existing encryption techniques. The difficulty of key size in encryption methods, which might provide employment issues, is the issue that this work attempts to solve. The suggested method seeks to offer a quick and reliable way to encrypt data using the strength of RSA and AES Cryptography together. The study's findings show that the suggested method is quick and secure. Implementation and performance analysis using Python 3.11 (64-bit) shows the efficacy of the suggested strategy. For implementation and performance analysis we use "Design of Technology Integrated Shredder Machine" project of data set size 4,616,263 bytes before compression. After compression, the file size reduced to 4,475,411 bytes. based on results,the proposed algorithm improve the original RSA key generation time is 3.336312 second or 100.00%,encryption time of AES by RSA 0.015691 second or 0.01%,the ***decryption*** *time of* ***RSA 0.015624*** *second or* ***0.01%****,the* ***encryption*** *time of compressed data by AES 0.000011 second or 0.00% and the decryption time of compressed data by AES 0.0000012 second or 0.01% but when we encrypt the by AES alone ,the encryption time is 0.07812s or 47.75% and the decryption time is 0.08546s or 52.25 % as the result of this performance investigation, the proposed or AES and RSA hybrid algorithm is very fast and secure than the individual once.*

Key Words: Advanced Encryption Standard, Cryptography, Decryption, Encryption

CHAPTER ONE

1. INTRODUCTION

1.1. Background of the study

A network of distant computers hosted online is referred to as a 'cloud' in the computing industry. The functions of these computers are data management, storage, and analysis. Through the use of the internet, cloud computing enables users to make use of services and assets, doing away with the necessity for local gear and software to perform computer tasks .The cloud makes it possible for remote access, configuration, and modification of hardware and software resources. Users can benefit from a range of advantages provided by cloud storage, such as increased service flexibility, less effort for storage management, location-independent data access, and platform independence. Scalability, efficiency, and flexibility are just a few of the many advantages of cloud computing. It enables businesses and people to instantly access strong computer capabilities without having to fork over a lot of cash for pricey hardware and software. By doing this, the computer won't ask to install any local programs. The initial costs are lower, and copies of the data are kept on several servers located in different geographic areas. Only changing the reference to the object's storage location is necessary to repair a failed service. Users back up their data and transfer it to an alternative cloud service provider. This situation calls for improved customer service. It provides information, help, and data storage for the whole internet [1]. Hybrid Cloud is the final option. Most cloud computing services are Infrastructure as a service (IaaS); Platform as a Service (PaaS) and Software as a service (SaaS) [2].

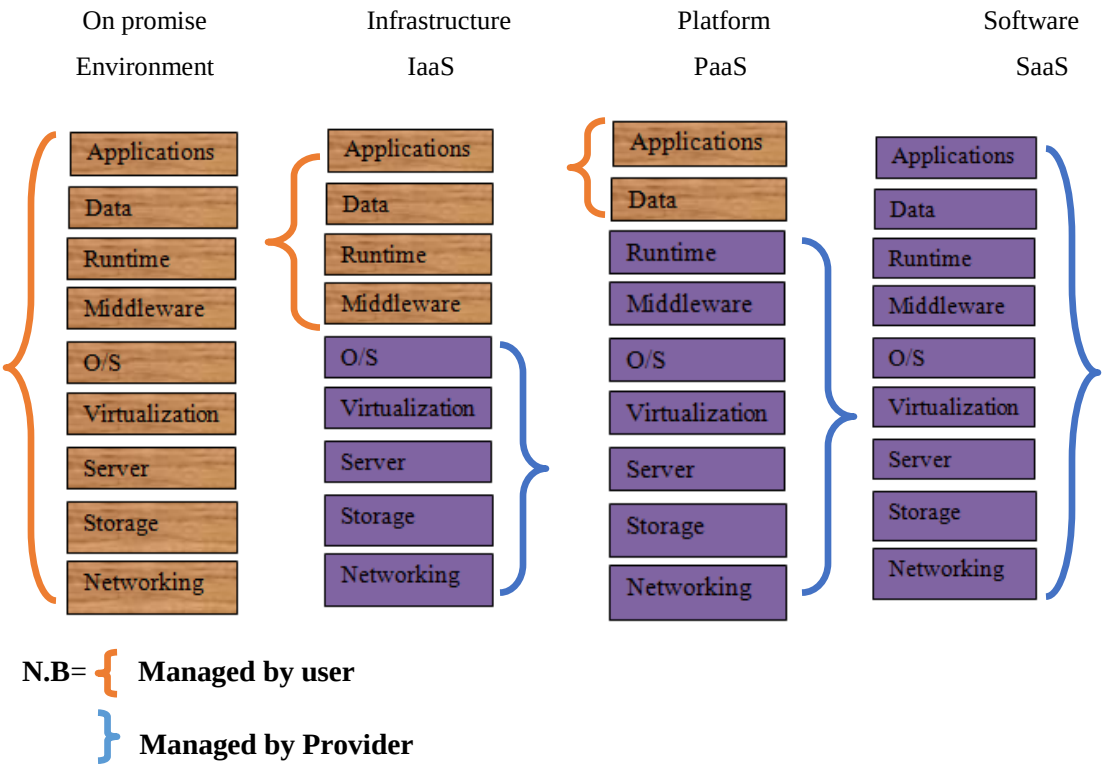


Figure 1.1 Model of Cloud Service

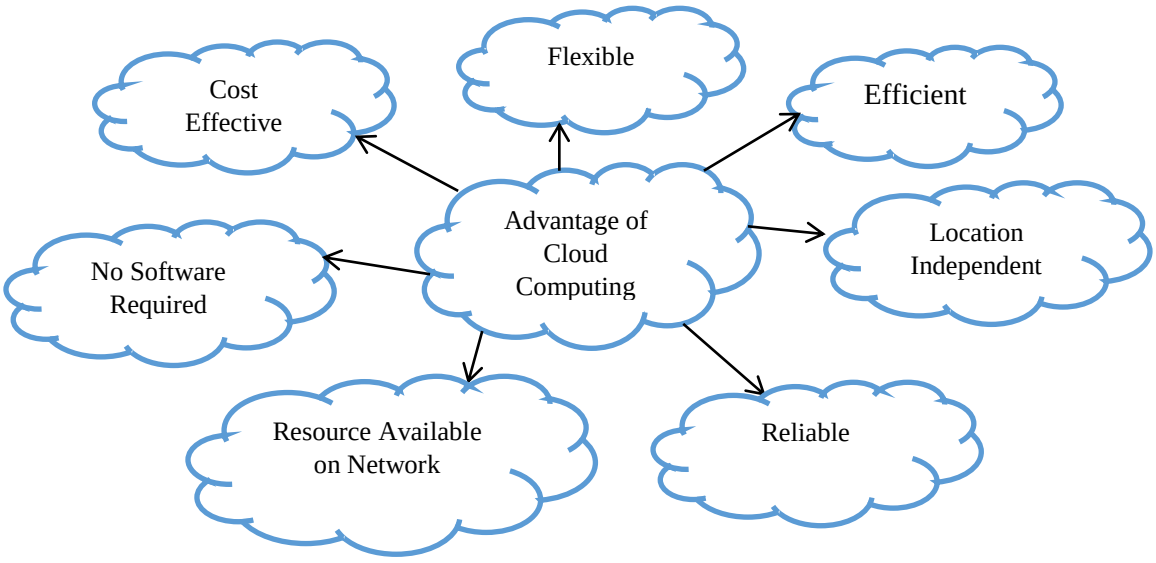


Figure 1.2 Advantage of Cloud Computing

The more organization and more individuals are storing their data on remote area called cloud. In the truth, cloud service providers provide a variety of services at affordable

prices raises questions about their reliability [3]. Cloud storage decreases the amount of gear and software that users need to utilize it; it may store your digital content, such as files, documents, images, videos, and music, on remote servers that can be accessed online. Instead of being stored locally on your computer or external hard drive, your data can be uploaded to a cloud storage provider's server and viewed from any device with an internet connection. Cloud storage companies frequently provide a choice of storage options and pricing alternatives in order to meet a range of demands and budgets. Popular cloud storage providers include Mega, Google Drive, Drop Box, Microsoft One Drive, and I Cloud.

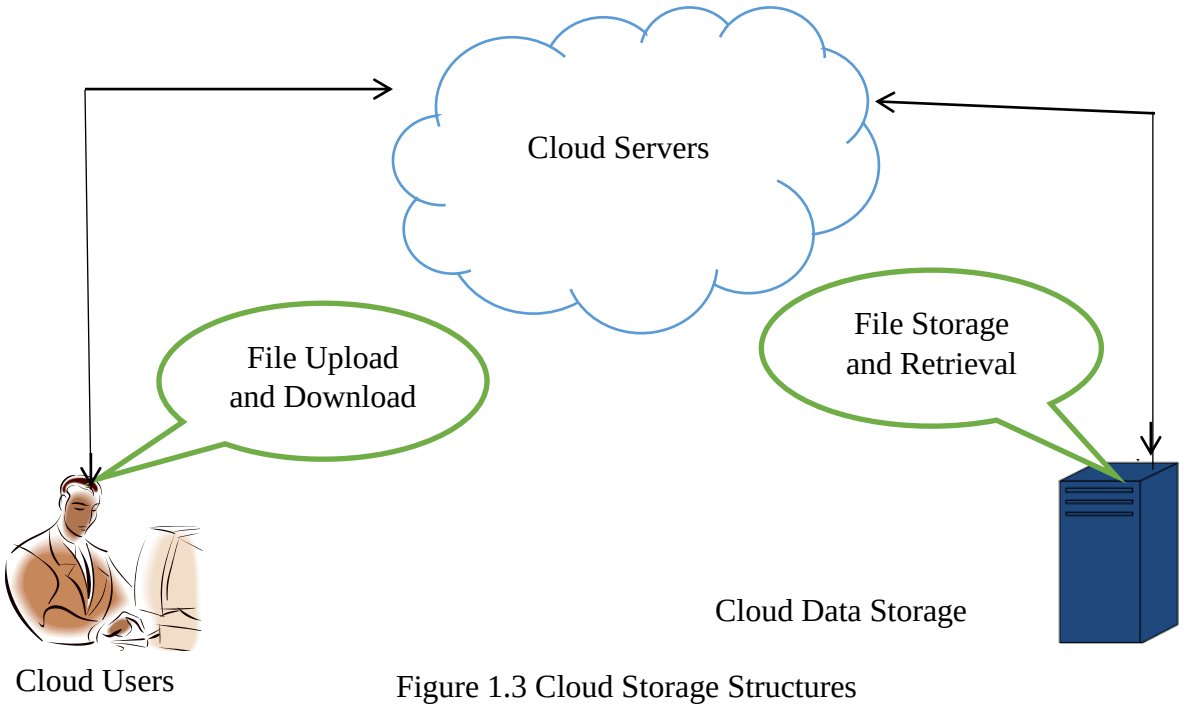


Figure 1.3 Cloud Storage Structures

There are several benefits to using cloud storage, including: Accessibility: Your files are available everywhere, on any device with an internet connection. Data backup and recovery: Because your data is stored off-site, even if your local device is lost, stolen, or damaged, you can still retrieve it. Sharing files with others enables real-time collaboration on papers. Effectiveness from a financial standpoint: Using the cloud to store your data is typically less expensive than getting and keeping physical storage devices. However, there are also potential drawbacks to cloud storage, such as possible security and privacy threats and data breaches. When choosing a trustworthy cloud

storage provider to protect your data, you need thoroughly consider your options and make an informed decision [4].

We utilize MEGA cloud storage and Cyber Ghost VPN to safely store our data in the cloud. MEGA asserts that before entering the company's servers, every data saved in the cloud is secured on the user's device. With Mega, we can obtain up to 20 GB of free cloud storage. It provides safe end-to-end encryption to assist prevent a privacy compromise. You can obtain the whole 20 GB without paying if we finish our duties. Sharing links in a fashion that requires a second portion of the URL, essentially acting as a password, before the receiver may decode the content, is an intriguing feature. Our files viewed and updated via a browser, the desktop sync client, or the mobile app, making it compatible with a number of operating systems like Android, iOS, Windows, and others. Cyber Ghost can be a useful tool for safeguarding user data by encrypting internet traffic, hiding the user's IP address, and watching out for online risks. By acting as a middleman between your device and the internet, a virtual private network (VPN) hide your real location and provide the impression that you are surfing from a different area. Through the usage of services like Cyber Ghost, which encrypt internet traffic and mask user IP addresses, user data protected [5].

Users of cloud storage manage how the resources are used because they are shared, lowering these risks. They need to make sure that only those with permission may access the data in order to reduce the chance of losing control of it. Make sure the data is exchanged and stored in the cloud securely as well. One of the most popular techniques for protecting data privacy is data encryption with cryptography [6], which provides a variety of options for safeguarding data to retain its private and confidentiality. Data encryption ensures that your privacy will be protected by the cloud service provider. Key management, encryption, and decryption are requirements for all cryptography schemes .AES and RSA are used in the encryption and decryption procedures. Text files are crucial for giving information, especially in remote sensing. Nowadays, text files are utilized and depended upon more and more for information storage and transfer. Text files are a particular type of digital or computer-based, non-executable file that are structured as lines of electronic text and contain letters, numbers, symbols, or a

combination of these. Text files are collections of textual data or information that can be read by humans [7]. Also kept in, plain-text or rich text forms. Although software developers, often employ user-created and stored documents to store programmed data.

All human language communications take the form of plain text or clear text. This includes any correspondence between you and me. This means that, if the message is not in any way formalized, anyone who understands the language may understand a message in plain text [8]. As a result, we must now utilize coding schemes to make sure that information is concealed from anyone for whom it is not meant, even those who can view the coded data. Digital text files make up a significant portion of our daily communication. The process of gathering, processing, and exchanging text files across numerous interpersonal and unrestricted communications has become more sensitive as a result of phones and computers. Text is preserved and later used by many applications [9].

There must be adequate security measures and protection against the fact that many fields, including the military and the medical industry, contain critical text information. Sensitive data must be encrypted, and encryption algorithms are made to assure data security, confidentiality, and access by authorized parties only. Processing power and storage capacity, however, are quite important on computers and even in the cloud. We use compression techniques to shrink the size of text for storage and transition bandwidth [10].

Text compression often reduces the total file size by identifying related strings within a text file and replacing them with a temporary binary representation. By identifying recurring sequences and swapping them out with shorter representations, computers can compress text similarly to how people do [11].

Compression is a method for reducing the size of data files, which leads to a decrease in the amount of storage space needed, faster transmission, and faster read/write file times. Both lossy and lossless compressions are types of compression. Certain original files are shrunk through lossy compression to reduce file size. It's crucial to reduce the number of colors in an image or music file as a consequence. After employing lossy compression to

compress the file, the quality is only slightly reduced, and the erased data cannot be recovered [7].

The quality of the files is not at all impacted because lossless compression does not result in data loss. When a file is lossless, it may be reconstructed from its original state. Text, executable software, spreadsheets, and other file kinds can all be compressed using this technique [12]. We choose lossless text compression over other methods because it provides the highest level of security and quality for the most text-intensive fields, such as the medical and military fields.

Encryption is the process of converting authentic content into an incomprehensible format so that it may be secured against unauthorized users by utilizing a cryptography technology. It serves as a safeguard against unauthorized access to sensitive data. As a result, obtaining the original data is incredibly difficult for an unauthorized user. The most effective algorithms must have completed testing in order to meet the security standards that protect the encryption process [13]. When a text input is subjected to a symmetric or asymmetric encryption technique, cypher text is produced.

The terms "crypt" and "graphein," both of which imply "study and writing" and "hidden or secret," respectively, are the roots of the word "cryptography," which has Latin origins. Sender and recipient both have serious concerns about information security during data transfer [14]. This is where cryptography is essential. As a result, one method of securing communication across a network is to use cryptography to transform readable (plain) text into unreadable (cypher) text. Applications in the military and healthcare utilize it to encrypt data (audio, video, image, and text).In the field of cryptography, there are several distinct types of encryption methods. Asymmetric and symmetric encryption techniques are both present in this work, though.

In symmetric encryption, the private key is the only key that may be used to encrypt and decode text. Modern times may see the use of a mix of letters, numbers, and other symbols to create a special or symmetric key. Before transmitting an encrypted communication to a recipient via the internet, the sender must first provide them the recipient's private key, which is used for encryption. The sender needs to find an alternative technique than using the internet to deliver the receiver this encryption key.

Symmetric encryption techniques are widely employed for encrypting or transmitting massive volumes of data [15]. International Data Encryption Algorithm (IDEA), Data Encryption Standard (DES), Rivest Cipher 4(RC4), Rivest Cipher (RC5), and Triple Data Encryption Standard (3DES) typical algorithms include the Caesar cypher, BLOWFISH, Advanced Encryption Standard (AES), and TWOFISH.

Asymmetric encryption uses two keys, a private key and a public key. A separate encryption key is used for both text encryption and decoding in order to provide secure communication between two parties. Sender must provide recipient access to the sender's public key for encryption in order for them to communicate .Asymmetric encryption algorithms are often used for public key encryption, private key decryption, private key signature, public key verification, small bits of sensitive information encryption, digital signature, and other reasons. Public key encryption is a term that is used to describe asymmetric encryption [16] RSA, DSA, and ECC are a few well-liked algorithms [17].

1.2. Research Problem

These days, digital text is utilized mostly through a network and has become an indispensable source of information. Text from many sources is presented to the globe on a daily basis. Data on a high level of secrecy are frequently found in the majority of this material. Cloud computing without a doubt offers a number of benefits, but there are also some security concerns. Since we are aware that someone else has access to our sensitive data and that we don't have complete control over our database, we must take precautions. Therefore, it's possible that hackers could access our private information or sensitive data if the security of a cloud service were to be breached. Attackers consistently try to extort the owner of these private texts in a variety of ways by attempting to steal, harm, or use them as leverage. Additionally, recent times have seen a lot of interest in the security of digital text. Since people can now exchange digital multimedia with others conveniently over the internet, the Internet's rapid development and widespread application of technology, it is necessary to suggest a strong method of protecting these texts against various types of attackers. Although numerous studies; were done using RSA, DES, AES, and other techniques to improve data security [18].

The suggested strategy's efficiency could have been more robustly demonstrated through an evaluation and comparison study. The first problem in this work, the data is encrypted before being temporarily stored on the cloud by those algorithm AES, RSA, DES, 3DES, the encryption before upload is the good work but, when we encrypt our data, the encryption and decryption time is high or the length of key is only for small data or the key is break. The second problem is again they pose a potential security risk during upload because our location is seen by others. The third problem is the memory size of our data we want to upload to cloud because of our data redundancy. The forth and the last problem in this work is where we upload our data (storage free size, end to end encryption) is the other problems in exist papers [19]–[21].

1.3. Research Questions

1. Which algorithm is better for text encryption and decryption?
2. Which algorithm is better for text compression and decompression?
3. Which cloud storage providers are better for store user sensitive data?
4. How we hide our IP-address when we upload our data to cloud or which VPN is more better?

1.4. Objective of the Study

1.4.1. General Objective

The aim of the thesis is to enhance text security using hybrid cryptography with LZW compression. This system will require an input that is successfully encrypted using hybrid algorithm techniques and store them anywhere by using cryptography algorithm both symmetric and asymmetric encryption by combining AES and RSA for more secure sensitive user data store on cloud.

1.4.2. Specific Objectives

- To select better algorithm for text encryption and decryption.
- To select better algorithm for text compression and decompression.
- To select better Cloud storage provider.
- To select better Virtual Private Network (VPN).

- To collect data from individuals and Department to encrypt and decrypt.
- To compress data for minimize its size.

