A PROJECT PROPOSAL REPORT

ON ENHANCING MESSAGE SECURITY THROUGH ENCRYPTION DECRYPTION TECHNIQUES AND STRATEGIES. FOR SECURITY AGENCIES

SUBMITTED BY

EMONG CHARLES

Under the guidance of

Dr. EMMANUEL EILU

In partial fulfillment for the award of the degree of

A Bachelor of Science in Information Technology

[BSc. Information Technology]

[2021 – 2024]

At

Department of Information Technology

FACULTY OF COMPUTING AND ENGINEERING

UGANDA CHRISTIAN UNIVERSITY

Table of Contents

DECLARATION

I EMONG CHARLES declare that this Report is my original work and has never been submitted for the award of a degree, diploma or any other academic qualification in any other university, college or institution before.

Signature 

Date 27/09/2024

APPROVAL

This report has been written and submitted for Examination with approval of the undersigned supervisor. This report has been written and submitted to the Faculty of Engineering and Information Technology of Uganda Christian University with my approval as the supervisor.

Academic Supervisor

Signature_____

Date____01/10/2024_____

DR Emmanuel Eilu

Department of Information Technology

Uganda Christian University

DEDICATION

This work is dedicated to Mrs Zaituna Tikabulamu my mother, Brother Timothy and the entire family for their mentorship, prayers and support.

ACKNOWLEDGEMENT

First and foremost, I would like to express my sincere appreciation and gratitude to God for granting me His protection. Many thanks to Mrs Zaituna Tikabulamu, Agaba Martin, Mr. Wilberforce Odaga, Mr Patrick Odong, Mr Edward Katende and others not mentioned for guiding me during my project. In my view, it was a unique and pleasant opportunity to work in a new field and to use the opportunity to reunite with my family back home.

LIST OF TABLES

ABSTRACT

This project centers on developing a secure communication system using the AES (Advanced Encryption Standard) encryption-decryption algorithm. As cyber threats increase and more sensitive data is exchanged online, safeguarding the privacy, integrity, and authenticity of digital communications is crucial. The project employs AES, a renowned symmetric key encryption method, to safeguard data from unauthorized access both in transit and storage.

The AES algorithm, specifically the 256-bit version, is selected for its blend of high security and computational efficiency. This makes it ideal for securing sensitive communications, from personal exchanges to corporate and military contexts. The system works by converting plaintext information, such as messages or files, into cipher text before transmission, ensuring it is unreadable by unauthorized users. The recipient then uses the same key to decrypt the data and recover the original information.

The system design integrates several essential security elements, such as:

- Key Generation and Management: The system securely generates and distributes encryption keys, with regular rotation to mitigate cryptographic vulnerabilities.
- User Authentication: Only authenticated users can access encrypted or decrypted information, supported by strong authentication mechanisms.
- Data Integrity: Techniques are implemented to ensure that transmitted data remains unaltered during transmission.
- Real-Time Secure Communication: The system enables real-time encryption and decryption for communication across networks like LAN and the internet, using socket programming to securely transfer data between devices.
- This project aims to showcase the practical application of AES encryption in securing real-world communications. A user-friendly interface is provided for managing encryption and decryption activities, supporting key updates, access to encrypted/decrypted data, and integrity checks. The system is also scalable, making it suitable for both personal and large-scale organizational use.

Throughout its development, the project addresses common challenges in encryption systems, including key management, balancing encryption performance, and maintaining compatibility with existing communication systems. Ultimately, this project delivers a secure and efficient solution for protecting sensitive data transmissions in today's digital landscape.

Keywords: AES encryption, secure communication, data protection, cryptography, symmetric encryption, key management, cyber security, network security.

CHAPTER ONE

1.0 Introduction

Chapter One of this project presents the Background Information to the Study highlighting the Problem Statement, Objectives, Scope and Significance to the Study.

1.1 Background to the Study

Currently, Security agencies in Uganda face a lot of challenges in trying to conceal security related information from one department to another. This is brought about manual transfer of message (use of courier), use of systems which are not more secured to deliver message. Such systems include: use of emails, whatsapp, and messenger, direct phone calls which makes sensitive information to be intercepted or accessed. Therefore, there is a need for an AES Encryption and Decryption System. An AES Encryption Decryption System is a symmetric key cryptosystem used to protect data confidentiality. It operates on fixed block sizes of 128 bits using keys of length 128, 192 or 256 bits. This system is comprised of the following elements namely:-

- Symmetric key generation, which uses a single key for both encryption and Decryption.
- Encryption process, which involves several steps organized into rounds (10, 12, 14 rounds for 128, 192, or 256 respectively).
- Decryption process, which is the reverse of the encryption process using the same key.
- Key schedule, is an algorithm that, given the cipher key, generates a series of round keys for use in each round of both Encryption and Decryption

According to Bruce Schneier, Encryption and Decryption Systems are fundamental to the modern digital Security and privacy.

1.2 Problem Statement

The rapid increase of cyber threats has created a pressing need for robust and efficient encryption and decryption tool. Security agencies in Uganda face significant challenges in ensuring the confidentiality, integrity, and availability of sensitive information amidst the evolving landscape of cyber threats. According to the Uganda Communications Commission (UCC), cybercrime incidents have been steadily increasing, with reported cases of hacking, data breaches, and online fraud impacting government agencies and

private organizations alike. Therefore, there is a critical need to research and develop an advanced encryption and decryption tool that addresses these shortcomings while ensuring high security, optimal performance and user friendly design.

1.3 Main Objective

To develop an encryption and decryption Application that enhances message Security through encryption and decryption Techniques and Strategies.

1.4 Specific Objectives

- To Study the current System used by security Agencies to manage security communication
- To design and implement encryption algorithms for message confidentiality. ¬ To develop decryption methods to securely access encrypted messages.
- To test and validate the encryption decryption system so as to check for errors and future development.

1.5 Scope

The System designed and developed will be used by Security Agencies in Uganda particularly in the Directorate of Communication and Information Technology for sending and receiving sensitive information from one Division to another, the system will also save keys and load key for encryption and decryption.

1.6 Significance:

- By stimulating communication channels with advanced encryption and decryption techniques, security agencies can mitigate the risks of data breaches, espionage, and unauthorized access to classified information, thereby bolstering national security.
- It enables Compliance with Regulatory Standards, my solution aligns with regulatory standards and data protection laws, ensuring that security agencies meet compliance requirements while exchanging sensitive information securely.
- Improved Operational Efficiency, Efficient communication processes and secure data exchange mechanisms enable security agencies to collaborate effectively, make informed decisions, and respond swiftly to emerging threats.

CHAPTER TWO

LITERATURE REVIEW

2.0 Introduction

Chapter One presented the background information to the study highlighting the objectives, scope and significance to the study. This chapter is about the literature review of the Encryption Decryption systems. It specifies what an Encryption Decryption system is, what it needs and how it works for its development.

2.1 Encryption Decryption System

According to Schneier (1995), Encryption and decryption systems have evolved from simple substitution ciphers used in ancient Egypt and the Roman Empire, such as the Caesar cipher, to complex mechanical devices like the Enigma machine used during World War II. The advent of computers brought significant advancements, with symmetric key encryption (like DES and AES) and asymmetric key encryption (such as RSA and ECC) becoming fundamental to modern cryptography. Hash functions, digital signatures, and Public Key Infrastructure (PKI) further enhance data integrity, authenticity, and secure communication. Quantum cryptography represents the future, leveraging quantum mechanics to ensure theoretically unbreakable security.

According to Schneier (1995) in his book, “Applied Cryptography”, defines Encryption and Decryption system as the process of converting plaintext to cipher text and the reverse process of transforming the cipher text back to its original plaintext form using a corresponding algorithm and key.

Other terminologies

- Plaintext: It is the original text which has to be encrypted.
- Cipher Text: It is the encrypted text. The text obtain after encoding the data with the help of a key is known as cipher text.
- Key: It is a word or value that is used to encrypt the plain text or decrypt the cipher text.
- Crypto Analyst: A crypto analyst is a person who is an expert in analyzing and breaking codes
- Security: The state of being free from danger or threat.

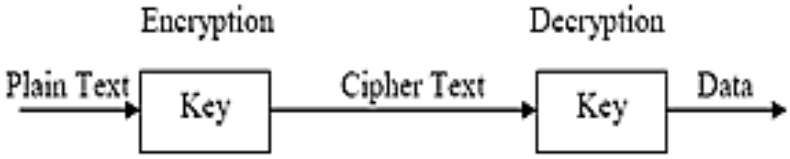


Figure 1: Cryptographic Model [2]

2.2 Types of encryption Decryption Systems

Various encryption and decryption systems are used to secure communications, data storage, and other sensitive information. The following are the main types of encryption and decryption systems commonly used.

2.2.1 Symmetric Key Encryption

According to Stallings, (2016). "Cryptography and Network Security: Principles and Practice." Pearson, Symmetric key encryption system involves using the same key for both encryption and decryption processes. This key must be securely shared between communicating parties. It is efficient in terms of computation but requires a secure channel for key exchange to prevent interception. Symmetric encryption involves several key components that work together to ensure the confidentiality and integrity of data.

a. **Encryption Algorithms used are**: - AES, DES, 3DES,

   According to Stallings, (2016), Symmetric encryption algorithms can be classified into two main types based on how they process the plaintext:

   - Block Ciphers, These algorithms divide the plaintext into fixed-size blocks (e.g., 128 bits for AES) and encrypt each block individually. Examples include AES, DES, and Blowfish.
   - Stream Ciphers, These algorithms encrypt plaintext one bit or byte at a time, generating a key stream that is combined with the plaintext. Examples include RC4 and Salsa20.

b. Key

   According to Stallings, (2016), The secret key is a crucial component that must be shared securely between the sender and the receiver. Both parties use this key for encryption and decryption. The security of the symmetric encryption depends heavily on the confidentiality and randomness of the key.

c. Initialization Vector (IV)

   According to Stallings, (2016), An IV is a random value used along with the key to ensure that identical plaintext blocks produce different cipher text blocks. This adds an additional layer of security by preventing pattern recognition. IVs are commonly used in block cipher modes like CBC (Cipher Block Chaining) and CFB (Cipher Feedback).

d. Padding

   According to Stallings, (2016), Block ciphers require the plaintext to be a multiple of the block size. Padding schemes (e.g., PKCS#7) are used to add extra bytes to the plaintext to make it the correct length. The padding is removed after decryption.

e. Modes of Operation

According to Stallings, (2016), Block ciphers can operate in different modes to achieve various security properties and functionalities. Some common modes of operation include:

- ECB (Electronic Codebook), simplest mode that encrypts each block independently. However, it is not recommended for use due to its susceptibility to pattern attacks.
- CBC (Cipher Block Chaining), each plaintext block is XORed with the previous cipher text block before being encrypted, providing better security than ECB.
- CFB (Cipher Feedback) and OFB (Output Feedback), these modes turn a block cipher into a stream cipher by encrypting smaller units of plaintext.
- CTR (Counter), Converts a block cipher into a stream cipher by encrypting counter values and XORing them with the plaintext, allowing for parallel processing.

f. Key Management

According to Stallings, (2016), Key management encompasses the procedures and protocols for generating, distributing, storing, and disposing of cryptographic keys. Effective key management is essential to maintain the security of the encrypted data and includes:

- Key Generation, The process of creating strong, random keys.
- Key Distribution, Securely sharing keys between parties.
- Key Storage, Safely storing keys to prevent unauthorized access.
- Key Rotation, Regularly updating keys to reduce the risk of compromise.
- Key Revocation, Removing or disabling keys when they are no longer needed or have been compromised.

2.2.2 Asymmetric Encryption

According to RSA (1977), Asymmetric encryption, also known as public-key cryptography, involves several key components that work together to ensure secure communication over an insecure channelwithout the need for sender and receiver to share a secret key beforehand.. Here are the primary components of asymmetric encryption:

a) Key Pair

According to RSA (1977), Asymmetric encryption uses a pair of keys: a public key and a private key.

- Public Key, This key is widely distributed and used for encryption. It can be shared openly without compromising security.
- Private Key, This key is kept secret by the owner and is used for decryption. Only the owner should have access to this key.

b) Encryption Algorithms

According to RSA (1977), these algorithms define the mathematical processes used to encrypt and decrypt data. Common asymmetric encryption algorithms include:

- RSA (Rivest-Shamir-Adleman), one of the first public-key cryptosystems, widely used for secure data transmission. RSA relies on the computational difficulty of factoring large integers.
- ECC (Elliptic Curve Cryptography), Uses elliptic curves over finite fields to provide similar security to RSA but with shorter key lengths, resulting in faster computations and reduced storage requirements.
- DSA (Digital Signature Algorithm), primarily used for digital signatures, based on the discrete logarithm problem.

c) Key Generation

According to RSA (1977), the process of generating a public and private key pair. This involves selecting two large prime numbers (for RSA) or a random point on an elliptic curve (for ECC), and then performing mathematical operations to produce the keys. Key generation needs to ensure that the keys are sufficiently large and random to prevent attacks.

d) Digital Certificates

According to RSA (1977), Digital certificates, often issued by a Certificate Authority (CA), bind a public key to the identity of its owner. These certificates include information such as the owner's name, the public key, the CA's digital signature, and the validity period. Certificate Authority (CA), a trusted entity that issues digital certificates. The CA verifies the identity of the certificate requester and signs the certificate to vouch for its authenticity. Certificate Revocation List (CRL), a list of certificates that have been revoked by the CA before their expiration date, often due to compromise or changes in the owner's status.

2.2.3 Hybrid Encryption

According to RSA Data Security (1990s), Hybrid is a cryptographic method that combines the strengths of both symmetric and asymmetric encryption to provide a secure and efficient method for encrypting and transmitting data. Here are the key components of hybrid encryption:

Hybrid Encryption Process

According to RSA Data Security (1990s), the encryption process in hybrid encryption involves two main steps:

I. **Symmetric Encryption of Data**

According to RSA Data Security (1990s), the plaintext data is encrypted using the symmetric key and the chosen symmetric encryption algorithm. This produces the cipher text, which is the encrypted version of the original data.

II. **Asymmetric Encryption of the Symmetric Key**

According to RSA Data Security (1990s), the symmetric key used in the previous step is encrypted using the recipient's public key and the chosen asymmetric encryption algorithm. This produces an encrypted symmetric key (also known as the wrapped key).

Hybrid Decryption Process

According to RSA Data Security (1990s), the decryption process in hybrid encryption also involves two main steps:

a) **Asymmetric Decryption of the Symmetric Key**

The recipient uses their private key to decrypt the encrypted symmetric key received from the sender. This recovers the original symmetric key used to encrypt the data.

b) **Symmetric Decryption of Data**

The recovered symmetric key is used to decrypt the cipher text. This produces the original plaintext data.

2.2.3.3 Applications of Hybrid Encryption

According to RSA Data Security (1990s), Hybrid encryption is widely used in various applications due to its combination of security and efficiency, including:

- Secure Email, Protocols like PGP (Pretty Good Privacy) use hybrid encryption to secure email communications.
- Secure Web Browsing, SSL/TLS protocols use hybrid encryption to secure HTTPS connections.
- Digital Payments, Hybrid encryption is used in securing transactions and sensitive data in online payment systems.
- File Encryption, Tools like Bit Locker and True Crypt use hybrid encryption to secure files and storage devices.

2.2.4 Digital Signature

According to Schneier (1995), A Digital signature is analogous to a handwritten signature or a stamped seal, but it offers far more inherent security. Digital signatures are cryptographic mechanisms used to verify the authenticity and integrity of digital messages or documents. They ensure that the message was created by a known sender (authentication) and that it was not altered in transit (integrity). This algorithm is used to create and verify the digital signature.

Common algorithms include RSA, DSA (Digital Signature Algorithm), and ECDSA (Elliptic Curve Digital Signature Algorithm).

2.2.4.1 Applications of Digital Signatures

According to Schneier (1995), Digital signatures are used in various applications to ensure secure and trustworthy digital communications, including:-

- Email Security, Ensuring that emails are sent by the purported sender and have not been altered.
- Software Distribution, Verifying the integrity and origin of software packages to prevent tampering.
- Financial Transactions, Securing online banking and electronic payments by verifying transaction details.
- Legal Documents, Ensuring the authenticity and integrity of electronic contracts and agreements.
- Block chain, Verifying transactions and maintaining the integrity of the block chain ledger.

2.2.5 Hash Function

According to Rivest (1992), a Hash Function is an algorithm that takes an input (or message) and returns a fixed size string of bytes. A hash function generates a fixed-size hash value (digest) from the original message. This hash value represents the message in a condensed form and is unique to the specific input data. Common hash functions include SHA-256, SHA-3, and MD5. A Hash Value (Digest) is a fixed-length string generated from the input message. Any change in the message will produce a different hash value, ensuring data integrity.

2.2.5.1 Message Digest

According to Rivest (1992), the result of the hash function applied to the original message. This digest is what gets encrypted with the sender's private key to form the digital signature.

2.2.6 Public Key Infrastructure (PKI)

According to Whitfield and Hellman (1976), PKI is the framework for creating a secure method for exchanging information based on public key cryptography. It includes a set of roles, policies, hardware, software, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates. PKI ensures the integrity, authenticity and confidentiality of data.

