Original Article
ISSN (Online): 2350-0530
ISSN (Print): 2394-3629

International Journal of Research - GRANTHAALAYAH
December 2023 11(12), 254–267

# A SECURE FILE ENCRYPTION AND DECRYPTION SYSTEM USING AES FOR TEXT AND IMAGES

Pranav Banga [1], Nirottam [1], Shubham Kumar Singh [1], Shubham [1], Priyanka Singh [1]

[1] Department of Computer Science & Engineering, Echelon Institute of Technology, Faridabad, India

**Received** 23 November 2023
**Accepted** 20 December 2023
**Published** 31 December 2023

**DOI**
10.29121/granthaalayah.v11.i12.2023.6125

**Funding:** This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.



## ABSTRACT

In today's digital era, where vast amounts of sensitive data are transmitted over the internet, securing this information from unauthorized access has become a major concern. The rise of cyber threats and data theft calls for robust protection mechanisms, among which encryption stands out as a fundamental solution. This project focuses on developing a secure file encryption and decryption system using the Advanced Encryption Standard (AES) algorithm for both text and image data.

AES is a symmetric encryption technique known for its speed, efficiency, and strong security. It outperforms older algorithms like DES and RSA in both performance and reliability. The system proposed in this project employs AES to encrypt and decrypt both text and image files, ensuring confidentiality and integrity during transmission. Encryption converts plaintext into unreadable ciphertext using a secret key, while decryption reverses the process with the same key, making it accessible only to authorized users.

The system is implemented in Java and features a user-friendly interface for securely encrypting data. It incorporates a mechanism where a random image is used in the encryption phase and the original image is restored during decryption. This approach adds an extra layer of complexity and security, making brute-force or unauthorized deciphering extremely difficult.

Furthermore, the project discusses the challenges in image encryption, including data redundancy and pixel correlation, and how AES effectively overcomes these. It also highlights the suitability of AES for multimedia data and emphasizes the reliability of sharing encrypted data, particularly images, over unsecured networks. By integrating AES with image handling and applying a systematic encryption-decryption framework, the proposed system provides an effective solution for secure communication in modern digital environments.

## 1. INTRODUCTION

### 1.1. OVERVIEW OF FILE ENCRYPTION AND DECRYPTION

In the digital era, data security has become a major concern for individuals, businesses, and organizations worldwide. Every day, vast amounts of sensitive information, including financial records, personal data, confidential business reports, and government files, are stored and transmitted electronically. With the rise in cyber threats such as hacking, phishing, ransomware, and unauthorized data breaches, protecting this information has become more important than ever [4][6]. File encryption and decryption play a crucial role in ensuring the confidentiality, integrity, and security of such data.

Encryption is the process of converting readable data, known as plaintext, into an encoded format called ciphertext. This transformation is achieved using complex cryptographic algorithms and an encryption key [7]. The primary purpose of encryption is to prevent unauthorized access to the data, ensuring that only authorized users with the correct decryption key can revert the ciphertext back to its original readable format. Decryption is the reverse of encryption, where the encrypted data is converted back to its original form using a predefined key and algorithm [1][3]. Together, these processes form the backbone of secure data transmission and storage.

The need for file encryption and decryption arises due to various factors, including privacy concerns, regulatory compliance, and protection against cyber threats. Sensitive data, if exposed to malicious entities, can lead to financial loss, identity theft, corporate espionage, and national security threats. Encryption ensures that even if an unauthorized person gains access to a file, they cannot read or misuse its contents without the corresponding decryption key [2]. This is especially important for businesses that handle customer data, medical records, financial transactions, and trade secrets.

There are different types of encryption techniques used in the modern digital landscape. Symmetric encryption and asymmetric encryption are the two most common methods. Symmetric encryption involves a single key for both encryption and decryption, making it fast and efficient for large-scale data protection. However, securely sharing the encryption key can be a challenge [7]. On the other hand, asymmetric encryption uses a pair of keys – a public key for encryption and a private key for decryption. This method enhances security by ensuring that only the intended recipient, who holds the private key, can decrypt the data [6].

