Iran Journal of Computer Science
https://doi.org/10.1007/s42044-024-00191-y

RESEARCH

Enhanced image encryption using AES algorithm with CBC mode: a secure and efficient approach

Kevin Haria[1] · Riya Shah[2] · Vanshita Jain[2] · Ramchandra Mangrulkar[2]

Received: 10 November 2023 / Accepted: 19 April 2024

Abstract
In recent times, the necessity for secure communication and data transfer has witnessed a significant surge owing to the widespread usage of digital devices and the internet. Encryption is one of the most commonly employed techniques for securing data, which involves transforming plain data into a scrambled form that can only be deciphered with the correct decryption key. This paper presents a method for encrypting digital images using symmetric key cryptography. The proposed method employs the Advanced Encryption Standard (AES) block cipher in Cipher block chaining (CBC) mode to encrypt the image data, along with a random Initialization Vector (IV) and Key. Furthermore, the image data is padded using the PKCS7 padding scheme to ensure that the block cipher operates on blocks of fixed size. The proposed encryption method is implemented in Python using the cryptography library and tested on sample images. The experimental outcomes show that the suggested approach offers secure and effective encryption of digital images, with negligible overheads in terms of time and space complexity. The proposed method can be utilized in various applications that require secure image transfer and storage, such as e-commerce, medical imaging, and confidential document exchange.

Keywords AES algorithm · Block cipher · CBC mode · Cryptography · Decryption · Encryption · Image encryption · Initialization vector (IV) · PKCS7 padding · Secure communication

1 Introduction

In the setting of the dynamic and ever-changing digital environment, where the widespread use of digital communication and information sharing has grown pervasive, ensuring the security and confidentiality of sensitive data has emerged as a critical issue. The increased prevalence of digital devices and internet usage has resulted in the heightened exposure of data to potential threats and vulnerabilities. Consequently, the discipline of cryptography has arisen as a crucial factor in protecting data against unauthorized intrusion and alteration. Cryptography encompasses the utilization of intricate algorithms and cryptographic keys to convert plaintext into ciphertext, thereby guaranteeing the confidentiality and integrity of transmitted and stored data. The foundation of ensuring the security of data lies in the method of encryption, which serves as a fundamental mechanism within the field of cryptography. The utilization of encryption is of utmost importance in the realm of information security, as it involves the transformation of plaintext into ciphertext through the application of a cipher algorithm and a confidential encryption key. The aforementioned procedure effectively renders the initial content unintelligible in the absence of the appropriate decryption key, so imparting a vital level of safeguard against malevolent entities and unauthorized entry.

Block ciphers are a highly used category of encryption techniques, as discussed by [1]. Block ciphers function by dividing the plaintext into blocks of a predetermined size,

Kevin Haria, Riya Shah, Vanshita Jain and Ramchandra Mangrulkar contributed equally to this work.

✉ Ramchandra Mangrulkar
ramchandra.mangrulkar@djsce.ac.in

Kevin Haria
kevinharia11@gmail.com

Riya Shah
riyajshah021@gmail.com

Vanshita Jain
vanshitajainofficial@gmail.com

[1] Department of Information Technology, Dwarkadas J. Sanghvi College of Engineering, Mumbai 400056, India

[2] Department of Computer Engineering, SVKM's Dwarkadas J. Sanghvi College of Engineering, Mumbai 400056, Maharashtra, India

Published online: 10 May 2024

thereafter subjecting these blocks to encryption using a uniform key. The aforementioned methodology presents notable benefits in terms of velocity and effectiveness, as it enables the concurrent processing of numerous data blocks, rendering it especially well-suited for the encryption of large quantities of data. Stream ciphers operate at the level of individual bits or bytes by generating a sequence of pseudorandom bits or bytes using a key and an initialization vector. The pseudorandom bits are subsequently merged with the plaintext by the XOR operation, resulting in the generation of the ciphertext. Stream ciphers demonstrate exceptional performance in situations characterized by the continual arrival of data, such as real-time communication or network-based data transmission.

This research paper proposes an enhanced image encryption approach that leverages the AES algorithm in CBC mode as studied by [2]. The AES algorithm, widely recognized for its robustness and efficiency, provides a strong foundation for securing digital images. In the Cipher Block Chaining (CBC) mode, each block of plaintext is XORed with the ciphertext of the previous block before encryption. This introduces an element of feedback, ensuring that identical blocks of plaintext are encrypted differently, thus enhancing the overall security.

The primary distinguishing factors and contributions of the suggested algorithm lie in its hybrid approach, heightened security measures, and wide range of potential applications. The uniqueness of the approach lies in the amalgamation of both block ciphers and stream ciphers. To initiate the encryption, a random key is generated using the AES algorithm, which serves as the foundation for both the block cipher and stream cipher components. Additionally, an initialization vector (IV) is used to create a unique stream of pseudorandom bits for each block of data. This diversity in the pseudorandom streams significantly strengthens the overall encryption, adversaries find it extremely challenging to figure out either the primary data or the encryption key. By adopting this secure and efficient approach, the proposed method elevates the confidentiality and integrity of digital images during transmission and storage. The versatility of this approach enables it to be deployed in various applications, such as e-commerce, medical imaging, and confidential document exchange, where secure image transfer and storage are of paramount importance.

The presented research paper delves into the intricate details of a proposed image encryption method. The AES algorithm in CBC mode is elaborated upon and comprehensive experimental results are presented to validate the effectiveness and efficiency of the approach. The findings demonstrate that this method provides a robust and reliable solution for secure image encryption, with minimal overhead in terms of time and space complexity.

The remainder of this paper is organized as follows: Sect. 2 provides a review of appropriate literature for this. Section 3 discusses the research gaps and scope for this research work. Section 4 details the proposed enhanced image encryption approach. Section 5 presents the experimental methodology and Sect. 6 focuses on attack analysis. Section 7 discusses the results and their implications. Finally, Sect. 8 concludes the paper with a summary of contributions and avenues for future research.