2.2.7 Quantum Cryptography

According to Bennett & Brassard (1984), Quantum Key Distribution (QKD), Uses quantum mechanics to securely exchange encryption keys. These systems and protocols form the backbone of modern applied cryptography, ensuring secure communications, data integrity, authentication, and privacy in various applications ranging from internet transactions to secure email and private communications.

2.3 Related Systems

There are huge amount of work done by the various researchers in the field of cryptographic algorithm for data security.

2.3.1 Elliptic Curve Cryptosystems for Protecting the Communication in Unsecure Network

According to Koblitz (1985) *et al*. proposed an elliptic curve cryptosystems for protecting the communication in unsecure network. Elliptic curves over finite fields of public key cryptosystems use the multiplicative group of a finite field. These systems offer same level of security as traditional cryptosystems like RSA but with much smaller key sizes, leading to faster computations and reduced storage requirements.

2.3.1.1 How it works

According to Koblitz (1985) *et al*, Elliptic curve cryptosystems operates by using mathematical properties of elliptic curves to create secure cryptographic keys in ECC, a user generates a public-private key pair, where the private key is a randomly chosen integer and the public key is a point on the elliptic curve obtained by multiplying the private key with a predefined base point on the curve. Encryption involves converting a plaintext message into a point a point on the curve and combining a cipher text pair. Decryption reverses this process using the recipient's private key to retrieve the original message point. The security of ECC stems from the Elliptic Curve Discrete logarithm problem (ECDLP), which makes deriving the private key from the public key computationally infeasible, thus ensuring secure communication even with smaller keys sizes compared to traditional cryptographic systems like RSA.

According to Koblitz (1985) *et al*, Algorithm used in this system included:-

- Elliptic Curve Diffie-Hellman (ECDH) for key exchange
- Elliptic Curve Digital Signature Algorithm (ECDSA) for Digital Signatures.
- Elliptic Curve Integrated Encryption Scheme (ECIES) for public

2.3.1.2 Strength of the system

According to Koblitz (1985) *et al*, the following are the strength of this system and they included:-

- High security with shorter key lengths compared to RSA.
- Efficient in terms of computational and memory resources.
- Smaller key sizes reduce bandwidth and storage requirements. Suitable for devices with limited resources.

2.3.1.3 Weakness of the system

According to Koblitz (1985) *et al*, the following are the weaknesses of this system and they included:-

- It was mainly based on the structure either of the multiplicative group or the multiplicative group of a finite field.
- Implementation complexity can be higher than other systems.
- Requires careful parameter selection and secure implementations to avoid vulnerabilities.
- Less mature than RSA in terms of adoption his

2.3.1.4 Conclusion

The elliptic curve cryptosystems (ECC) highlights their high security and efficiency with smaller key sizes, making them ideal for resource-constrained devices. Despite their advantages, ECC requires careful implementation to prevent vulnerabilities.

2.3.2 Rijndael method/algorithm for protecting sensitive unclassified government information

According to Daemen and Rijmen (2001), The Rijndael algorithm, selected as the Advanced Encryption Standard (AES) by the U.S. National Institute of Standards and Technology (NIST), was announced as the official standard in 2001. It was chosen for its strong security and efficiency among 15 competing algorithms.

Algorithm Used

According to Daemen and Rijmen (2001), The Rijndael algorithm employs a series of transformations that ensure confusion and diffusion of data:

- Confusion is achieved through the Sub Bytes operation using a non-linear S-box.
- Diffusion is provided by the Shift Rows and Mix Columns operations, spreading the influence of each plaintext bit over many cipher text bits.

2.3.2.1 How It Operates

According to Daemen and Rijmen (2001), Rijndael operates as a symmetric key block cipher, meaning it uses the same key for both encryption and decryption. It processes data in fixed-size blocks of 128 bits and supports key sizes of 128, 192, or 256 bits. The algorithm involves multiple rounds of data transformation, where each round consists of four primary operations:

- Sub Bytes, Non-linear substitution step where each byte is replaced with another byte using an S-box.
- Shift Rows, Transposition step where the rows of the state are shifted cyclically.
- Mix Columns, Mixing operation which combines the bytes of each column.
- AddRoundKey, Each byte of the state is combined with a round key derived from the cipher key.

The number of transformation rounds varies based on the key length: 10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys.

2.3.2.2 Strength of the system

According to Daemen and Rijmen (2001, the strength of the system are:-

- Strong encryption standard with high performance. Widely adopted and standardized (NIST AES).
- Efficient in both hardware and software implementations.
- Flexible key sizes (128, 192, 256 bits).

2.3.2.3 Weakness of the system

According to Daemen and Rijmen (2001, the system had the following weakness:-

- The performance of Rijndael algorithm based on speed of encryption, decryption process and keyset up time.
- While the algorithm itself was secured, it was prone to side-channel attacks such as timing, power analysis, and electromagnetic attacks.

2.3.2.4 Conclusion

Rijndael, as the AES standard, was a robust and efficient encryption algorithm that became the cornerstone of modern data security practices, particularly for protecting sensitive unclassified government information. Its combination of strong security, performance, and flexibility made it a preferred choice in various applications. Despite its strengths, careful implementation and key management were crucial to maintaining the security of AES-encrypted data.

2.3.3 TACIT encryption technique for secure routing

According to Prosanta Gope *et al*. (2015) proposed a new block cipher cryptographic symmetric key algorithm named TACIT encryption technique for secure routing. TACIT (Topology Aware Cryptographic Information Transmission) encryption techniques enhanced secure routing in communication networks. These techniques were designed to address security challenges specific to the routing of data across complex network topologies.

2.3.3.1 How It Operates

According to Prosanta (2015), TACIT encryption techniques involve encrypting routing information and data packets in a manner that leverages the network's topology. The core idea is to use cryptographic keys that are aware of the network's structure, enabling secure and efficient routing. The operations typically involve:

- Utilizing the knowledge of the network's structure to optimize encryption and routing decisions.
- Distributing cryptographic keys in a manner that ensures only authorized nodes can decrypt and forward data packets.
- Adapting to changes in the network topology to maintain secure routes even as the network evolves.

Algorithm Used

According to Prosanta (2015), TACIT encryption typically employs a combination of:

- Symmetric Key Encryption, for efficient encryption and decryption of data packets.
- Asymmetric Key Cryptography, for secure key distribution and management.
- Topology-Aware Key Management, ensuring that keys are distributed based on the network's structure, allowing only legitimate nodes to participate in routing.

2.3.3.2 Strength of the system

According to Prosanta (2015), the strength of this system are:-

- It used an independent approach with suitable mathematical which was assumed to be computationally secured.
- Key distribution system was being applied on a secure policy based routing.

2.3.3.3 Limitations of the system

According to Prosanta (2015), the strength of this system are:-

- It was limited to conversion of text file.

2.3.3.4 Conclusion

TACIT encryption techniques provided a robust framework for securing routing in communication networks by controlling topology awareness. These techniques enhanced security optimized routing efficiency, and offered scalability and resilience. However, the added complexity and key management overhead presented challenges that needed to be carefully managed. As networks continued to grow and evolve, TACIT techniques offered a promising approach to maintaining secure and efficient communication routes.

2.3.4 National Identification System of NITA Uganda with Embedded AES Encryption

According to (NITA-U, 2014).The National Information Technology Authority - Uganda (NITA-U) developed the national identification system to manage the registration, identification, and verification of Ugandan citizens and residents. This system, known as the National Identification and Registration Authority (NIRA), was significantly enhanced around 2014 to improve data security and operational efficiency. As part of these enhancements, the Advanced Encryption Standard (AES) was embedded into the system to secure sensitive personal data.

2.3.4.1 How AES Works in the National Identification System

According to (NITA-U, 2014). AES ensures that sensitive data such as personal identification numbers, biometric data, and demographic information are encrypted before being stored or transmitted. The encryption process involves the following steps:

- SubBytes**: Each byte in the 128-bit block is substituted with a corresponding byte from a substitution table (S-box).
- ShiftRows, The rows of the block are shifted cyclically to the left by different offsets.
- Mix Columns, The columns of the block are mixed using a linear transformation to obscure the data further.
- AddRoundKey, Each byte of the block is combined with a byte from the round key using the XOR operation.

These steps are repeated for a specified number of rounds (10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys), resulting in highly secure encrypted data that is challenging for unauthorized parties to decipher.

2.3.4.2 Strengths of AES in the National Identification System

According to (NITA-U, 2014). The strength of this system embedded in the National Identification System are:-

- AES is highly secure and resistant to known cryptographic attacks, making it ideal for protecting sensitive identification data.
- AES is efficient in both software and hardware implementations, ensuring that encryption and decryption processes do not significantly impact the system's performance.
- The algorithm's support for different key lengths allows NITA-U to choose the appropriate balance between security needs and computational efficiency.
- AES is an internationally recognized standard (FIPS PUB 197), ensuring that the national identification system complies with global security standards and best practices.

2.3.4.3 Weaknesses of AES in the National Identification System

According to (NITA-U, 2014). The weaknesses of this system embedded in the National Identification System are:-

- Effective key management is critical to maintaining the security of AES. Challenges include secure key generation, distribution, storage, and periodic rotation.
- Poor implementation of AES can introduce security vulnerabilities. Ensuring that AES is implemented correctly and securely is essential to avoid potential weaknesses.
- Higher security configurations, such as using 256-bit keys, require more computational resources, which can impact the system's performance, especially in resource-constrained environments.

2.3.4.4 Conclusion

The integration of AES into the national identification system of NITA Uganda significantly enhances the security of personal identification data, protecting it from unauthorized access and cyber threats. AES's strengths in security, efficiency, flexibility, and compliance make it an excellent choice for the system's needs. However, effective key management, secure implementation, and resource optimization are crucial to fully leverage AES's benefits and ensure the system's integrity and performance.

2.3.5 The Uganda Revenue Authority (URA) with Embedded AES Encryption

According to URA (2010). The Uganda Revenue Authority (URA) customs system is designed to manage and streamline customs operations, ensuring efficient and secure processing of goods entering and leaving Uganda. Around 2010, the URA integrated the Advanced Encryption Standard (AES) into its customs system as part of an extensive upgrade to bolster data security and protect sensitive information.

2.3.5.1 How AES Works in the URA Customs System

According to URA (2010, AES is a block cipher that encrypts data in fixed blocks of 128 bits and supports key sizes of 128, 192, and 256 bits. In the URA customs system, AES ensures that sensitive data, such as transaction records, personal information, and trade details, are encrypted before being stored or transmitted. This encryption process involves multiple rounds of transformations, depending on the key size used:

- Sub Bytes, Each byte in the data block is substituted with a corresponding byte from a substitution table (S-box).
- Shift Rows, The rows of the data block are shifted cyclically to the left by different offsets.
- Mix Columns, The columns of the data block are mixed using a linear transformation to further obscure the data.

- AddRoundKey, Each byte of the data block is combined with a byte from the round key using the XOR operation.

These steps are repeated for a specified number of rounds—10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys—resulting in highly secure encrypted data that is difficult for unauthorized parties to decipher.

2.3.5.2 Strengths of AES in the URA Customs System

According to URA (2010), the benefits of the system are:-

- AES is highly secure and resistant to known cryptographic attacks, making it ideal for protecting sensitive customs data.
- AES is efficient in both software and hardware implementations, ensuring that encryption and decryption processes do not significantly impact the performance of the URA customs system.
- The algorithm's support for different key lengths allows the URA to choose the appropriate balance between security needs and computational efficiency.
- AES is an internationally recognized standard (FIPS PUB 197), ensuring that the URA customs system complies with global security standards and best practices.

2.3.5.3 Weaknesses of AES in the URA Customs System

According to URA (2010, the benefits of the system are:-

- Effective key management is critical to maintaining the security of AES. This includes challenges such as secure key generation, distribution, storage, and periodic rotation.
- Poor implementation of AES can introduce security vulnerabilities. Ensuring that AES is implemented correctly and securely is essential to avoid potential weaknesses.
- Higher security configurations, such as using 256-bit keys, require more computational resources, which can impact the system's performance, especially in resource-constrained environments.

2.3.5.4 Conclusion

The integration of AES into the URA customs system significantly enhances the security of customs data, protecting it from unauthorized access and cyber threats. AES's strengths in security, efficiency, flexibility, and compliance make it an excellent choice for the URA's needs. However, effective key management, secure implementation, and resource optimization are crucial to fully leverage AES's benefits and ensure the system's integrity and performance.

2.3.6 Stannic bank Uganda online banking platforms with embedded AES and RSA Algorithms

According to Stanbic (2024), Stanbic Bank Uganda has progressively developed and upgraded its online banking platform over the years. The bank's commitment to enhancing digital banking services has seen continuous improvements, with significant upgrades being made in recent years to incorporate advanced encryption technologies such as AES (Advanced Encryption Standard) and RSA (Rivest-Shamir-Adelman).

2.3.6.1 How It Works

According to Stanbic (2024), Stanbic Bank's online banking platform integrates AES and RSA encryption to ensure the security of financial transactions and customer data. AES is used for encrypting bulk data within the platform. When a customer performs a transaction, their data is encrypted using a symmetric key algorithm. This ensures that sensitive information such as account details and transaction history remains confidential during transmission over the internet. RSA is utilized for secure key exchange and to establish secure connections. When a user logs into the online banking platform, RSA encryption is used to securely exchange keys between the user's device and the bank's server. This ensures that the communication channel is protected against eavesdropping and man-in-the-middle attacks.

2.3.6.2 Benefits of Embedded Algorithms in the System

According to Stanbic (2024), here are the benefits of AES and RSA in the Stanbic Bank online banking platform:-

- The combination of AES and RSA provides robust security for online transactions and customer data, protecting against unauthorized access and data breaches.
- These encryption methods ensure that data transmitted between customers and the bank remains intact and unaltered, maintaining the integrity of financial transactions.
- By employing industry-standard encryption algorithms, Stanbic Bank builds and maintains trust with its customers, assuring them that their personal and financial information is secure.
- The use of advanced encryption helps Stanbic Bank comply with national and international data protection regulations, reducing the risk of legal and financial penalties.

2.3.6.3 Weaknesses of Embedded Algorithms in the System

According to Stanbic (2024), here are the weaknesses of AES and RSA in the Stanbic Bank online banking platform:-

- Implementing and maintaining advanced encryption algorithms can be complex and costly, requiring specialized knowledge and resources.

- Encryption and decryption processes can introduce latency, potentially affecting the performance of the online banking platform.
- Ensuring the secure generation, distribution, and storage of encryption keys can be challenging. Any compromise in key management can undermine the security of the entire system.
- Strong security measures may sometimes impact user experience, requiring additional authentication steps that can be perceived as inconvenient by users.

2.3.6.4 Conclusion

Stanbic Bank Uganda's online banking platform exemplifies the integration of advanced encryption technologies to enhance security and protect customer data. By employing AES for data encryption and RSA for secure key exchange, the bank ensures robust protection against cyber threats, fostering customer trust and regulatory compliance. Despite the challenges of complexity, cost, and performance overhead, the benefits of maintaining a secure and trustworthy online banking platform far outweigh the potential drawbacks. This commitment to security highlights Stanbic Bank's dedication to providing safe and reliable digital banking services in Uganda.

2.4 COMPARISON OF RELATED SYSTEMS

Table 1: Comparisons for the Related Systems

FEATURES	STRENGTH	WEAKNESS	TECHNOLOGY
Elliptic Curve Cryptosystems (ECC)	➢ High security with shorter key lengths compared to RSA. ➢ Efficient in terms of computational and memory resources. ➢ Smaller key sizes reduce bandwidth and storage requirements. ➢ Suitable for devices with limited resources.	➢ Implementation complexity can be higher than other systems. ➢ Requires careful parameter selection and secure implementations to avoid vulnerabilities. ➢ Less mature than RSA in terms of adoption history.	➢ System based
TACIT Encryption Techniques for Secure Routing	➢ Enhances security by integrating cryptographic techniques with trust-based routing. ➢ Protects against various network attacks,	➢ Can introduce significant computational and communication overhead due to trust management. ➢ Complexity in trust calculation and maintenance.	➢ IoT System

	including spoofing and eavesdropping. ➢ Provides both data and route security in dynamic networks.	➢ Less standardized and widespread compared to ECC and AES.	
Rijndael Method (AES)	➢ Has Strong encryption standard with high performance. ➢ Widely adopted and standardized (NIST AES). ➢ Its Efficient in both hardware and software implementations. ➢ Flexible key sizes (128, 192, 256 bits).	➢ Symmetric key requirement means secure key distribution and management are critical. ➢ Vulnerable to side-channel attacks if not properly implemented. ➢ Larger key sizes can affect performance in resource-constrained environments.	➢ Web based
National Identification System of NITA Uganda with Embedded AES Encryption	➢ High level of data protection and confidentiality. ➢ Compliance with international security standards. ➢ Strong authentication processes.	➢ Complex implementation and maintenance. ➢ High cost of encryption infrastructure. ➢ Potential performance impact due to heavy encryption.	➢ Embedded On a web based System
The Uganda Revenue Authority (URA) with Embedded AES Encryption	➢ Secure data transmission and storage of customs-related information. ➢ Prevention of data breaches and unauthorized access. ➢ Compliance with national and international trade security standards.	➢ Complexity in managing and maintaining encryption keys. ➢ Possible performance issues during high transaction volumes. ➢ High implementation costs.	➢ Embedded On a web based System
Stanbic bank Uganda online banking platforms with embedded AES	➢ Enhanced security for online and mobile banking transactions. ➢ Protection against cyber threats and fraud.	➢ Complexity and cost of implementing and maintaining encryption algorithms.	➢ Embedded On a web based System

and RSA Algorithms	➢ Compliance with financial regulations.	➢ Potential latency introduced by encryption and decryption processes. ➢ Challenges in secure key management.