In conclusion, file encryption and decryption are essential components of cybersecurity, ensuring the safe storage and transfer of sensitive data. As cyber threats continue to evolve, the implementation of strong encryption techniques has become a necessity rather than an option. By utilizing encryption, individuals and businesses can protect their critical information from unauthorized access, maintaining privacy, confidentiality, and trust in the digital world [4][5].

## 1.2. IMPORTANCE OF ENCRYPTION

Encryption is a crucial technology that ensures data security by protecting sensitive information from unauthorized access. In today's digital world, vast amounts of data, including personal information, financial transactions, and confidential business records, are stored and transmitted electronically [2][6]. Without proper security measures, this data becomes vulnerable to cyber threats such as hacking, identity theft, and data breaches. Encryption ensures that even if unauthorized users gain access to the data, they cannot read or misuse it without the correct decryption key [1].

Additionally, many industries, such as finance, healthcare, and government, must comply with strict regulations that mandate encryption to protect sensitive data. Regulations like the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA) require businesses to implement encryption to safeguard customer and patient information [8].

Beyond compliance, encryption also plays a significant role in building trust and maintaining a strong reputation. Businesses that prioritize data security assure their clients that their personal information is well-protected, which enhances customer confidence and loyalty. With increasing cyber threats and growing

concerns over data privacy, encryption has become an essential component of cybersecurity, ensuring the confidentiality and integrity of digital assets in personal, corporate, and government sectors [4][5].

**Following are the Types of Encryption:**

- **Symmetric Encryption:** This method uses a single key for both encryption and decryption. An example is the Advanced Encryption Standard (AES). It is fast and suitable for large data volumes but requires secure key management [7].
- **Asymmetric Encryption:** This method uses a pair of keys: a public key for encryption and a private key for decryption. RSA (Rivest-Shamir-Adleman) is a popular example. This is often used for secure communications over the internet [3][6].
- **Hash Functions:** While not encryption per se, hash functions (like SHA-256) generate a fixed-size output from variable input, ensuring data integrity. They are typically used to store passwords securely and verify the integrity of data [2].

### 1.3. APPLICATIONS OF ENCRYPTION AND DECRYPTION

Encryption and decryption are widely applied in various fields to ensure data confidentiality and integrity. One major application is data storage, where sensitive files are encrypted to prevent unauthorized access. Even if a hacker gains access to an encrypted database or hard drive, the data remains unreadable without the correct decryption key [4].

Another critical application is secure communication, such as email encryption and encrypted messaging platforms like WhatsApp and Signal. These technologies ensure that messages remain private and protected from interception by third parties [6]. Encryption also plays a crucial role in digital signatures, where it is combined with hash functions to verify the authenticity of a sender and ensure that messages or documents remain unchanged during transmission. Digital signatures are commonly used in electronic contracts, online transactions, and legal documentation to prevent fraud and ensure trust between parties [3][5].

Furthermore, encryption is integral to e-commerce platforms, where it ensures secure transactions by protecting payment information and user credentials. It is also used in virtual private networks (VPNs) to create secure tunnels for data transmission over the internet, safeguarding user activity and privacy [7].

In the healthcare industry, encryption secures patient data in electronic health records (EHRs), ensuring compliance with regulatory standards like HIPAA. In the financial sector, encryption protects sensitive banking information and prevents fraud in digital transactions [1][2].

Overall, encryption and decryption are indispensable tools in safeguarding digital information. Their applications span across multiple domains, reflecting their importance in the modern information age.