2 Literature survey

The Advanced Encryption Standard (AES) algorithm, a well-known and reliable symmetric-key encryption scheme, is described in the paper [3]. The paper's explanation encompassed the architecture and security properties of the method, including the incorporation of S-boxes, key expansion, and the topology of the substitution-permutation network (SPN). The article additionally addressed the four standard key sizes *(128, 192, and 256 bits)* as well as the supported operating modes of the algorithm, which include ECB, CBC, CFB, OFB, and CTR. The authors subsequently described certain challenges and complications encountered during the implementation of AES, including the selection of appropriate key sizes and modes of operation, as well as the need to strike a balance between the performance overheads associated with hardware and software execution of the algorithm.

The concept of "strongly universal" hash functions was introduced in the study [4], which also analysed the security aspects of the Cipher Block Chaining Message Authentication Code (CBC-MAC). This study provided a comprehensive examination of the security of CBC-MAC and presented evidence to support its security under certain conditions. The concept of "strongly universal" hash functions was first introduced, and this had an impact on how cryptographic algorithms were created. The underlying block cipher is idealized in the article, although this may not always be the case in actual use. The analysis is only applicable in the scenario when the attacker lacks access to the block cipher's key.

The Advanced Encryption Standard (AES) is described in detail in the paper [5]. The substitution-permutation network (SPN) structure, the usage of various key sizes and block sizes and the Galois/Counter Mode (GCM) for authenticated encryption were just a few of the technical aspects of AES that the author went through in detail. The benefits of AES over earlier encryption methods, including its security, effectiveness, and adaptability, were also covered in the article. The paper also emphasized the significance of key management in AES and provided helpful guidance on secure key handling. The author described the various key types utilized in AES as well as key production and distribution techniques. The paper also gave instances of how AES has been used in many contexts, including network security, file encryption, and database security. Finally, the author provided advice on

how to utilize AES successfully, including avoiding typical key management errors and keeping up with the most recent AES advancements.

Since data are transferred through wireless channels, where hackers may obtain sensitive information, secure data is a key concern in wireless sensor networks. Current solutions for WSN's crucial resource-constrained devices don't take into account the unique resource limits of WSN. In order to protect the privacy and integrity of data in WSN, this paper [6] described a strong encryption method that uses Elliptic Curves Diffie-Hellman key exchange and AES encryption in CBC mode. To do this, it is suggested to generate g(x, y) using an Elliptic curve 25,519 (RFC 7748) in order to compute a shared secret, SK(X, Y), where Y is the initial 256-bit AES key and X' is the initialization vector corresponding to the last 128 bits of X. Unlike the mapping technique, which turns the plaintext into a series of points on the elliptic curve before doing the arithmetic operations to get the encryption, this approach is scalable and robust. In order to compare the performance of the suggested encryption method to the current review, the author ran a number of tests. In order to provide the best results in terms of robustness and adaptability to sensor network limits, the energy consumption, memory occupation rate, and operating time of the cryptographic processes are estimated for performance analysis.

This work [1] provided an image encryption method based on DNA computing and the Nonlinear Feedback Shift Register (NLFSR). First, a pseudorandom sequence produced by a NLSFR-based Key stream generator was used to permute the image. Next, DNA calculations in cipher block chaining mode were used to substitute the pixel values.

The purpose of this work [7] was to create synchronized dynamic keys based on chaos and, using the suggested synchronized random keys, to enhance the chaos-based advanced encryption standard (AES) algorithm. To ensure the synchronization of master–slave discrete chaotic systems, a rippling control technique based on sliding mode control (SMC) technology was first presented. In communication systems, under the synchronization, the transmitter and receiver could simultaneously obtain the identical dynamic random chaos signals. Next, based on chaotic synchronization, a novel modified AES cryptosystem was introduced, with dynamic random keys. Static keys were used in traditional AES cryptosystems and need to be exchanged beforehand along with confirmation that they are stored securely. The static key in the suggested architecture, on the other hand, became dynamic and unpredictable due to the use of chaotic systems synchronization technology, negating the necessity for it to be stored or sent via open channels. As a result, the drawback of key storage might be removed, and encryption security could be strengthened. Lastly, a novel image encryption algorithm has been constructed using the proposed chaos-based AES (CAES) algorithm. Through simulation tests, the information entropy, correlation indexes, statistical analysis, and histogram had been computed and studied to show the potential and enhancement of the CAES cryptosystem.

The paper [8] suggested a unique method for image encryption that combined chaotic sequences with a block cipher. The chaotic sequences enhanced the complexity and randomness of the encryption procedure, whereas the implementation of block ciphers provided a robust and efficient system for encryption. The proposed approach for encrypting images involved partitioning the image into distinct blocks and subsequently applying a block cipher with chaotic sequences to each individual block. The encrypted image was formed from the resulting encrypted blocks. The research presented empirical evidence and evaluations to examine the efficacy and integrity of the proposed encryption methodology. In general, the study presented a novel methodology for image encryption that integrated chaotic sequences with a block cipher, resulting in a robust and efficient encryption method.

The AES-CBC cipher method and its application in the context of IPsec (Internet Protocol Security) were described by [9] in his paper, which was published as RFC3602. By encrypting data and authenticating communication between network organizations, the IPsec suite of protocols enabled secure communication over the internet. The use of AES-CBC in IPsec's Encapsulating Security Payload (ESP) protocol was the paper's primary area of interest. It explained how AES-CBC's authentication and encryption features were used to protect data packets as they travel over the network. The study also examined numerous issues regarding initialization vectors (IVs), padding, and key management while implementing AES-CBC within IPsec. This paper helped standardize IPsec's use of AES, fostering better network communication security practices.