2.5 Conclusion

This chapter mainly described the literature review of the Encryption Decryption systems, highlighting various types and the related systems, their Strengths and Weaknesses, ECC's efficiency, AES's widespread use, key management issues, and TACIT's specialized applications. Continued research is needed to optimize these systems and address their limitations, enhancing overall cryptographic security. This foundation paves the way for future advancements in secure communications.

CHAPTER THREE

Research Methodology

3.0 Introduction

The methodology is designed to ensure the system's effectiveness, security, and user-friendliness. The methodology encompasses distinct patterns of research, systematic approaches to data collection, robust techniques for data analysis, and a suite of tools for design and implementation. The Methodology was in relation to the object of the proposed Encryption Decryption System.

3.1 System Study and Analysis

To determine the system and user requirements, as well as the system inputs and outputs, a variety of fact-finding techniques were employed in this project. These techniques played a crucial role in defining the system's expected functionalities. The methods used included:

3.2 Data Collection techniques

Several methodologies can be used to collect data for an encryption and decryption app project, depending on the specific objectives, resources, and constraints. Here are some commonly used methodologies:

3.2.1 Interview

Interviews provide a flexible method for engaging with stakeholders, adaptable to structured, unstructured, or semi-structured formats. By conducting comprehensive interviews with stakeholders such as IT officers, COMSEC officers, directors, and other security operatives, I sought to identify and define both functional and non-functional requirements specific to my encryption and decryption system. By utilizing a combination of semi-guided and unguided interview techniques, and employing both closed and open-ended questions, I gained valuable insights into the operational dynamics, strengths, challenges, and information flow of the current system. This approach facilitated a deep understanding of the system's complexities, ensuring precise requirement specification crucial for developing a robust encryption and decryption solution.

3.2.2 Observation

I personally conducted direct observations of operational tasks of the current system, such as encryption and decryption processes, key management practices, and system performance under various loads. These observations included assessing storage methods, congestion levels, and other relevant operational dynamics. I documented these observations using pen and paper to ensure thorough and accurate recording of all findings.

3.2.3 Reviewing existing documents

I undertook an extensive investigation into encryption systems, examining various sources related to the technology to gather crucial details. Additionally, a thorough literature review provided the essential information required for their research into encryption methodologies and practices

3.2.4 Questionnaires

I employed questionnaires as a method to gather comprehensive insights from users of the encryption system at MODVA. These surveys were strategically distributed among selected system users to collect valuable statistical data. By analyzing the feedback received, the researchers identified specific challenges related to encryption processes and system usability. This approach helped prioritize improvements necessary to enhance the system's efficiency and effectiveness in safeguarding data. The findings from the questionnaire, guided the development process by highlighting user requirements and areas for enhancement within the encryption system framework.

3.3 Data Analysis Methods

I used analysis software like Microsoft Excel for quantitative data and Microsoft Word for qualitative data to carefully gather and record observations about the encryption decryption system. I looked at things like how fast data gets encrypted and decrypted, how the system reacts under different amounts of use, and what users said about how easy it is to use. I made graphs and charts to show these things clearly. By doing this, I found patterns, figured out where the system could work better, and found ways to make it faster and more user-friendly. This helped me make smart choices to improve the system overall.

3.4 System Analysis and Design

The analysis and design of the encryption decryption system involved carefully defining inputs, processes, and outputs while following system rules. This method ensured that data was securely transformed while keeping it confidential and intact. Tools like data flow diagrams showed how information flowed through the system during encryption and decryption. At the same time, entity-relationship diagrams clarified how different parts of the system connected and depended on each other. These diagrams were crucial in developing a strong encryption decryption system that met technical standards and matched organizational security needs and user expectations.

3.4.1 System Analysis

In encryption decryption systems, systems analysis involves identifying and analyzing data to understand what the system needs to do. This includes defining both functional tasks, like securely encrypting and decrypting data and managing cryptographic keys, and non-functional aspects, such as ensuring security, reliability, scalability, and ease of use. These requirements form the basis for designing a strong system that meets technical standards and meets the needs of the organization and its users. This systematic approach ensures data protection, reduces security risks, and improves operational efficiency, fostering ongoing innovation in data security strategies.

3.4.1.1 Functional Requirements:

A. User Authentication

- Users must be authenticated before accessing the secure communication platform.
- The system should provide Support for multi-factor authentication methods such as passwords, biometrics, or security tokens.
- The system should be able to manage user roles and permissions based on authentication credentials.

B. Message Encryption and Decryption

- The system shall be able to encrypt outgoing messages and decrypt
- incoming messages using. Advanced Encryption Algorithm
- The system shall provide Support for symmetric encryption techniques.
- The system shall be able to Seamlessly integrate encryption and decryption processes into the messaging workflow.

C. Key Management

- The System shall Generate, distribute, and store encryption keys in a secure manner.
- The System shall provide Key rotation and expiration policies to enhance security.
- The System shall provide Key recovery mechanisms to facilitate data access in case of key loss or compromise.

D. Secure Transmission

- The System shall use secure communication protocols (e.g., SMTP) for data transmission over networks.
- The System shall be able to provide Protection against man-in-the-middle attacks and eavesdropping.

E. User Interface

- The System shall comprise of Intuitive and user-friendly interface for encryption, and encrypted messages.
- The System shall provide clear indication of message encryption status and options for managing encryption settings.
- The System shall provide Accessibility features to accommodate users with disabilities

F. Logging and Auditing

- The System shall Log user activities, including login attempts, message encryption, and decryption operations.
- The System shall track security events, monitor system usage, and detect anomalies.
- The System shall Comply with data retention policies and regulatory requirements for audit logs.

3.4.1.2 Non-functional requirements

A. Performance

- The System shall efficiently handle message encryption and decryption operations to minimize latency.
- The System shall support a large number of users and messages without degradation in performance.

- The System shall optimize resource to minimize system resource utilization and maximize output.

B. Availability

- The System shall be available and reliable to ensure uninterrupted access.
- The System shall provide redundant and failover mechanisms to mitigate service disruptions and minimize downtime.
- The System shall provide regular maintenance and updates to address security vulnerabilities and performance issues.

C. Security

- The System shall implement industry-standard security practices and protocols to protect against unauthorized access and data breaches.
- The System shall provide regular security assessments and penetration testing to identify and remediate vulnerabilities.
- The System shall comply with data protection regulations and privacy laws governing the handling of sensitive information.

D. Interoperability

- The System shall be compatible with existing communication systems and messaging protocols used within the organization.
- The System shall integrate third-party applications and services for seamless data exchange and collaboration.
- The System shall provide support for cross-platform operation on desktop, and web-based environments.

E. Scalability

- The System shall be able to scale horizontally and vertically to accommodate increasing user demands and message volumes.
- The System shall provide flexible architecture that allows for easy expansion and deployment in distributed environments.

3.4.2 System Design

Process modeling involved the creation of Data Flow Diagrams (DFDs) to illustrate how data moves and interacts across different parts of the encryption decryption system. These diagrams provided a visual representation of the entire data flow process, from where data enters the system to how it is processed and eventually produced as

outputs. They were crucial in developing detailed process models that outlined every step and interaction within the system. Information for constructing these DFDs was derived from the Data Dictionary, ensuring a thorough understanding of how data flows and changes within the system.

3.4.2.1 Data flow diagrams

DFDs (Data Flow Diagrams) are graphical representations of the flow of data through an information system. They are often used to model the processes involved in a system, including encryption and decryption applications. Here's a simplified DFD for an encryption and decryption application:

3.4.2.1 DFD0: Context Level Diagram

The DFD0, or Context Level Diagram, provides an overview of the system from a high-level perspective, showing the interactions between the system and external entities. In the case of an encryption and decryption application, the main external entities could be users, data sources, and possibly external storage.

A. Components:

I. External Entities

- Users: Individuals interacting with the encryption and decryption application.
- Data Source: The source of the data to be encrypted or decrypted.
- External Storage: Where encrypted or decrypted data might be stored.

B. Processes

- Encryption/Decryption System: Represents the core functionality of the application.

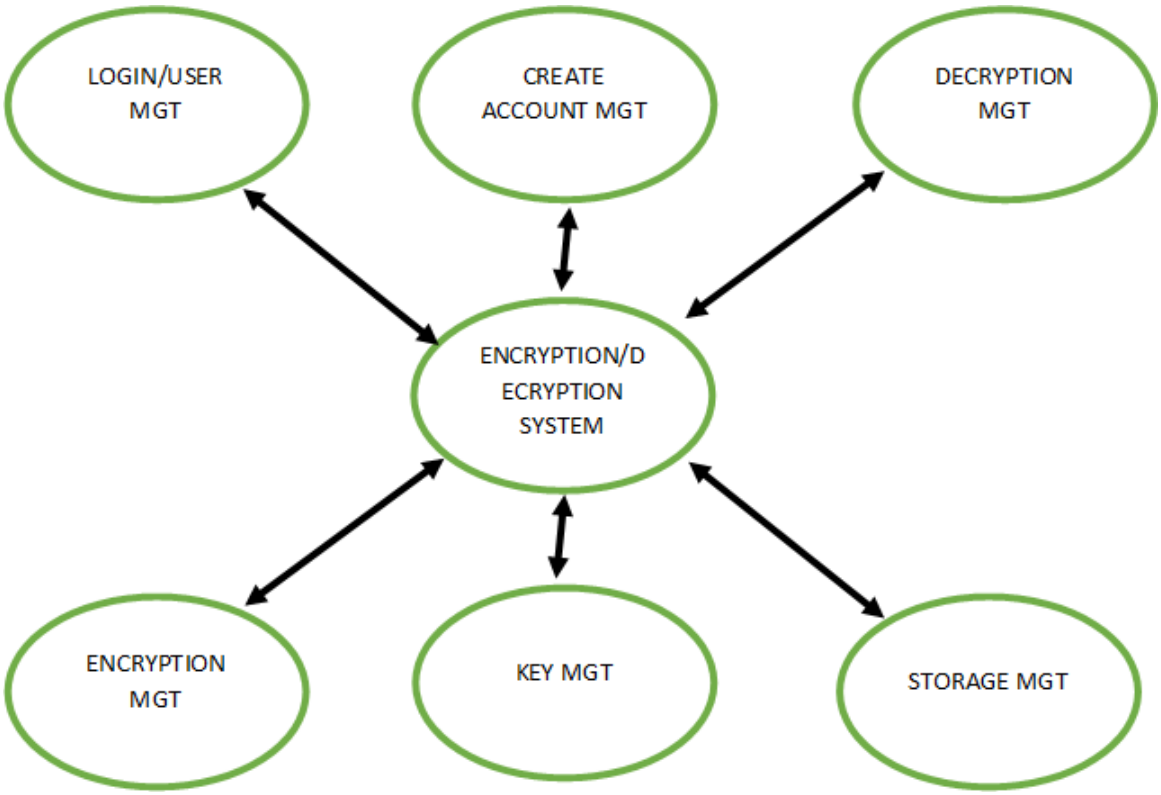
C. Data Flows

- Input Data: Represents data flowing into the system for encryption or decryption.
- Encrypted/Decrypted Data: Data flowing out of the system after encryption or decryption.

D. Data Stores:

- Encrypted Data Store: Stores encrypted data before or after processing.
- Decrypted Data Store: Holds decrypted data before final delivery or storage.
- Key Storage: Repository for securely storing encryption keys.

ILLUSTRATION OF DFD0



3.4.2.2 DFD1: Level 1 Data Flow Diagram

The DFD1 elaborates on the processes and data flows identified in the DFD0, breaking them down into more detail.

A. Components:

I. Processes

- Input Processing: Responsible for receiving data from external entities and preparing it for encryption or decryption.
- Encryption Process: Takes plaintext data and converts it into ciphertext.
- Decryption Process: Takes cipher text data and converts it back into plaintext.
- Output Processing: Prepares encrypted or decrypted data for output to external entities.

II. 2. Data Stores

- Plaintext Data Store: Stores data before encryption.
- Cipher text Data Store: Stores data after encryption.

III. Data Flows

- Input Data: Flow of plaintext data into the system.

- Encrypted Data: Flow of cipher text data out of the encryption process.
- Decrypted Data: Flow of plaintext data out of the decryption process.

ILLUSTRATION OF DFD1

3.4.2.3 DFD2: Level 2 Data Flow Diagram

The DFD2 provides even more detail, breaking down the processes and data flows identified in DFD1 into sub-processes and data transformations.

A. Components:

Sub-Processes

I. Key Generation:

- Sub-process responsible for generating encryption and decryption keys.
- Encrssyption Algorithm: Sub-process that implements the encryption algorithm.
- Decryption Algorithm: Sub-process that implements the decryption algorithm.

II. **Data Stores**

- Key Store: Stores encryption and decryption keys securely.

III. **Data Flows**

- Key Generation Data Flow: Flow of data related to key generation process.
- Encrypted Data Flow: Flow of data during encryption process. Decrypted Data Flow: Flow of data during decryption process.

These three levels of Data Flow Diagrams provide a comprehensive understanding of the encryption and decryption application, detailing the processes, data flows, and interactions between various components of the system. This structured approach helps in visualizing the system's functionality and identifying potential areas for improvement or optimization.