1.5. Significance of the Study

More organization and individual use Cloud Computing for store or save their sensitive data on remotely space. Detouring this time, every organization and individuals (healthcare, military, telecommunication, medicine, transportation and business institutions) are more success if secure their data by using hybrid algorithm (RSA and AES). Therefore our work can be applied in these areas. This research work helps and satisfies different organization by improving real time text encryption and decryption in different fields. Some of the significant is:-

- Optimize the key generation of AES and RSA algorithm.
- Minimize the time to encrypting and decryption.
- Fast and secure communication.
- Provide greater security performance in text encryption without difficulty.

1.6. Scope and Limitation of the Study

This Research is discussed the AES and RSA for text encryption performance, security and its shortcoming. Then it attempt to propose an algorithm overcome these shortcomings. The research is limited to improving text security and speed of text encryption, decryption and key generation of AES, RSA and LZW compression. The proposed algorithm performance is compared to that of original AES and RSA because the new algorithm is taking the advantages of them to gather rather than single of them.

We design secure user data stored on cloud by cryptography algorithm. It focuses on Securing text user data by using AES and RSA encryption techniques during storing and retrieval of text document stored on cloud. Comparison to other algorithms is not allowed. As a result, comparisons with other algorithms based on various metrics are not included in this study. The algorithm's ability to encrypt and decode text files is first verified by the research, and then the algorithm's encryption and decryption times are examined. Lack of free cloud tools is the study's restriction; laptop PCs are used for

installation and testing. As a result, the researcher is compelled to employ a relatively limited set of data.

1.7. Organization of the Study

There are 7 sections to this Thesis. The Review of Literature in my subject is expressly noted in Chapter 2. The Methodologies of Research are explained in chapter 3 Proposed Architecture and System Design was presented in chapter 4; Results and Analysis in Chapter 5. Conclusion and Future works in chapter 6 and submission of Reference in chapter 7.

CHAPTER TWO

2. LITERATURE REVIEW

2.1 Introduction

This chapter presents securing of sensitive user data stored on cloud by encrypts the data using AES and RSA algorithm and Text compression. An overview of the Cryptography is provided at the outset of the chapter's content. The Algorithm both Symmetric and Asymmetric, LZW compress discussed in details; furthermore, the various approaches of Encryption and decryption are explained in detail in this chapter. Lastly, in this study, some related Cloud security by Cryptography algorithm is discussed in detail.

Importance of Security

One of the key problems in the computer environment is security, which limits the data availability, confidentiality, and integrity. To achieve strong security and shield the text data from hostile assaults, several researchers and academics have offered various security schemes. Modern applications such as defense databases, bank finances, mobile computing, personal communication, etc. all place a high priority on text data security. Text share cryptography might be crucial, especially for applications that need authentication based only on shared keys kept by many parties [22].

We require robust techniques to safeguard our data so that it is not exposed to an attacker since security in text transmission is difficult during the transformation of text for purpose of communication. Text should be secured before being sent from the sender to the recipient in order to prevent this. Researchers suggested several approaches and ways to maintain security for text data transfer, with encryption and data concealing being two of the most often utilized methods [35].Security is ensured through encryption. The issue with encryption, however, is that after several attempts, hackers and attackers may start to lessen their activity. Data concealing is another technique, and the data might be in the form of text, images, sounds, or videos [23], [24].

2.1. Introduction to Cryptography

Making a system of protocols to turn plain text into cipher is the goal of cryptography. It is a technique used to ensure that all of a file's contents are transmitted with the utmost confidentiality and are unaltered. Information privacy must be maintained through cryptography. In order to decrypt a file that has been encrypted using cryptography, a hacker needs the keys, so the file cannot be read by hackers even after it has been compromised. So, a crucial tool for protecting communication during transmission is cryptography. The fact that cryptography uses undisclosed methods to transport information over networks prevents hackers from easily deciphering sent data is another benefit [25].

The original message, secret message, or information that has been encrypted but is still legible is referred to as Plain Text in cryptography. Cipher text, also known as encrypted text, is data that has been transformed from the original data with the use of a key. Key: The method or process of turning original data into unreadable form with the use of a key is known as encryption, whilst the opposite process is known as decryption. The string of words or any value that is utilized for converting readable data to unreadable and inverse is known as the key. Cryptography is often divided into two categories: symmetric cryptography and asymmetric cryptography. When using symmetric or private key cryptography, only one key is needed for both data encryption and decryption; however, when using asymmetric or public key cryptography, two keys are required: one for data encryption (private key), and the other for data decryption (public key) [26].

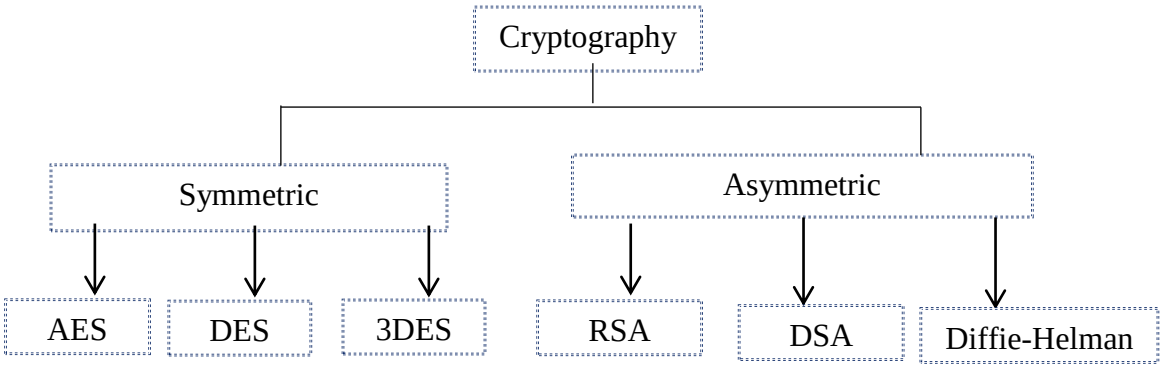


Figure 2.1 Classifications of Cryptography Algorithms

When data is being transmitted over a network, cryptography provides authentication for the data. We must transform the data into a distorted format in order to protect our sensitive information from unauthorized individuals. Thus, using various algorithms like International Data Encryption Algorithm (IDEA),Data Encryption Standard (DES),Triple Data Encryption Standard (3DES),Data Structure (DS), Blowfish and ECC to encrypt text data, cryptography is the most effective and well-known method for doing.

2.1.1. Symmetric Cryptography

The symmetric encryption techniques, as we just established, encrypt and decode data using the same single key. Each algorithm has a unique way for encryption and decryption data, and each one will employ a block of data that is a defined size and a key that is a fixed size for both operations [8].

These algorithms only let the entry of the English alphabet, certain symbols, and numeric numbers. As a result, the output (cipher text) will be created as a document using only special characters, alphabets, numerals, or any combination of these. This algorithm's primary advantage is its low computational complexity and quick processing speed during encryption. Compared to the asymmetric approach, symmetric is quicker. When the encryption key and decryption key used in an algorithm are the same, the algorithm is said to as symmetric and uses a single key. For the encryption and decryption processes to work, the key must be made available during communication [27].

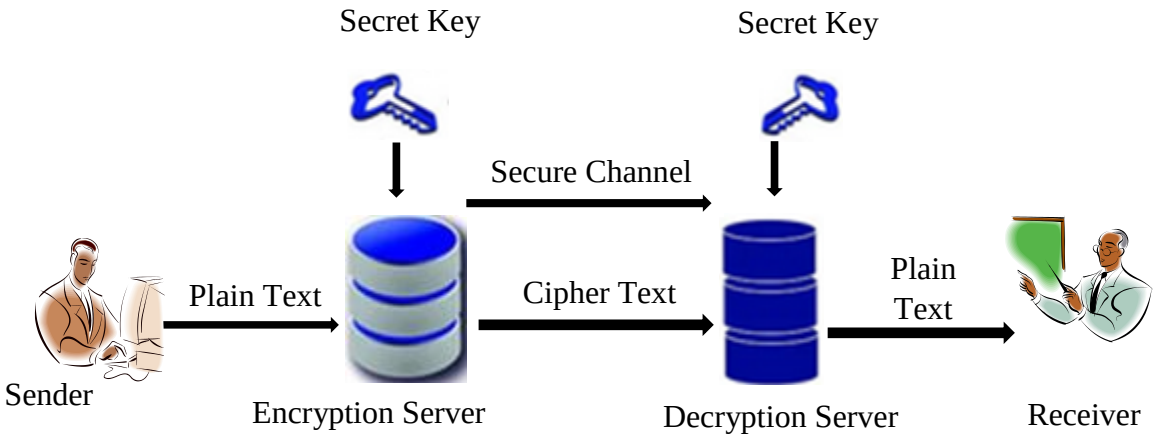


Figure 2.2 Symmetric Key Cryptography

Due to its simplicity of usage, symmetric cryptography has lately grown in popularity and is now used in a variety of applications. Key sharing can occasionally be insecure with symmetric approaches since only one key is used on both sides of the transaction. Although symmetric key cryptography is quicker than public key cryptography, all private key cryptography will be useless if the adversary already knows the secret key. As a result, the key must be kept a secret while communicating. The secret key might be physically exchanged between these parties. However, it is not an acceptable solution for parties to physically exchange keys when they are separated by a geographical barrier. While other researchers use DES, 3DES, BLOWFISH, and AES to encrypt and decrypt text, we instead use a combination of symmetric and asymmetric key cryptography because there is only one key that can be used for both encryption and decryption [28].

2.1.2. Asymmetric Cryptography

Asymmetric key encryption is a method that allows for the use of different keys for encryption and decryption. In this instance, the public key set is the first key set. Secondly, we kept it a secret. Similar terms for "public key encryption" might be used to describe both of these keys [29].

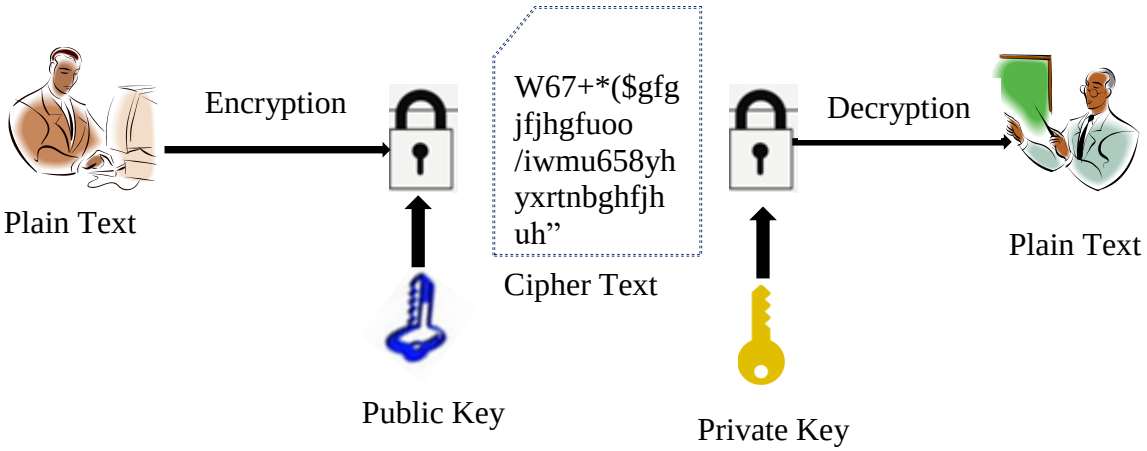


Figure 2.3 Asymmetric key Cryptography

Asymmetric key cryptography is essential for transferring secure data even when both communication parties are unable to agree on a certain secret technique. A longer key is used to improve data security while it is being transferred. Public cryptography's disadvantage is that it performs less quickly than private key encryption. By combining

the two, we can protect our file more effectively. For these theses, we select RSA from asymmetric key cryptography and AES from symmetric key [30], [31].

Table 2.1 Comparison between Symmetric and Asymmetric Cryptography

Symmetric	Asymmetric
Use algorithm like AES,DES,3DES,RC4	Use RSA, Diffie-hellman, DSA Algorithm, ECC
Symmetric encryption is fast in execution	Asymmetric encryption is low in execution because of high computational burden
Use only one key for both encryption and decryption process.	Use two different keys one for encryption and one for decryption process.
The best benefit of this algorithm is confidentiality.	Best benefit is confidentiality, authenticity and non-repudiation.
Simple form of encryption.	Complex form of encryption due to high computational.
The decoded text is less than or equal to the original text.	The cipher text is larger or equal to the plain text.
Used to encrypt large amount of data's	Used to encrypt small amount of data's
Secure for both provider and user.	Secure only for user.

To implement the encryption mechanism, several techniques are available. There are DES, 3DES, Elgama, Diffie-Hellman, and more algorithms. The diffie-helmman, DSA, RSA, and AES are discussed next [32].

First used in 1976, the Diffie-Hellman key agreement technique was developed by Whitfield Diffie and Martin Hellman. For generating shared secrets via insecure networks, it is the first practical method. Diffie Hellman is, in other words, an encrypted end-to-end key exchange system that provides sufficient information to two parties to allow for secure communication without the need for a secret key to be disclosed. It is used to exchange keys between sender and receiver. One private key and one public key

are used in these key exchange systems. In order to create the secret key utilized in the encrypted key exchange, two parties' public keys and private keys are combined. The parties, who may not have previously met, make use of the shared secret key in their subsequent communications [33].

The message is encrypted by the sender using both private and public keys. The recipients then decrypt the encrypted message using their own private key as well as the sender's.

However, the DH key exchange does not allow for the use of signing or digital signatures. Key exchange is susceptible to man-in-the-middle attacks since it doesn't verify any parties involved in the key exchange.

A. Digital Signature Algorithms

DSA is the name of the other public key cryptography system that is employed for data authentication and integrity checking. DSA was performed to enable SHA-based signature creation and validation. Using its private key, a sender can create a digital signature to encrypt a message before sending it to a recipient [34].

The recipient can then use the sender's public key to validate the signature once they have received the message. When it comes to signing and confirming, it is compatible with DSA. However, the computation and validation required by DSA are substantial. Only used for authentication, the data is not encrypted. Cryptocurrency using DES Cryptography algorithm.

B. DES Encryption Cryptography

It stands for the 1977-founded Data Encryption Standard. Digital data is encrypted using a symmetric key technique known as DES. DES encryption divides the plain-text into two equal halves and then uses a 56-bit key and 64 bits of plain-text to create a 64-bit cipher text, or encrypted form of the data. The DES encryption method only employs 56 bits of the key length, even though the block size is 64 bits (the remaining 8 bits are used simply as check bits). No of the key length, DES needs 16 rounds of the same procedures. When used against DES, linear crypto analytic attacks are highly potent. It is susceptible to brute force assaults because of the weak keys, the limited

amount of operations in DES, and the ban on permutation combinations. It is less secure than AES because of its short key length of 56 bits, which is inadequate to protect the bulk of current applications that rely on encryption [35].

C. 3DES Encryption Cryptography

Triple Data Encryption Algorithm, sometimes referred to as 3DES or TDES, is its full name. This symmetric-key block encryption applies the DES algorithm three times to each block. It has a 112 or 168-bit key length and a 64-bit block size. It is based on DES and makes use of the Feistel network. Since DES's key length is too short, 3DES was developed as a more secure replacement. Even though the DES algorithm is iterated through three times with three keys, 3DES cannot be considered secure without the usage of three unique keys.

Standard DES's flaws became increasingly apparent, leading to the use of 3DES in a number of applications. It was one of the most often employed encryption methods before AES became well-known. Modern cryptography techniques and super computers have led to certain severe weaknesses in 3DES, just as the DES.

D. AES Encryption Cryptography

2001 saw the development of the Advanced Encryption Standard. After triple-DES was found to be slow, AES, which is six times faster than triple-DES, was created. Today, it ranks among the most popular symmetric block cipher algorithms. It works with bytes instead of bits. AES is yet another type of encryption that prevents malevolent actors [36]. It is among the most effective encryption techniques in use today. AES effectively strikes a balance between speed and security, allowing us to continue using the internet without interruption.

AES is a form of symmetric encryption since it uses the same key for both encryption and decryption. The three distinct AES key sizes are 128, 192, and 256 bits. Depending on the length of the key, there are several key combinations. It uses the substitution permutation network and differs structurally from earlier encryption techniques. Consequently, we choose to employ AES [37].

The essential objectivity to fulfill the following fundamental security services: -

Confidentiality: is to prevent the protected data from being disclosed without authorization. The possibility of a growth in the number of access points due to the fact that several devices and apps can access cloud storage raises the risk of unauthorized disclosure. Therefore, new techniques like encryption must be used in order to guarantee the privacy of the data saved in cloud storage [38].

Integrity: Protecting data from being altered by unauthorized individuals is referred to as maintaining data integrity. Authorization procedures are used since it is a severe problem in the cloud environment. For each authenticated user to prevent access from unauthorized users, the authorization provides the access privileges [39]. The protected data must be assured that only authorized entities are able to access it, nonetheless, due to the rise in access points and system entities. An effective way to guarantee data integrity in a cloud context is through the implementation of a digital signature

Availability: Information accessibility refers to enabling authorized parties to access the data when required. These security services may be offered using a variety of more powerful tools and approaches that cryptography offers. One cryptography method is encryption, which uses a set of guidelines known as an encryption algorithm and an encryption key to convert data into cypher text. Data may seem radically different from the initial data and appear random as a result of this. The receiver will then get the cypher safely, and using a decryption key, they will be able to use it to restore the original data. Both the data encryption methods AES and RSA are examples of data communication. Advanced Encryption Standard (AES) and RSA (Rivest, Shamir, and Adleman) are both acronyms for encryption technologies [40].

2.1.3. AES Algorithm

NIST, the National Institute of Standards and Technology, is responsible for publishing AES. Advanced Encryption Standard, which goes by the abbreviation AES, is a Symmetric Encryption algorithm. Two Belgian Cryptographers named Joan Daemen and Vincent Rijmen created AES. AES is helpful for converting plain text into cipher text or other unintelligible forms. Only the right password can decode encrypted text when the original or plain text is required

AES has a 128-bit minimum key length and is suitable for a variety of applications. Furthermore to having a good length, the key also resists timing and power attacks, despite the fact that there were few available resources. With the Advanced Encryption Standard (AES), various techniques are used to provide security. AES is better, but it falls short of the essential level of security in some situations, such as those involving brute force attacks. As a result, Rivest-Shamir-Adleman (RSA), a further layer, is incorporated into AES. The security issues caused by Trojan horses and brute force attacks are eliminated by the RSA method. The data is more securely guarded from assaults as a result of the two-layer strategy. The AES utilizes various three keys with their respective rounds and a constant block size of 128 bits.192 bit keys have 12 rounds, 128 bit keys have 10, and 256 bit keys have 14. AES has 3 Block ciphers, namely [41]:

AES is a block cypher with a symmetric encryption technique. The AES algorithm's operation may appear complicated, but it is actually rather easy to comprehend. AES contains three block cyphers, including:

AES-128: A key is used to perform the encryption and decryption procedures. This block cypher, which is also the least secure of the three accessible blocks, encrypts and decrypts messages using a key that is 128 bits long. Despite the fact that AES-128 has never been compromised, its resistance to brute-force attacks is concerning. AES-128 has some security flaws, yet it is still highly quick and effective at encrypting data. The technique encrypts data in 10 cycles with 128-bit keys [42].