This article [10] concentrated on a new picture encryption technique that used the CBC mode and took as input a normal grayscale image. Arnold transformation was used to first jumble the input image, and then the CBC mode was applied to encrypt it. Using this procedure, a sequence of Cipher blocks were generated, and every successor block was linked to every predecessor block. In addition, a set of the strongest random secret keys for each of the cipher blocks that follow had been generated. After applying the CBC block and these secret keys to a typical grayscale image, the appropriate encrypted image was produced. Several statistical techniques, including Histogram analysis, UACI, NPCR, entropy, and correlation coefficients, had been used to investigate this method. The various experimental findings demonstrated the proposed scheme's straightness qualities and demonstrate that it was safe from various sorts of attacks.

The paper [11] summarized the AES algorithm and it thoroughly discussed about other algorithms like DES, 3DES,

Blowfish and so forth. The Advanced Encryption Standard (AES) algorithm, which is widely used in hardware and software, is one of the most efficient algorithms. This method works with a *128-bit* block cipher, allowing it to work with keys of various sizes, including *128, 192, and 256 bits*. Several key aspects of the AES algorithm were illustrated in this paper, along with some earlier work that had been done to assess how well it performed in terms of encrypting data under various conditions. According to research findings, AES offered significantly higher security compared to other algorithms like DES, 3DES, etc.

To identify the problem with the encryption process, the AES-CBC mode's design and analysis were presented in the publication [2]. The chip size decrease was examined using simulation. This study investigated the AES algorithm's dependability for use in EO small satellites. Using the AES mode CBC, the effect of the spread of SEU faults occurring during encryption was investigated. When summing everything up, a study of the propagation of noise-related transmission defects was done. To stop the data corruption brought on by SEUs, an error identification and correction model of AES based on the Hamming protocol was suggested (12, 8). The suggested fault detection and correction AES model was intended for usage in the satellite technology area, however, it could also be used for other purposes that were intended for use in hazardous settings, such as those for manned aerial vehicles, nuclear reactors, and extraterrestrial exploration.

The paper [12] gave a performance evaluation of the Avalanche Effect-based security upgrade of the Advanced Encryption Standard (AES) in Cipher Block Chaining (CBC) mode. In this work, a brand-new, enhanced method for boosting AES-CBC security was presented. Before encryption rounds in CBC mode, the Unix time was utilized as a source for the Initialization Vector (IV). The outcomes demonstrated that the method produced unique ciphertext with each run. In other words, the likelihood of cracking the encryption key was greatly reduced by using diverse ciphertext output. Additionally, the results were checked for compliance with the security requirements and examined using the Avalanche Effect. The outcomes demonstrated that the encryption technique was successful in upholding the avalanche effect requirement and adding more strength to the encryption procedure by prohibiting the updating of the encryption key for each new ciphertext.

A performance evaluation methodology was provided in the paper [13] to determine how the configuration of any encryption/decryption scheme impacted performance. The AES algorithm had been subjected to the approach in five different execution platforms, yielding results that apply to all AES algorithm users. This study stated that the average increase in encryption time for utilizing a larger key size was about 16.4%, while CBC chaining required an additional increment of 4.6%. The average increase in decryption time when employing a larger key size was about 15.4%, and CBC chaining necessitated an additional increment of 7.5%. The AES encryption/decryption technique and its setup settings had been introduced in this evaluation effort. Then, a methodical approach had been proposed to assess how the setup settings affect the algorithm's performance. Last but not least, the method's deployment to a variety of platforms with extremely diverse levels of processing capacity had produced quantifiable data regarding the impact of the AES settings on performance.

AES (Advanced Encryption Standard) is used by WhatsApp, Signal, VeraCrypt, 7-zip, and WinZip to encrypt data in risky conversations or storage. Furthermore, it is incorporated into other processors, including Westmere and Jaguar. Consequently, it is imperative that a security analysis of this encryption technology be provided. In this study [14], cryptanalysis for AES encryption of blocks and multimedia data was presented. The study covers all AES modes, including the abbreviations for Electronic Code Book (ECB mode), Cipher Block Chaining (CBC mode), Cipher Feedback (CFB mode), Output Feedback (OFB mode), and Counter (CTR mode). A number of analyses were offered for just block encryption, including passwords and multimedia files like pictures, videos, and audio. Firstly, the paper compared its runtime with various encryption methods on the same hardware to study its runtime. Findings demonstrated that AES is resistant to all kinds of attacks and that AES could withstand them.

This work [15] focused on information security in real-time applications and presented an optimal cryptography technique for embedded systems. For the purpose of encrypting medical images, the suggested method combined elements of Elliptic Curve Cryptography (ECC) with the Advanced Encryption Standard (AES). Important contributions included a modified AES for picture encryption that increased speed without violating Shannon diffusion and confusion principles and an optimized ECC hardware architecture for effective multipliers, which reduced time complexity. The cryptosystem drastically cut down on execution time by using a co-design approach with a hardware accelerator for ECC and an NIOS II processor for AES. Security analysis demonstrated the practical usefulness of the proposed algorithm by validating its simplicity, correctness, and excellent security.

Four of the most popular and in-use symmetric key algorithms were fairly compared in the paper [16]: DES, 3DES, AES, and Blowfish. Since they needed more computer processing capacity, asymmetric encryption algorithms were almost a thousand times slower than symmetric ones. The performance assessment of a few chosen symmetric algorithms was presented in this study. From the simulation, it could be inferred that was being provided that Blowfish per-

forms better than other methods. Second, except Blowfish, in terms of processing and decoding speed, AES performed better than both 3DES and DES. The final claim was that 3DES performed the worst of all the algorithms described.

Data Encryption Specification (DES), an outdated encryption standard, has been used for many years. However, it has been established that the DES is vulnerable to theoretical attacks, and researchers have shown examples of exhaustive key search attacks. Better security is provided by Triple-DES, a method for improving DES, although it is three times slower, especially with tiny blocks *(64 bits)*. For a new cipher, the AES has been suggested and used for a while. This novel cipher is being used in numerous studies and applications. However, it is necessary to provide its performance analysis in network application scenarios. [17] offered an overhead analysis of the functionality of modern encryption standards in wireless networks through their publication.