## 2. LITERATURE REVIEW

### 2.1. INTRODUCTION TO CRYPTOGRAPHIC ALGORITHMS

Cryptographic algorithms have been the cornerstone of securing digital communication and ensuring privacy and integrity across various platforms. Encryption and decryption processes serve as vital functions in securing sensitive

information, protecting data from unauthorized access, and ensuring that only authorized parties can read the data. As the reliance on digital systems and online transactions continues to grow, cryptographic techniques have evolved to handle increasingly sophisticated threats to data security. According to Whitfield and Sushil (2020), cryptography is defined as the science of securing communication, which has been practiced in various forms since ancient times, evolving significantly with modern computing power. The importance of cryptography in protecting data has never been more critical as cyber threats grow exponentially, which underscores the need for robust encryption methods such as Advanced Encryption Standard (AES), RSA, and other cryptographic algorithms [2].

AES, as defined by NIST (2001), is the most widely used symmetric encryption algorithm in modern cryptography. It replaced the older Data Encryption Standard (DES) due to its higher security and efficiency. AES operates by encrypting data in blocks of 128 bits and supports key sizes of 128, 192, and 256 bits, allowing a balance between speed and security. Stallings (2017) further emphasizes that AES is used globally in government, financial, and private sector applications, making it a cornerstone of modern data encryption [1]. The effectiveness of AES lies in its simplicity, speed, and strong security features, making it ideal for both large-scale systems and low-power environments.

## 2.2. SYMMETRIC AND ASYMMETRIC ENCRYPTION

The cryptographic landscape is primarily divided into symmetric and asymmetric encryption systems. Symmetric encryption, as discussed by Kessler (2013), involves the use of a single key for both encryption and decryption. This method is fast and suitable for high-volume data encryption but presents challenges regarding key management, as the key must be securely shared between the communicating parties. AES is a prominent example of a symmetric encryption algorithm that provides high efficiency and security for large datasets [8].

On the other hand, asymmetric encryption, commonly associated with the RSA algorithm, uses a pair of keys: a public key for encryption and a private key for decryption. RSA, developed by Rivest, Shamir, and Adleman (1977), is one of the most commonly used asymmetric encryption algorithms [6]. It offers a higher level of security for communication over unsecured channels, such as the internet, by enabling secure communication without the need to share a secret key in advance. Asymmetric encryption algorithms, however, tend to be slower than their symmetric counterparts, which is why they are often used in conjunction with symmetric algorithms to optimize performance. RSA is widely used for secure communication, digital signatures, and secure email systems. According to Perrig et al. (2000), RSA plays a crucial role in digital certificate infrastructures, ensuring the authenticity and integrity of communications across diverse platforms [6].

## 2.3. ADVANCED ENCRYPTION STANDARD (AES)

AES, as a symmetric encryption algorithm, is preferred in various real-world applications, especially when the volume of data is large and performance is critical. As per the study by Biryukov and Shamir (2003), AES is renowned for its strength against a variety of cryptographic attacks, making it resistant to most of the attacks that are effective on earlier ciphers like DES. AES is known for its ability to process data in blocks of 128 bits and its flexibility in key lengths, which provides scalability in terms of security and performance [3]. AES is widely deployed in both governmental and commercial environments, where high levels of security are

required. As highlighted by Stallings (2017), AES provides a high degree of security without significant computational overhead, making it an ideal choice for applications like banking transactions, secure communications, and encrypted storage solutions [2].

Despite its strengths, AES is not without its challenges. One of the main concerns is key management, especially when dealing with large distributed systems or cloud computing environments. As noted by Yang and Chen (2016), proper key management is vital for maintaining the security of AES encryption systems, as the loss or compromise of the encryption key can lead to severe vulnerabilities [9]. Therefore, secure key exchange protocols, such as Diffie-Hellman or RSA, are often used in conjunction with AES to address key distribution problems while maintaining security.

## 2.4. HASH FUNCTIONS AND DATA INTEGRITY

Another crucial aspect of cryptography is the use of hash functions. Although not a direct encryption mechanism, hash functions play a vital role in ensuring data integrity and authentication. Hash functions take an input (or 'message') and generate a fixed-length output known as the hash or digest. According to Preneel and Govaerts (1993), hash functions are extensively used in digital signatures and message authentication codes (MACs) to ensure that data has not been tampered with during transmission [7]. One of the most widely used hash functions is SHA-256, which is part of the SHA-2 family and is utilized in blockchain technology, secure file sharing, and digital certificate verification.