AES-192: It uses a key that is 192 bits long for the encryption and decryption of messages. It is more resistant to brute-force attacks as compared to AES-128 because it has a longer key, thereby more secure. Despite this, AES-192 is not commonly used, and people tend to lean towards AES256. To use the algorithm for encryption data the process consists of 12 rounds for a 192-bit long key. AES-256: It uses a key 256 bits long for the encryption and decryption of messages. This block is more secure when compared to the AES-128 and AES-192 because of the long length of the encryption key. The longer the encryption key, the more difficult to hack [43]. AES-256 is consequently far more resistant to brute-force attacks than the two earlier blocks. AES is an asymmetric block cypher, which means that only the sender and the recipient of the message have

access to the secret key used for encryption and decryption. The key used to encrypt the communication is often the same key used to decode it at the other end. The procedure takes 14 cycles to employ for a 256-bit key to encrypt data. This demonstrates that the message was encrypted using one of the three encryption keys. The encrypted message is created in the "cypher text" and the encryption procedure is carried out in the “cipher” [8].

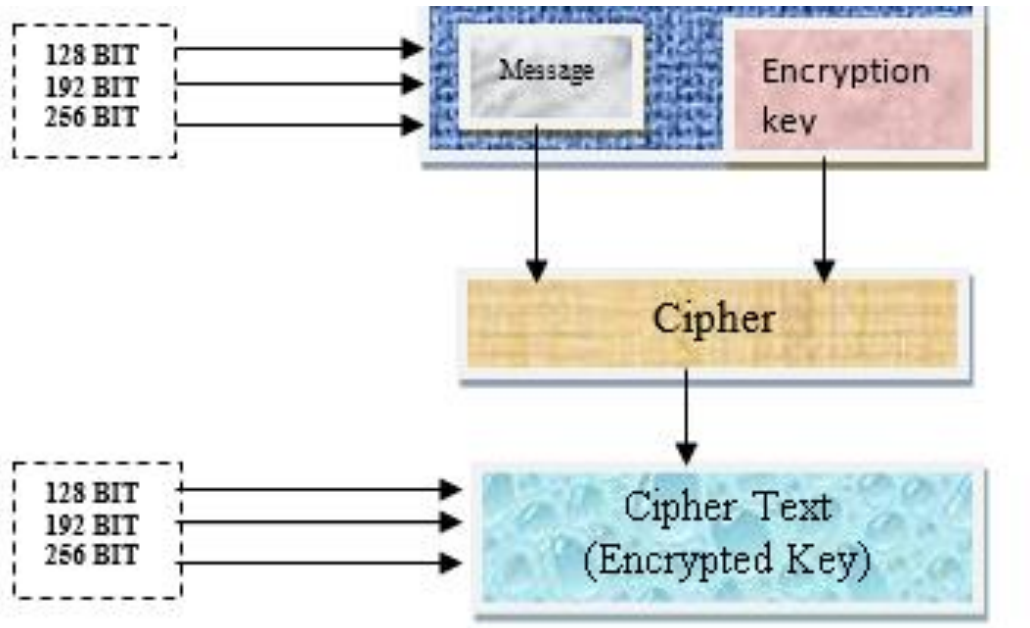


Figure 2.4 Flow of AES Encryption

1. **Sub Byte Steps**

This is the technique through which the text or data is encrypted and rendered unintelligible. The 16 inputs of the 4x4 matrix array are changed byte per byte during the sub bytes step. The sub byte S (i, j) is used to replace the (i, j) with the aid of an 8 bit replacement box.

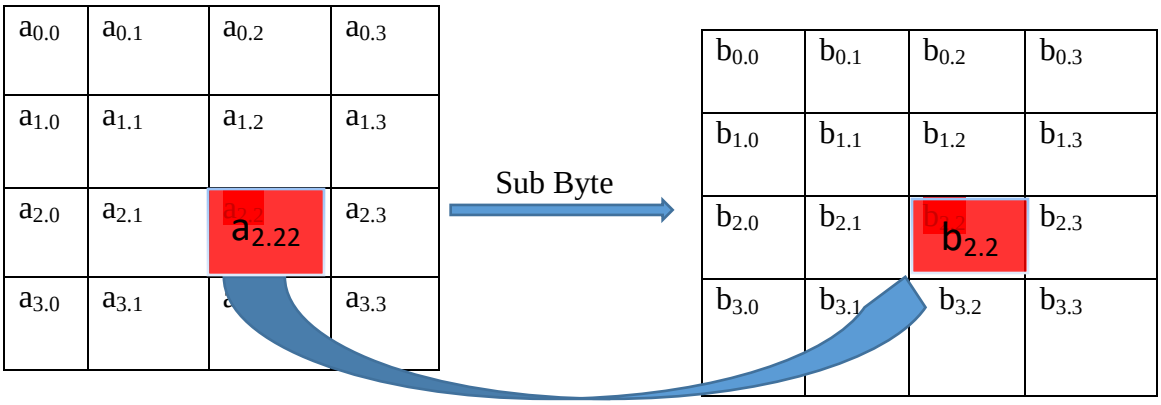


Figure 2.5 Sub Bytes

2. Permutation Steps

Shift rows come next, after sub bytes. On the row's current state, this step operates. Every byte is left-shifted in a circular fashion. While the second, third, and fourth rows all move one byte to the left, the first row remains unchanged. The procedure is depicted in the following graphic.

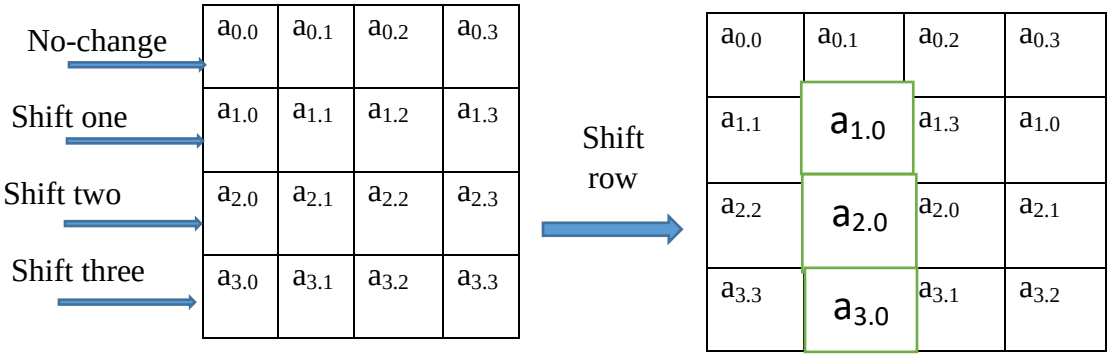


Figure 2.6 Shift Row

3. Mixing the Column Steps

A fixed polynomial c(x) is multiplied with each column of the state in the mix columns step. We get a new matrix with 16 new bytes after the multiplication [8], [44], [45].

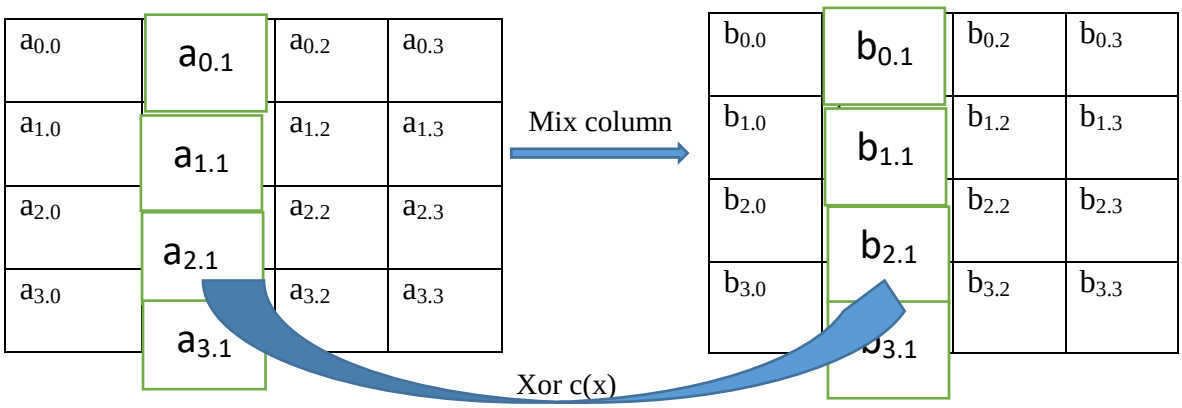


Figure 2.7 Mix Columns

1. Add Round Key Steps

With this stage, the text encryption process is finished. The state and the sub key is linked during this phase. Each cycle of the main key generates a sub key using Rijndael's key scheduling. The state's size and the sub key's size will be equal. Each byte of the state and the corresponding byte of the sub key are combined using bitwise XOR [44].

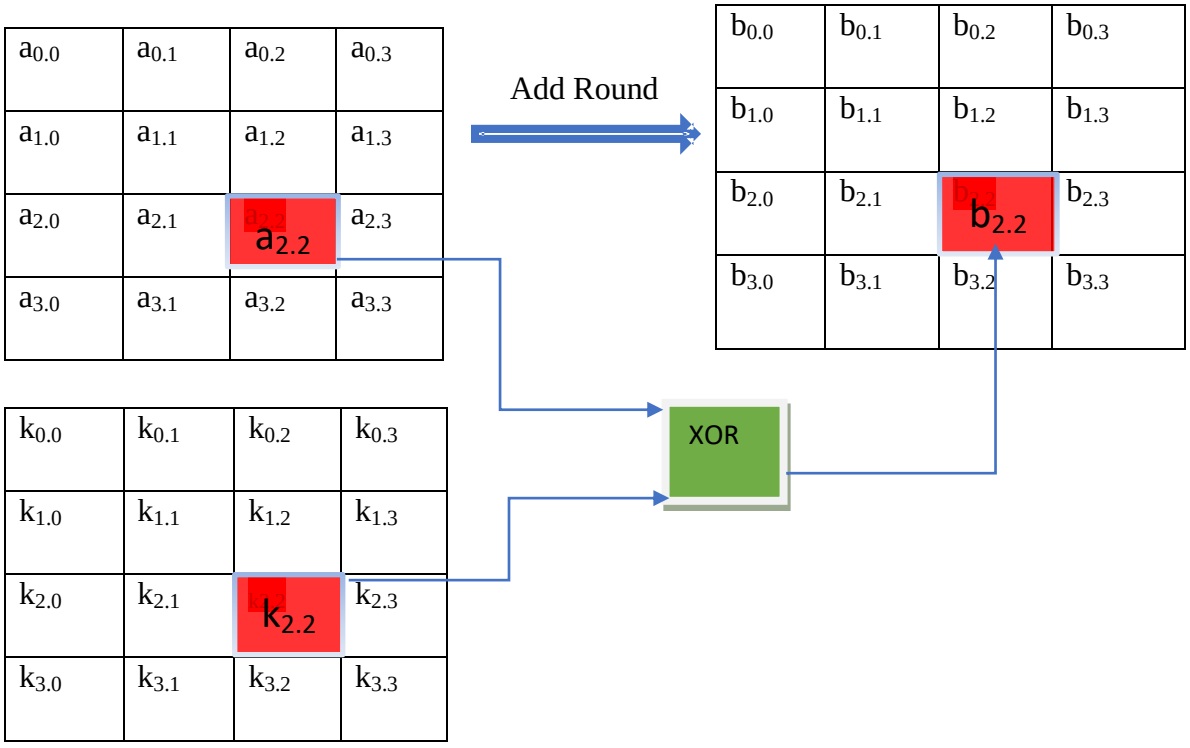


Figure 2.8 Add Round Key

2.1.4. RSA Algorithms

Rivest-Shamir-Adleman, or RSA, is one of the most well-known asymmetric key algorithms for key exchange or data encryption. Two keys are used in asymmetric encryption; one is used for encryption and the other for decryption. When the key value is large, RSA security is improved. The key pair is made up of two very big prime numbers, the product of which yields [8]. Finding the N factorial is hence difficult if key is a large number. The RSA process is divided into three steps. There are descriptions of phases 1, 2, and 3 of Encryption and Decryption. Key Generation Process of RSA. RSA involves two key public and private. Public used to encrypt the data and private use for decrypt the encrypted data [45]–[47].

The generation of key takes the following step:

- Step 1: Select two prime numbers, x and y, and $x \neq y$.
- Step 2: Calculate $n = x * y$. Calculate $\varphi(n) = (x - 1) \times (y - 1)$.
- Step 3: Select integer e such that GCD $(e, \varphi(n)) = 1$ such that $1 < e < \varphi(n)$.
- Step 4: Calculate the private key, d = e^ (-1) (mod φ (n)).
- Step 5: Then public key = {e, n}.
- Step 6: Private Key = {d, n}.

1. Encryption Process of the RSA

Encryption is the process of converting plain text (original) message to cipher text (unreadable) form by using the formula. Encryption: $C = m^e \bmod n$.

2. Decryption of the RSA

Decryption is process of converting unreadable data to original.Decryption:$M = C^d \bmod n$.

Where M=plain text and C=cipher text.

2.1.5. LZW Compression

A type of lossless compression method called LZW shrinks files without sacrificing their quality. Abraham Lempel and Jacob Ziv were the authors, and Terry Welch later released it in 1984.It is the best way for all-purpose data compression since it is simpler and more

flexible than other lossless comparison methods. Compression also has no errors. The approach depends on the concept that integer codes take up less memory than string literals and has no effect on our data. The LZW algorithm first reads character sequences, groups them into strings of recurrent patterns, and then converts them. They are connected because both compression and encryption seek to reduce file size without compromising file quality [48], [49].

2.1.6. Text Security

Security is by far the biggest issue with text storage. Security concerns are particularly common in the cloud computing environment since data is stored across several computers and storage media. Text security used in cloud computing is more complicated than text security used in more traditional methods. Because of this, the confidentiality must be ensured by the encryption and decryption procedures in order to shield our data from illegal access and alteration and to feel secure or confident about it being kept on the cloud [50]–[52].

Texts are used in a variety of applications, including the military, medical systems, classified papers, internet transmitted texts, and more. These texts are now given online from a variety of sources. Only users who have been verified should have access to or get these texts since they include certain important and secret information that has to be safeguarded from leakage [53]–[56].

AES and RSA are examples of symmetric and asymmetric encryption techniques. When we employ both of them, the data is more secure since only those who have been authenticated are allowed access [57].

2.1.7. Text Compression

Text compression is either the act of compressing the original text into a straightforward symbol or the process of transforming texts into symbols that are smaller than the original text using certain encoding techniques. By using text compression, data may be transmitted using less bandwidth and in a smaller amount of space than before [58].

They come in two varieties: lossy and lossless compression. Lossless compression preserves the original data, or it can reduce the size of text without compromising the

quality of the text, whereas lossy compression results in some information from the original material are being lost. It uses text-based information. The most common techniques are run-length encoding, Huffman coding, Shanon Fanon, LZW, and arithmetic encoding.

2.2. Related Works

In the related works, different related researches which are the same area and scope are reviewed. There have been different researches done on the security of cloud storage with AES, RSA, and other algorithms for encryption /decryption purposes. Some of the research works which has been done in AES/RSA are stated as follows.

2.2.1. Implementation using AES &RSA Algorithm

In [59] the area of cloud computing data security, the study presents multiple advances. In order to improve the safety and confidentiality of data in the information center of the cloud server, it suggests a hybrid solution that combines stenography with standard key cryptography methods. To uphold the notion of data security, the division and merger concept is presented. To guarantee that only people with permission are allowed to see the encrypted data, the paper discusses access control and sensitive data segregation. An extra degree of security is provided by the application of public-key and symmetric-key encryption techniques. The efficiency gain of the suggested algorithms as a matter of runtime when compared to current encryption techniques is also highlighted in the conclusion. The lack of an evaluation and a comparison study, which would have offered stronger proof of the efficacy and efficiency of the suggested strategy, is one of the paper's shortcomings. High-level security by hybrid of public-key cryptography techniques is recommended to be investigated further. In general, the study advances the field of safe cloud data management; however, more investigation and assessment are required to fully confirm its efficacy.

[60] The two modules that make up the suggested system are the upload and download modules. The Upload Module allows users to create encryption keys, securely upload files, identify themselves, and encode what they upload using their public key. While temporary documents are deleted, the data that has been encrypted is kept in the user's cloud document directory. Users can decrypt cloud data by entering their secret private

key and username in the Download Module. After decoded, the user receives a download link. Restrictions: Input Security during Upload: The fact that the data is first encrypted and then momentarily stored on the cloud presents one possible drawback. If suitable safety precautions are not in place, there is a chance that data will be open to assaults during this short window of time. Before uploading, the data should be encrypted to reduce this risk.

[61] The work makes a contribution by emphasizing the necessity of adequate safeguards in cloud computing and the significance of encryption techniques in protecting cloud data. It offers a comparison of several algorithms according to a number of factors, including encryption time, memory usage, and execution time. Finding each algorithm's advantages and disadvantages through analysis makes it easier to choose the best protection strategy for the cloud. Restrictions: The paper's main shortcoming is that it solely compares the encoding methods that have been stated, ignoring other crucial facets of information safety including handling keys, authentication processes, and safe transmission protocols. A comprehensive strategy for cloud security of data would take these more variables into account in addition to the encryption scheme choice.

[62] To overcome the drawbacks of utilizing RSA alone, the study offers a hybrid encryption method that combines the AES and RSA algorithms. This is a significant advance. The suggested technique mitigates the issues related to RSA restricted data limitation and high electrical usage and allows for the safekeeping of higher data volumes by utilizing AES encryption for the encryption of data and RSA to encoding its AES keys. The paper's disregard for outsiders watching the data encoding and upload process, however, could be a weakness. It is advised to utilize a Virtual Private Network, or VPN, to conceal the system's local addresses while uploading and encrypting data in order to increase security. By encrypting connection between the user's machine and the cloud, a VPN offers an additional layer of security, protecting the confidentiality and integrity of data while it is being transmitted.

[63] In order to encrypt and decrypt image files of various sizes, the researchers in this work separately built the AES & RSA methods employing MATLAB. They discovered that the length of time needed for both algorithms' encryption and decryption increased

with file size. Both AES and RSA encryption processes, they found, typically required less time to complete than their decoding counterparts. They did point out, though, that RSA required less time to encrypt data than AES. RSA demonstrated greater speed and security, but AES scored higher when it comes of expense, safety, and execution, according to their evaluation. The research makes the argument that greater anticipated execution duration could arise from the use of a combination encryption algorithm that combines RSA and AES. But it's also anticipated to offer improved data security. Regretfully, neither the anticipated time frame nor the degrees of safety attained through the mixed technique are made clear in the study.

[64] The proposed method combines data security and BWT compression, with compression levels based on file size and repetitive sequences. It addresses vulnerabilities related to key and plain-text attacks and is suitable for secure file transfer, cloud storage, messaging, IoT, finance, and healthcare. BWT lacks an adaptive dictionary, requiring character rearrangement and computationally intensive steps. LZW uses an adaptive dictionary-based approach, dynamically building during compression, capturing recurring patterns more effectively. This approach offers faster compression speeds and reduces computational complexity.

[65] This thesis explores compression algorithms for time series data, comparing PMC-MR and Swing filters. It suggests a hybrid approach, exploring alternative algorithms like Slide filter. The research also explores different data types, multivariate techniques, and timestamp compression beyond Gorilla. We recommend Lempel-Ziv-Welch (LZW) for text compression, as it efficiently encodes repeated patterns and Burrows-Wheeler Transform (BWT) for efficient encoding of repetitive patterns. Future research aims to advance time series compression and improve efficiency.