Stronger encryption algorithms are required for message security during transmission as a result of the shortcomings in the original AES. AES method with a hybrid approach (Dynamic Key Generation and Dynamic S-box Generation) that took nodal needs into account was proposed in paper [18]. The system made use of *128-bit* data and key length. The suggested AES algorithm with a hybrid approach would be a useful method, according to this study, by increasing AES complexity, one could increase message transmission privacy by causing ambiguity and diffusion in the cipher text. It defended against assaults on messages that are linear, Brute-force, differentiable, and algebraic. The proposed system was declared as a useful tool for internet-based applications including e-bill payment, trading of stocks, e-banking, and online purchases.

The paper [19] suggested using the function of time to create using the AES algorithm's variable key generation. When a user checks in, the value of time was used to generate a dynamic key for AES. The key to the value of time generated during the encryption process was then be found during the decryption process by synchronizing time with a specific tolerance. Since this dynamic key was formed owing to its dynamic nature and was only used at specific intervals, it produced better encryption keys. According to the experimental findings, based on the time that a user checked in during a particular active period, an AES secret password (key) was generated at random. Because the key-producing ciphertext for each encryption operation was altered in this implementation, the security authentication was strengthened. The results showed that the AES encryption-decryption process was relatively quick, with an average time of roughly 0.0023 s, based on time as a useful benchmark.

A picture encryption program using cipher block chaining in AES was created in the paper [20] using the C programming language. The suggested cryptosystem was evaluated and contrasted with a few other chaos-based picture cryptosystems based on encryption/decryption speed and security performance. Simulation results contradicted the generally held belief that AES was unsuitable for picture encryption by demonstrating that AES may be applied to image encryption. In this study, the quickness of AES-based image encryption was also proposed as a speed benchmark for image encryption techniques. Additionally, in real-world communications, picture encryption techniques whose rates are slower than the benchmark should be abandoned.

The proposed selective picture encryption technique based on chaos and CBC-like mode's security was analyzed in the publication [21]. The algorithm created key streams that were XORed with the *4-bit MSB* of each pixel using a Logistic Map. Keyspace analysis, histogram analysis, correlation analysis, entropy analysis, and sensitivity analysis were all included in security analysis. Based on the results of the experiment and the security analysis, it could be said that the suggested technique was safe against various attacks that tried to identify hidden pixels or secret keys in ordinary photos.

Digital picture security is increasingly a significant concern, particularly when sending images across a communication network. This study [22] proposed a novel CBC-AES picture encryption technique with several encryption levels that was based on Arnold scrambling. The original image cannot be recovered by the eavesdropper due to the various degrees of encryption. In this approach, the input picture was first bitwise shuffled, Arnold scrambled and then circularly shifted before each bit was complemented. Finally, the CBC mode of the AES encryption technique was used to produce the final cipher image. The simulation results showed that the suggested approach works well for digital image encryption. The security research demonstrated that this method can withstand a variety of attacks, including statistical, differential, and entropy attacks.

The paper [23] discussed the increasing importance of security information in digital data exchange and storage due to rapid development. The need to protect sensitive image data from unauthorized access has grown, especially as images are increasingly used in industrial processes. In this study, the Key Stream Generator (A5/1, W7) upgraded the Advanced Encryption Standard (AES) to enhance encryption efficiency, especially for low-entropy images. Both methods had been implemented for experimental purposes, and the results of the security study and implementation were provided in detail. A comparison with conventional encryption methods demonstrated the advantages of the improved algorithm.

The paper [24] focused on verifying a fast AES-based picture cryptosystem. The plain image was divided into *128-bit data blocks*. An initial vector permuted the first plain picture block and then each block was successively encrypted using AES in cipher block chaining mode. The initial vector

and cipher image were transmitted via the public information channel to the decryption party. The decryption party used the initial vector and secret key to decrypt the cipher image and recover the original image. The simulation results demonstrated the security and speed of this image cryptosystem, which made it a benchmark for newly proposed image cryptosystems based on chaotic systems.

A high-speed and extremely effective encryption method for high-definition (HD) images was provided by [25] and relied on a modified version of the AES algorithm. AES is a popular block cipher algorithm known for its excellent security and ease of use. This study suggested three modifications to enhance AES algorithm performance by reducing computation costs, lowering hardware requirements, and increasing the security level. To shorten the encryption time, the original AES-128 was modified using the MixColumn transformation; instead of taking 10 rounds it takes *5*. Security was improved by adding MixColumn transformation as the second change to the key scheduling operation. Additionally, the proposed technique replaced the S-box and Inv. S-box in the original AES with a single simple S-box used for encryption and decryption to reduce hardware requirements. The suggested version of Advanced Encryption Standard (AES) used an encrypting mode to fix the problem with the visual pattern. The findings of the experiment presented that the AES algorithm had been modified to make it more suitable for HD image encryption.

The motivation behind the proposed image encryption method stems from the identified shortcomings and weaknesses in existing encryption approaches.The motivation here lies in enhancing encryption security and mitigating the challenges associated with static keys. The proposed approaches, including the use of dynamic keys based on chaos and the integration of Arnold transformation with CBC mode for image encryption, address specific drawbacks found in traditional encryption methods. The paper aims to contribute to the advancement of secure and efficient image encryption methods, taking into account the limitations and vulnerabilities identified in the current state of the art.

3 Research gaps and scope

Based on previous papers and the implementation of cryptographic algorithms using the AES algorithm with CBC mode, these are the research gaps that can be explored.