ILLUSTRATION OF DFD2

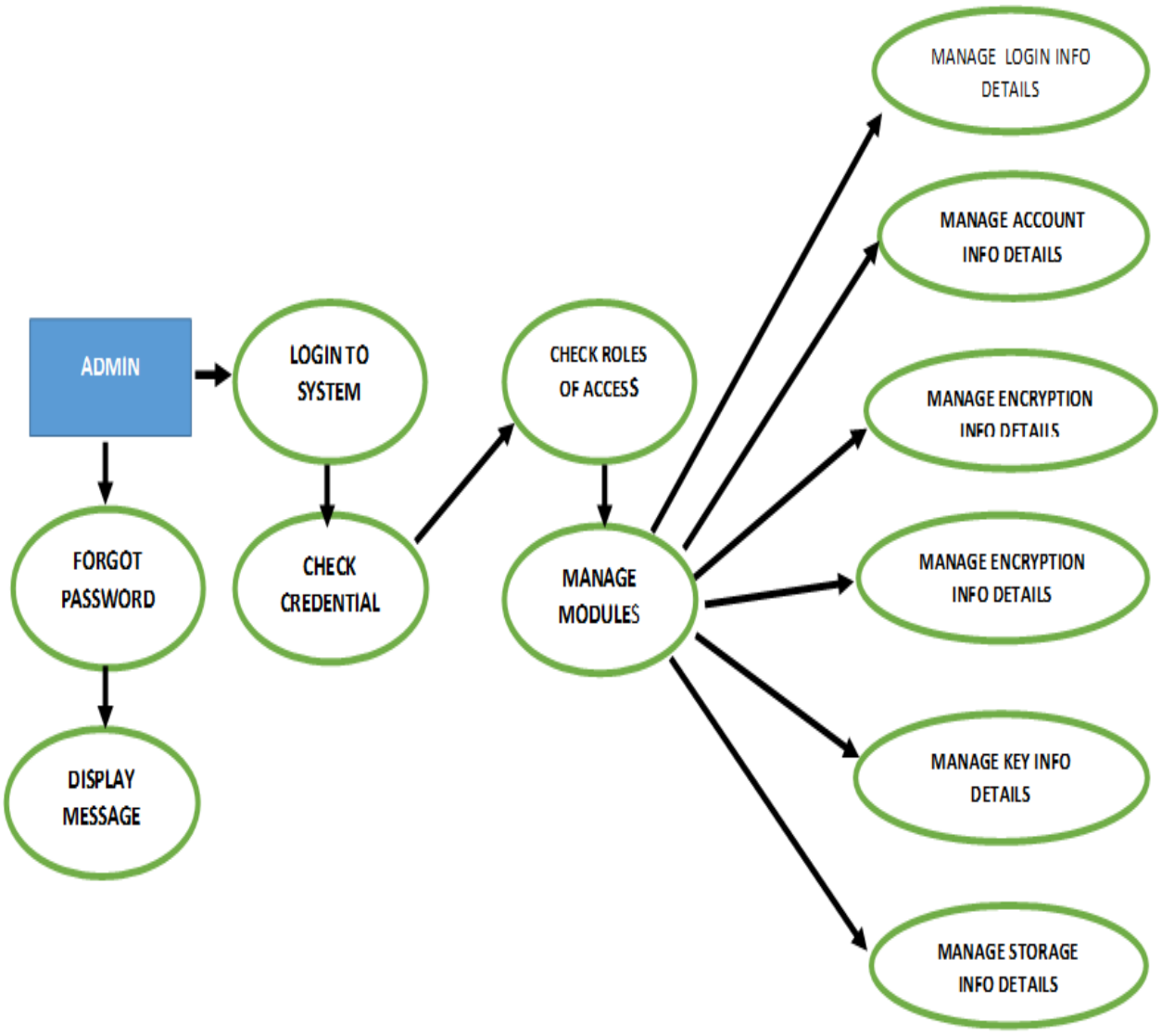


FIG 1 showing Processing flow diagram of AES algorithm

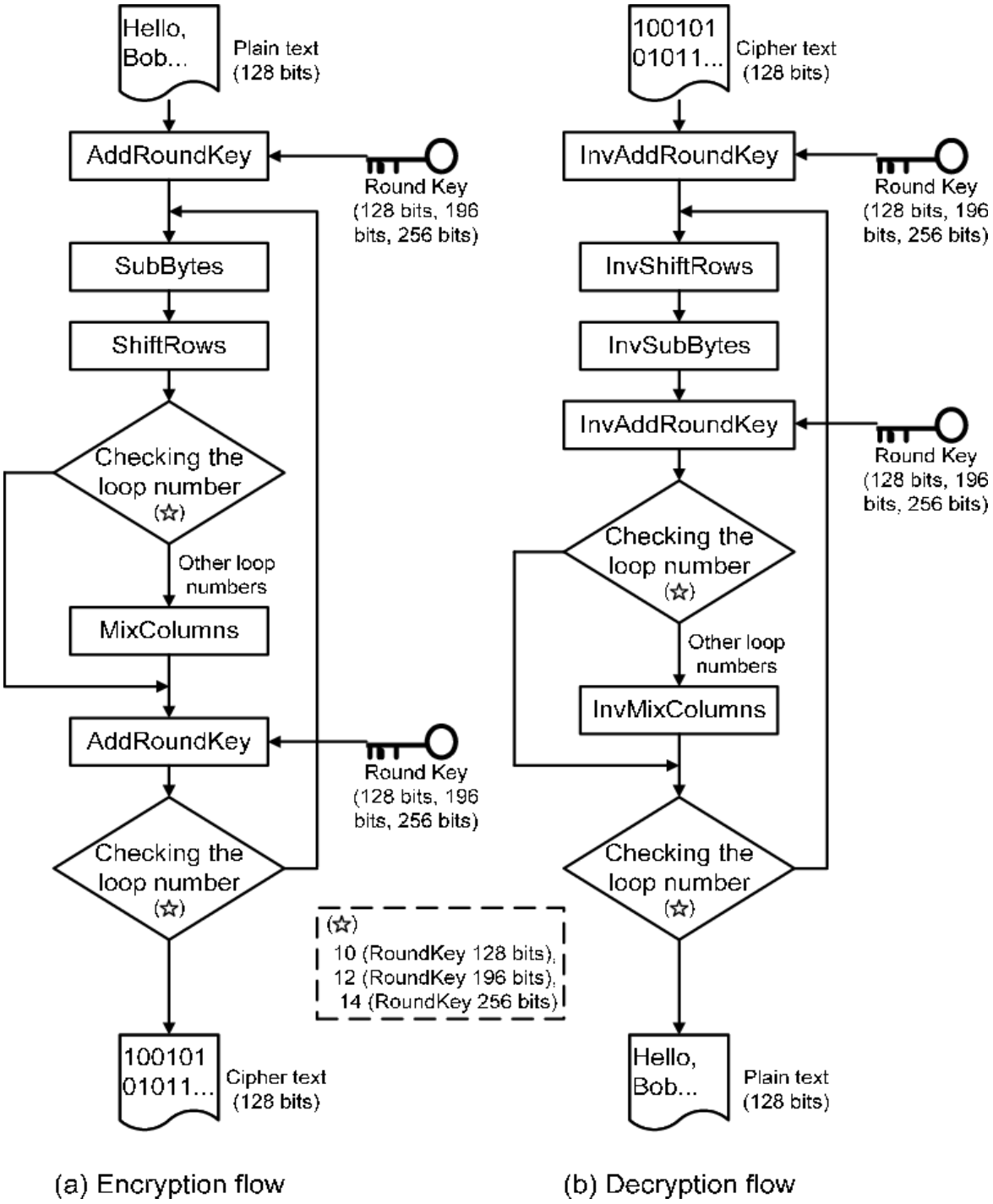
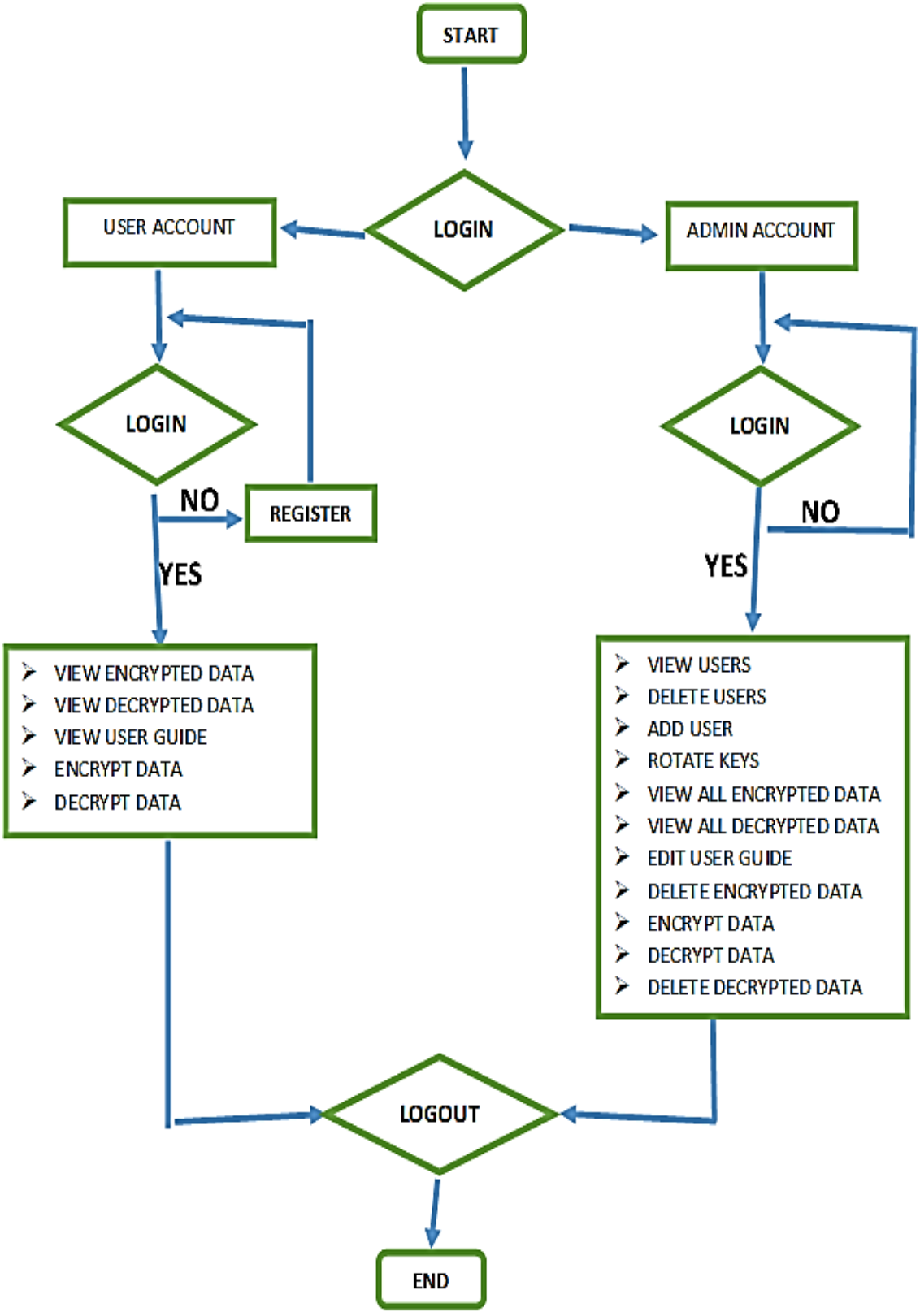


Figure 3 depicts the system flowchart of the encryption and decryption system.



Data modeling, on the other hand, utilized Entity-Relationship Diagrams (ERDs) to define the data requirements and relationships within the encryption decryption system. ERDs depicted entities (like data objects or tables), their attributes (characteristics or properties), and the relationships between them. This visualization helped in establishing the structure of relationships and attributes needed for the system's relational database schema. It enabled effective design and organization of data storage, retrieval, and manipulation processes within the system, ensuring efficient handling and utilization of data resources.

3.5 System Implementation

The system implementation phase is where the designs of the database and the application are turned into working systems. This phase involves creating the actual database and the application programs. I used the Data Definition Language (DDL) of the selected Database Management System (DBMS) for this task. DDL commands help in defining, modifying, and managing the database structure, ensuring it matches the design specifications.

First, I set up the database schema using DDL commands to create tables, indexes, and relationships. Then, I developed application programs that interact with the database, handling data input, processing, and retrieval. These programs ensure the application performs as required. During this phase, I also conducted thorough testing to identify and fix any issues, ensuring the system operates efficiently and without errors.

3.5.1 Implementation Tools

Various tools were used during the implementation stage to aid in the development of the system. The main tools included XAMP/Apache server, MySQL, PHP, VScode, and the Windows operating system. Each tool was chosen for its specific capabilities and contributions to the development process.

3.5.1.1 Visual studio code

Visual Studio Code (VSCode) is a widely-used and highly adaptable code editor created by Microsoft. It has become extremely popular in web development because of its powerful features and extensive integrations. These make VSCode an excellent environment for building, testing, and deploying web applications. It supports various programming languages and offers a wide range of extensions that enhance its

functionality. This versatility makes VSCode an ideal choice for projects involving technologies like Apache, MySQL, and PHP. Below is a detailed explanation of how VSCode operates and how it integrates with these components.
In my project, I used VScode extensively for coding. Vscode syntax is straightforward and flexible, allowing for quick development and deployment. Using the VScode, I integrated HTML, PHP, and CSS content that is used run.

3.5.1.2 XAMPPServer

XAMPPServer is a complete server package that provides a solid environment for web development. It includes Apache, MySQL, and PHP, which are crucial for creating dynamic web applications. Apache is a widely used web server that allows developers to host and manage web pages.

Once XAMPPServer is installed, it configures Apache on your computer, making it possible to save web pages in a designated directory that can be accessed through the machine's IP address. This configuration provides developers with the ability to test and improve their web applications locally before making them live on the internet. The inclusion of MySQL and PHP in this setup allows for smooth communication between the server, database, and application code, creating an integrated and efficient development environment.

Apache

Apache HTTP Server, commonly referred to as Apache, is a highly popular and powerful web server software. It is responsible for handling requests from clients, such as web browsers, and serving web pages in response to those requests. Here's how it works:
Installation and Setup: When XAMPPServer is installed, Apache is set up on your local machine. This allows your computer to act as a server, hosting web pages that can be accessed locally or over a network.

- **Hosting Web Pages**: Apache serves web pages stored in a specific directory on your machine. In XAMPPServer, this directory is typically the "htdocs" folder. You place your HTML, PHP, and other web files in this directory.
- **Handling Requests**: When a web browser requests a page from the server, Apache processes the request and delivers the corresponding web page. For example, if you enter "http://localhost/index.php" in your browser, Apache will look for the "index.php" file in the "htdocs" folder and serve it to the browser.

- **Configuration**: Apache is highly configurable. You can modify its settings using the "httpd.conf" configuration file to manage aspects like directory permissions, URL rewriting, and server modules.

3.5.1.3 PHP

In my project, I used PHP extensively to build the content management system. PHP's syntax is straightforward and flexible, allowing for quick development and deployment. PHP scripts run on the server, generating HTML content that is sent to the client's browser. This server-side processing enables the creation of interactive and dynamic web applications, where content can be generated and modified based on user input and database queries.

PHP (Hypertext Preprocessor) is a server-side scripting language that is embedded within HTML to create dynamic web pages. Here's how PHP works within XAMPPServer:

- **Server-Side Processing**: When a browser requests a PHP page, Apache processes the request and passes the PHP code to the PHP interpreter. The interpreter executes the PHP code on the server.
- **Dynamic Content**: PHP can generate dynamic content based on user interactions, database queries, and other logic. For example, a PHP script can display a personalized greeting based on user input or retrieve and display data from a MySQL database.
- **Database Interaction**: PHP includes functions to connect to MySQL databases, execute SQL queries, and handle the results. This allows your web application to store and retrieve data as needed.
- **Embedding in HTML**: PHP code is embedded within HTML tags using the "<?php ... ?>" syntax. This allows you to mix PHP code with HTML to create dynamic web pages.

3.5.1.4 MySQL

MySQL is a popular open-source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL) for database operations. It is known for its speed, reliability, and flexibility, making it an ideal choice for web applications. MySQL supports multi-user and multi-tasking operations, allowing it to handle large amounts of data and multiple concurrent users efficiently.

In my implementation, I used MySQL to manage the database, storing all the data required by the application. Its open-source nature allows for customization and scalability, ensuring the database can grow and adapt to changing needs. I used SQL commands to define the database schema, insert and retrieve data, and manage relationships between tables. MySQL's robustness and performance were crucial in ensuring the reliability and efficiency of the system.

Here's how it works within XAMPPServer:

- Database Creation: You can create databases using MySQL to store various types of data needed by your web application, such as user information, product details, and transaction records.
- Structured Query Language (SQL): MySQL uses SQL for database operations. SQL commands allow you to create tables, insert data, retrieve data, update records, and delete entries. For example, the command "CREATE TABLE users (id INT, name VARCHAR(100));" creates a new table called "users."
- Data Handling: MySQL is designed to handle multiple users and large amounts of data efficiently. It supports indexing, which speeds up data retrieval, and transactions, which ensure data integrity.
- PHP Integration: PHP scripts can connect to MySQL databases using built-in functions. This integration allows your web application to dynamically interact with the database, such as displaying user data on a webpage or processing form submissions.

3.5.1.5 HTML

HTML, or Hypertext Markup Language, is the standard language for creating web pages and web applications. It provides the structure and layout for web content, using elements such as headings, paragraphs, links, and lists. HTML also supports embedding multimedia elements like images and videos and integrates with other web technologies like CSS and JavaScript.

In my project, I used HTML to build the front-end interface of the content management system. HTML tags and attributes were used to define the structure and presentation of web pages, ensuring a consistent and user-friendly interface. The use of HTML allowed us to create a responsive and interactive web application, enhancing the overall user experience.

3.5.1.6 CSS (Cascading Style Sheets)

CSS (Cascading Style Sheets) is integral to system implementation as it defines the visual presentation and layout of web-based applications. By styling HTML elements, CSS ensures consistency in design across all system components, enhancing usability and user experience. It enables developers to create responsive designs that adapt seamlessly to various devices and screen sizes, improving accessibility and user engagement. CSS also supports modular design practices, facilitating code reusability and maintainability. Through frameworks and preprocessors, such as Bootstrap and Sass, CSS streamlines development by providing ready-made styles and tools for customization, branding, and performance optimization. Overall, CSS plays a critical role in shaping the aesthetic appeal, functionality, and accessibility of systems, making it essential for creating intuitive and visually appealing user interfaces.

In summary, the system implementation phase involved the careful and systematic development of the database and application programs. By using a range of tools, including XAMP/Apache server, PHP, MySQL, HTML, and CSS, I was able to create a robust and functional system. Each tool played a vital role in different aspects of the development process, contributing to the successful realization of the project.

3.6 System Testing and Validation

3.6.1 Testing

Testing within the system implementation process is a critical phase aimed at verifying the functionality and performance of application programs. The primary objective is to uncover errors or bugs in the software and ensure that it operates as expected under various conditions. Initially, the testing process involved executing the application programs and systematically identifying any faults that emerged. Each identified issue was meticulously corrected, and the testing cycle repeated until the system consistently performed according to the specified requirements and performance benchmarks.

- **System Performance and Efficiency Testing**: The testing phase included rigorous evaluation of the system's performance metrics such as response times, throughput, and resource utilization. This ensured that the system could handle expected workloads efficiently without performance degradation. Metrics like CPU usage, memory consumption, and disk space utilization were monitored to optimize system efficiency and ensure smooth operation.

- **Compatibility Testing**: Another crucial aspect of testing was verifying the system's compatibility across different operating environments. Compatibility tests were conducted on various operating systems like Windows XP, Linux distributions, and Windows 7. This process involved deploying the system on each platform and evaluating its functionality to confirm seamless operation across diverse environments. By ensuring compatibility, the system could reach a broader user base and maintain consistent performance across different platforms.
- **Security Testing**: Security testing focused on assessing the system's resilience against potential threats and vulnerabilities. This included testing for vulnerabilities that could be exploited by remote attacks and evaluating the effectiveness of authentication mechanisms. Measures such as penetration testing, vulnerability scanning, and authentication scenario testing were employed to identify and address security weaknesses. By prioritizing security testing, the system was fortified against unauthorized access and data breaches, safeguarding sensitive information and ensuring compliance with security standards.

3.6.2 Validation

Validation within the system implementation context was a comprehensive evaluation process to confirm that the Encryption Decryption System effectively met its intended purpose and satisfied the needs of its users. This phase focused on verifying whether the system aligned with the identified user requirements, functional specifications, and non-functional criteria.

- **Validation of User Requirements**: The validation process involved engaging end-user representatives who interacted directly with the system. These representatives tested the system functionalities and workflows to validate that they met the intended user requirements. Feedback from end users was collected and analyzed to ensure that the system's features and usability aligned with user expectations and operational needs.
- **Functional and Non-functional Validation**: Beyond user requirements, validation encompassed a thorough assessment of the system's functional capabilities and non-functional aspects such as performance, reliability, and scalability. Functional validation verified that the system performed tasks accurately and efficiently,

meeting operational goals like financial transaction processing and account management. Non-functional validation ensured that the system could handle concurrent user sessions, maintain data integrity under load, and scale effectively to accommodate future growth.