[66] The paper discusses data compression techniques and their effectiveness. Huffman Coding is a superior technique due to its better compression ratio, while Arithmetic Coding is the most powerful but slower. Combining techniques can enhance compression ratios. Arithmetic Coding reduces channel bandwidth and transmission time. However, LZW compressions require a larger dictionary and text data characteristics. Cloud storage presents inherent risks, making the placement of sensitive data potentially risky. To

ensure comprehensive security, multiple measures, including access control, encryption, auditing, and redundancy, are essential.

[5] This paper reviews challenges in enhancing cloud storage security, evaluates existing approaches, and discusses emerging technologies. The next stage of research aims to introduce a comprehensive framework that emphasizes integrated security layers in a dynamic and localized manner, providing security on demand. Customization of security measures based on specific requirements is crucial, and future research should focus on developing a tailored framework for diverse cloud systems and resources. we recommend Mega cloud .Mega cloud storage offers enhanced privacy and security through end-to-end encryption, ensuring data is only accessible to the user. This provides greater control and ownership over data, unlike public cloud storage services. Mega also provides generous free storage, making it suitable for individuals or small-scale users with limited budgets, allowing them to store a substantial amount of data without additional costs.

[44] In this study, the researchers focused on the security and encryption methods employed by ten mobile cloud storage applications, namely 4Shared, One Drive, Mega, SurDoc, Cubby, ADrive, Safe Sync, Team Drive, Wuala, and Just Cloud. The methodology involved analyzing intercepted communications to determine the encryption protocols used by these applications. The researchers utilized Wireshark to check for plaintext in the intercepted packets and employed a histogram analysis for text files. Additionally, various cryptography protocols were examined, including IPsec, AES encryption, SSL, and TLS. The findings revealed that 4Shared employed its own security cryptography protocol with 128-SSL encryption, Mega utilized AES-128-bit and RSA-2048 encryption, and Just Cloud implemented 256-bit AES encryption. Other applications, such as Wuala, SurDoc, ADrive, and One Drive, also demonstrated encryption measures to ensure data security. From ten cloud storage application we select the mega cloud storage because it utilized AES 256 and RSA2048 and also we encrypt and decrypt the sensitive user data by using AES 256 and RSA2048.

[45] The paper highlights the importance of selecting encryption techniques and key management in cloud storage services for system security, confidentiality, and privacy. Different levels of encryption provide varying protection against attacks. End-to-end

encryption mega cloud storage is effective, but server-side encryption be challenging. The findings help users choose the best service provider based on their security needs and data integrity.

[8] Although the proposed system acknowledges the importance of implementing encryption and decryption techniques to secure data in the cloud, there are still areas that require further enhancement. The research suggests expanding the key length of the AES algorithm to enhance security, increasing storage node availability, minimizing memory usage, and reducing costs. Additionally, the system could benefit from selecting storage nodes from different cloud providers to diversify risk and optimize cost calculations. Furthermore, future work should focus on improving the speed of the decryption process and exploring compression techniques to reduce the size of cover images and the key length can be expanded using any other key generation algorithm. The AES algorithm uses 128 bits. This includes ten rounds or cycles of AES algorithm. They recommend as, in future this can be extended to 192 or 256 bits. If 192 bit key is used, the number of cycles will be 12. When the key size is 256 bits there are 14 rounds. The increased key size can produce more number of keys and also the security can be enhanced.

2.2.2. LZW Compression Algorithm

Huffman encoding, arithmetic coding, run length encoding, entropy encoding, Lempel-Ziv-Swelch, deflation and chain codes algorithm are types of lossless compression. From these compression LZW algorithm is the most in the terms of compression ratio, types of data compressed, and faster computation time. Moreover it is simple and good compression, dynamic code word table built for each files and decompression creates the code table.

2.2.3. RSA and AES Cryptography

The RSA and AES cryptography or asymmetric and symmetric cryptography algorithm is used to make fast and more secure the data when we use the RSA for Encryption of the AES key and then after AES key is encrypted by RSA, the plain text message is encrypted by AES key already encrypted by RSA.so the combination of symmetric and asymmetric cryptography algorithm is more beneficiary than the single ones. When we

encrypt the data by AES we get 098765, and in RSA we get 0987654, but in hybrid means AES and RSA we get 09876509876,

2.3. Summary of Related Work

Table 2.2 Summary of related work

References	Title	Contribution	Limitation
[59]	Secure File Storage on Cloud Using Cryptography.	The research proposes a hybrid solution that integrates stenography with conventional key cryptography techniques.	The suggested strategy's efficiency could have been more robustly demonstrated through an evaluation and comparison study.
[60]	Enhancing The Data Security In Cloud By Implementing Hybrid (RSA & AES) Encryption Algorithm.	The proposed system includes upload and download modules.	The data is encrypted before being temporarily stored on the cloud, which may pose a potential security risk during upload.
[61]	Secure User Data in Cloud Computing Using Encryption Algorithms	The work underscores the importance of robust security measures in cloud computing and the significance of encryption techniques in safeguarding cloud data.	The comparison of encoding methods overlooks other crucial aspects of information safety, such as handling keys, authentication processes, and safe transmission protocols.

[62]	A Comparative Analysis of Cryptography Algorithms:	The study presents a hybrid encryption method that combines AES and RSA algorithms, using RSA alone.	To protect data encoding and upload process from outsiders, it is recommended to use a Virtual Private Network (VPN).

Generally: To securely store sensitive data on the cloud, several challenges must be addressed. Firstly, trust in the cloud's security is crucial. AES alone has limitations, such as the same key for encryption and decryption. RSA is suitable for small data sizes. Combining AES and RSA provides the best solution. However, uploading data exposes our location. Using a VPN hides our location, but it can introduce redundancy in the uploaded data. To overcome these challenges, a proposed approach involves creating the sensitive data, compressing it using LZW lossless compression.

CHAPTER THREE

3. METHODOLOGY

3.1. Introduction

To meet the research objectives, a large number of publications, books, journals, and the internet where searched through. Materials on the use of cryptography techniques to safeguard user data stored in the cloud were also looked at. The proposed method confirms that different levels of confidentiality are enforced at the site of data processing using double encryption, and that unauthorized disclosure is prevented. This proposed system used a variety of methods to determine the security of user data kept on the cloud. The relevance of several components of the proposed system is described, as well as the techniques to be used in their design. It covers the study design, data types and sources, methods and techniques, algorithms, data analysis and presentation, among some other things. Cryptography is used to keep user data safe in the cloud. Python and other tools are used to implement Encryption. The method described in the study protects the data from unauthorized disclosure and modifications. Additionally, the method ensures availability and confidentiality. Data is better protected using the suggested hybrid method. As a result, it stands out.

3.2. Methods for implementation

This study used Python for implement a solution used hybrid (AES and RSA) encryption algorithms for data security. LZW lossless compression algorithm was used for efficient data storage. Mega cloud storage was used for secure storage, providing end-to-end encryption. Cyber Ghost VPN was used to protect online activities and maintain privacy. These methods ensure data security, reduce size, and protect against unauthorized access.collect the data from individual(Daniel Moti).

3.3. Development Tools

Different tools and libraries are used to perform the implementation and analysis results. In this paper, the python programming language, System type:64-bit operating system, x64-based processor; Edition: Windows 10 Pro, Processor: Intel(R) Core(TM) i3-8100 CPU @ 3.60GHz 3.60 GHz ,Installed RAM: 4.00 GB (3.83 GB usable), System type:64-

bit operating system, x64-based processor. This Research work explores cloud data security using various methods, techniques, and algorithms.

3.2.1 Data set Description

Cryptography algorithm is need a data or sensitive data to encrypt and decrypt that data for security. We have collected the sensitive data from Daniel Moti his project. The original PDF file consists of 80 pages and has a size of 4,616,263 bytes before compression. After compression, the file size reduced to 4,475,411 bytes. The content of the PDF file is a project titled "Design of Technology Integrated Shredder Machine" and was submitted to the Department of Mechanical Engineering at Gambella University by Daniel Moti Adudgna.

3.4. Algorithm Implementation

This study suggests that employing Hybrid (RSA and AES) encryption techniques in conjunction is one of the potential security measures for protecting cloud storage. A secure system may offer speed, scalability, and security, which is the recommended technique. The implementation of the two algorithms together is as follows: shipments to import:

CHAPTER FOUR

4. PROPOSED SYSTEM AND ARCHITECTURE

4.1. Overview

The system, design, and architecture provided in this chapter are discussed in the methods section.

4.2. Proposed System AES-RSA Design For Secure the User Data

The AES-RSA technique is used to encrypt and decode provided data. Text files are encrypted using the AES technique, and the AES key is encrypted using the asymmetric RSA algorithm to prevent third parties from confirming the transfer between clients or clients and servers and to make it more difficult for attackers to access. In this approach, instead of using distinct stages for each stage of execution, the original data enters the system and is produced as encrypted data using an AES key and an encrypted RSA public key. The decryption procedure is completed at the receiver. The inverse of an encryption process is a decryption process. This encrypted data is decrypt using an RSA private key that is created by the sender to acquire the original file, which is the same size as the original data.

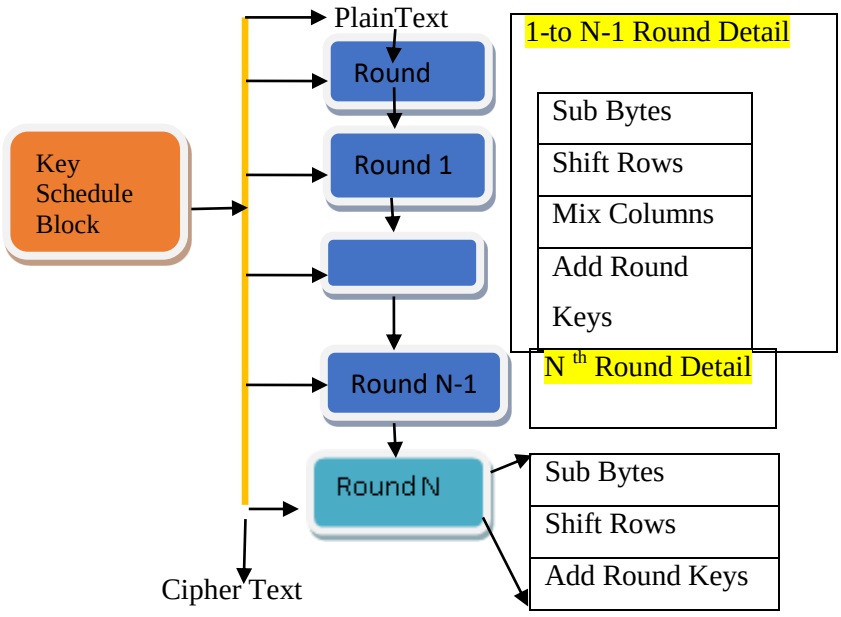


Figure 4.1 AES Grouping Algorithm

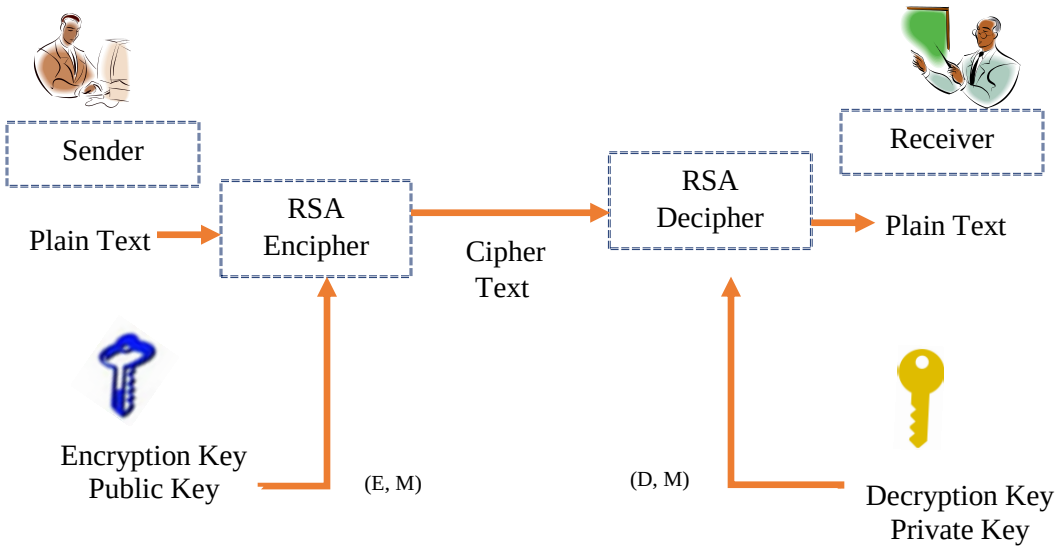


Figure 4.2 RSA Algorithm and Encryption

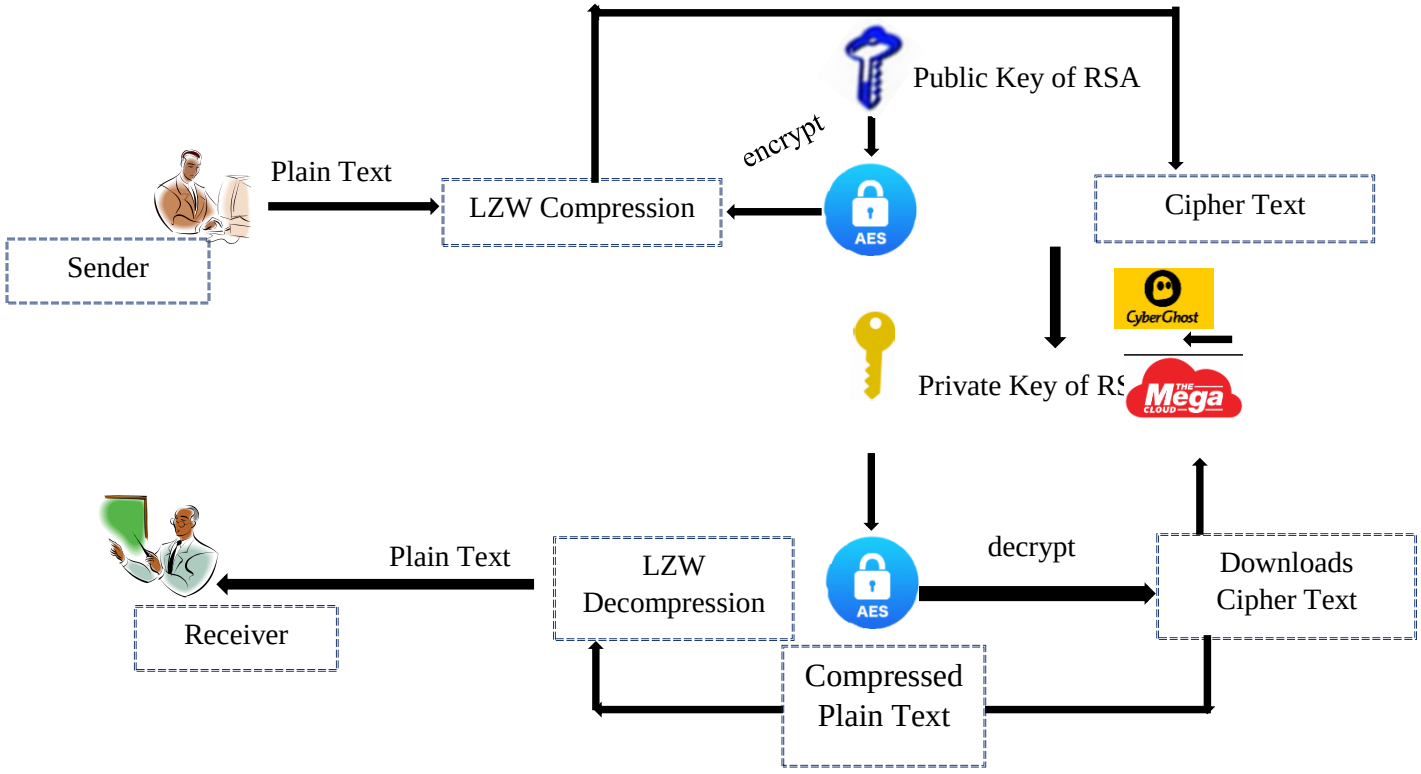


Figure 4.3 Model Architecture of Secure User Data

On the sender's end, compress the text, encrypt the AES key using RSA, and then encrypt the AES message using the encrypted AES key before sending to the Cloud storage

provider (MEGA). On the receiver side, get the cipher text, decode the AES key using RSA, decrypt the cipher text using the AES key, decompress the text, and obtain the original text.

During the receiver need to download the files from the cloud, we make use of the Cyber Ghost virtual private network (VPN) service, which can protect user information by encrypting internet traffic and disguising the user's IP address. It mainly concentrates on evaluating the effectiveness of text data encryption and decryption methods utilizing the AES and RSA algorithms, LZW compression, MEGA cloud storage, and Cyber Ghost VPN for safe storage and internet access. The paper highlights how these methods increase algorithm strength, key generation, and decryption speed to guarantee the security and privacy of sensitive user data.

4.3. Experimental Setup

The tools used for implementation were installed on a personal laptop computer Dell 11th Gen Intel(R) Core(TM) i3-1165G7 @ 2.80GHz 1.69 GHz Installed RAM 16.00 GB (15.7GB usable), 64-bit operating system, x64-based processor. The operating system is Windows 10 Professional, 64 bits. The latest version of Python 3.10.

4.4. Implementation Algorithm

Key Generation of RSA

The RSA key pair produced by this script may be used to encrypt and decode data. To create the key pair, the script makes use of the RSA module included in the Crypto Public Key package. First, the Crypto Public Key package's RSA module is imported. The generate() function of the RSA object is then used to build an RSA key pair with a key size of 2048 bits. This produces a new RSA key object that contains the public and private keys. The script then employs its export_key() function to obtain the private and public keys from the RSA key object. The public key is taken from the public key() function of the RSA object and placed in a different variable, whereas the private key is directly retrieved from the RSA object and stored in a variable.

The script then saves the private and public keys to separate files with the names "Private.pem" sand "Public.pem," respectively.

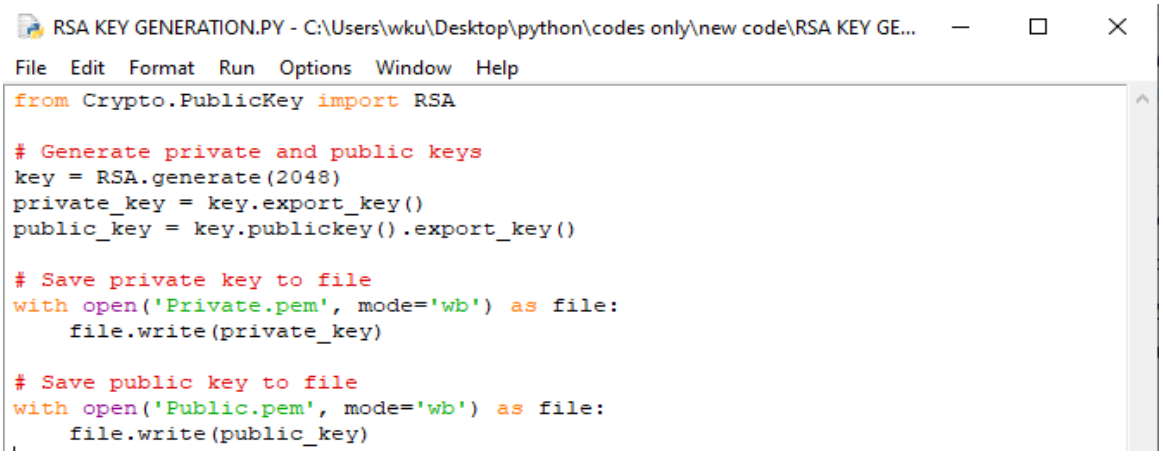


Figure 4.4 Key Generation of RSA

To generate AES Key

The receiver must compute the following fundamental phases in the keys generation process: Since we are using AES 256 keys on these theses, the first sender sends the original message, which is then XOR with the key and substituted using an S-box. Next, we shift the output of the S-box, after which we shift the mix columns and obtain the first's cipher text. Finally, we add the round keys, and we repeat the process up to 14 rounds. Using the OS.u random() method, this script creates a 256-bit AES key and saves it to a binary file called "key.bin".