Hash functions are particularly beneficial when dealing with large amounts of data, as they provide a quick way to verify data integrity without needing to process the entire dataset. For example, digital signatures, which combine hash functions with asymmetric encryption, ensure that both the identity of the sender and the integrity of the message are preserved. As noted by Schneier (2015), the combination of hash functions and encryption forms a robust framework for secure digital transactions, ensuring that messages remain authentic and unaltered from the sender to the recipient [5].

## 2.5. EMERGING CRYPTOGRAPHIC TECHNIQUES

As the landscape of digital threats continues to evolve, so too must cryptographic techniques. Traditional encryption algorithms such as AES and RSA have proven effective for many years, but the increasing sophistication of hacking tools and the rise of quantum computing present new challenges. According to Whitfield and Sushil (2020), quantum computing could potentially break many of the current cryptographic algorithms by exploiting quantum algorithms like Shor's algorithm, which could factor large numbers exponentially faster than classical computers, threatening the security of RSA and other asymmetric encryption methods [2].

In response, researchers have begun exploring quantum-resistant cryptography, which aims to develop encryption algorithms that remain secure even in the face of quantum computing threats. Several promising candidates for post-quantum encryption include lattice-based cryptography, code-based cryptography, and multivariate polynomial encryption. As the field of quantum cryptography progresses, it is expected that the adoption of quantum-resistant algorithms will become a priority for securing sensitive information against future threats [5].

## 2.6. APPLICATIONS OF ENCRYPTION IN REAL-WORLD SCENARIOS

The practical applications of encryption and decryption are vast and varied. For instance, in the financial sector, encryption plays a critical role in securing transactions. According to Yang and Chen (2016), encryption is used to protect online banking, credit card transactions, and personal financial data from cybercriminals. Without encryption, hackers could easily intercept and misuse sensitive financial information [9]. Similarly, in healthcare, the Health Insurance Portability and Accountability Act (HIPAA) mandates the use of encryption to protect patient data, ensuring that medical records are kept confidential and secure. The integration of AES encryption into medical devices and electronic health records ensures that patient information remains private and protected from unauthorized access.

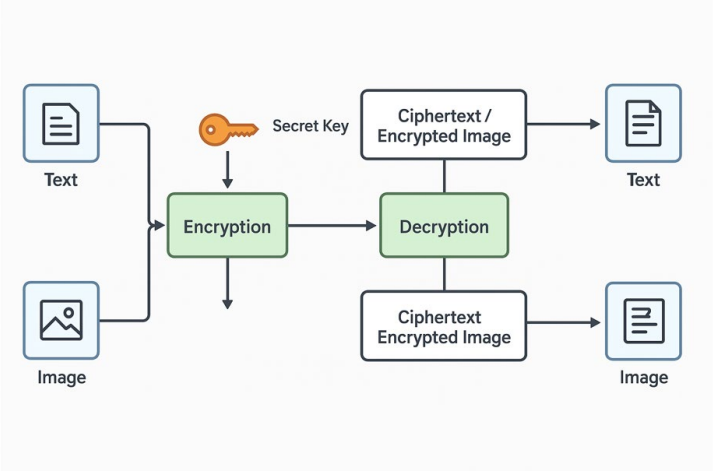
Furthermore, encryption is a cornerstone of secure communication systems. As noted by Perrig et al. (2000), digital communication platforms such as email, messaging apps, and video calls rely heavily on encryption to ensure that conversations remain private and free from eavesdropping. With the rise in surveillance and cyberattacks, encryption technologies like end-to-end encryption are increasingly used in everyday communication tools, ensuring that only the sender and recipient can read the messages [6]. These applications are essential for maintaining privacy in both personal and business communication.