- **Continuous Improvement**: Throughout the validation phase, continuous improvement and refinement of the system were prioritized based on user feedback and testing outcomes. Adjustments and enhancements were made iteratively to optimize system performance, address any identified gaps, and enhance overall user satisfaction. By validating the system against comprehensive criteria and user expectations, stakeholders could confidently deploy the Online Financial Transfer Management System, knowing it was robust, reliable, and aligned with organizational goals.

3.6.3 Conclusion

In conclusion, this chapter detailed the diverse methodologies used throughout the research, covering research patterns, data collection approaches, analysis techniques, and system design tools. It began by highlighting both qualitative methods (like observation, interviews, Questionnaires, and Literature review) and quantitative methods (such as data analysis) to ensure a thorough understanding of the study's goals. Data collection was accurately planned, using direct observations, structured interviews, and online questionnaires to gather relevant and reliable data. The analysis was conducted, ensuring robust and credible findings. Additionally, the chapter examined the tools and technologies used for system design and implementation, such as integrated development environments (IDEs), version control systems, and database management systems, emphasizing their critical role in the project's success. This comprehensive approach laid a strong foundation for the system's effective design and implementation.

CHAPTER FOUR

4.0 System Study, Analysis and Design

This chapter concerns the study of the existing system, analysis of the requirements for the system, process and data modeling.

4.1 The study of the Existing System

Through a comprehensive analysis of current Wireless Messaging Terminals (WMTs), including interviews, observations, and document reviews, researchers identified several issues. Users experienced delays in message delivery and inefficiencies in communication processes. These problems led to communication slowdowns and hindered operational effectiveness, demonstrating the need for an improved an Encryption Decryption system.

An in-depth examination of the WMT system produced a flow chart that details the communication workflow. This chart outlines the process from the initial terminal setup to its full network integration, aiming to enhance message transmission efficiency and reliability (see Figure 4.1)

4.1.1 Workflow for the Wireless Messaging TerminalProcesses

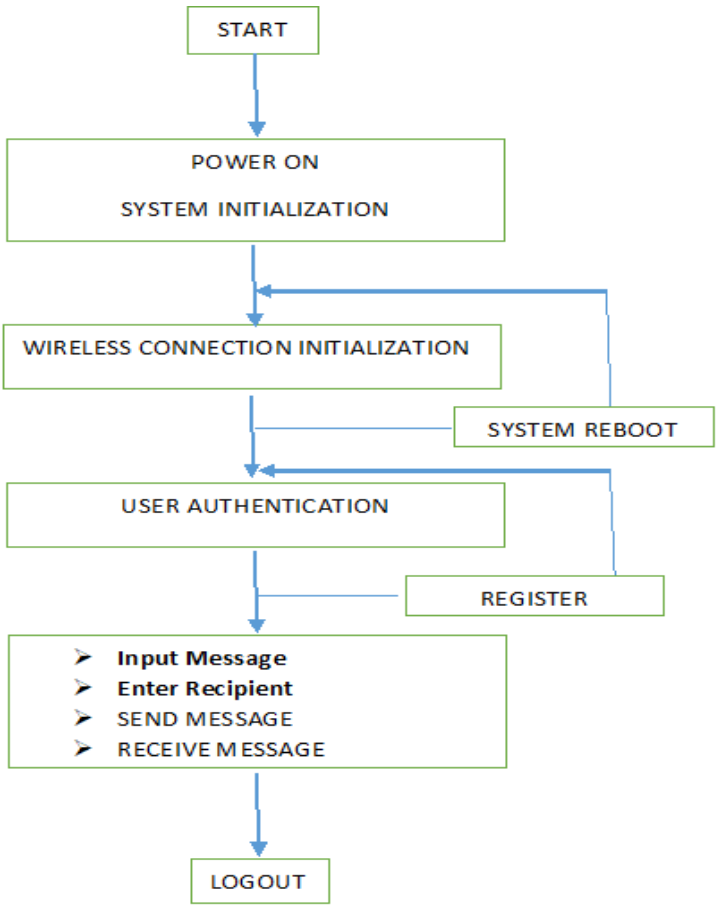


Figure 4. 1: Flow chart for the Wireless Messaging Terminal system

4.1.2 Strength of the existing System

- **Mobility**: Portable and usable in various locations without physical connections.
- **Real-time Communication**: Enables instant message delivery and receipt.
- **Ease of Deployment**: Quick to set up and integrate into existing systems.
- **Cost-Effective**: Reduces the need for physical wiring and infrastructure.
- **Versatility**: Supports multiple wireless protocols for different environments.
- **Enhanced Efficiency**: Improves communication processes and operational efficiency.

4.1.3 Weakness of existing System

- **Signal Reliability**: Performance can be affected by signal strength and interference.
- **Security Concerns**: Susceptible to security threats; requires encryption and secure authentication.
- **Limited Battery Life**: Batteries need frequent recharging or replacement.
- **Network Dependency**: Relies on a stable wireless network connection.
- **Higher Initial Cost**: Upfront cost can be significant, though operational costs may be lower.
- **Maintenance and Upgrades**: Requires regular maintenance and updates.
- **Compatibility Issues**: May face compatibility problems with existing systems or devices

4.2 Data analysis results

Researchers employed various data collection techniques to gather detailed information on Wireless Messaging Terminals (WMTs). These techniques included surveys, interviews, observations, and document reviews. The aim was to identify challenges and performance metrics related to the current WMT systems. The collected data was then analyzed to produce comprehensive reports and visual representations.

4.2.1 The tabular representation of the challenges associated with the current Wireless messaging Terminal system

s/no	**Challenge**	**Frequency**	**Impact**	**Number of respondents out of 5**	**Percentages**
	Delayed Message Delivery	High	Reduced efficiency	3	60
	Signal Interference	Moderate	Communication errors	2	40
	Security Vulnerabilities	High	Risk of data breaches	3	60
	Battery Life Concerns	High	Service interruptions	3	60

High Initial Costs	Moderate	Financial strain	5	100
Maintenance and Compatibility Issues	Moderate	Operational complexity	3	60

The Graphical Representation of the Challenges faced by the current financial management system.

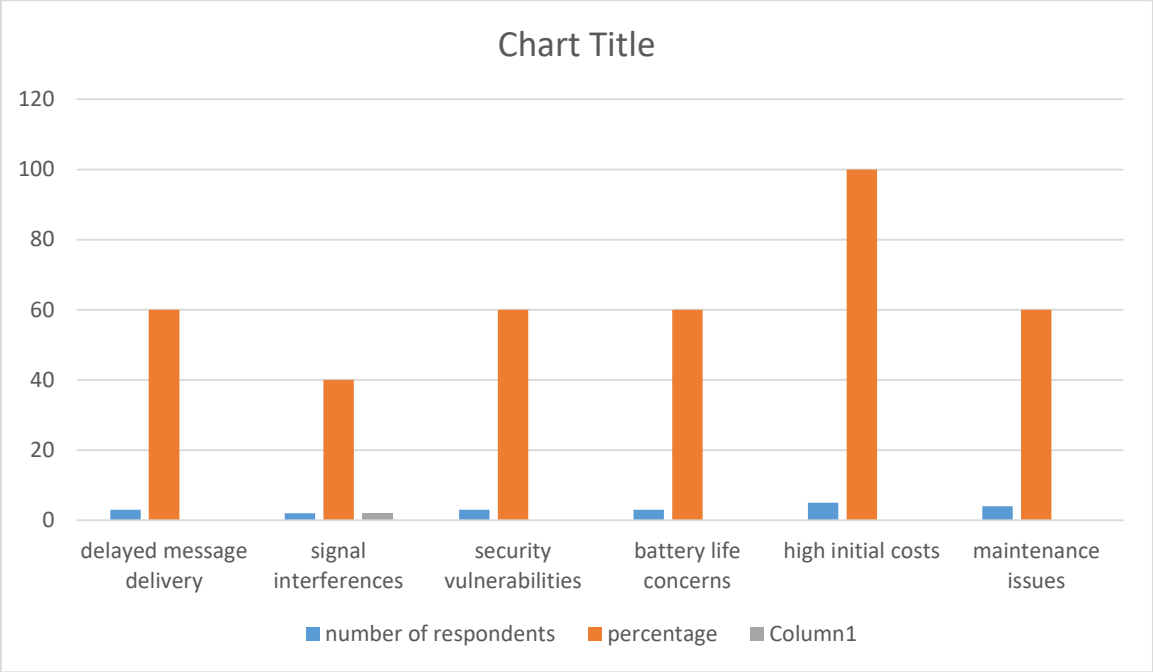


Figure 4. 2: A graphical presentation of the challenges faced by the current Wireless messaging Terminal system.

4.3 User Requirements

- The system should have a user-friendly interface that is easy for all users to navigate.
- The system must enable instant sending and receiving of messages.
- The system should be portable and usable in various locations.
- The system should have a long-lasting battery to minimize the need for frequent recharging.
- The system must provide secure messaging with encryption and authentication features.
- The system should operate consistently in different environmental conditions.
- The system should support various wireless communication protocols like GSM and Wi-Fi.
- Users should have access to customer support for troubleshooting and assistance.
- The system should offer options for data backup and recovery.
- The system must be compatible with existing communication systems and devices.

4.3.1 Functional requirements

According to the tools used to collect data from the users, the following functional requirements were met;

- Users must be authenticated before accessing the secure communication platform.
- The system should support multi-factor authentication methods such as passwords, biometrics, or security tokens.
- The system should manage user roles and permissions based on authentication credentials.
- The system shall encrypt outgoing messages using the Advanced Encryption Algorithm.
- The system shall decrypt incoming messages using the Advanced Encryption Algorithm.
- The system shall provide support for symmetric encryption techniques.
- The system shall seamlessly integrate encryption and decryption processes into the messaging workflow.
- The system shall generate, distribute, and store encryption keys in a secure manner.
- The system shall provide key rotation and expiration policies to enhance security.
- The system shall provide key recovery mechanisms to facilitate data access in case of key loss or compromise.
- The system shall use secure communication protocols (e.g., SMTP) for data transmission over networks.
- The system shall provide protection against man-in-the-middle attacks and eavesdropping.
- The system shall have an intuitive and user-friendly interface for encryption and encrypted messages.
- The system shall provide a clear indication of message encryption status and options for managing encryption settings.
- The system shall provide accessibility features to accommodate users with disabilities.
- The system shall log user activities, including login attempts, message encryption, and decryption operations.
- The system shall track security events, monitor system usage, and detect anomalies.
- The system shall comply with data retention policies and regulatory requirements for audit logs.

4.3.2 System requirements

System requirements encompass the necessary specifications to implement specific functionalities within the system. These requirements detail both the system's overall design and its individual properties. They include the necessary hardware and software components, as outlined below

4.3.2.1 Hardware Requirements

Category	Requirement
Hardware Requirements	
Processor	Dual-core processor with 2.0 GHz or higher
Memory (RAM)	Minimum 4 GB (8 GB or higher recommended)
Storage	Minimum 500 GB HDD or SSD
Network Interface	Ethernet or Wi-Fi capable of secure network connections
Graphics	Basic graphics card to support the user interface
Security Module	Trusted Platform Module (TPM) for hardware-based key storage (optional but recommended)

4.3.2.2 Software Requirements

Software Component	System Requirement	Justification
Operating System	- Windows 10 or higher - Linux (Ubuntu 18.04 or higher, CentOS 7 or higher) - macOS 10.13 or higher	Operating System
Programming Language	HTML - Java - C++ or any language supporting cryptographic libraries -PHP CSS	Programming Language
Cryptography Libraries	- OpenSSL	Cryptography Libraries
Database	- MySQL - PostgreSQL or any other relational database system	Database
Web Server	- Apache - Nginx or equivalent	Web Server
Communication Protocols	- SMTP - HTTPS and other secure communication protocols	Communication Protocols

4.3.2.3 Security Requirements

Security Requirements	
Encryption Standards	- AES (Advanced Encryption Standard) with 128-bit, 192-bit, and 256-bit keys
Key Management	- Secure key generation, distribution, and storage mechanisms
Authentication	- Multi-factor authentication support

4.4 System Design

In the system design phase, process modeling involved use of Data Flow Diagrams (DFD), and Data modeling involved use of Entity Relationship Diagrams (ERD).

4.4.1 Architectural Design for the System

The architectural design shows how the Secure communication encryption Decryption system is comprised of the different subsystems namely Data collection, Data Processing, Data Storage and Data Display. The figure below shows an flowchart diagram of the secure communication encryption Decryption System.

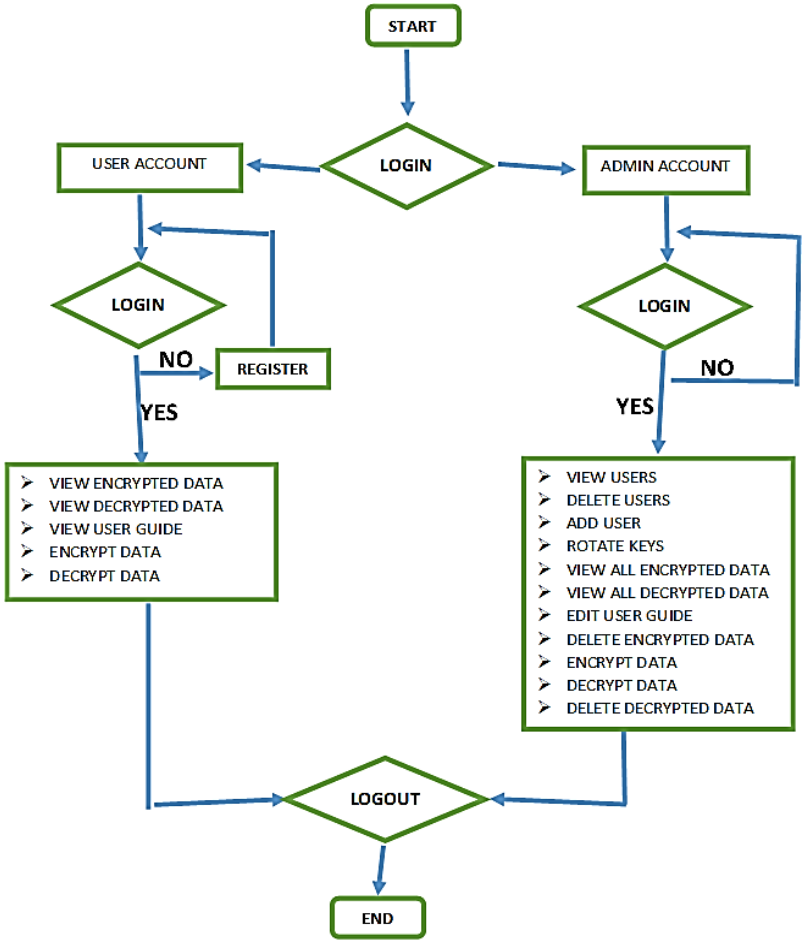
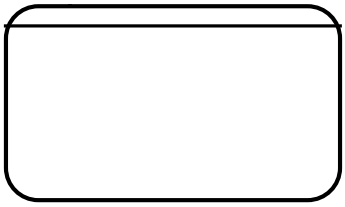


Figure 4. 3: The Architectural Design for AES Encryption Decryption system

4.4.2 Process Modeling

These diagrams illustrate the flow of information within the AES Encryption/Decryption system, showing how data progresses from initial input through various encryption stages, secure storage locations, and decryption processes.

4.4.2.1 Key Symbols

Symbol	Name
	
	Data store
	Data flow
	Process

4.4.3 Data Flow Diagrams (DFD).

As a key tool for system analysts, Data Flow Diagrams (DFDs) are crucial for depicting the movement of data within a system. They utilize various symbols to represent different elements of the system, including processes, data stores, data flows, and external entities. These symbols help to clearly illustrate how data interacts and is managed throughout the system.