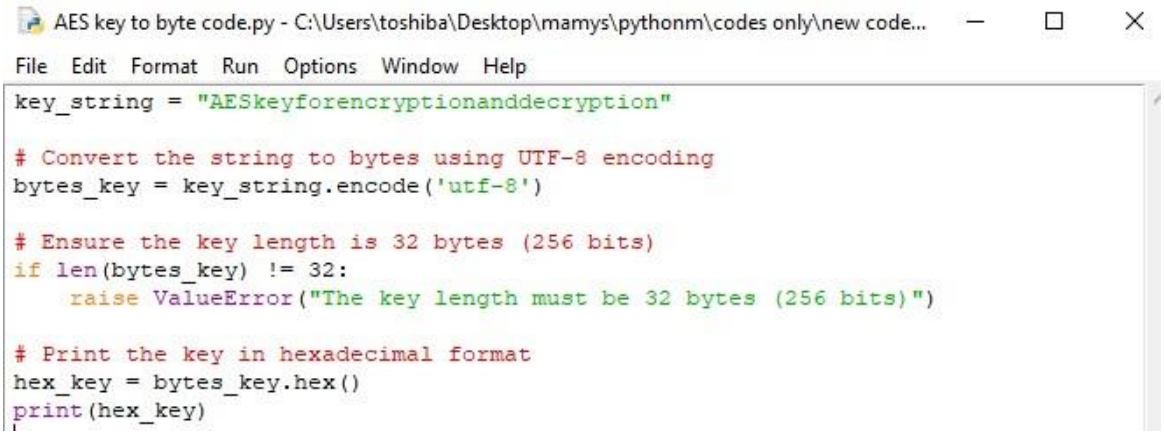


Figure 4.5 Key generation of AES

To Encrypt the AES Key by Public Key of RSA

This Python program imports an RSA public key from a.pem file, loads an encrypted AES key from a binary file called "encrypted_aes_key.bin," uses the RSA private key to decrypt the AES key, and then saves the decrypt AES key to the binary file "decrypt_aes_key.bin". The decrypt information is kept in the file 'decrypt_aes_key.bin'. Overall, utilizing an RSA private key to decrypt an AES key, this script offers a quick and effective method for doing so. The decrypt key can then be saved to a file and used to encrypt and decrypt data.

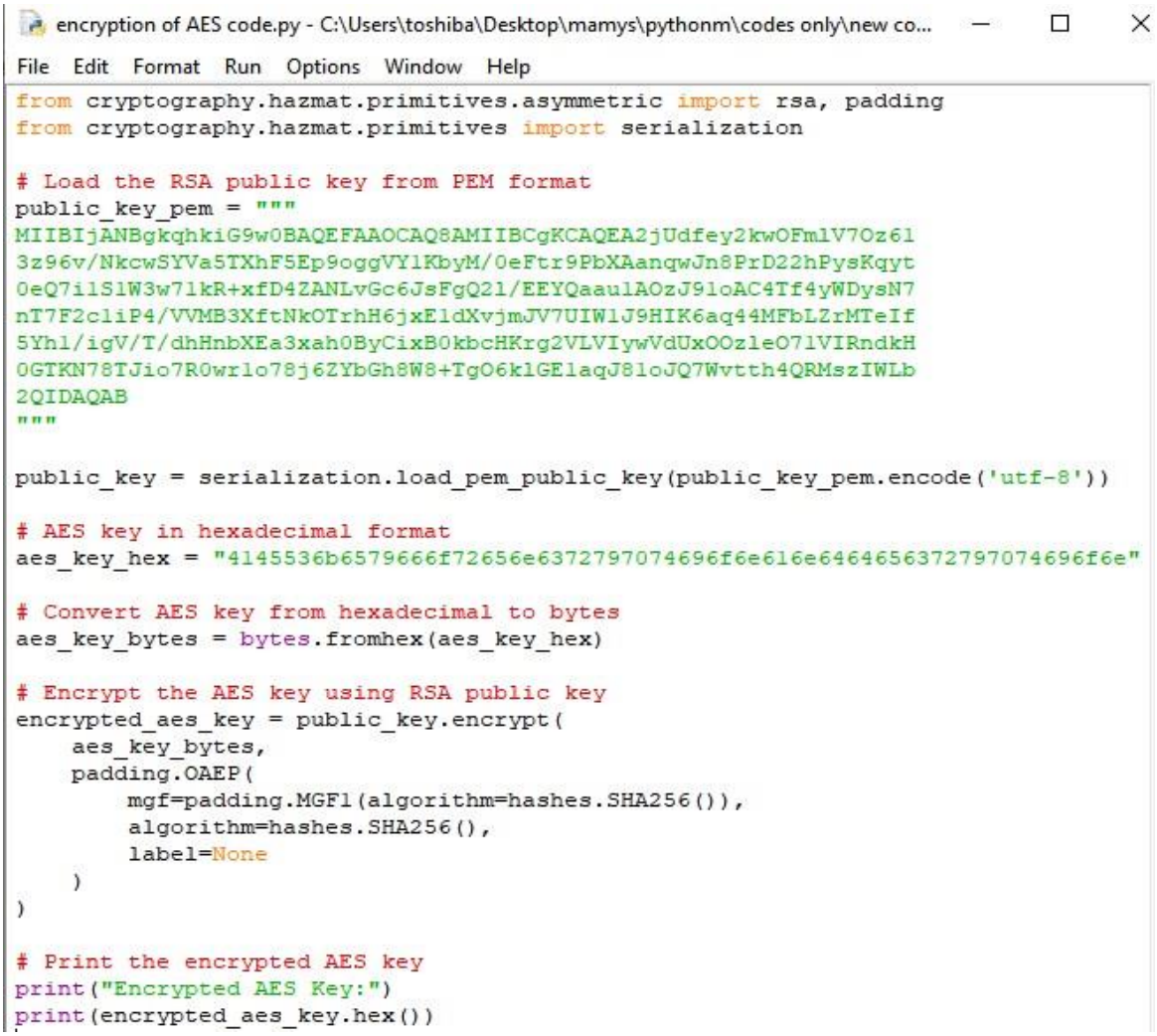


Figure 4.6 Encrypt the AES Key by Public Key of RSA

To Compress Data by LZW loss less Compression

This Python program imports data from the "WHO-COVID-19-global-data.csv" CSV file, transforms the data to a string, and then compresses the string using the LZW technique. The compressed string is then saved to the "compressed_data.bin" binary file.Using the write() method and the open() function in binary write mode, the compressed data is written into the file 'compressed_data.bin' for storage. Using the LZW method, this script offers a quick and easy approach to compress data without sacrificing data quality, which can result in large file size reductions.

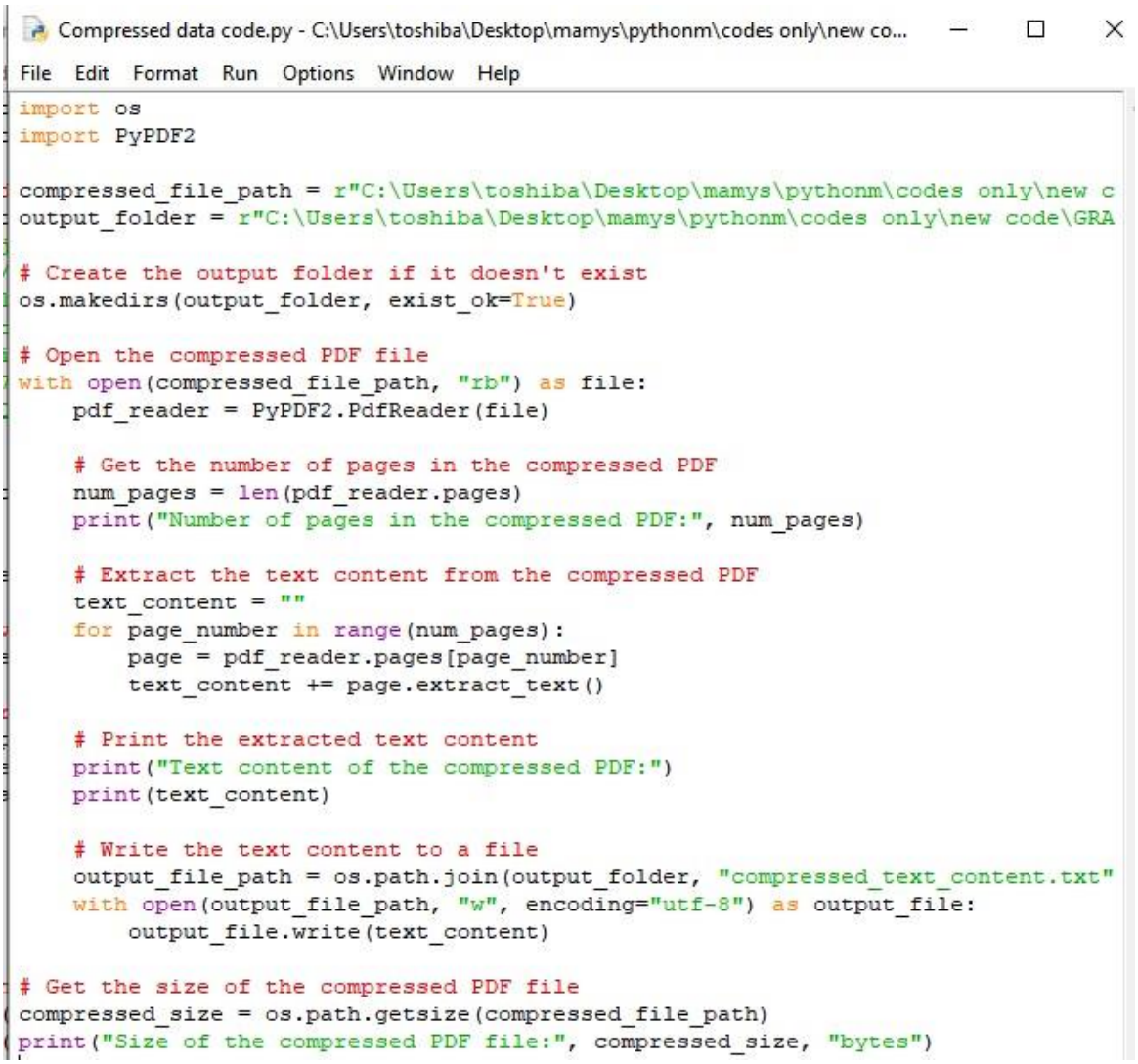


Figure 4.7 Compress Data by LZW loss less Compression

To Encrypt the Compressed by Encrypted AES

This Python program loads the RSA private key from a.pem file, loads the encrypted AES key from a binary file called "encrypted_aes_key.bin," decrypt the AES key using the RSA private key, loads the compressed data from a binary file called "compressed_data.lzw," pads the compressed data to the block size of AES, creates an initialization vector (IV), and then encrypts the padded data using the AES With the use of an IV, this script offers a quick and easy way to encrypt data using AES-CBC mode with a decrypt AES key, protecting both the secrecy and integrity of the data.

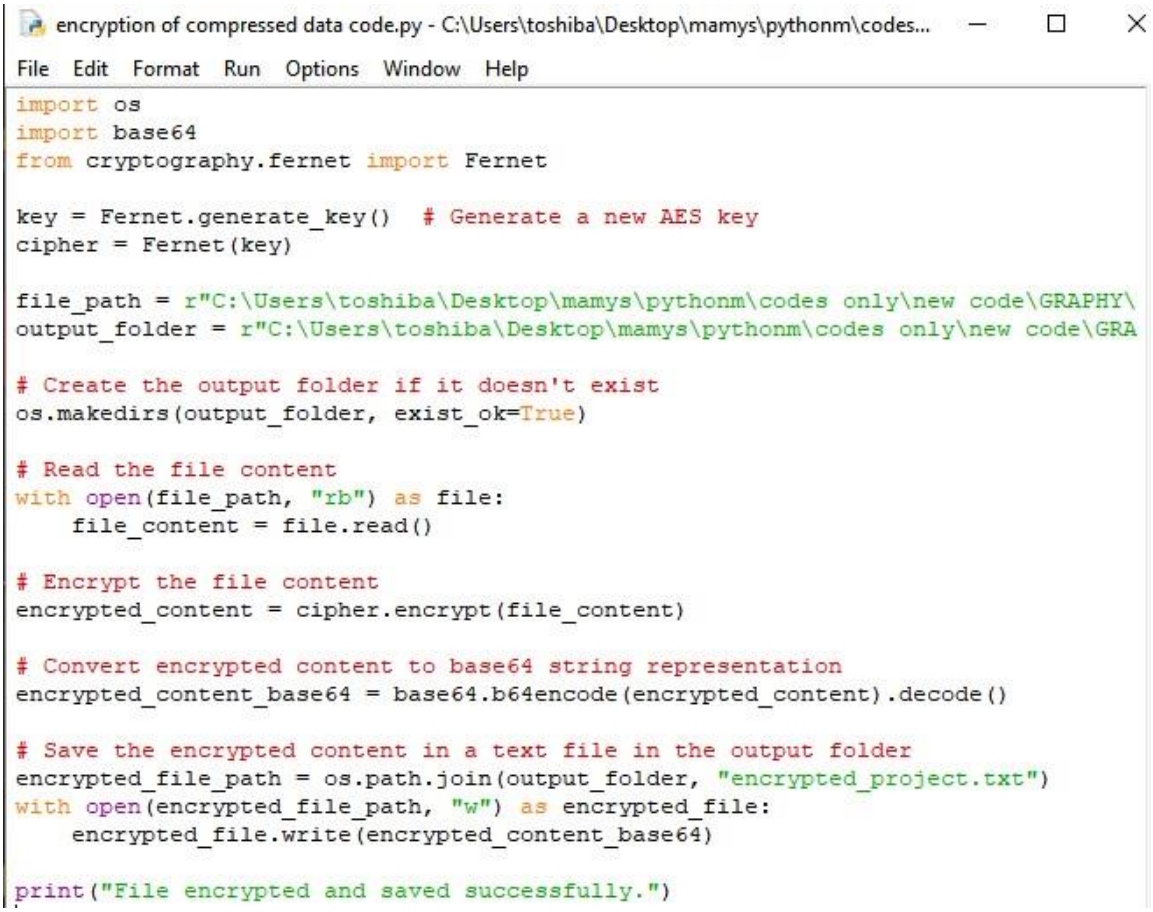


Figure 4.8 Encrypt the Compressed by Encrypted AES

To Decryption of Encrypted Data

This Python program loads the AES key from the 'aes_key decryption.bin' binary file, reads the encrypted data from the 'encrypted_data.bin' binary file, creates an AES cypher

object with the key and mode, decrypts the data using the cypher object, and saves the decrypted data to the 'decrypted_data.txt' text file. With the use of an initialization vector (IV) and an AES key, this script offers a quick and easy method for decrypting data that has been encrypted in the AES-CBC mode. This helps to protect the confidentiality and integrity of the data.

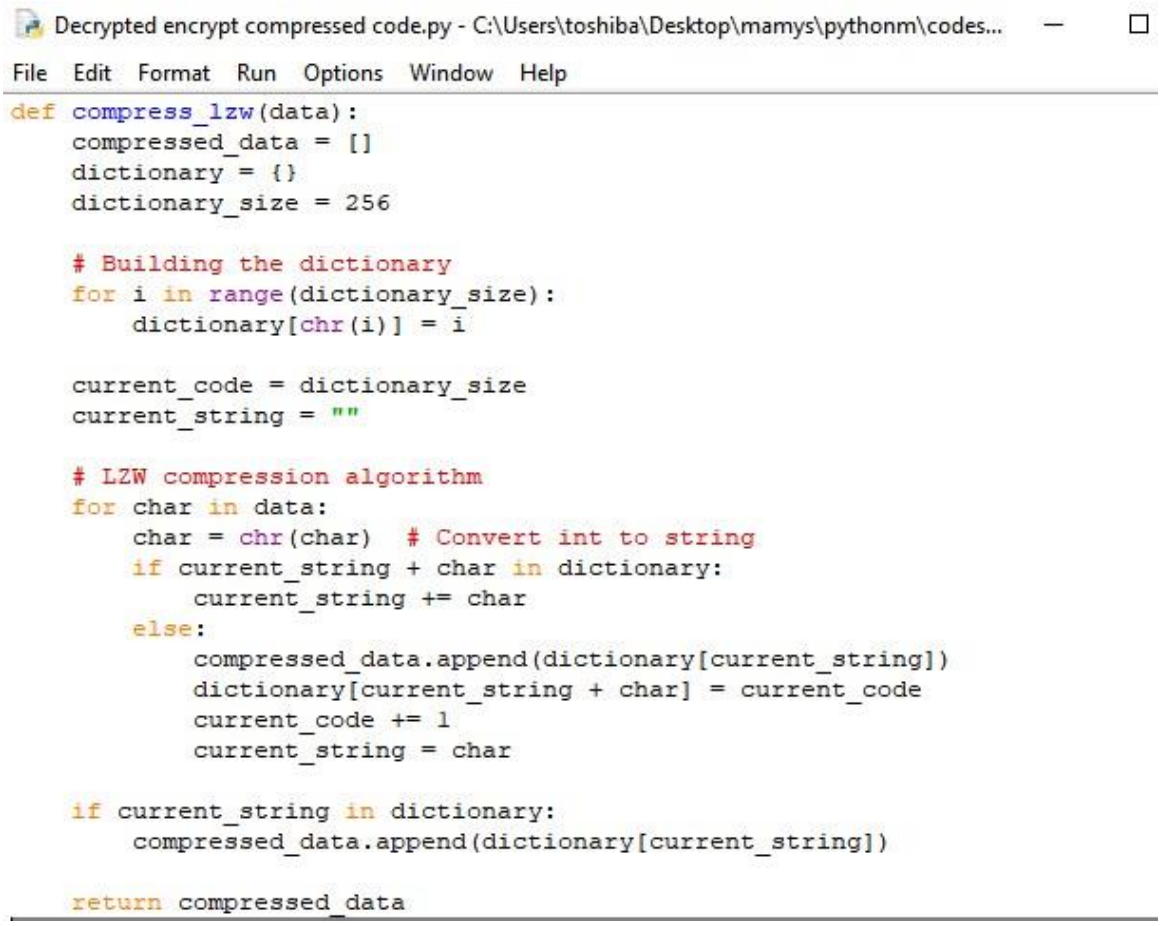


Figure 4.9 Decryption of Encrypted Data

To Decompression of Compressed Data

This Python program loads the AES key from the AES_key decryption.bin' binary file, reads the encrypted data from the 'encrypted_data.bin' binary file, creates an AES cypher object with the key and mode, decrypt the data using the cypher object, and saves the decrypt data to the 'decrypt_data.txt' text file. AES-ECB mode is not as safe as other modes like AES-CBC since it might result in patterns in the encrypted data. This script

offers a quick and easy solution to decrypt data that has been encrypted using AES-ECB mode with an AES key.

DECOMPRESSION of COMPRESSED DATA.PY - C:\Users\wku\Desktop\python\codes only\ne...