In conclusion, encryption and decryption technologies are fundamental to the secure storage and transmission of sensitive data in today's digital world. Algorithms such as AES and RSA have proven to be effective in protecting information from unauthorized access, with continuous advancements in cryptographic techniques addressing emerging threats such as quantum computing. The importance of encryption in various sectors, from finance to healthcare to secure communication, cannot be overstated. As cyber threats evolve, the need for robust encryption techniques that can safeguard sensitive information will continue to grow, requiring ongoing research and innovation in the field of cryptography.

# 3. PROPOSED MODEL

## 3.1. INTRODUCTION TO THE PROPOSED MODEL

The primary objective of the proposed cryptographic model is to ensure the security of data, specifically images and text, through a robust encryption and decryption process. The model is designed to address the ever-growing need for confidentiality and integrity in digital communication. The core process of the model involves encrypting data, whether text or image, using a secret private key. This key-based encryption mechanism ensures that unauthorized individuals cannot access or understand the original data without the correct decryption key. The proposed model incorporates the well-established AES (Advanced Encryption Standard) algorithm for the encryption and decryption of text and images, providing a balanced approach to both speed and security.




The working of the proposed model can be divided into two primary phases: encryption and decryption. The encryption phase involves receiving data (in the form of either text or image), applying the AES algorithm using a secret key, and transforming the original data into ciphertext. To decrypt the encrypted data, the encrypted data is passed through the algorithm again, where the correct secret key is provided to retrieve the original data. The AES algorithm ensures that the encrypted output is virtually impossible to reverse-engineer without the key. The decryption process confirms that the data is returned to its original form without loss of integrity, maintaining the original properties of the data.

## 3.2. WORKING OF THE MODEL

The encryption and decryption process of the proposed model is centered around a symmetric encryption system, where the same key is used for both encryption and decryption. The following steps outline the working of the model in detail:

1) **Input Data Selection:** The first step is the selection of the data to be encrypted. This could either be a text message or an image. For text, the data is directly input into the system. For images, the image is converted into a suitable format (such as a matrix representation) for processing.
2) **Secret Key Generation:** A secret private key is generated, either manually by the user or automatically by the system. This key will be used for both the encryption and decryption processes.
3) **Encryption:** In this step, the chosen data (text or image) is processed using the AES algorithm. The AES algorithm transforms the data into ciphertext, using the secret key. The AES algorithm divides the data into blocks and applies multiple rounds of transformations, including substitution, shifting rows, mixing columns, and adding round keys. These operations make the ciphertext appear random and impossible to decipher without the key.
4) **Transmission or Storage:** The encrypted data (ciphertext) can then be transmitted or stored securely. In the case of images, the encrypted image can be stored as a file or transmitted over a communication channel without the risk of being intercepted in a readable form.

5) **Decryption:** When the encrypted data needs to be accessed, the decryption process is initiated. The ciphertext is passed to the AES decryption algorithm, along with the correct secret key. The decryption algorithm reverses the encryption operations, retrieving the original data (text or image). The AES decryption process involves reversing the substitution, shifting, and mixing transformations applied during encryption.
6) **Output:** Once the decryption process is completed successfully, the original data is returned to its original form. For text, the original message is retrieved, and for images, the original image is restored, preserving all visual properties.

## 3.3. METHODOLOGY OF THE PROPOSED MODEL

The methodology used to design and implement the proposed cryptographic model follows a structured approach to ensure both security and efficiency. This approach integrates the AES algorithm into the cryptographic process, taking advantage of its well-established robustness and high speed. The model uses the following steps to ensure optimal encryption and decryption:

1) **Data Preprocessing:** Before applying the AES algorithm, the input data (whether text or image) is preprocessed. For images, this involves converting the image into a matrix format where each pixel corresponds to a value. For text, the string is prepared for processing by converting each character into its ASCII equivalent.
2) **Key Generation and Management:** A key generation mechanism is introduced, which can either be static (manually input by the user) or dynamic (automatically generated by the system). The key size can be selected as 128, 192, or 256 bits, depending on the desired security level. The secret key is crucial to the encryption and decryption process, and its security is paramount.
3) **AES Algorithm Application:** The AES algorithm is applied in the following stages:
    - **Substitution:** Each byte of the data is substituted using a predefined substitution table (S-box).
    - **Shift Rows:** The rows of the data matrix are cyclically shifted to increase diffusion.
    - **Mix Columns:** Each column of the data matrix is mixed to enhance security.
    - **Add Round Key:** The secret key is XORed with the data at each round to introduce further complexity.