4.4.3.1 The Context Level DFD

This high-level diagram shows the AES Encryption Decryption system as a single process and its interactions with external entities such as users and networks. Here, the user interacts with the AES Encryption Decryption system to send and receive encrypted messages, and the network provides connectivity for secure message transmission. It's as shown below:

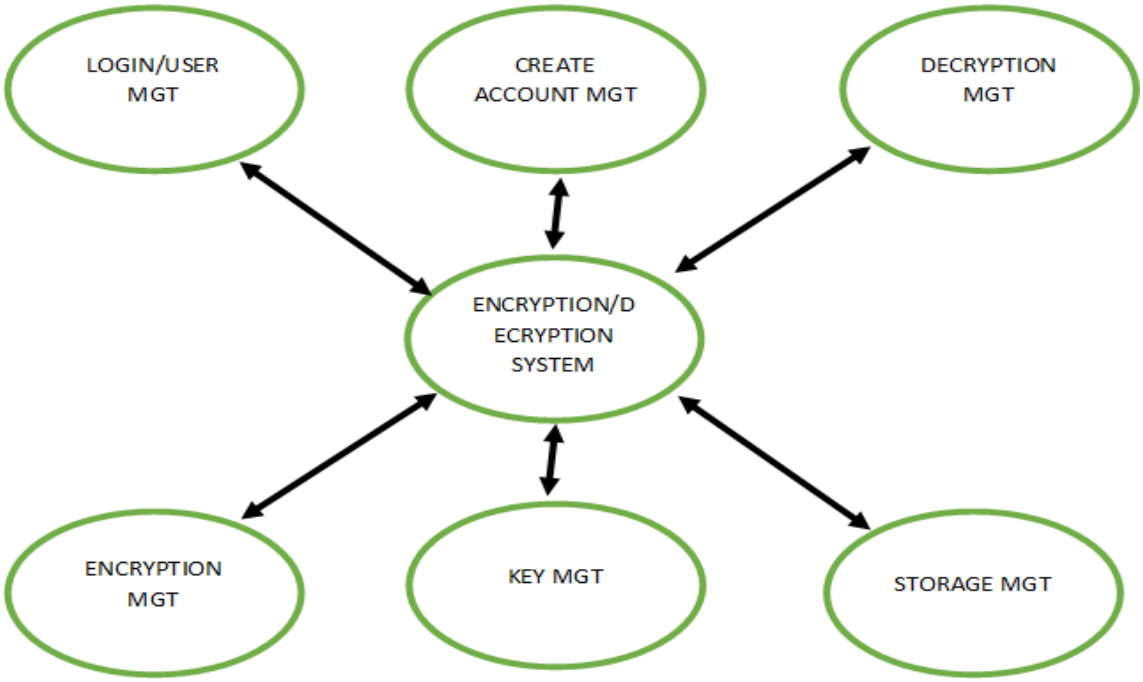


Figure 4.4: **Context Diagram for the AES Encryption Decryption System.**

In **Figure 4.3**, a user accesses the AES Encryption Decryption system, and upon successful authentication, can request and receive encrypted resources. Feedback is promptly delivered to the user. Similarly, the administrator logs into the system, and once authenticated, can perform data queries and obtain immediate responses.

4.4.3.2 The Level 1 DFD for the AES Encryption Decryption System

Figure 4.5: Level 1 DFD for the AES Encryption Decryption System

Description for the Level 1 DFD:

In this subsection, there are tables describing all the design objects used in developing the system. They include Processes, Data flows, Data stores, and External entities.

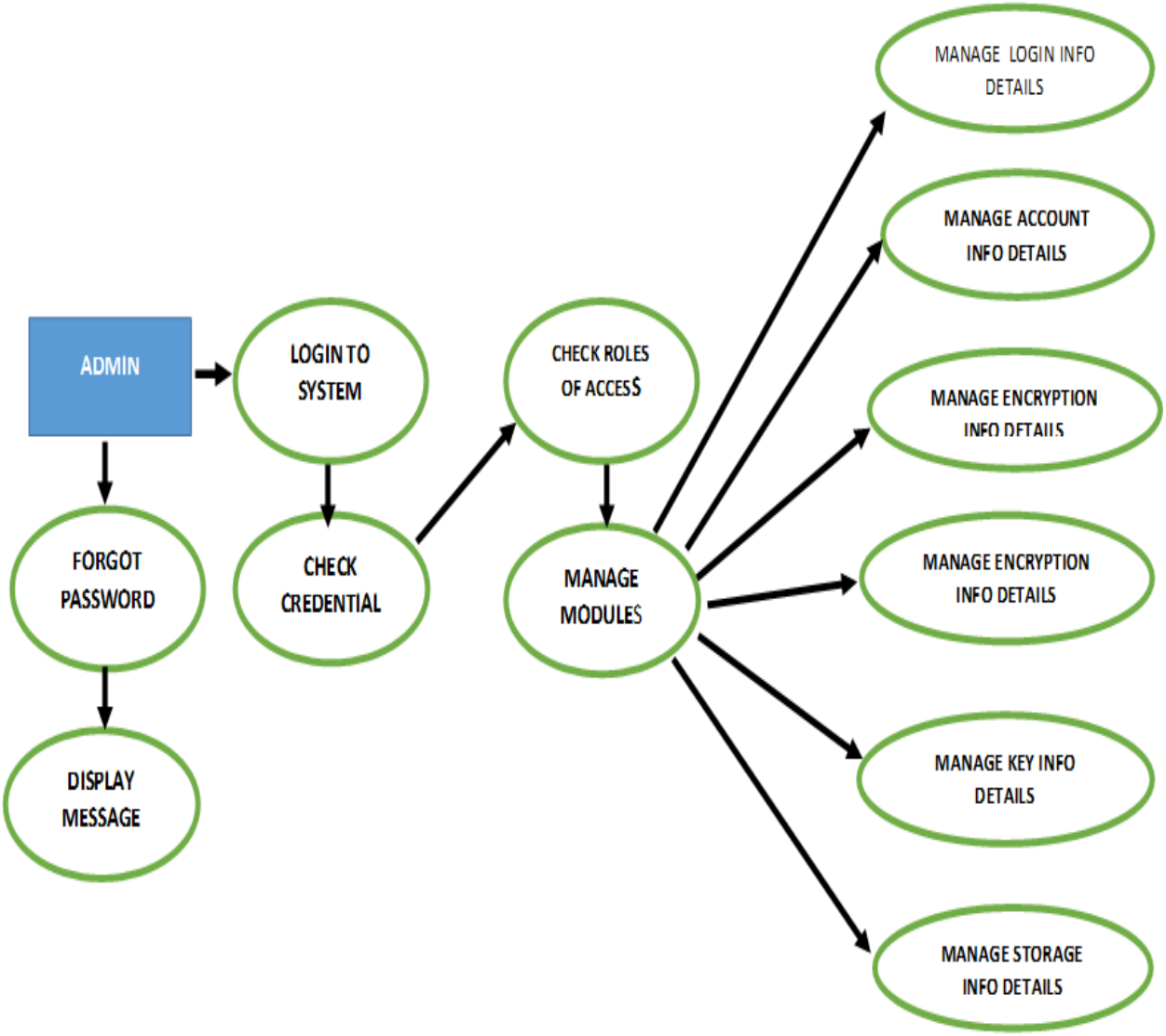
Here is a table for the Description of DFD Level 1 of the AES Encryption Decryption System based on the specified components: User Login Management, Account Management, Key Management, Encryption, Decryption Management, and Storage Management.

Process	Description	Inputs	Outputs	Data Stores
1. User Login Management	Manages the user login process, including authentication and authorization of users.	User credentials (ID, Password)	Authentication result, Access tokens	User Database
2. Account Management	Handles user account creation, updates, and deletion, including role and permission management.	User data (Name, Email, Role)	Account creation confirmation, Updates	User Database
3. Key Management	Manages the generation, distribution, and storage of encryption keys. Includes key rotation and recovery.	Key generation requests, Key rotation requests	Encryption keys, Key recovery instructions	Encryption Keys
4. Encryption Management	Encrypts messages and data using AES before storage or transmission.	Message content, Encryption parameters	Encrypted messages, Confirmation of encryption	Message Database
5. Decryption Management	Decrypts incoming encrypted messages for user access and display.	Encrypted message, Decryption parameters	Decrypted message, Decryption confirmation	Message Database
6. Storage Management	Manages the storage and retrieval of encrypted and decrypted messages, and system logs.	Data storage requests, Retrieval requests	Stored messages, Logs	Message Database, Error Logs

4.4.3.3 The Level 2 DFD

The DFD2 provides even more detail, breaking down the processes and data flows identified in DFD1 into sub-processes and data transformations.

ILLUSTRATION OF DFD2



These three levels of Data Flow Diagrams provide a comprehensive understanding of the encryption and decryption application, detailing the processes, data flows, and interactions between various components of the system. This structured approach helps in visualizing the system's functionality and identifying potential areas for improvement or optimization.

Here's the updated table for the **AES Encryption Decryption System**, with the Message entity replaced by Encrypted Text and Decrypted Text:

4.4.3.4 Identification of Entities and Their Attributes for the AES Encryption Decryption System

Entity	Attribute	Description
User	User ID	Unique identifier for each user.
	Username	The name chosen by the user for login.
	Password	Securely stored password for authentication.
	Email	Contact email address of the user.
	Phone Number	Contact phone number of the user.
	Role	The user's role (e.g., regular user, administrator).
	Last Login Time	Timestamp of the user's last login.
Encryption Key	Key ID	Unique identifier for each encryption key.
	User ID	Reference to the user associated with the encryption key.
	Algorithm	Encryption algorithm used (e.g., AES).
	Public Key	Public encryption key used for asymmetric encryption (if applicable).
	Private Key	Private encryption key used for asymmetric decryption (if applicable).
	Secret Key	Secret key used for symmetric encryption.
	Created At	Timestamp of when the key was generated.
Encrypted Text	Encrypted Text ID	Unique identifier for each encrypted text record.
	Sender ID	ID of the user who sent the encrypted text.
	Receiver ID	ID of the user who will receive the encrypted text.
	Encrypted Content	The encrypted text or data.
	Timestamp	Date and time when the encrypted text was created.
	Status	Status of the encrypted text (e.g., sent, failed).
Decrypted Text	Decrypted Text ID	Unique identifier for each decrypted text record.
	Encrypted Text ID	Reference to the associated encrypted text record.
	Decrypted	The decrypted text or data.

Entity	Attribute	Description
	Content	
	Timestamp	Date and time when the decrypted text was obtained.
Error Log	Error ID	Unique identifier for each error log entry.
	Error Type	Type or category of the error (e.g., network, authentication).
	Error Description	Description of the error encountered.
	Timestamp	Date and time when the error occurred.
	Resolution Status	Status of the error resolution (e.g., resolved, unresolved).
System Log	Log ID	Unique identifier for each system log entry.
	Log Type	Type of log (e.g., operation, security).
	Log Message	Detailed message about the system operation or event.
	Timestamp	Date and time when the log entry was created.
Network	Network ID	Unique identifier for each network configuration.
	Interface	Network interface used (e.g., Ethernet, Wi-Fi).
	IP Address	IP address of the network configuration.
	Subnet Mask	Subnet mask used for network segmentation.
	Gateway	Gateway IP address for routing.
	Status	Current status of the network (e.g., active, inactive).
Recipient	Recipient ID	Unique identifier for each recipient.
	Recipient Name	Name of the recipient.
	IP Address	IP address associated with the recipient.
	Created At	Timestamp of when the recipient was added.

4.4.3.5 Modeling Relationships between Entities

Entity Relationships:

User and Encryption Key

Relationship: One-to-Many

Each user can have multiple encryption keys (e.g., for different encryption algorithms or key rotations). The User entity has a one-to-many relationship with the Encryption Key entity.

User and Encrypted Text

Relationship: One-to-Many

Each user can send multiple encrypted texts. The User entity has a one-to-many relationship with the Encrypted Text entity.

User and Decrypted Text

Relationship: One-to-Many

Each user can receive and decrypt multiple encrypted texts. The User entity has a one-to-many relationship with the Decrypted Text entity.

Encrypted Text and Decryption Key

Relationship: Many-to-One

Each encrypted text may be decrypted using a specific key. The Encrypted Text entity has a many-to-one relationship with the Encryption Key entity, implying that multiple encrypted texts can be decrypted with the same key.

Decrypted Text and Encrypted Text

Relationship: One-to-One

Each decrypted text corresponds to a specific encrypted text. The Decrypted Text entity has a one-to-one relationship with the Encrypted Text entity, meaning each decrypted text is linked to a single encrypted text.

Error Log and System Log

Relationship: One-to-Many

Each system log entry may generate multiple error logs. The System Log entity has a one-to-many relationship with the Error Log entity.

Network and Recipient

Relationship: One-to-Many

Each network configuration can be associated with multiple recipients for communication purposes. The Network entity has a one-to-many relationship with the Recipient entity.

Diagram Representation:

Below is a textual representation of the relationships:

User

⟶ **Encryption Key** (1-to-Many)

⟶ **Encrypted Text** (1-to-Many, as Sender)

⟶ **Decrypted Text** (1-to-Many, as Receiver)

Encryption Key

⟶ **Encrypted Text** (Many-to-One)

Encrypted Text

⟶ **Decrypted Text** (1-to-1)

System Log

⟶ **Error Log** (1-to-Many)

Network

→ **Recipient** (1-to-Many)

Entity-Relationship Diagram (ERD)

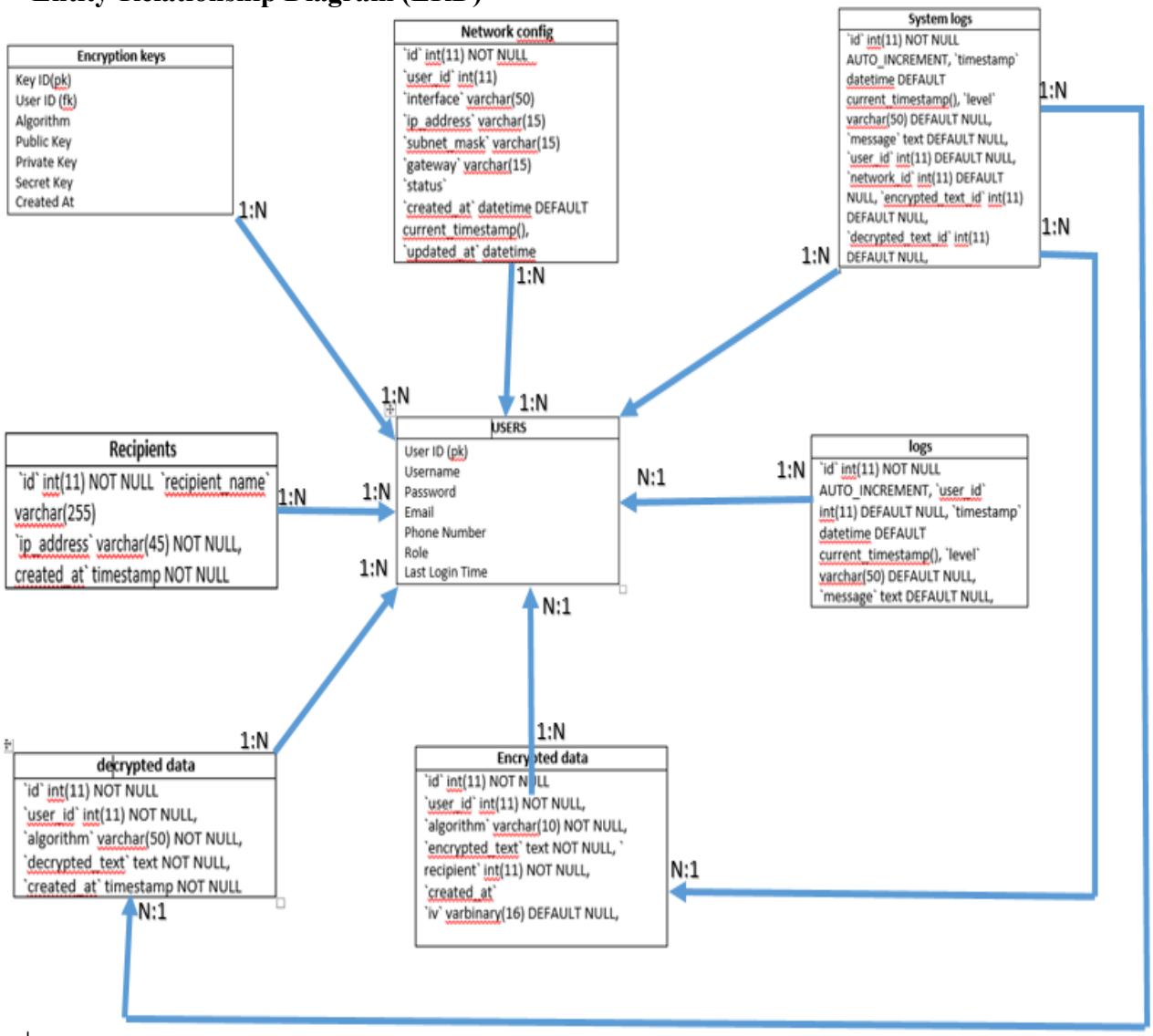


Figure 4. 4: The Entity Relationship Diagram

4.5 Conclusion

In summary, this chapter was mainly based on the study of the existing system, analysis of the requirements for the system, processes and data modeling.

CHAPTER FIVE

5.0 System Implementation, Testing, and Validation

This section outlines how the design models of the AES Encryption Decryption System are implemented and demonstrates the various outcomes produced by the system. It includes screenshots to illustrate how the AES Encryption Decryption System presents results in response to user commands.

5.1 System Functions

The AES Encryption Decryption System facilitates various functions for different types of users, including administrators, and users. Each role has specific functions and access rights within the system.

5.1.1 Functions Provided to All Users

The AES Encryption Decryption System ensures secure access for all users by requiring authentication through usernames and passwords. This security measure helps safeguard the system and data.

- **Authentication:** Users must log in using their credentials to access system features.
- **Message Management:** Users can encrypt and decrypt messages.
- **User Interface Access:** Provides a clear and responsive interface for interacting with encryption and decryption functions.

5.1.2 Functions Provided to the Users

Authenticated users can perform the following functions within the AES Encryption Decryption System:

- **Encrypt Messages:** Compose and encrypt messages for secure transmission.
- **Decrypt Messages:** Decrypt received messages for reading.
- **View Message History:** Access and review past messages sent and received.