File Edit Format Run Options Window Help

```
import os
from Crypto.Cipher import AES

# Load AES key from file
with open('aes_key decryption.bin', 'rb') as file:
    aes_key = file.read()

# Load encrypted data from file
with open('encrypted_data.bin', 'rb') as file:
    encrypted_data = file.read()

# Decrypt data
cipher = AES.new(aes_key, AES.MODE_ECB)
decrypted_data = cipher.decrypt(encrypted_data)

# Save decrypted data to file
with open('decrypted_data.txt', 'wb') as file:
    file.write(decrypted_data)
```

Figure 4.10 Decompression of Compressed Data

4.5. Security proof

AES-256 Security Proof:

Key Schedule of AES-256:

The key schedule of AES-256 expands a given 256-bit key into a set of round keys, which are used in the encryption and decryption process. The key schedule involves several steps, including key expansion, sub key generation, and round constant generation. Here is a simplified formula for the key schedule of AES-256:

W [0] = First 32 bits of the 256-bit key

W [1] = Next 32 bits of the 256-bit key

...

W [7] = Last 32 bits of the 256-bit key

For i = 8 to 59:

if i mod 8 = 0:

W[i] = W [i-8] ⊕ Sub Word (Rot Word (W [i-1])) ⊕ Rcon [i/8]

Else:

W[i] = W [i-8] ⊕ W [i-1]

Here, Sub Word is a byte substitution operation that applies the S-box substitution to each byte of a word, Rot Word rotates the bytes of a word, and Rcon [i/8] is a round constant derived from the Rijndael's finite field. The symbol "⊕" represents the bitwise XOR operation.

Round Function of AES-256:

The round function of AES-256 is applied in each round of the encryption and decryption process. It consists of several operations, including byte substitution, shift rows, mix columns, and key addition. Here is a simplified formula for the round function of AES-256:

Sub Bytes (state) = Sub Word (state) =Apply byte substitution using the S-box

Shift Rows (state) = Shift Rows (state) = Perform row shifting operation

Mix Columns (state) = Mix Columns (state) = Apply column mixing operation

Add Round Key (state, round Key) = state ⊕ round Key = Perform key addition

Here, Sub Word is the byte substitution operation, Shift Rows shifts the rows of the state matrix, Mix Columns applies the column mixing operation, and Add Round Key performs the key addition operation. The symbol "⊕" represents the bitwise XOR operation.

RSA Security Proof:

The RSA Encryption Process:

RSA encryption involves the use of a public key (N, e) and a private key (N, d).

The public key consists of the modulus, N, and the public exponent, e.

The private key consists of the modulus, N, and the private exponent, d.

To encrypt a plain-text message, M, using the public key: $C \equiv M^e \pmod{N}$.

The resulting cipher text, C, can be transmitted over an insecure channel.

The RSA Decryption Process:

To decrypt the cipher text, C, using the private key: $M \equiv C^d \pmod{N}$.

The original plain-text message, M, is recovered using the private exponent, d.

Security of RSA:

The security of RSA relies on the difficulty of factoring the modulus, N, into its prime factors.

If an attacker can factor N and determine p and q, they can compute $\varphi(N)$ and subsequently compute the private exponent, d, from the public exponent, e.

However, factoring large numbers is believed to be a computationally difficult problem, especially when the numbers are sufficiently large (e.g., in RSA-2048, N is a 2048-bit number).

Current best-known factoring algorithms, such as the General Number Field Sieve (GNFS), become impractical for large key sizes.

.

CHAPTER FIVE

5. RESULTS AND ANALYSIS

5.1. Analysis of Results

In this section, we have discussed experimental details and evaluation results of Securing user data stored on cloud and different experiments were carried out with AES,RSA and LZW Compression.

5.2. Performance evaluation in python code result

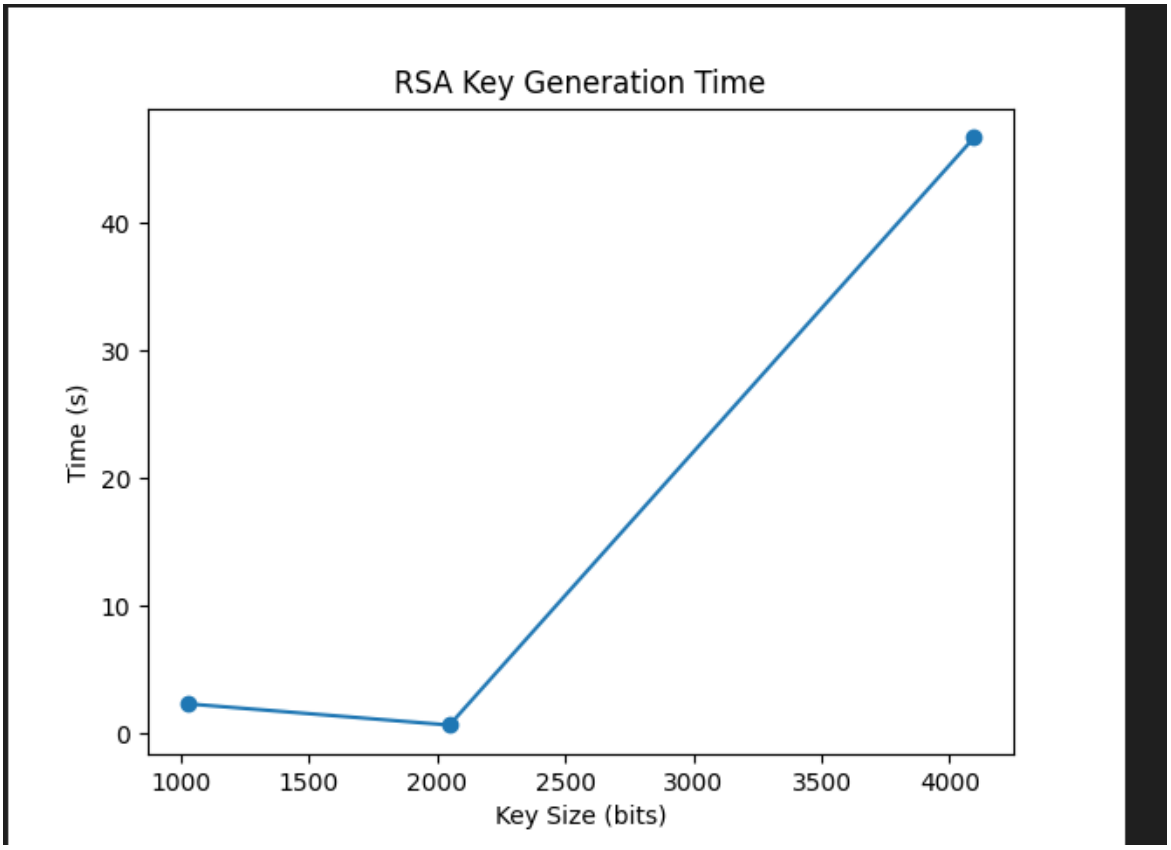


Figure 5.1 RSA Key Generation

This Python program calculates the time required to produce each key pair using the Crypto.PublicKey module to generate RSA key pairs of various sizes. The findings are then plotted by the script using the Matplotlib package.

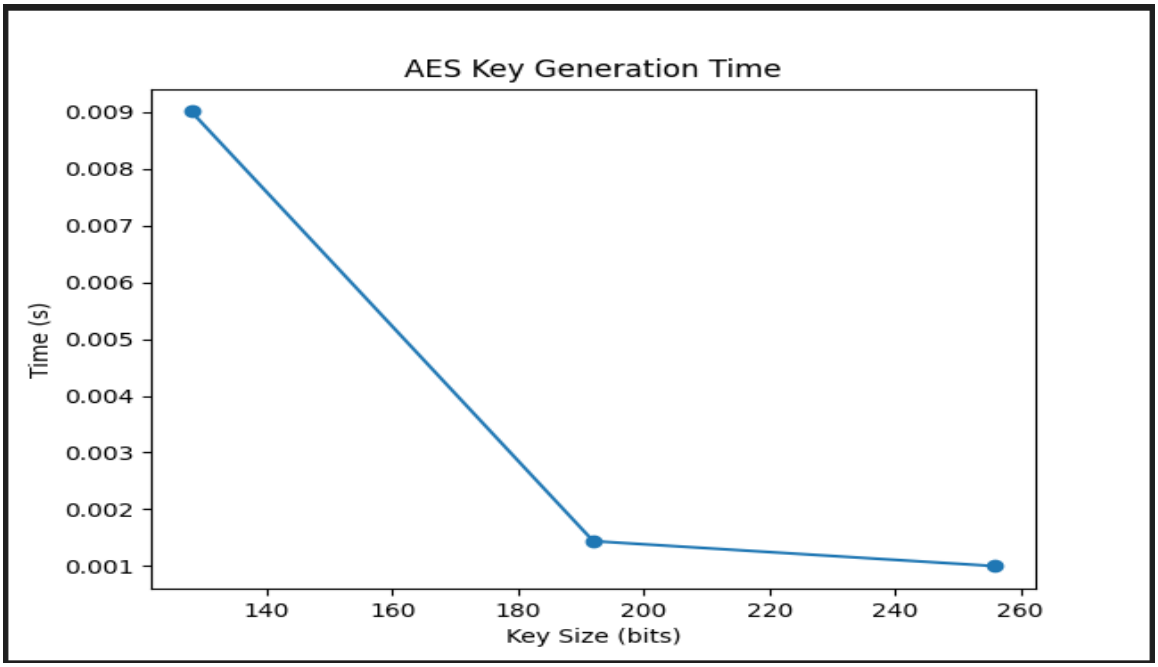


Figure 5. 2 AES Key Generation

This Python program creates AES keys of various sizes, calculates how long it takes to create each key, and then uses the matplotlib module to visualize the data.

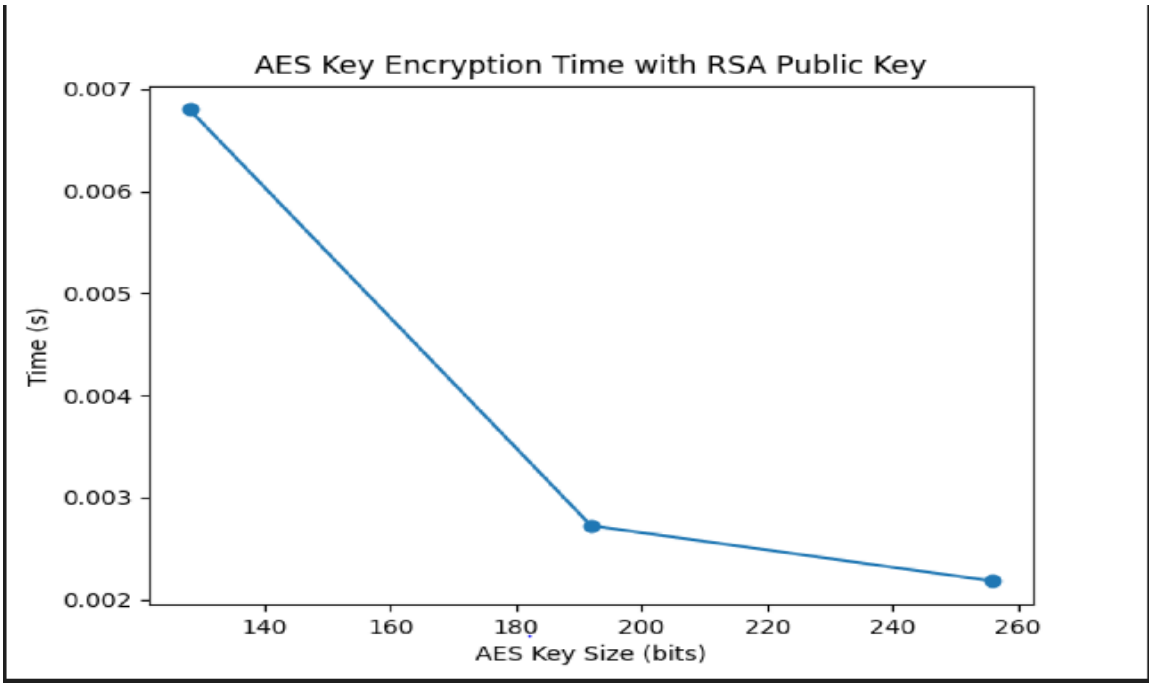


Figure 5. 3 AES Key Encryption by RSA Public Key

This Python program calculates the amount of time it takes

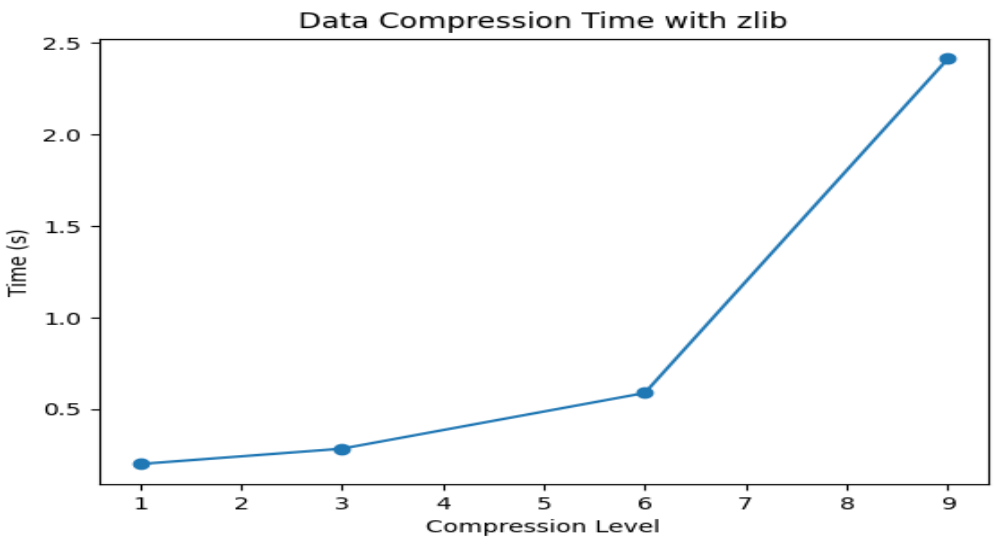


Figure 5.4 Data Compression by LZW

To encrypt various-sized AES keys with an RSA public key, then uses the matplotlib module to visualize the findings.

This Python program estimates the time it takes to compress data at each level using the zlib library, shows the results using the matplotlib library, and compresses data from a CSV file using various compression levels.

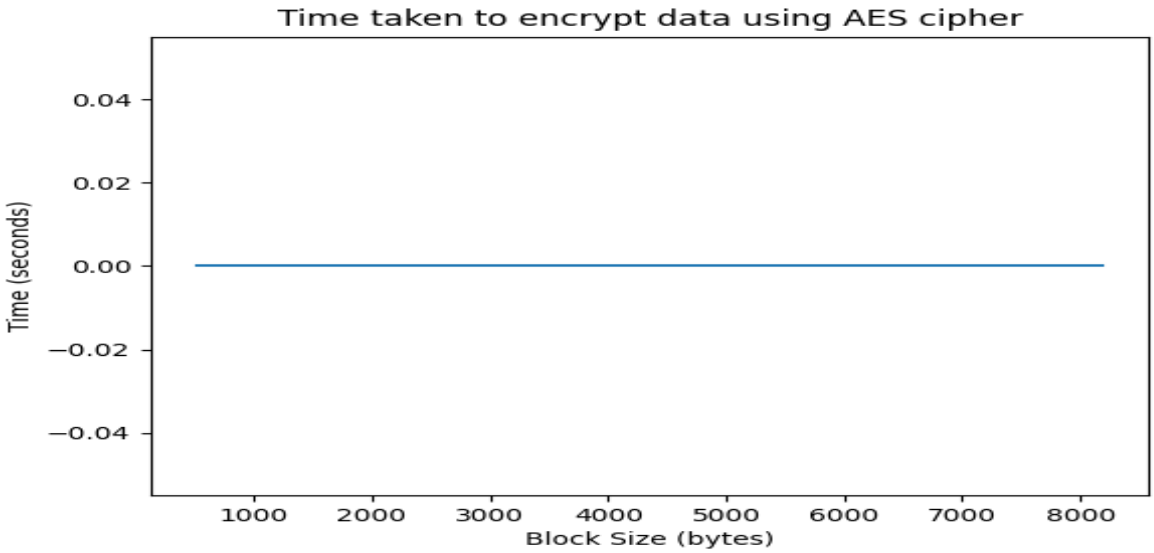


Figure 5.5 AES Encrypt the Compressed Data

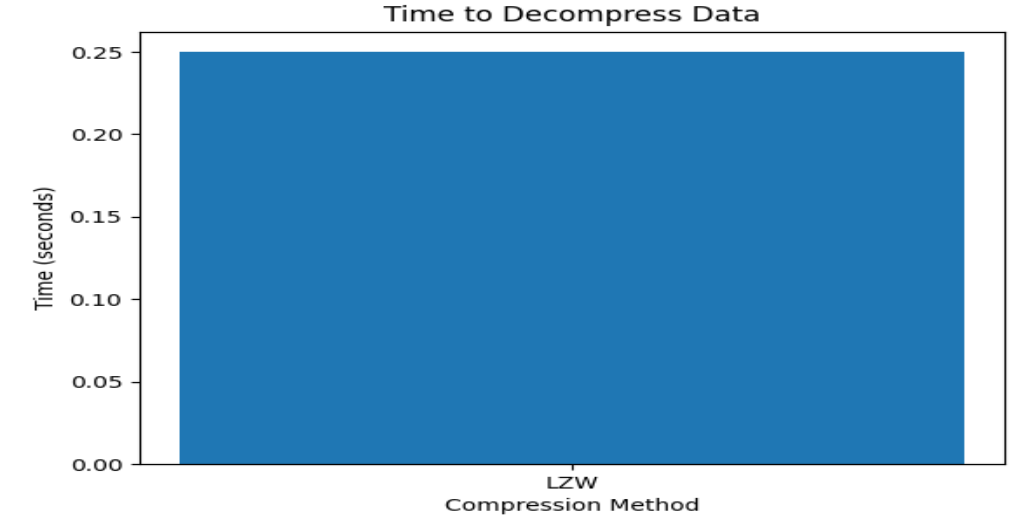


Figure 5.6 Compressed Data

The compression technique and the amount of time it took to decompress the data are both displayed on the x- and y-axes of the bar graph, respectively.

This program estimates the time it takes to encrypt the data at each block size, compresses the data using AES encryption, and then shows the results using the matplotlib package.

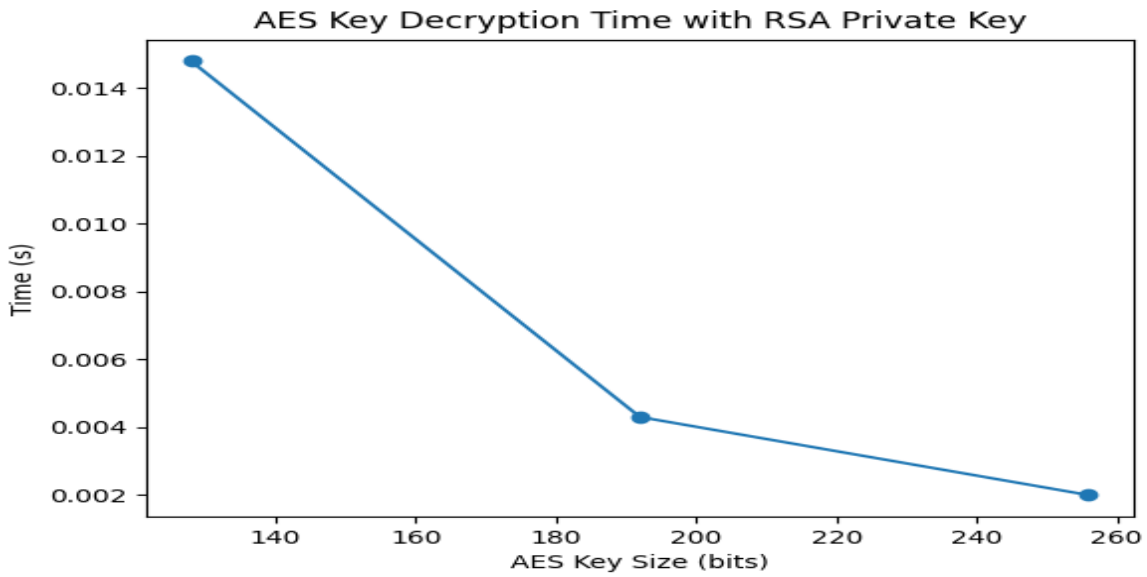


Figure 5.7 AES Key Decryption by RSA Private Key

This Python program evaluates the amount of time it takes to decode AES keys of various sizes, graphs the results using the matplotlib package, and employs RSA decryption to do.