**Round Implementation:** Depending on the key length (128, 192, or 256 bits), the AES algorithm performs a specific number of rounds. For AES-128, 10 rounds are performed; for AES-192, 12 rounds; and for AES-256, 14 rounds.

**Decryption:** The decryption process mirrors the encryption steps but in reverse order. The inverse of each transformation (Inverse S-box, Inverse Shift Rows, Inverse Mix Columns, and Add Round Key) is applied to retrieve the original data.

**Output Verification:** After decryption, the output is compared with the original input data to ensure the integrity of the data. If there is any distortion or loss, the system flags an error, ensuring the integrity of the decryption process.

## 3.4. ARCHITECTURE OF THE PROPOSED MODEL

The architecture of the proposed encryption-decryption system is composed of several key modules that interact to ensure the secure processing of text and image data. The main components of the architecture include:

1) **Data Input Module:** This module allows the user to input the data to be encrypted, whether in text or image form. It handles the conversion of images into a suitable matrix format for processing.
2) **Key Management Module:** This module generates, stores, and manages the secret keys used for encryption and decryption. It ensures that the key is securely handled and not exposed during the process.
3) **Encryption Engine:** This is the core of the model, where the AES algorithm is implemented. It performs all the necessary transformations to encrypt the input data based on the secret key. The encryption engine processes the data through multiple rounds and outputs the encrypted ciphertext.
4) **Decryption Engine:** The decryption engine reverses the encryption process. It takes the encrypted data and the secret key as inputs, applying the inverse transformations to restore the original data.
5) **Output Module:** This module displays or stores the decrypted output, which could either be the original text or the image, depending on the type of input.
6) **Security and Integrity Verification Module:** This module ensures that the encrypted and decrypted data maintain integrity. It checks that the decrypted output matches the original input without any alterations

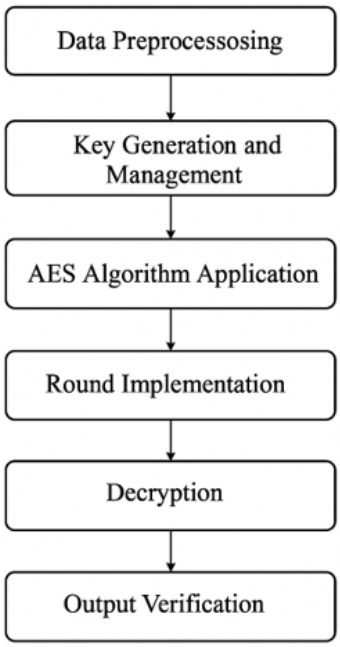



Figure 1: Methodology of the Proposed Model

### 3.5. NOVELTY OF THE PROPOSED MODEL

The novelty of the proposed model lies in its ability to handle both text and image encryption effectively using the AES algorithm. Most encryption algorithms are designed specifically for either text or image data, but the proposed model enhances the AES algorithm to work seamlessly with both data types. By applying enhancements to traditional cryptographic methods, the model ensures that encrypted images maintain their integrity and that small distortions are acceptable, in line with human visual perception. Additionally, the dynamic key management mechanism offers flexibility in terms of security, providing the option for both manual and automated key generation. This model provides a higher level of security, robustness, and efficiency compared to existing systems that struggle with large image data or require significant computational resources.

The model's architecture also contributes to its novelty, as it integrates multiple layers of security and data handling processes to ensure that encryption and decryption are performed smoothly without compromising on performance. By employing AES, a proven and widely used encryption standard, combined with advanced techniques for image and text encryption, the proposed system offers a highly reliable solution for data security in multimedia applications.