5.1.3 Functions Provided to the Administrator

The system administrator has comprehensive access to manage and oversee all system operations in addition to that of the users

- **System Monitoring:** View and analyze all messages and system performance metrics.
- **User Management:** Add, update, or remove user accounts as needed.
- **Error Handling:** Review and address system errors or issues reported by the system.
- **System Configuration:** Adjust system settings and configurations to ensure optimal performance.

5.2 System Map

Component	Description	Interactions
User Interface	Provides screens and views for user interaction.	- Connects to Authentication Module for login. - Connects to Encryption/Decryption Module for message processing. - Connects to Error Handling Module for error notifications.
Wireless Connection Module	Manages network connectivity (e.g., LAN, Wi-Fi).	- Provides connectivity to User Interface, Encryption/Decryption Module, and Authentication Module.
Encryption/Decryption Module	Handles encrypting, decrypting, and storing messages.	- Interacts with Wireless Connection Module for message transmission. - Provides data to User Interface and System Monitoring Module.
Authentication Module	Manages user login and authentication.	- Validates credentials from User Interface. - Provides access to Encryption/Decryption Module based on authentication status.
System Monitoring Module	Monitors system performance and status.	- Interacts with Encryption/Decryption Module and Error Handling Module to provide performance data and error logs.
Error Handling Module	Detects and reports system errors.	- Provides error information to User Interface. - Interacts with System Monitoring Module to log and handle errors.
Logs Module	Records system activities, errors, and user actions.	- Interacts with Authentication Module, Encryption/Decryption Module, and Error Handling Module to log relevant events and activities.
Database	Stores user data, encrypted/decrypted messages, and system configurations.	- Provides data to Encryption/Decryption Module, User Interface, and System Monitoring Module. - Updates records based on user actions and system processes.

Figure 5.2: tabular representation of System Map

5.3 Sample Screen-shots

5.3.1 System home page

Figure 5.2 illustrates the homepage, where all authorized users can log into the AES Encryption Decryption System to access their respective areas and complete their tasks. When an administrator selects the login option, they are directed to a specific login page designed for administrative access, as depicted in the screenshot below.

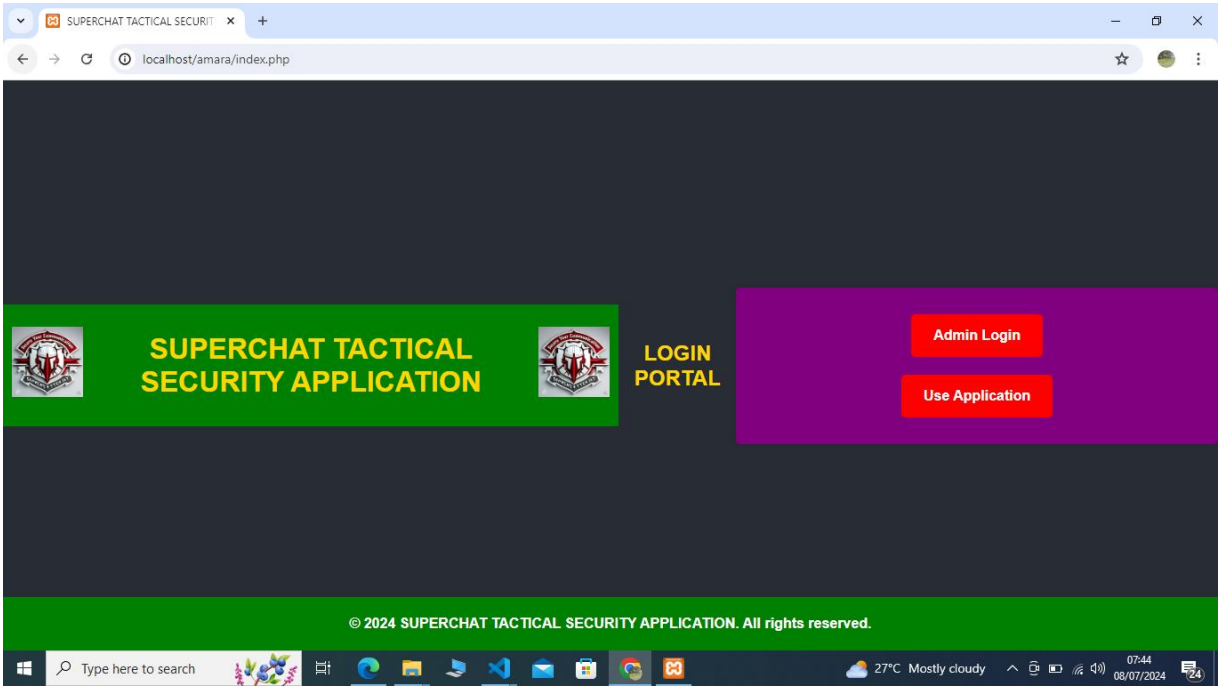


Figure 5. 1: System home page

5.3.2 Administrator's login page

Figure 5.3 depicts the login page for administrators. Here, the administrator selects the "Admin" option and enters their password to gain access to the AES Encryption Decryption System. Upon successful login, the administrator can view and manage various features such as account types, user accounts, password settings, and mini-statements. If the administrator inputs incorrect credentials, access to the system will be denied.

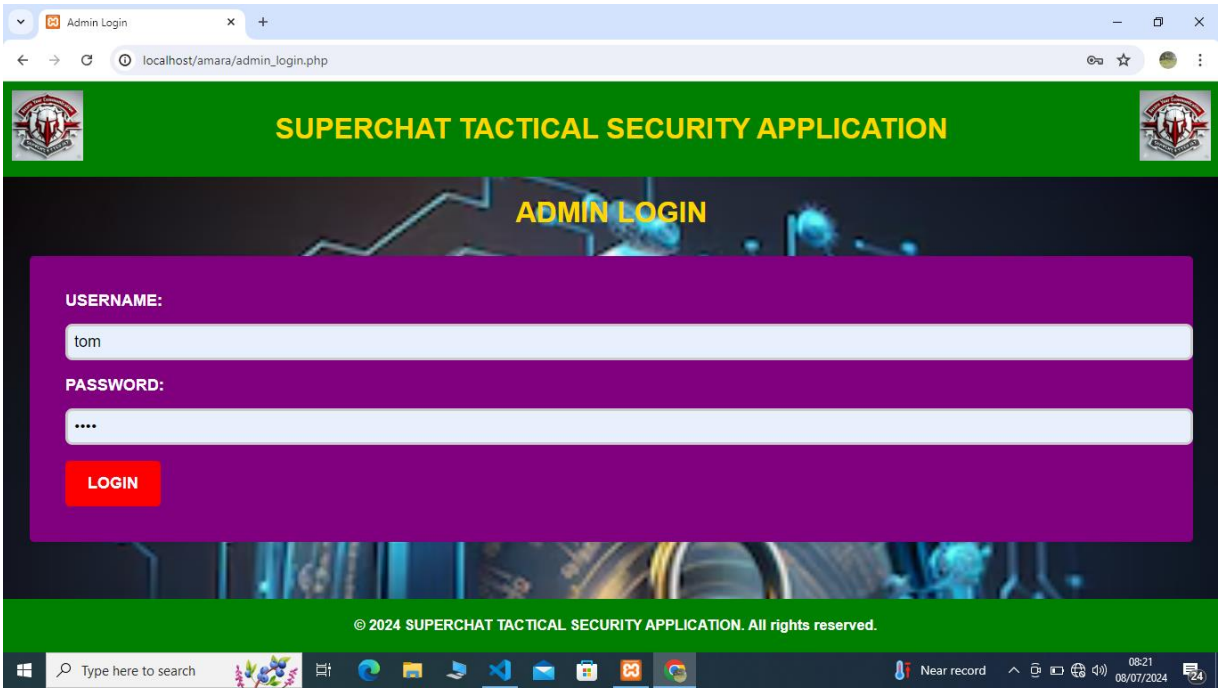


Figure 5. 2: Administrator login page

5.3.3 Administrative view page

Figure 5.4 illustrates the administrator's dashboard following a successful login into the AES encryption and decryption system. The dashboard features several navigation links, highlighted in red, that enable the administrator to manage encryption keys, oversee encrypted and decrypted data, configure encryption algorithms, and handle user accounts and their access permissions.

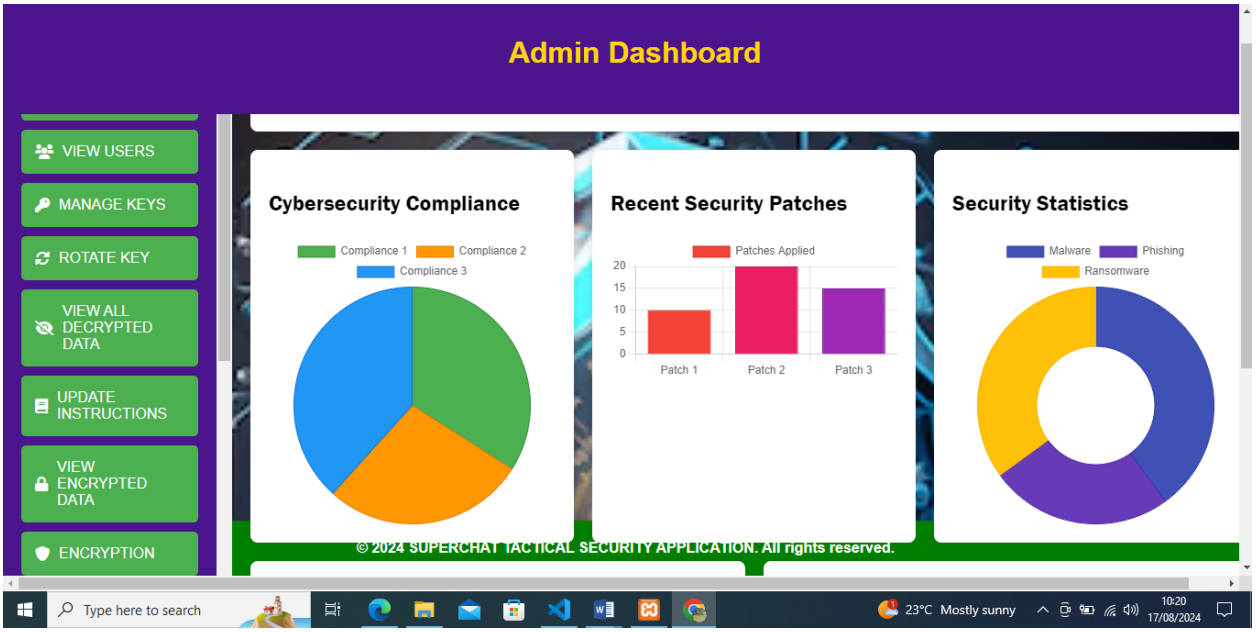


Figure 5. 3: Administrative view page

5.3.4 user's login page

Figure 5.9 illustrates the user login page for the AES encryption and decryption system. Users select the user application option and enter their password to log in. Upon successful authentication, users can access their encrypted messages, perform decryption, and manage their encrypted data.

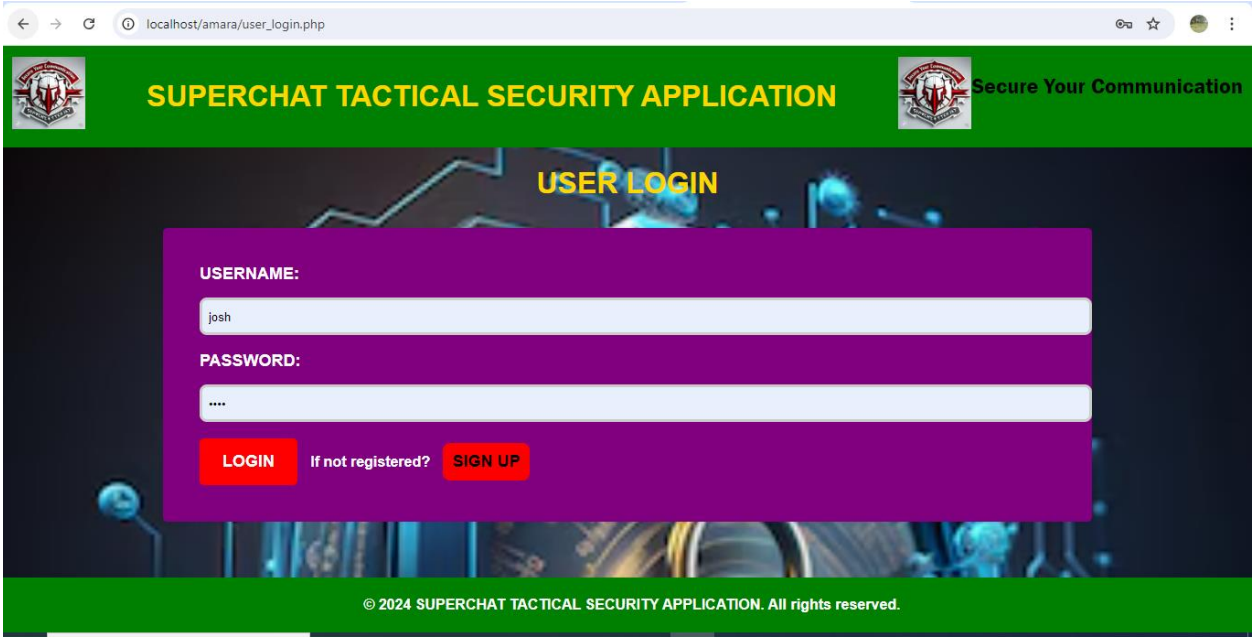


Figure 5. 4: User login page

5.3.5 Logged in user's account

Figure 5.10: Displays a user logged into their AES encryption and decryption system account. From this interface, the user can view their account details, encrypt and decrypt messages, manage encryption keys, and access their encrypted data and decryption history.

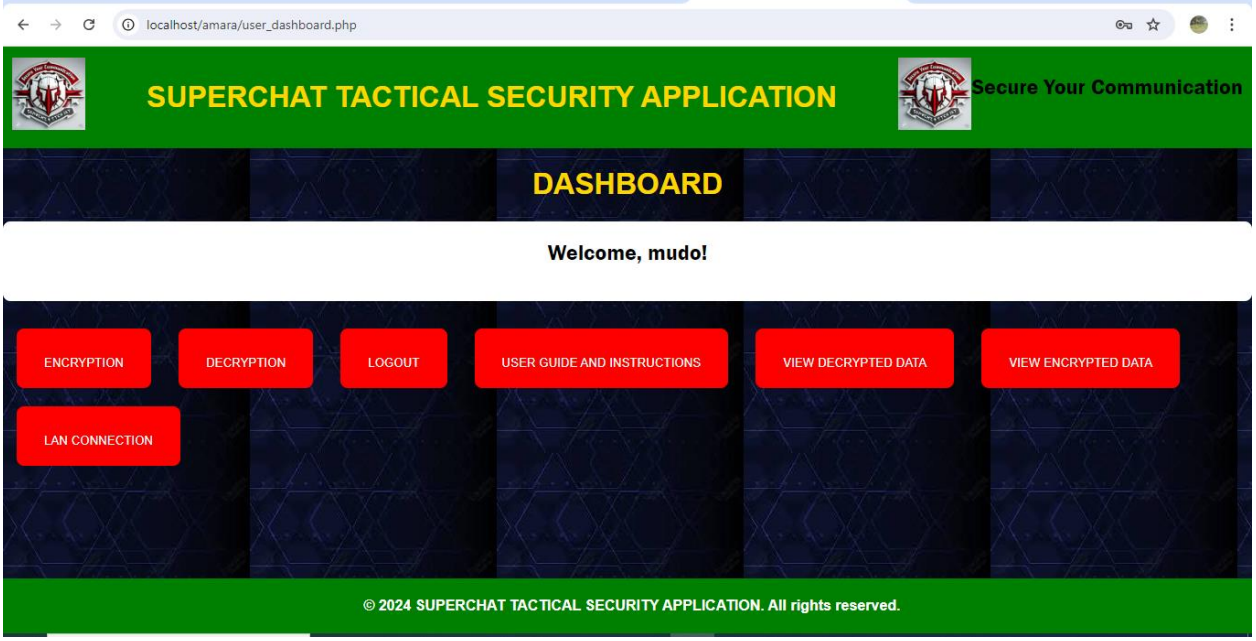


Figure 5. 5: Logged in customer account

5.3.6 encryption pages

Figure 5.11: Displays the Encryption page where a user can securely encrypt their data before sending it to another user. This page allows the user to select the encryption algorithm, input the data, and generate the encrypted text for secure transmission.