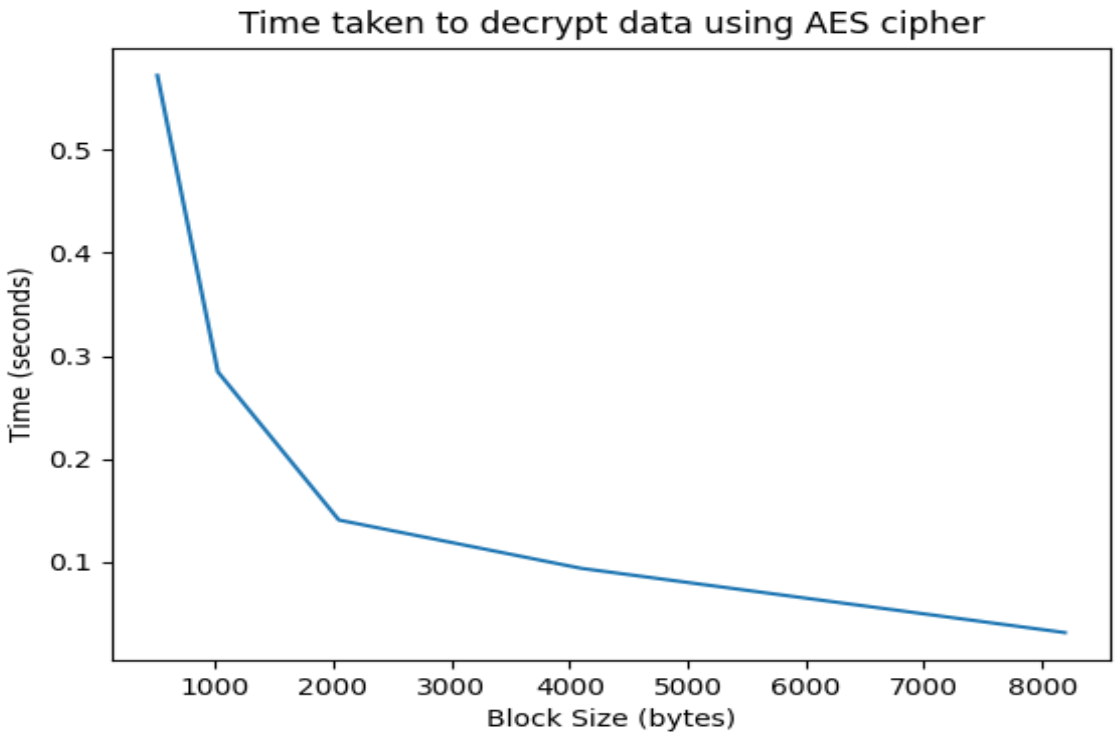


Figure 5.8 Data Decryption Time of AES

This Python program decrypt encrypted data with different block sizes using AES, calculates how long it takes to decrypt the data at each block size, then graphs the results using the matplotlib package.

5.3 Discussion of Results

The study's findings show that the suggested method is quick and secure. Implementation and performance analysis using Python 3.11 (64-bit) shows the efficacy of the suggested strategy. For implementation and performance analysis we use “Design of Technology Integrated Shredder Machine” project of data size of 4,616,263 bytes before compression. After compression, the file size reduced to 4,475,411 bytes. based on results, the

proposed algorithm improve the original RSA key generation time is 3.336312 second or 100.00%,encryption time of RSA 0.015691 second or 0.00%,the decryption time of RSA 0.015624 second or 0.00%,the encryption time of AES 0.000000 second or 0.00% and the decryption time of AES 0.000000 second or 0.00% but when we encrypt the by AES only, the encryption time is 0.07812s or 47.75% and the decryption time is 0.08546s or 52.25 % as the result of this performance investigation, the proposed hybrid (AES and RSA) algorithm is very fast and secure than the individual once.

RQ1 Which algorithm is better for text encryption and decryption for text data security?

- For encryption and decryption of sensitive user data we use AES and RSA cryptography algorithm. The encryption time of data by AES is 0.07812s.

RQ2 Which algorithm is better for text compression and decompression to improve the performance and security?

- We use LZW loss less compression. The compression of 4,616,263 bytes data is 4,475,411 bytes when we compress by LZW loss less compression.

RQ3 Which cloud storage provider are better for store user sensitive data?

- We use Mega cloud storage for store our data. It provides safe end-to-end encryption to assist prevent a privacy compromise.

RQ4 How we hide our IP-address when we upload our data to cloud or which VPN is more better?

- We use Cyber Ghost VPN. It is useful tool for safeguarding user data by encrypting internet traffic, hiding the user's IP address, and watching out for online risks.

CHAPTER SIX

6. CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion of Research work

In this research, our suggested technique is made more safe and quick thanks to improvements in algorithm strength, key generation, and decryption speed provided by hybrid (AES, RSA) and better encryption speed provided by LZW algorithm. The suggested technique can therefore be used in text security-demanding environments like hospitals and defenses.

This study's goal was to assess the efficiency of text data encryption and decryption techniques. The study employed the hybrid (AES and RSA) encryption and decryption techniques as well as the LZW loss less compression and decompression algorithm. Additionally, the study assessed how well MEGA cloud and Cyber Ghost VPN worked for storing private user information and connecting to the internet, respectively.

The study's main conclusions were that both AES and RSA or hybrid offered excellent levels of security for the encrypting and decryption of text data; with AES being quicker than RSA for big text data. So we take the advantage of both AES and RSA. For text data, LZW lossless offered the ideal compromise between compression ratio and decompression speed. Finally, for storing sensitive user data and connecting to the internet, MEGA cloud and Cyber Ghost VPN both offered excellent levels of protection. In general, the study advances knowledge of practical strategies for safeguarding text data as well as user security and privacy.

The hybrid technique, which combines the Advanced Encryption Standard (AES) and the Rivest-Shamir-Adleman (RSA) algorithm, is widely used in modern cryptographic systems. Protocols like Transport Layer Security (TLS), Pretty Good Privacy (PGP), Secure Sockets Layer (SSL), and OpenPGP use this approach. TLS uses AES for symmetric encryption, while PGP uses AES for data encryption and RSA for key exchange and digital signatures. SSL provides a secure channel between web browsers and servers.

6.2 Recommendations and future work

6.2.1 Recommendations

- Strong Passwords: Encourage users to create strong, unique passwords for their accounts.
- Regular Software Updates: Advice users to keep their operating systems, applications, and antivirus software up to date.
- Secure Wi-Fi Usage: Encourage users to connect to trusted and secure Wi-Fi networks, especially when accessing sensitive information.

6.2.2 Future works

One may expand our algorithm in the Future research recommendations include the following:

1. **Performance assessment:** Although the study evaluated the effectiveness of text data encryption and decryption algorithms, future research might concentrate on more thoroughly assessing these approaches' performance in various settings and circumstances. For instance, performance might be assessed using various text data formats audio, video, image, hardware, and software platforms.
2. **Alternative compression algorithms**: Although it was shown that LZW LOSSLESS provided a reasonable compromise between compression ratio and decompression speed for text data, future study might investigate alternative compression algorithms and assess their efficacy for various types of text data. This could make it easier to find fresh, possibly more effective approaches to text data compression.
3. **Security risks and threats**: Although the study emphasized the need for organizations to use strong encryption to guard against data breaches and unauthorized access, future research could concentrate on identifying and evaluating the particular security risks and threats that organizations face when dealing with text data. This could provide guidance for the creation of more focused security measures.

REFERENCES

[1] P. Soni and R. Malik, “A hybrid cloud security model for securing data on cloud,” *CEUR Workshop Proc.*, vol. 2889, pp. 118–125, 2021.

[2] V. S. Mahalle and A. K. Shahade, “Enhancing the data security in Cloud by implementing hybrid (Rsa & Aes) encryption algorithm,” *2014 Int. Conf. Power, Autom. Commun. INPAC 2014*, no. I, pp. 146–149, 2014, doi: 10.1109/INPAC.2014.6981152.

[3] R. Adee and H. Mouratidis, “A Dynamic Four-Step Data Security Model for Data in Cloud Computing Based on Cryptography and Steganography,” *Sensors*, vol. 22, no. 3, pp. 1–23, 2022, doi: 10.3390/s22031109.

[4] F. Lakrami, N. Elkamoun, and M. El Kamili, “Advances in Ubiquitous Networking,” *Lect. Notes Electr. Eng.*, vol. 366, pp. 287–300, 2016, doi: 10.1007/978-981-287-990-5.

[5] F. Yahya, V. Chang, R. J. Walters, and G. B. Wills, “Security challenges in cloud storages,” *Proc. Int. Conf. Cloud Comput. Technol. Sci. CloudCom*, vol. 2015-Febru, no. February, pp. 1051–1056, 2015, doi: 10.1109/CloudCom.2014.171.

[6] R. M. Balajee, H. Mohapatra, and K. Venkatesh, “A comparative study on efficient cloud security, services, simulators, load balancing, resource scheduling and storage mechanisms,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1070, no. February, p. 012053, 2021, doi: 10.1088/1757-899x/1070/1/012053.

[7] H. B. Pethe and S. R. Pande, “Comparative Study and Analysis of Cryptographic Algorithms AES and RSA,” *Int. J. Adv. Res. Comput. Sci. Manag. Stud.*, vol. 5, no. 1, pp. 48–56, 2017, [Online]. Available: www.ijarcsms.com

[8] Smitha Nisha Mendonca, “Data Security in Cloud using AES,” *Int. J. Eng. Res.*, vol. V7, no. 01, pp. 205–208, 2018, doi: 10.17577/ijertv7is010104.

[9] R. Arora and A. Parashar, “Secure User Data in Cloud Computing Using Encryption Algorithms,” *Int. J. Eng. Res. Appl.*, vol. 3, no. 4, pp. 1922–1926, 2013.

[10] T. Hidayat and R. Mahardiko, "A Systematic Literature Review Method On AES Algorithm for Data Sharing Encryption On Cloud Computing," *Int. J. Artif. Intell. Res.*, vol. 4, no. 1, 2020, doi: 10.29099/ijair.v4i1.154.

[11] S. Sa'idu, P. Taneja, and K. Shreya, "A Comparative Analysis of Cryptographic Algorithms : AES & RSA and Hybrid Algorithm for Encryption and Decryption," *Int. J. Innov. Sci. Res. Technol.*, vol. 7, no. 8, pp. 1725–1732, 2022.

[12] T. D. B. Weerasinghe, B. Eng, and A. (Sl, "IFRSA's International Journal Of Computing|Voll|issue |April Analysis of a Hybrid Cipher Algorithm for Data Encryption," pp. 397–401, [Online]. Available: www.ifrsa.org

[13] R. Ahlswede, A. Ahlswede, I. Althöfer, C. Deppe, and U. Tamm, "Data compression," *Found. Signal Process. Commun. Netw.*, vol. 10, no. 02, pp. 9–38, 2014, doi: 10.1007/978-3-319-05479-7_2.

[14] D. Barman* and M. B. Ahamed, "Improved LZW Compression Technique using Difference Method," *Int. J. Innov. Technol. Explor. Eng.*, vol. 9, no. 5, pp. 87–92, 2020, doi: 10.35940/ijitee.e2216.039520.

[15] S. Kaur, "Design and Implementation af LZW Data Compression Algorithm," *Int. J. Inf. Sci. Tech.*, vol. 2, no. 4, pp. 71–81, 2012, doi: 10.5121/ijist.2012.2407.

[16] R. Rahim *et al.*, "An application data security with lempel-ziv welch and blowfish Lesson Study for Learning Community View project A systematic literature review on attribute independent assumption of Naive Bayes: research trend, datasets, methods and frameworks View proje," *Int. J. Eng. Technol.*, vol. 7, no. 2, pp. 71–73, 2018, [Online]. Available: https://www.researchgate.net/publication/325626057

[17] A. T. Zahary, [1]wafaa A N A Al-Nbhany, and A. Zahary, "A Comparative Study among Cryptographic Algorithms: Blowfish, AES and RSA Internet of Things (IoT) View project MANETs and VANETs View project Wafaa A N A Al-Nbhany central statistical organization A Comparative Study among Cryptographic Algorithms: Blowf," *Int. Arab Conf. Inf. Technol.*, no. December,

2016, [Online]. Available: https://www.researchgate.net/publication/325106192

[18] A. Singh, M. Marwaha, B. Singh, and S. Singh, “Comparative Study of DES, 3DES, AES and RSA,” *Int. J. Comput. Technol.*, vol. 9, no. 3, pp. 1162–1170, 2013, doi: 10.24297/ijct.v9i3.3342.

[19] A. Al Hasib and A. A. M. M. Haque, “A comparative study of the performance and security issues of AES and RSA cryptography,” *Proc. - 3rd Int. Conf. Converg. Hybrid Inf. Technol. ICCIT 2008*, vol. 2, no. November 2001, pp. 505–510, 2008, doi: 10.1109/ICCIT.2008.179.

[20] R. Marqas, S. M. Almufti, and R. Rebar, “Comparing Symmetric and Asymmetric cryptography in message encryption and decryption by using AES and RSA algorithms,” *J. Xi’an Univ. Archit. Technol.*, vol. XII, no. III, 2020, doi: 10.37896/jxat12.03/262.

[21] J. Kenyatta, M. Ndego, M. W. Kimwele, and R. Shamir Adleman, “Eveque Mutabaruka Enhancing Data Security By Using Hybrid Encryption Technique (Advanced Encryption,” *s J. Comput. Sci.*, vol. 2, no. 5, pp. 2349–5391, 2015.

[22] B. H. Lee, E. K. Dewi, and M. F. Wajdi, “Data security in cloud computing using AES under HEROKU cloud,” *2018 27th Wirel. Opt. Commun. Conf. WOCC 2018*, no. September, pp. 1–5, 2018, doi: 10.1109/WOCC.2018.8372705.

[23] H. Yang, “Application of Hybrid Encryption Algorithm in Hardware Encryption Interface Card,” *Secur. Commun. Networks*, vol. 2022, 2022, doi: 10.1155/2022/7794209.

[24] Y. Huang and P. Mishra, “Trace Buffer Attack on the AES Cipher,” *J. Hardw. Syst. Secur.*, vol. 1, no. 1, pp. 68–84, 2017, doi: 10.1007/s41635-017-0004-3.

[25] P. Derbez, P. A. Fouque, and J. Jean, “Improved key recovery attacks on reduced-round AES in the single-key setting,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7881 LNCS, pp. 371–387, 2013, doi: 10.1007/978-3-642-38348-9_23.

[26] M. Panjwani, A. Satpute, A. Kamble, N. Ukey, V. Makde, and Y. Mehere,

"Securing Data in a Cloud Using Aes," *Int. J. Res. Anal. Rev.*, vol. 7, no. 1, pp. 2348–2350, 2020, [Online]. Available: www.ijrar.org

[27] M. Tyagi, M. Manoria, and B. Mishra, "Analysis and Implementation of AES and RSA for cloud," *Int. J. Appl. Eng. Res.*, vol. 14, no. 20, p. 3918, 2019, doi: 10.37622/ijaer/14.20.2019.3918-3923.

[28] H. Abroshan, "A Hybrid Encryption Solution to Improve Cloud Computing Security using Symmetric and Asymmetric Cryptography Algorithms," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 6, pp. 31–37, 2021, doi: 10.14569/IJACSA.2021.0120604.

[29] J. B. Awotunde, A. O. Ameen, I. D. Oladipo, A. R. Tomori, and M. Abdulraheem, "Evaluation of four encryption algorithms for viability, reliability and performance estimation," *Niger. J. Technol. Dev.*, vol. 13, no. 2, p. 74, 2017, doi: 10.4314/njtd.v13i2.5.

[30] M. A. Panhwar, S. Ali Khuhro, G. Panhwar, and K. A. Memon, "A Study of Symmetric and Asymmetric Cryptographic Algorithms," *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 19, no. 1, p. 48, 2019.

[31] A. Kajal and G. Jambheshwar, "Enhanced Cloud Storage Security Using ECC - AES A Hybrid Approach," no. August, 2018, doi: 10.18231/2454-9150.2018.0593.

[32] D. Kodzo, M. Hodowu, D. R. Korda, and E. Danso Ansong, "An Enhancement of Data Security in Cloud Computing with an Implementation of a Two-Level Cryptographic Technique, using AES and ECC Algorithm," *Int. J. Eng. Res. Technol.*, vol. 9, no. 09, pp. 2278–0181, 2020, [Online]. Available: http://www.ijert.org

[33] D. J. K. Mantri* and R. Mishra, "Secure Data Transmission using Goldbach Codes and RSA Algorithm," *Int. J. Recent Technol. Eng.*, vol. 8, no. 4, pp. 5–8, 2019, doi: 10.35940/ijrte.b2638.118419.

[34] P. View Project Lemaire, L. Vossaert, J. Jansen, and J. Naessens, "RSA

Encryption Algorithm Optimization to Improve Performance and Security Level of Network Messages A Logic-Based Framework for the Security Analysis of Industrial Control Systems. Automatic Control and Computer Sciences. View project RSA Encryption Algor," *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 16, no. 8, p. 55, 2017, [Online]. Available: https://www.researchgate.net/publication/308553822

[35] E. Bala, A. Aminat, and E. Christopher, "Hybrid Data Encryption and Decryption Using Rsa and Rc4," vol. 10, no. 10, 2019, [Online]. Available: http://www.ijser.org

[36] C. Gunavathi and K. Premalatha, "A_comparative_analysis_of_swar.PDF," vol. 6, no. 1, pp. 358–373, 2014.

[37] F. Kpeky, F. Abed-Meraim, E. M. Daya, and O. D. Samah, "Modeling of hybrid vibration control for multilayer structures using solid-shell finite elements," *Mech. Adv. Mater. Struct.*, vol. 25, no. 12, pp. 1033–1046, 2018, doi: 10.1080/15376494.2017.1365987.

[38] U. Patel, A. Patel, and F. Suthar, "Science and Applications the Study of Digital Signature," no. October, pp. 0–6, 2019.

[39] O. Chok and S. Herath, "Computer Security Learning Laboratory: Implementation of DES and AES Algorithms using Spreadsheets," 2004, [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.107.5405

[40] MOH HENG HUONG, "IMPLEMENTATION OF (AES) ADVANCED ENCRYPTION STANDARD ALGORITHM IN COMMUNICATION APPLICATION," *Lincolin Arsyad*, vol. 3, no. 2, pp. 1–46, 2014, [Online]. Available: http://journal.stainkudus.ac.id/index.php/equilibrium/article/view/1268/1127

[41] M. A. Chavan, M. A. Jadhav, M. Shraddha Kumbhar, M. I. Joshi, and M. I. Joshi, "Data Transmission using RSA Algorithm," *Int. Res. J. Eng. Technol.*, pp. 2008–2010, 2019, [Online]. Available: www.irjet.net

[42] S. Mall and S. K. Saroj, “A new security framework for cloud data,” *Procedia Comput. Sci.*, vol. 143, pp. 765–775, 2018, doi: 10.1016/j.procs.2018.10.397.