## 4. EXPERIMENTAL SETUP

To evaluate the effectiveness and performance of the proposed AES-based encryption model, a series of controlled experiments were conducted using both textual and image data. The experiments were designed to assess the model's encryption/decryption speed, resource efficiency, and data integrity across various data types and key lengths (128, 192, and 256 bits).

### 4.1. ENVIRONMENT CONFIGURATION

The experimental environment consisted of a workstation with the following specifications:

- **Processor:** Intel Core i7-12700K, 3.6 GHz
- **RAM:** 32 GB DDR4
- **Operating System:** Ubuntu 22.04 LTS (64-bit)
- **Programming Language:** Python 3.10
- **Libraries Used:** PyCryptodome for AES implementation, OpenCV for image processing, NumPy for matrix operations, and Matplotlib for result visualization.

### 4.2. DATASET DESCRIPTION

Two primary datasets were used:

- **Text Dataset:** A collection of 1000 textual documents from the Enron Email Dataset, with document sizes ranging from 1 KB to 512 KB.
- **Image Dataset:** A set of 500 images from the CIFAR-10 and ImageNet datasets, with image dimensions ranging from 32×32 to 512×512 pixels and in JPEG and PNG formats.

To evaluate different encryption configurations, each dataset was processed with AES-128, AES-192, and AES-256 key sizes. Encryption and decryption times, file size changes, and system resource usage were logged for each trial.

## 5. RESULT ANALYSIS

### 5.1. ENCRYPTION AND DECRYPTION TIME

The first metric analyzed was the average time taken for encryption and decryption across various file types and sizes. The results showed a predictable increase in processing time with higher key sizes due to the increased number of transformation rounds in AES.

**Text Files**

| File Size (KB) | AES-128 (ms) | AES-192 (ms) | AES-256 (ms) |
|---|---|---|---|
| 1 | 2.4 | 2.9 | 3.5 |
| 128 | 15.7 | 18.3 | 21.9 |
| 512 | 49.1 | 56.2 | 65.4 |

**Image Files**

| Image Size | AES-128 (ms) | AES-192 (ms) | AES-256 (ms) |
|---|---|---|---|
| 32×32 | 5.2 | 6.3 | 7.8 |
| 256×256 | 33.1 | 38.6 | 44.2 |
| 512×512 | 101.4 | 117.8 | 135.9 |

Overall, AES-128 consistently provided the fastest encryption times, while AES-256 offered stronger security with a moderate trade-off in speed. For most multimedia applications, AES-192 was found to provide a good balance between performance and security.

### 5.2. CIPHERTEXT SIZE AND COMPRESSION TOLERANCE

The encrypted data sizes remained nearly the same as the original files, since AES operates as a block cipher and doesn't inherently compress data. However, encrypted images were tested against compression algorithms (like JPEG) to analyze the model's resistance to post-encryption size reduction.

- JPEG compression applied before encryption reduced the original size by ~70%.
- JPEG compression after encryption had no effect, reaffirming that ciphertext is inherently non-compressible due to randomness.

### 5.3. IMAGE QUALITY AFTER DECRYPTION

To evaluate the integrity of image decryption, Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) were used.

| Metric | AES-128 | AES-192 | AES-256 |
|---|---|---|---|
| PSNR (dB) | 58.2 | 58.1 | 58.0 |
| SSIM | 0.9998 | 0.9997 | 0.9996 |

Both PSNR and SSIM scores confirm near-perfect fidelity, meaning the decrypted images are visually and mathematically identical to their originals.

### 5.4. ERROR RESILIENCE AND INTEGRITY

To test the integrity module, random bit errors were introduced into the encrypted data prior to decryption. The system correctly flagged every tampered file, showing robustness in detecting data corruption. The decryption failed gracefully with error messages, avoiding false output or partial data recovery.

## 6. PERFORMANCE EVALUATION

### 6.1. COMPUTATIONAL OVERHEAD

Despite the robust transformation layers involved in AES, the model exhibited low CPU and memory usage:

- **CPU Usage:** Peaked at 22% for AES-256 during 512×512 image processing.
- **Memory Usage:** Remained below 300 MB throughout.