Figure 5. 6: encryption page

5.3.7 Decryption page

Figure 5.12: Displays the User's Decryption page, where a user can view the decrypted details of their encrypted data

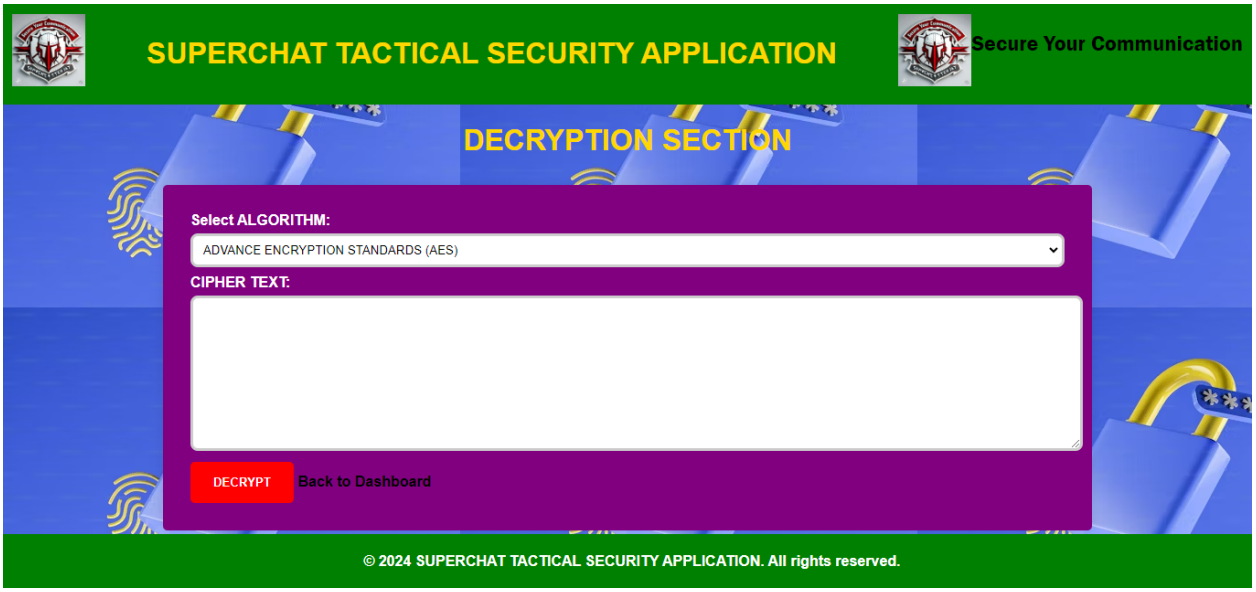


Figure 5. 7: Customer Mini-Statement Page

5.4 System Testing and Validation Results

In this section, we discuss the results from testing and validating the AES Encryption Decryption System. The goal was to identify errors, verify the system's adherence to user requirements, and gather user feedback to ensure the system meets their needs.

5.4.1 System Testing Results

The AES Encryption Decryption System was tested to identify and correct errors, ensuring it met the specified performance and functional requirements. The testing process involved:

- **Error Detection:** The system was tested with various inputs to identify faults. These issues were addressed, and the testing cycle was repeated until the system functioned as expected.
- **Data Validation:** The system was evaluated for its ability to handle valid and invalid data. Incorrect inputs triggered appropriate error messages, confirming that the system correctly identifies and reports errors. The system passed all tests, demonstrating that it operates according to user specifications and performance expectations.

5.4.2 Validation Results

System validation involved presenting the AES Encryption Decryption System to users to gather feedback on its performance and to confirm that it met their needs. The validation process included:

- **User Feedback:** Users tested the system and provided input on whether it met their requirements. This involved checking the system's input and output data to ensure completeness and accuracy.
- **Database Accuracy:** The system's database was scrutinized to ensure it conformed to standards and operated correctly under defined conditions.
- **User Satisfaction:** Feedback indicated that the system was user-friendly, fast, and effective in meeting user needs.
- A questionnaire was distributed to capture detailed user responses, with the results summarized in Table 15.

Table 9: System Validation

Feature	Number of Users Out of 5	Percentage of Users
Functionality	4	80%
Performance	3	60%
Security	5	100%
Usability	2	40%
Compatibility	4	80%
Documentation	2	40%
Compliance	3	60%

The results showed that most users found the system to be easy to learn, user-friendly, and effective in improving transaction processes and reducing delays. Overall, the system met the

users' needs and requirements, and the feedback was overwhelmingly positive.

5.5 Conclusion

In summary, this chapter detailed the functionalities offered by the AES Encryption Decryption System to all user roles, including users and administrators, and included various screenshots of the system. It also covered the testing and validation procedures, where the AES Encryption Decryption System was scrutinized for errors and evaluated against the specified user requirements. The results of these assessments were compiled and reviewed.

.

CHAPTER SIX

6.1 Summary

The AES Encryption/Decryption system has successfully met its objectives of enhancing data security through advanced encryption methods. Designed to replace traditional, less secure data handling techniques, the system now allows users to encrypt and decrypt sensitive information using the AES algorithm, ensuring data confidentiality and integrity. The system supports secure data transmission between users, offering a robust mechanism to protect sensitive information from unauthorized access. Administrators have full control to oversee the system's functionality, manage encryption keys, monitor user activities, and ensure the secure operation of the system.

6.2 Recommendations

To further enhance the AES Encryption/Decryption system, it is recommended to explore additional security features and improvements.

A. Research into emerging encryption technologies could help address any current limitations and integrate new capabilities, such as quantum-resistant algorithms.
B. Additionally, expanding the system's compatibility with a broader range of applications and platforms would improve its usability and accessibility. Developing comprehensive training materials and support resources will also be beneficial in ensuring users can fully utilize the system's security features.

6.3 Future Work

Future enhancements for the AES Encryption/Decryption system should focus on:

i. **Enhanced Encryption Features:** Introducing advanced encryption options, such as hybrid encryption techniques and real-time encryption for data in motion.

ii. **Integration with Other Systems:** Ensuring compatibility with other data management and communication systems for seamless and secure data exchange.

iii. **User Experience Improvements:** Refining the user interface and overall user experience to make the system more intuitive, especially for non-technical users.

6.4 Conclusions

The AES Encryption/Decryption system has effectively achieved its goals, providing a modern and secure solution for data encryption and decryption. By moving from traditional methods to an advanced encryption platform, it has significantly improved data security for all users. The system's features, security measures, and administrative controls represent a substantial advancement in how data protection is managed and implemented across various platforms and applications.

7.0 References

Abdelhalim, M. B., El-Mahallawy, M., Ayyad, M. and Elhennawy, A. (2012). Design & Implementation of an Encryption Algorithm for use in RFID System. *International Journal of RFID Security and Cryptography (IJRFIDSC),* Vol. 1, Issues 1-4, pp. 51 – 57.

Adesanya, O. (2002). The impact of information technology on information dissemination. In Madu, E.C. and Dirisu, M.B. (Eds.), *Information science and technology for library schools in Africa* (pp.10-24). Ibadan, Nigeria: Evi-Coleman.

Introna, L. D. (1992). *Towards a Theory of Management Information.* Unpublished DCom Dissertation, University of Pretoria.

Madji, A. and Lin, Y. H. (2007). Simple Encryption/Decryption Application. *Internation Journal of Computer Science and Security,* Vol. 1, Issue (1), pp. 33 – 40.

Meyer, H. W. J. (2000). *The transfer of agricultural information to rural communities.* Unpublished doctoral dissertation, University of Pretoria, Pretoria, S. Africa.

Nwosu, I. (2004). Digital public relations: concept and practice, In Nwokocha, J. (Ed.). *Digital public relations: New techniques in reputation management* (pp. 33-34). Lagos, Nigeria: Zoom Lens Publishers.

Ogbomo, M. O. and Ogbomo, E. F. (2008). Importance of Information and Communication Technologies (ICTs) in Making a Heathy Information Society: A Case Study of Ethiope East Local Government Area of Delta State, Nigeria. *Library Philosophy and Practice 2008,* ISSN 1522-0222, pp 1 – 8.

Stallings, W. (2011). *Cryptography and Network Security: Principles and Practice* (5th ed.). NY, US: Prentice Hall.

Woherem, E.R. (2000). *Information technology in the Nigerian banking industry*. Ibadan, Nigeria: Spectrum Books.

8.0 Appendices

Appendix I: Interview schedule sample questions

Can you describe the current authentication process and any challenges faced?

- ☐ Well-defined and Effective
- ☐ Adequate but Needs Improvement
- ☐ Ineffective and Needs Overhaul
- ☐ Not Implemented
- ☐ Other (please specify): _____

What encryption algorithms are currently used, and what are the strengths and weaknesses of these methods?

- ☐ AES (Advanced Encryption Standard) – Strong Security, High Performance
- ☐ RSA (Rivest-Shamir-Adleman) – Secure but Slower Performance
- ☐ ECC (Elliptic Curve Cryptography) – Efficient and Strong Security
- ☐ DES (Data Encryption Standard) – Outdated, Weak Security
- ☐ Other (please specify): _____

How are user roles managed, and what improvements would you suggest for this process?

- ☐ Role-Based Access Control (RBAC) – Effective
- ☐ Attribute-Based Access Control (ABAC) – Flexible but Complex
- ☐ Policy-Based Access Control (PBAC) – Effective but Requires Maintenance
- ☐ Manual Role Management – Prone to Errors
- ☐ Other (please specify): _____

Can you provide examples of how message encryption impacts your daily communication workflows?

- ☐ Enhanced Security and Confidentiality
- ☐ Additional Layer of Security with Minimal Impact
- ☐ Significant Impact on Workflow Efficiency
- ☐ Minimal Impact on Workflow
- ☐ Other (please specify): _____

What key management practices do you currently follow, and what changes would you recommend?

- ☐ Centralized Key Management – Effective

- [ ] Decentralized Key Management – Flexible but Complex
- [ ] Automated Key Rotation – Secure and Efficient
- [ ] Manual Key Management – Risky
- [ ] Other (please specify): _____

How does the system ensure secure data transmission, and what vulnerabilities have you encountered?

- [ ] Robust Encryption Protocols – Few Vulnerabilities
- [ ] Secure Protocols with Occasional Issues
- [ ] Basic Protocols – Several Vulnerabilities
- [ ] Outdated Protocols – Numerous Vulnerabilities
- [ ] Other (please specify): _____

What features are most important to you in a user interface for managing encryption?

- [ ] Ease of Use
- [ ] Performance
- [ ] Customization Options
- [ ] Comprehensive Reporting
- [ ] Other (please specify): _____

Can you explain your current logging and auditing processes? How effective are they?

- [ ] Comprehensive and Effective
- [ ] Adequate but Could be Improved
- [ ] Basic and Ineffective
- [ ] Minimal Logging and Auditing
- [ ] Other (please specify): _____

What performance issues have you experienced with encryption operations, if any?

- [ ] None
- [ ] Minor Latency
- [ ] Moderate Latency
- [ ] Significant Latency
- [ ] Not Sure

How do you handle system availability and what measures are in place to prevent

downtime?

- ☐ Redundant Systems and Failover Mechanisms
- ☐ Regular Backups and Maintenance
- ☐ Load Balancing and Scalability Measures
- ☐ Minimal Downtime Strategies
- ☐ Other (please specify): _____

What security standards do you adhere to protect sensitive information?

- ☐ ISO/IEC 27001
- ☐ NIST Cybersecurity Framework
- ☐ GDPR Compliance
- ☐ HIPAA Compliance
- ☐ Other (please specify): _____

How important is interoperability with other systems in your organization, and can you share any specific experiences?

- ☐ Crucial – Seamless Integration Required
- ☐ Important – Minor Integration Issues
- ☐ Somewhat Important – Several Integration Challenges
- ☐ Not Very Important – Limited Integration
- ☐ Other (please specify): _____

What scalability challenges do you foresee as your user base and message volume increase?

- ☐ Performance Degradation
- ☐ Resource Constraints
- ☐ Increased Complexity
- ☐ Integration Challenges
- ☐ Other (please specify): _____

Feel free to adjust the options or add more details as needed for your specific use case!

Appendix II: **Questionnaires**

User Authentication

Authentication Methods:

What authentication methods are currently implemented for user access?

- ☐ Passwords
- ☐ Biometrics (e.g., fingerprints, facial recognition)
- ☐ Security Tokens (e.g., OTPs, hardware tokens)
- ☐ Multi-Factor Authentication (MFA)
- ☐ Other (please specify): _____

How effective do you find these methods in ensuring secure access?

- ☐ Very Effective
- ☐ Effective
- ☐ Neutral
- ☐ Ineffective
- ☐ Very Ineffective

User Roles:

How are user roles and permissions currently managed within the system?

- ☐ Role-Based Access Control (RBAC)
- ☐ Attribute-Based Access Control (ABAC)
- ☐ Policy-Based Access Control (PBAC)
- ☐ Ad-hoc Role Management
- ☐ Other (please specify): _____

What challenges do you face in role management?

- ☐ Defining Clear Roles
- ☐ Effective Permission Assignment
- ☐ Lack of Management Tools
- ☐ Scalability Issues
- ☐ Other (please specify): _____

Message Encryption and Decryption

Current Encryption Techniques:

What encryption methods are used for outgoing and incoming messages?

- [ ] AES (Advanced Encryption Standard)
- [ ] RSA (Rivest-Shamir-Adleman)
- [ ] ECC (Elliptic Curve Cryptography)
- [ ] TLS/SSL (Transport Layer Security/Secure Sockets Layer)
- [ ] Other (please specify): _____

How satisfied are you with the current encryption solutions?

- [ ] Very Satisfied
- [ ] Satisfied
- [ ] Neutral
- [ ] Dissatisfied
- [ ] Very Dissatisfied

Integration:

How seamlessly are encryption and decryption processes integrated into your messaging workflows?

- [ ] Fully Integrated
- [ ] Mostly Integrated
- [ ] Partially Integrated
- [ ] Poorly Integrated
- [ ] Not Integrated

Are there any gaps in this integration that affect usability?

- [ ] Usability Issues
- [ ] Performance Issues
- [ ] Security Issues
- [ ] Compatibility Issues
- [ ] Other (please specify): _____

Key Management

Key Management Practices:

How are encryption keys generated, distributed, and stored in your current system?

- [ ] Centralized Key Management
- [ ] Decentralized Key Management

- [ ] Automated Key Rotation
- [ ] Manual Key Management
- [ ] Other (please specify): _____

What challenges do you encounter with key management?

- [ ] Key Generation
- [ ] Key Distribution
- [ ] Key Storage
- [ ] Key Rotation
- [ ] Other (please specify): _____

Key Recovery:

What mechanisms are in place for key recovery in case of key loss or compromise?

- [ ] Backup and Restore
- [ ] Key Recovery Service
- [ ] Manual Recovery Procedures
- [ ] No Mechanism
- [ ] Other (please specify): _____

Secure Transmission

Transmission Protocols:

What secure communication protocols are used for data transmission?

- [ ] HTTPS (Hypertext Transfer Protocol Secure)
- [ ] SFTP (Secure File Transfer Protocol)
- [ ] VPN (Virtual Private Network)
- [ ] SSL/TLS
- [ ] Other (please specify): _____

Have there been any instances of security breaches related to these protocols?

- [ ] None
- [ ] Rare
- [ ] Occasionally
- [ ] Frequently
- [ ] Not Sure

User Interface

Usability Assessment:

How user-friendly do you find the current interface for managing encrypted messages?

- ☐ Very User-Friendly
- ☐ User-Friendly
- ☐ Neutral
- ☐ Not User-Friendly
- ☐ Very Not User-Friendly

What improvements would enhance usability for users?

- ☐ Simplify Navigation
- ☐ Improve Visual Design
- ☐ Enhance Performance
- ☐ Add User Training Materials
- ☐ Other (please specify): _____

Logging and Auditing

Activity Logging:

What user activities are logged in the current system?

- ☐ All User Activities
- ☐ Key Activities Only
- ☐ Critical Events Only
- ☐ Customizable Logging
- ☐ None

How effectively does the system monitor security events and system usage?

- ☐ Highly Effective
- ☐ Moderately Effective
- ☐ Slightly Effective
- ☐ Not Effective
- ☐ Unsure

Non-functional Requirements

Performance Metrics:

How does the current system handle message encryption and decryption operations?

- [ ] Excellent
- [ ] Good
- [ ] Average
- [ ] Poor
- [ ] Very Poor

Are there any latency issues during peak usage?

- [ ] None
- [ ] Minor
- [ ] Moderate
- [ ] Severe
- [ ] Not Sure

System Reliability:

How often does the system experience downtime or service disruptions?

- [ ] Very Reliable
- [ ] Reliable
- [ ] Occasionally Unreliable
- [ ] Frequently Unreliable
- [ ] Not Reliable

What redundancy measures are in place to ensure system availability?

- [ ] Failover Systems
- [ ] Load Balancers
- [ ] Backup Systems
- [ ] No Redundancy Measures
- [ ] Other (please specify): _____

Security Practices:

What security protocols are implemented to protect against unauthorized access?

- [ ] Encryption
- [ ] Access Controls
- [ ] Regular Audits
- [ ] Intrusion Detection Systems

- [ ] Other (please specify): _____

How frequently do you conduct security assessments?

- [ ] Daily
- [ ] Weekly
- [ ] Monthly
- [ ] Quarterly
- [ ] Annually

Compatibility Issues:

How compatible is the current system with existing communication systems and protocols?

- [ ] Fully Compatible
- [ ] Mostly Compatible
- [ ] Partially Compatible
- [ ] Poorly Compatible
- [ ] Not Compatible

Have you faced challenges with integrating third-party applications?

- [ ] None
- [ ] Minor Issues
- [ ] Moderate Issues
- [ ] Major Issues
- [ ] Not Sure

Scalability Needs:

How well does the current system scale with increasing user demands?

- [ ] Excellent Scalability
- [ ] Good Scalability
- [ ] Average Scalability
- [ ] Poor Scalability
- [ ] Very Poor Scalability

What challenges do you anticipate as user volumes grow?

- [ ] None
- [ ] Performance Degradation

- [ ] Increased Complexity
- [ ] Resource Constraints
- [ ] Other (please specify): _____