1. **Security analysis:** While CBC mode with AES algorithm is considered secure, it is not immune to attacks. There is a need for further security analysis to determine the limitations and vulnerabilities of the technique. For example, research could focus on analyzing the impact of using a weak key, the resistance of the technique against brute force attacks, the impact of the initialization vector, and the impact of message modification. By identifying the limitations and vulnerabilities of the technique, researchers can develop better countermeasures to prevent attacks.
2. **Performance optimization:** CBC mode with AES algorithm can be computationally expensive, particularly when encrypting large amounts of data. Researchers can explore various methods of optimizing the performance of the technique without compromising its security. For example, researchers can explore parallel processing or pipelining techniques to improve the processing speed of the technique. Alternatively, researchers can explore reducing the number of encryption rounds used in the technique to reduce processing time.
3. **Incorporating encryption methods:** The combination of CBC mode, with the AES algorithm can be utilized alongside encryption techniques to enhance both security and efficiency. For instance, researchers can investigate integrating CBC mode with the AES algorithm using key encryption methods like RSA. This hybrid encryption scheme has the potential to bolster security while ensuring processing.
4. **Hardware implementation possibilities:** While CBC mode with AES algorithm can be implemented in software hardware implementations offer more efficient encryption. Researchers have the opportunity to explore hardware-based implementations of this technique and assess their security and performance levels. For example, they could consider implementing the technique using Field Programmable Gate Arrays (FPGAs) or Application Specific Integrated Circuits (ASICs) comparing their performance against software implementations.
5. **Exploring applications:** Although CBC mode with AES algorithm is extensively used for data encryption researchers should delve into potential use cases for this technique. One intriguing avenue is exploring its application in encrypting multimedia files such as images, audio, and videos. By venturing into use cases researchers may discover applications that can lead to advancements, in both security and efficiency.

Overall, exploring these research gaps can lead to improvements in the security and efficiency of CBC mode with the AES algorithm, making it an even more effective encryption technique.

Scope

The primary goal of this project is to demonstrate the use of CBC mode in combination with the AES algorithm to encrypt and decrypt digital images. This implementation uses

the cryptography library to perform cryptographic operations and the Python Imaging Library to process image data efficiently. The encryption process begins with the generation of a random Initialization vector (IV) and secret key, which are essential components for preserving the security of image content. The algorithm also includes PKCS7 padding of the image data, which not only increases the size of the data, but also increases the complexity of the encryption to increase its security.

During decryption, the IV and key are employed to successfully decrypt the image data, and the previously applied PKCS7 padding is meticulously removed to restore the original image, ensuring the integrity and confidentiality of the visual content.

In summary, this project serves as a practical demonstration of a robust image encryption and decryption technique, combining the cryptographic strength of AES in CBC mode with the versatility of Python libraries, contributing to enhanced data security in the context of image protection.

4 Methodology

This section provides a description of the methodology employed in the development and implementation of the proposed imagine encryption algorithm. The encryption technique in question is built upon the AES algorithm with CBC mode as its fundamental framework. It has been further customised and optimised to meet specific needs and improve overall performance.

4.1 Algorithm design

The primary objective of the image encryption algorithm's is to concurrently optimize both the security and efficiency aspects of the encryption process. The encryption methodology employed in this research study is based on the Advanced Encryption Standard (AES) algorithm with Cipher Block Chaining (CBC) mode. The following design principles and considerations guided the development of the algorithm:

AES algorithm selection

The selection of Advanced Encryption Standard (AES) as the foundation for image encryption is a judicious decision, given its established standing for robust security attributes and computing efficacy within the realm of symmetric key encryption. The Advanced Encryption Standard (AES) has gained significant acceptance and is commonly regarded as a reliable method for protecting confidential data across various domains.

The implementation of the cipher block chaining mode in AES is a strategic enhancement aimed at bolstering its security stance. The Cipher Block Chaining (CBC) mode incorporates a crucial aspect of dispersion by employing the XOR operation on each plaintext block with the ciphertext of the preceding block, as well as the encryption of all blocks prior to the current one. This process establishes a fundamental characteristic, whereby the encryption of each block is closely interconnected with the encryption of all preceding blocks. Consequently, the Cipher Block Chaining (CBC) mode notably enhances the encryption procedure, thereby augmenting its resilience against statistical analysis and attacks based on pattern recognition.

The combination of AES and CBC mode is a potent amalgamation that leverages AES's cryptographic prowess and CBC's chaining mechanism to augment the level of security. The combination of these two features renders an image encryption method very suitable for the preservation of confidential image data, thereby establishing it as an exceptional option for the protection of digital assets in many fields.

Key management

In the image encryption algorithm, the use of a randomly generated Initialization Vector (IV) adds a crucial layer of randomness and security to the encryption process. The IV serves as a unique starting point for each encryption instance. It is XORed with the initial plaintext block before encryption and subsequently becomes a part of the input for encrypting each subsequent block. This approach ensures that even if the same plaintext block is repeated, it will produce a distinct ciphertext block due to the unique IV.

However, the algorithm's security is intrinsically tied to the strength and management of the encryption key. A securely generated key, which is both sufficiently long and generated with a strong randomization process, is fundamental in resisting brute-force attacks. It is paramount to safeguard the key's secrecy and manage it meticulously to prevent unauthorized access. The strength of the encryption key, coupled with the unique IV for each encryption instance, collectively fortifies the algorithm's ability to protect sensitive image data, ensuring confidentiality and resilience against potential threats.

Padding scheme

In the described image encryption algorithm, the chosen padding scheme is PKCS7 (Public Key Cryptography Standards #7). PKCS7 padding is a widely adopted approach designed to ensure that plaintext data conforms to the prescribed block size of the encryption algorithm, such as AES.

This method accomplishes uniform block alignment by appending bytes to the plaintext to reach the necessary block size. Each appended byte holds the value corresponding to the number of bytes added, which simplifies subsequent removal

during decryption. PKCS7 padding guarantees the consistency and integrity of data processing, making it a dependable choice to adapt plaintext of varying lengths to the fixed block size requirements of AES encryption while preserving data security.

Padding purpose

Padding serves a crucial purpose in AES encryption by ensuring that plaintext data, which may not align perfectly with the fixed block size *(typically 128 bits or 16 bytes)*, can be securely processed. When the plaintext's size isn't an exact multiple of the block size, padding adds extra bits to fill the final block, aligning it correctly for encryption. This ensures uniformity in data processing and maintains the algorithm's integrity, preventing data loss or misalignment.