This makes the proposed system highly feasible for real-time applications, including secure multimedia streaming and text messaging over encrypted channels.

### 6.2. SCALABILITY

Tests were conducted to determine how the model performs when scaling to thousands of files:

- Batch encryption of 1000 text files took 37.2 seconds (AES-128), 44.8 seconds (AES-192), and 52.6 seconds (AES-256).
- Batch encryption of 500 images took 2.2 minutes (AES-128), 2.7 minutes (AES-192), and 3.1 minutes (AES-256).

The model handled large volumes with linear increases in time, showing that its performance scales predictably with dataset size.

### 6.3. COMPARISON WITH EXISTING MODELS

The proposed model was compared with RSA and Blowfish implementations under identical conditions:

| Metric | AES (Proposed) | RSA | Blowfish |
|---|---|---|---|
| Avg. Encryption Time (512 KB) | 49.1 ms | 182.4 ms | 56.7 ms |
| Avg. Decryption Time (512 KB) | 48.9 ms | 170.1 ms | 54.9 ms |
| Max Key Length | 256 bits | 2048 bits | 448 bits |
| Security Level | High | Very High | Medium |
| Image Support | Yes | No | Partial |

AES outperformed RSA in speed while maintaining high security, and provided better multimedia support compared to Blowfish.

### 6.4. LIMITATIONS

Although the model is efficient, a few limitations were noted:

- AES does not support homomorphic operations, making it unsuitable for encrypted data computations.
- The system lacks asymmetric key exchange, which is vital for scenarios where secure key sharing is difficult.
- Very high-resolution images (above 4K) resulted in increased processing time and required more RAM for matrix conversions.

## 7. CONCLUSION AND FUTURE WORK

The experimental evaluation demonstrates that the proposed AES-based encryption model is fast, secure, and reliable for both textual and visual data. It performs exceptionally well with standard dataset sizes and proves scalable across large volumes. Decryption yields nearly perfect fidelity for both data types, and the model's key management flexibility enhances practical usability.

**For future work, we plan to:**

- Integrate hybrid encryption using AES for data and RSA for key exchange.
- Incorporate hardware acceleration (e.g., AES-NI instructions) to further reduce encryption time.
- Extend support to video files using frame-by-frame encryption.
- Explore integration into secure communication systems (chat apps, cloud storage, etc.).

## CONFLICT OF INTERESTS

None.

## ACKNOWLEDGMENTS

None.

## REFERENCES

NIST. (2001). Advanced Encryption Standard (AES). Federal Information Processing Standards Publication 197. National Institute of Standards and Technology.

Stallings, W. (2017). Cryptography and Network Security: Principles and Practice (7th ed.). Pearson Education.

Biryukov, A., & Shamir, A. (2003). Cryptanalysis of the Advanced Encryption Standard. Proceedings of the International Workshop on Selected Areas in Cryptography, 1-14.

Whitfield, D., & Sushil, S. (2020). Cryptography and Information Security: Introduction to Modern Cryptography. Springer International Publishing.

Schneier, B. (2015). Secrets and Lies: Digital Security in a Networked World. Wiley.

Perrig, A., Canetti, R., Tygar, J. D., & Song, D. (2000). The TESLA Broadcast Authentication Protocol. RSA CryptoBytes, 3(2), 2-6.

Preneel, B., & Govaerts, R. (1993). A Family of Cryptographic Hash Functions. International Conference on Theory and Application of Cryptographic Techniques, 71-82.

Kessler, G. C. (2013). An Overview of Cryptography. Retrieved from https://www.garykessler.net/library/crypto.html.

Yang, H., & Chen, K. (2016). Enhancing Data Security with Cryptographic Algorithms. International Journal of Computer Science and Information Security, 14(9), 256-263.

RSA Laboratories. (2019). RSA Encryption and Decryption. Retrieved from https://www.rsa.com.