[43] S. Malhotra and W. Singh, “An efficacy analysis of data encryption architecture for cloud platform,” *Procedia Comput. Sci.*, vol. 218, pp. 989–1002, 2022, doi: 10.1016/j.procs.2023.01.079.

[44] A. Aminnezhad, P. Khanmohamadi Hezaveh, M. Khodadadi, and A. Tan, “A Study of Ten Popular Android Mobile Cloud Storage Applications,” *Res. J. Appl. Sci. Eng. Technol.*, vol. 13, no. 7, pp. 533–543, 2016, doi: 10.19026/rjaset.13.3013.

[45] A. Osman, Y. Al Moaiad, and A. Osman, “ENCRYPTION TYPES AND KEY MANAGEMENT IN CLOUD STORAGE PROVIDERS Multimedia View project Cloud Computing View project ENCRYPTION TYPES AND KEY MANAGEMENT IN CLOUD STORAGE PROVIDERS,” vol. 8, no. September, pp. 2–6, 2020, [Online]. Available: https://www.researchgate.net/publication/344131451

[46] T. B. I. Guy-Cedric and S. R., “A Comparative Study on AES 128 BIT AND AES 256 BIT,” *Int. J. Sci. Res. Comput. Sci. Eng.*, vol. 6, no. 4, pp. 30–33, 2018, doi: 10.26438/ijsrcse/v6i4.3033.

[47] S. Delfin, “CLOUD DATA SECURITY USING AES ALGORITHM,” pp. 1189–1192, 2018.

[48] U. T. Pius, C. Onyebuchi, O. P. Chinasa, and E. F. Adoba, “A Cloud-Based Data Security System using Advanced Encryption (AES) and Blowfish algorithms,” *Available online www.jsaer.com J. Sci. Eng. Res. 59 J. Sci. Eng. Res.*, vol. 5, no. 6, pp. 59–66, 2018, [Online]. Available: www.jsaer.com

[49] R. F. S. L. Et.al, “Improvement of RSA Algorithm Using Euclidean Technique,” *Turkish J. Comput. Math. Educ.*, vol. 12, no. 3, pp. 4694–4700, 2021, doi: 10.17762/turcomat.v12i3.1889.

[50] R. Thilagavathy and A. Murugan, “Secure the Cloud Data Transmissionusing an Improved RSA Algorithm,” *Indian J. Sci. Technol.*, vol. 10, no. 12, pp. 1–6, 2017,

doi: 10.17485/ijst/2017/v10i12/103770.

[51] S. R. Patel and G. R. Patel, “Study on Improvements in RSA Algorithm,” *Int. J. Eng. Dev. Res.*, pp. 142–145, 2013, [Online]. Available: www.ijedr.org

[52] H. Vardhan Singh, A. Dhama, G. Kumar, and A. Kumar Sharma, “Enhanced Advanced Encryption Standard (E-AES): Using ESET,” *Int. Res. J. Eng. Technol.*, pp. 1839–1844, 2017, [Online]. Available: www.irjet.net

[53] U. Khurana and A. Koul, “Text Compression and Superfast Searching,” p. 11, 2005, [Online]. Available: http://arxiv.org/abs/cs/0505056

[54] D. Klinc, C. Hazay, A. Jagmohan, H. Krawczyk, and T. Rabin, “On compression of data encrypted with block ciphers,” *IEEE Trans. Inf. Theory*, vol. 58, no. 11, pp. 6989–7001, 2012, doi: 10.1109/TIT.2012.2210752.

[55] N. Anand, “Using AES Algorithm Encryption and Decryption of Text File, Image and Audio in Openssl and Time Calculation for Execution,” vol. 22, no. 6, pp. 39–44, 2020, doi: 10.9790/0661-2206013944.

[56] A. Rayarapu, N. V. Krishna, and D. Mundhra, “Securing Files Using AES Algorithm,” *Int. J. Comput. Sci. Inf. Technol.*, vol. 4, no. 3, pp. 433–435, 2013.

[57] I. A. Awan, M. Shiraz, M. U. Hashmi, Q. Shaheen, R. Akhtar, and A. Ditta, “Secure Framework Enhancing AES Algorithm in Cloud Computing,” *Secur. Commun. Networks*, vol. 2020, 2020, doi: 10.1155/2020/8863345.

[58] Z. Lu and H. Mohamed, “A Complex Encryption System Design Implemented by AES,” *J. Inf. Secur.*, vol. 12, no. 02, pp. 177–187, 2021, doi: 10.4236/jis.2021.122009.

[59] L. V. Batista and M. M. Meira, “Texture classification using the Lempel-Ziv-Welch algorithm,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3171, pp. 444–453, 2004, doi: 10.1007/978-3-540-28645-5_45.

[60] N. Jacob, P. Somvanshi, and R. Tornekar, “Comparative Analysis of Lossless Text

Compression Techniques," *Int. J. Comput. Appl.*, vol. 56, no. 3, pp. 17–21, 2012, doi: 10.5120/8871-2850.

[61] D. Yehya, M. Joudi, and A.-L. Sousi, "AES Encryption : Study & Evaluation," *Cryptogr. Netw. Secur.*, no. December, pp. 1–27, 2020.

[62] M. Senthil Murugan and T. Sasilatha, "Implementation of advanced encryption standard algorithm on steganography," *Int. J. Recent Technol. Eng.*, vol. 7, no. 5, pp. 225–230, 2019, doi: 10.15662/IJAREEIE.2016.0506068.

[63] N. S. M. Shamsuddin, S. A. Pitchay, and S. A. Pitchay, "Implementing Location-Based Cryptography on Mobile Application Design to Secure Data in Cloud Storage," *J. Phys. Conf. Ser.*, vol. 1551, no. 1, 2020, doi: 10.1088/1742-6596/1551/1/012008.

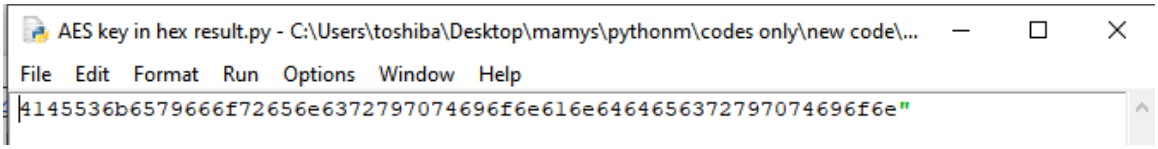
[64] M. B. Begum, N. Deepa, M. Uddin, R. Kaluri, M. Abdelhaq, and R. Alsaqour, "An efficient and secure compression technique for data protection using burrows-wheeler transform algorithm," *Heliyon*, vol. 9, no. 6, p. e17602, 2023, doi: 10.1016/j.heliyon.2023.e17602.

[65] J. Hughes, "Comparison of lossy and lossless compression algorithms for time series data in the Internet of Vehicles Jämförelse av destruktiva och icke-förstörande komprimeringsalgorithmer för tidsseriedata inom fordonens internet," 2023, [Online]. Available: www.liu.se

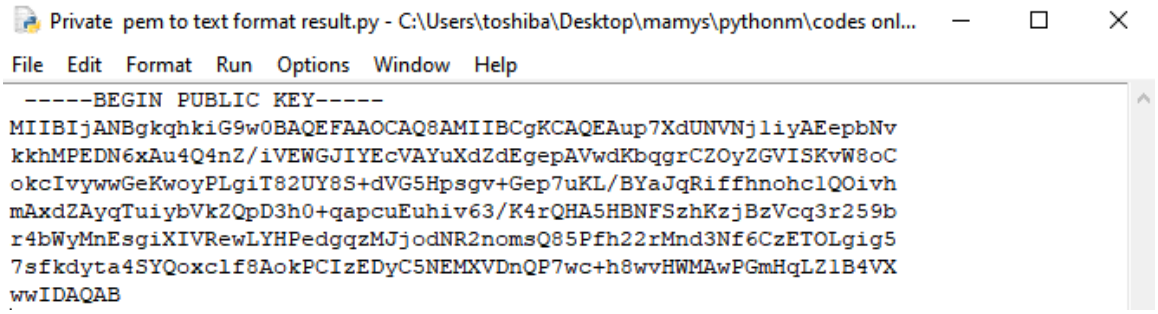
[66] H. Jani and J. Trivedi, "A Survey on Different Compression Techniques Algorithm for Data Compression," vol. 2, no. 3, pp. 364–368, 2014.

APPENDICES

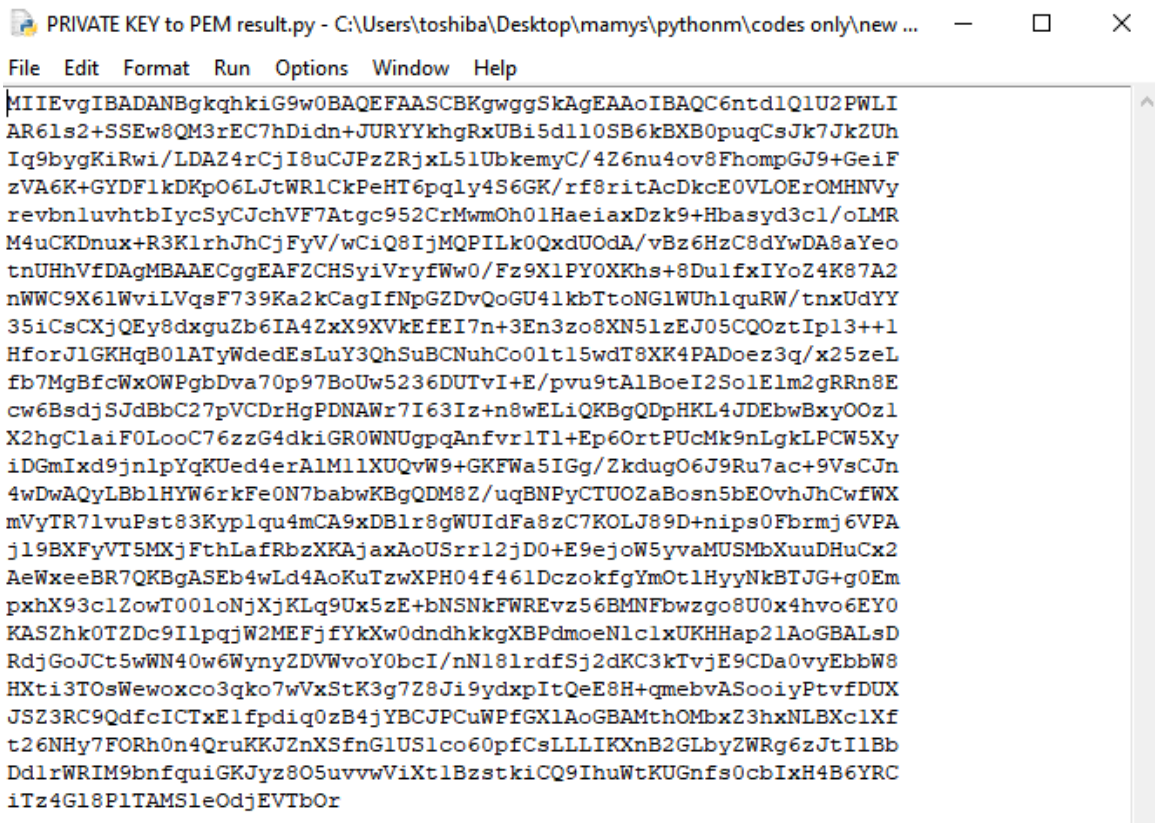
Appendix A: AES key in hexadecimal



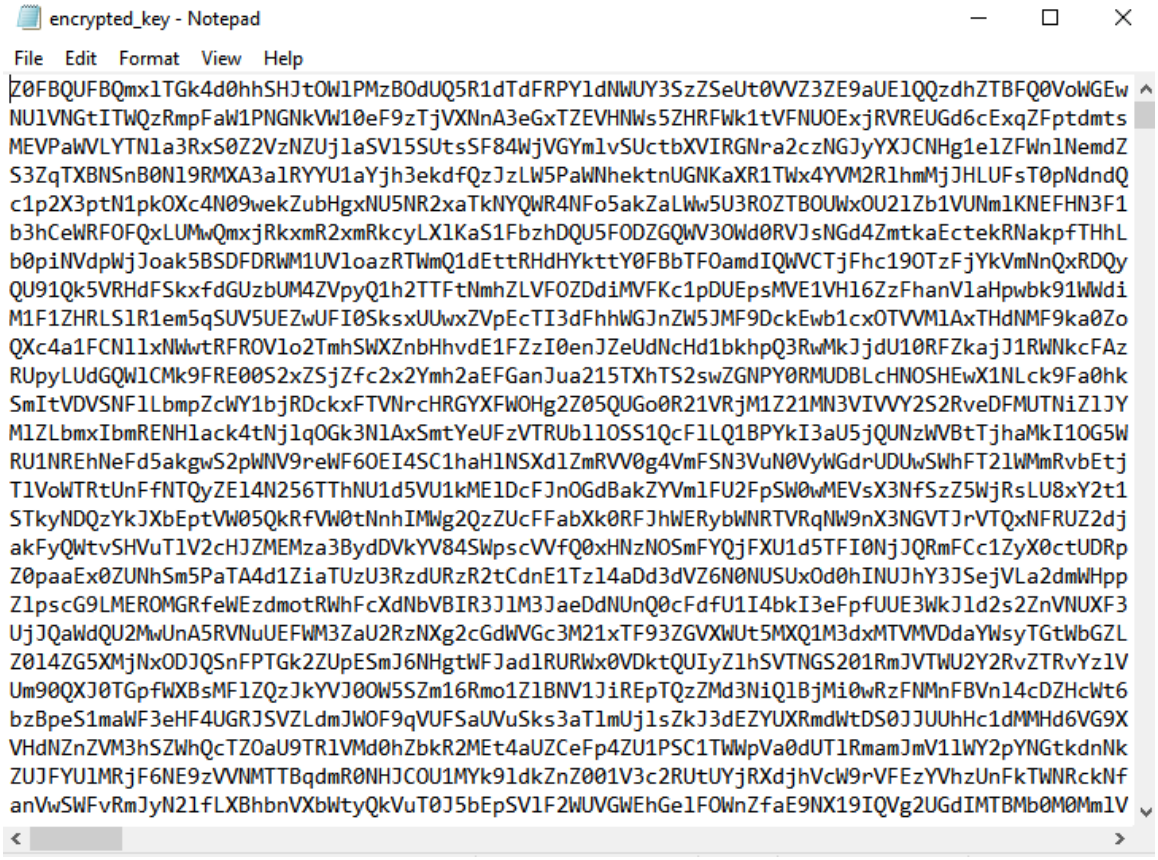
Appendix B: The Result of Public Key Generation of RSA



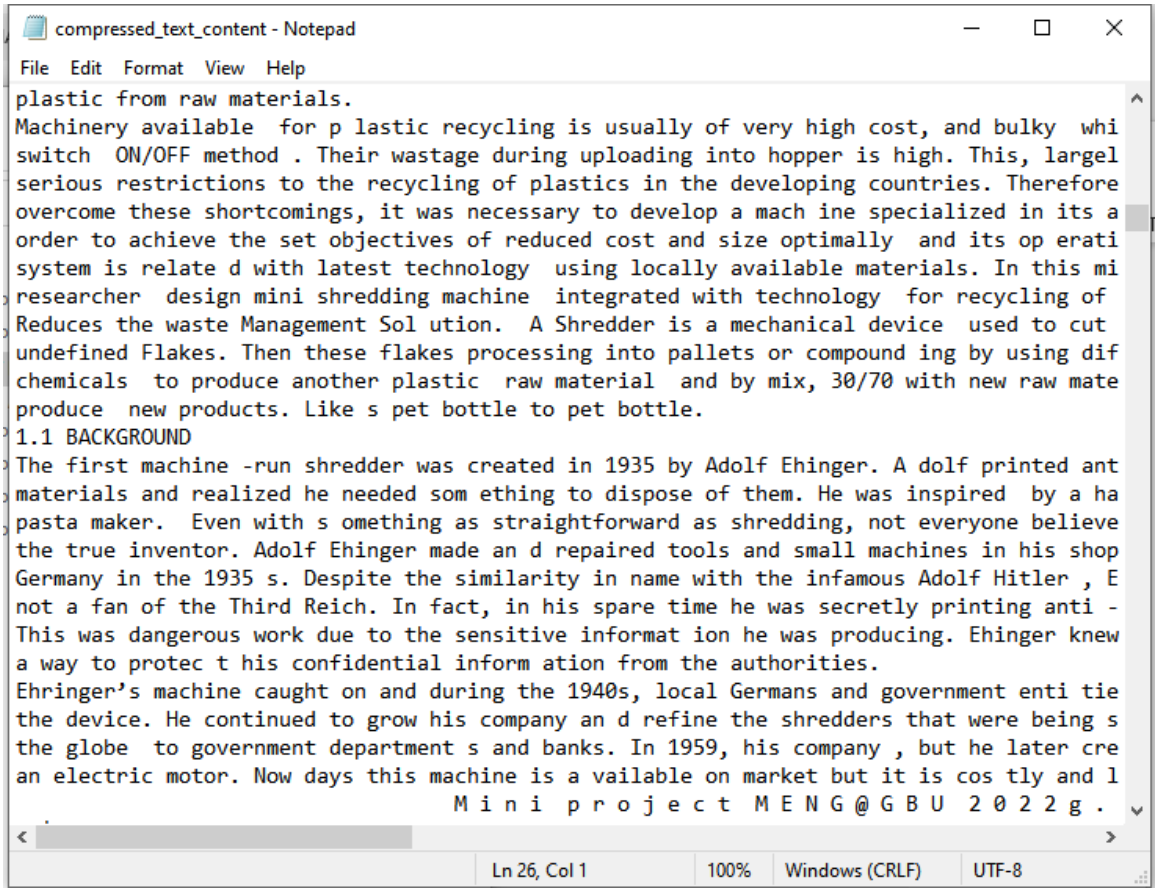
Appendix C: The Result of Private Key Generation of RSA



Appendix D: The Result of Encrypted AES Key by Public Key of RSA



Appendix E: The Result of Compress Data by LZW Loss less Compression



Appendix F: The Result of Encrypt the Compressed by Encrypted AES

Appendix G: The Result of Decryption of Encrypted Data

ACKNOWLEDGMENT

First, I would like to thanks to almighty god for his being with me in all aspect of my life. After that I wish to give my deepest gratitude to GBU for arranging this opportunity to have a real-life experience on the practical works related to theoretical knowledge. Next, I would like to express my heart full thank to my hosting company for accepting my internship request and allow me to practice in their company. Besides this I would like to extend my heart full thanks to my advisor Getnet B. (MSc.) for devotion of his precious time by giving valuable suggestion, comments and systematic guidance for the development of this mini project from starting to ending. Also, I would like to express my deepest gratitude to all workers in daily water company specially Abiy Bakale (Electro-Mech Engg) and my Adviser Ketama (Machanic) for your willful and unlimited help during my internship period and for your advices to guiding and helping me during this all time. I am dearly obliged to my heart brother John Moti.A (BSc) and Mikiyas Bemnat (Arct.) for devotion of your precious time by editing and