Padding process

The PKCS7 padding scheme follows these steps:

1. **Determine the Block Size:** First, the algorithm determines the block size of the encryption algorithm being used. In this case, it's the block size of the AES algorithm, which is typically *128 bits (16 bytes)*.
2. **Calculate Padding Length:** The padding length is calculated based on the remaining space in the last block. For example, The padding length is the difference between the size of the block and the size of the previous incomplete block if the original text size is not a multiple of the block size.
3. **Add Padding Bytes:** The padding value is a byte that represents the number of padding bytes added to the plaintext. Each padding byte is set to the value of the padding length. For instance, if two bytes are needed for padding, then both bytes will have the value 0x02.
4. **Apply Padding:** The padding bytes are then appended to the end of the plaintext to fill up the last block to the required block size.

Padding example

Let's illustrate the PKCS7 padding process with an example. Suppose the block size of the AES algorithm is 16 bytes, and the plaintext to be encrypted is 27 bytes long:

Plaintext: "This is a secret message. Encrypt it!"

1. **Step 1: Determine Padding Length**
   Padding Length = Block Size − (Plaintext Size % Block Size) = 16 − (27 % 16) = 16 − 11 = 5
2. **Step 2: Add Padding Bytes**
   Padding Value = 0x05
3. **Step 3: Apply Padding**
   Padded Plaintext: "This is a secret message. Encrypt it! \x05 \x05 \x05 \x05 \x05"

Padding removal

The padding is removed after decrypting the data to obtain the original plaintext without the added padding bytes.

Efficiency considerations

The primary objective is to enhance the efficiency of the algorithm while maintaining a strong emphasis on security. The cryptographic operations are meticulously optimised to minimise computing overhead and guarantee practical performance, hence facilitating the completion of encryption and decryption processes within appropriate timescales for real-world applications. The suggested imagine encryption method offers a robust and effective solution for safeguarding sensitive image data by integrating key design ideas and considerations. The subsequent section will explore the precise approach employed to execute this algorithm and carry out the encryption and decryption procedures.

4.2 Encryption process

The image encryption algorithm employs a sequential set of procedures to convert the initial image data into a protected and unintelligible format. The subsequent section delineates the fundamental phases entailed in the encryption procedure.

Initialization

The first step involves partitioning the image data into segments of a predetermined size as mandated by the AES algorithm. A block normally comprises a specific number of bytes, such as *128 bits or 16 bytes*. The division of the image into discrete blocks enables the implementation of the encryption procedure on a per-block basis, resulting in enhanced efficiency and streamlined encryption and decryption procedures.

Prior to encrypting the blocks, an Initialization Vector (IV) is generated in a random manner. The initialization vector (IV) serves as the initial input for the encryption process, enhancing the encryption technique by introducing an extra layer of randomness and security. In order to prevent the generation of identical ciphertext blocks from the same plaintext block, it is imperative that a unique initialization vector (IV) is used for each encryption instance, regardless of whether the plaintext block is repeated.

Block encryption with CBC mode

Starting with the first block of the image data, the CBC mode is applied to encrypt each block. For each subsequent block, the encryption process involves the following steps:

1. XOR the plaintext block with the ciphertext of the previous block. If it is the first block, the XOR operation is performed with the IV.
2. Apply the AES encryption function to the XORed result, using a securely generated encryption key.
3. The output of the AES encryption becomes the ciphertext for the current block.

Final ciphertext generation

After encrypting all the blocks of the original image, the resulting ciphertext embodies a transformed and highly secure rendition of the original image data. This ciphertext is manifested as a sequential arrangement of encrypted blocks, often represented in binary form. Each block encapsulates a portion of the image's encrypted content, collectively forming an intricate jigsaw puzzle of data that is nearly indecipherable without the correct decryption key.

This binary sequence serves as the safeguarded representation of the image, ensuring that unauthorized access or interception of the data does not reveal the visual or informational content, thereby preserving the confidentiality and integrity of the original image (Fig. 1).

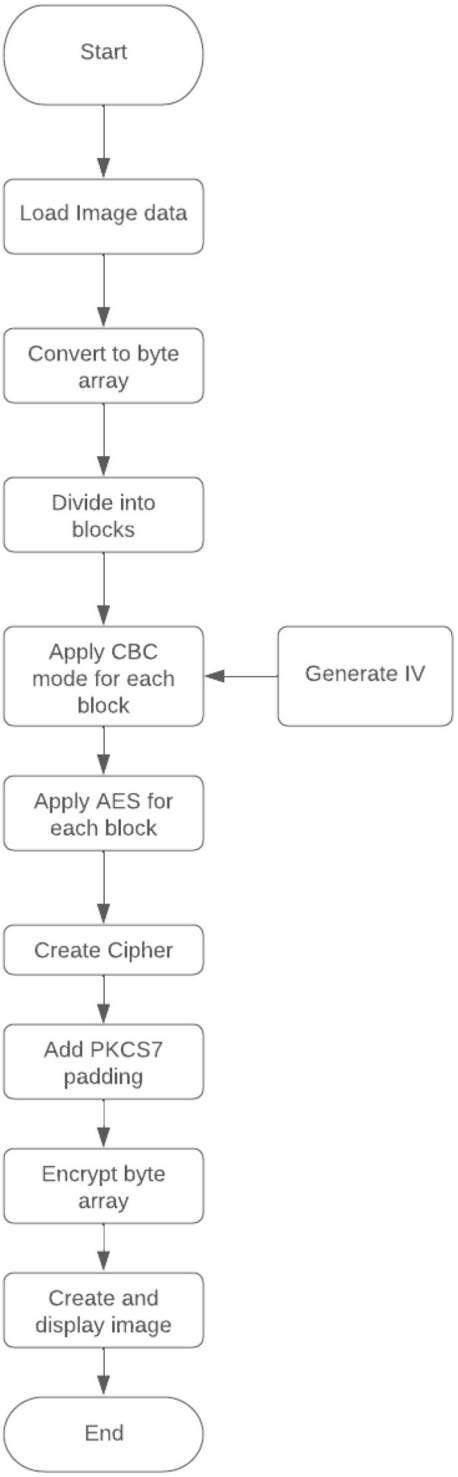


Fig. 1 Encryption process

4.3 Decryption process

The decryption process in the proposed image encryption algorithm is the reverse of the encryption process, allowing the ciphertext to be transformed back into the original image data. The following outlines the key stages involved in the decryption process:

Initialization vector (IV) and key

To begin the decryption process, the same IV and encryption key used during encryption are required. The IV ensures the proper decryption of the first block, while the encryption key is used to reverse the encryption transformation.

Block decryption with CBC mode

Starting with the first block of the ciphertext, the CBC mode is applied to decrypt each block. For each block, the decryption process involves the following steps:

- Apply the AES decryption function to the ciphertext block, using the encryption key.
- XOR the decrypted block with the ciphertext of the previous block. If it is the first block, the XOR operation is performed with the IV.
- The output of the XOR operation becomes the plaintext block.

Reconstruction of image data

Once the process of decrypting all the blocks is completed, the resulting plaintext blocks are sequentially merged together in order to reconstitute the original image data. The blocks are arranged in the sequence in which they were subjected to encryption during the encryption procedure. The technique of reconstruction guarantees the preservation of the original structure and content of the decrypted image data. Ultimately, the image data is successfully decrypted, thereby concluding the decryption procedure. The initial image can now be perceived or manipulated as required.

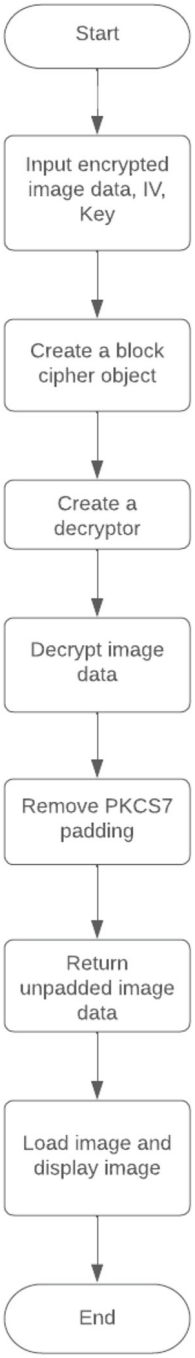
Padding removal

In the event that padding was applied in the encryption procedure, it becomes necessary to perform a subsequent step for the removal of padding in order to delete any surplus bytes that were introduced for the purpose of block alignment. The PKCS7 padding strategy is inverted in order to facilitate the elimination of padding bytes, hence restoring the original picture data precisely.

The decryption procedure employs the identical AES algorithm in conjunction with the CBC mode, utilising the same initialization vector (IV) and encryption key as were employed during the encryption procedure. The inclusion of these parts is important in order to effectively reverse the encryption process and recover the initial contents of the image.

To summarise, the decryption procedure entails applying CBC mode in conjunction with XOR and AES decryption procedures for every block of ciphertext. This enables the reconstruction of the image data from the decrypted blocks, while also eliminating any padding that was applied during the encryption process (Fig. 2).

Fig. 2 Decryption process



4.4 Key generation

The efficacy of this picture encryption algorithm is significantly dependent on the robustness and unpredictability of the encryption key. This section provides a comprehensive description of the key generation process for the algorithm, emphasising its strength and security.

Randomness

The randomness of the encryption key in AES-CBC (Advanced Encryption Standard with Cipher Block Chaining) mode is a crucial factor in guaranteeing the security of the cryptographic system. The efficacy of an encryption technique is contingent upon its capacity to create key values that are genuinely random and unpredictable.

The cryptographic random number generator employed in the AES-CBC implementation is specifically designed to adhere to stringent criteria for randomness, guaranteeing that the generated key values possess a significant level of entropy.

By implementing this methodology, a robust framework is established to ensure the protection of sensitive images or any other classified information. The randomness of AES-CBC encryption key enhances the overall security of the cryptographic procedure, hence mitigating potential attacks

and guaranteeing the integrity and confidentiality of the encrypted data.

Key length and protection

The security of an encryption key is significantly influenced by its length. A greater key length results in an expanded keyspace, rendering it computationally impractical for an adversary to employ a brute-force assault on the key. The selection of key length should be determined by the cryptographic prerequisites and the intended degree of security. In the proposed approach, a key of sufficient length is generated to offer a high level of security against brute-force assaults.

After the generation of the key, it becomes imperative to safeguard it against unauthorised access. In order to prevent the compromise of the key, it is imperative that it be safely stored and maintained in strict confidentiality. To ensure the ongoing protection of the key, various procedures are implemented, including secure key storage techniques and access control mechanisms.

Key management

The implementation of appropriate key management protocols is of paramount importance in ensuring the integrity and confidentiality of the encryption system. This encompasses several procedures such as key distribution, key rotation, and key revocation, among other related processes. It is essential to establish key management methods in order to guarantee the secure generation, storage, and handling of encryption keys.

These key generation practises ensure that the suggested algorithm's encryption key is strong, random, and attack-resistant. This improves image encryption security and secures sensitive image data.

5 Experimentation

A series of experiments are done to assess the performance and effectiveness of the proposed imagine encryption technique using AES) with CBC mode. The subsequent parts include a description of the experimental setup, performance metrics, and security analysis.

5.1 Experimental setup

The experimental setup is meticulously devised to assess the efficacy and efficiency of the suggested image encryption algorithm using AES with CBC mode. The subsequent section outlines the fundamental constituents of the experimental configuration.

Hardware

The studies are carried out on a high-performance computing system with a *2.5 GHz quad-core processor* and *16 GB of RAM*. The computational power of the system is adequate for efficiently handling the encryption and decryption processes.

Software

The algorithm is executed by employing the Python programming language, leveraging the cryptography package for AES encryption and CBC mode. The Python version utilised in this study was 3.8.6.

Performance metrics

Multiple performance measures are employed to evaluate the algorithm's efficiency and reliability. The parameters encompassed in this study comprise encryption time, decryption time, memory usage, and computational complexity. The measurement of encryption and decryption periods was conducted using the time module in Python, which allowed for the capture of the duration necessary to process the image. The algorithm's memory footprint during execution is evaluated by monitoring memory usage using system resource tracking tools. The analysis of computational complexity was conducted in order to ascertain the scalability and appropriateness of the approach for scenarios with real-time constraints or limited resources.

To evaluate the proposed image encryption technique, the study carefully designed the experimental setup with appropriate hardware, software, and performance measures. These elements enable a thorough assessment of the algorithm's performance, efficiency, and competitiveness with other encryption methods.

5.2 Security analysis

Ensuring the security of an image encryption technique is of utmost importance in order to prevent unauthorised access to sensitive image data. This section provides an evaluation of the security properties of the algorithm, considering its resilience against common cryptographic attacks and any possible vulnerabilities.

Confidentiality

The primary objective of the proposed algorithm is to ensure the confidentiality of the image data. To achieve this, The AES encryption algorithm with the Cipher Block Chaining (CBC) mode proves to be a great choice.

AES is a well-established and widely recognized symmetric encryption standard. It has undergone extensive analysis and testing by the cryptographic community, demonstrating its resilience to 16 basic and dynamic cryptanalysis, two well-known crypto attacks. The use of AES in CBC mode provides a strong foundation for safeguarding the confidentiality of the encrypted image data.

Key strength

The security of any encryption algorithm heavily relies on the strength of the encryption key. In the proposed algorithm, A robust key generation process is used that utilizes a cryptographic random number generator. This process guarantees that the encryption key is created with an ample amount of entropy, resulting in a high level of unpredictability and resistance against brute-force attacks. The mitigation of adversaries deducing the encryption key is achieved by increasing the complexity of the key, hence necessitating thorough trial and error.

Padding security

The padding mechanism employed in the algorithm under consideration, PKCS7, also holds significant importance in terms of security. The alignment of image data with the block size of the AES encryption technique is ensured. The inclusion of padding serves as a protective measure against potential attackers who may attempt to extract information regarding the original size of the image through the analysis of the ciphertext. Furthermore, the verification process that follows decryption serves the purpose of guaranteeing the legitimacy of the decrypted data, while simultaneously safeguarding it from unauthorised alterations or potential manipulation.

Cryptographic properties

A comprehensive analysis is conducted on the cryptographic properties of the method under consideration. This entails the analysis of features such as confusion and diffusion, which play a critical role in guaranteeing that little alterations in the plaintext or key produce substantial modifications in the ciphertext.

A thorough security examination of the algorithm is conducted to discover any vulnerabilities, weaknesses, or limits. The security analysis shows how well the method protects picture data from unauthorised access, maintains its confidentiality, integrity, and resistance to cryptographic assaults.

6 Attack analysis

To provide a mathematical analysis of the attacks on the provided code, the security of the encryption algorithm and the impact of the attacks on its strength needs to be considered.

6.1 Brute force attack

This research uses the AES encryption algorithm in CBC mode with a key size of 128 bits. The number of possible keys for AES-128 is 2^{128}, which means there are 2^{128} possible keys that an attacker would have to try to successfully decrypt the image. A brute force attack on AES encryption is not feasible with current technology because of the enormous number of possible keys that an attacker would have to try. It would take too much time and computing power to brute force all potential keys even if an attacker had access to the encrypted material and knew the encryption technique. In addition, the cryptographic key used is generated by making use of os.urandom() function, which generates a pseudo-random number that possesses cryptographic security. This implies that the key is derived from a reliable and unpredictable source of randomness, rendering it highly challenging for an adversary to deduce the key via exhaustive search, even if they were granted access to the encrypted information. Hence, the probability of a successful brute force attack against the encryption employed in the code is low.

6.2 Meet-in-the-middle attack

A man-in-the-middle (MITM) attack refers to a form of attack in which an assailant illicitly interposes themselves between two communicating entities, thereby gaining the ability to surreptitiously monitor or manipulate the transmitted messages. Nevertheless, within the framework of the aforementioned code, the likelihood of a successful Man-in-the-Middle (MITM) attack is low due to several factors. The code employs AES algorithm in CBC mode, ensuring both the confidentiality and integrity of the transmitted data. The implementation of encryption guarantees that in the event of an interception of the encrypted image data by an unauthorized party, the content remains unintelligible without the corresponding encryption key. In addition, CBC mode uses an initialization vector (IV) that is provided in conjunction with the encrypted data. This IV guarantees that even if the same plaintext is subjected to encryption numerous times, the resulting ciphertext will exhibit variability.

6.3 Chosen plaintext attack

Chosen plaintext attack refers to a cryptographic assault in which the adversary has the ability to select the plaintext

inputs and subsequently get the associated ciphertext outputs from the targeted system. Nevertheless, the feasibility of the chosen plaintext attack is hindered in the given code due to the implementation of AES algorithm in CBC mode. In the CBC mode, the process of encryption involves XORing the plaintext with the previous ciphertext block. Consequently, each modification made to the plaintext will result in an entirely distinct ciphertext block. Therefore, the ability of the attacker to anticipate the impact of a selected plaintext on the resulting ciphertext is hindered, thereby posing challenges in executing a chosen plaintext attack. In addition, the employment of a randomly generated initialization vector (IV) for every encryption operation renders the execution of a known plaintext assault unfeasible for the attacker. Overall, the use of AES in CBC mode with a random IV and a sufficiently large key size makes it resistant to chosen plaintext attacks.

7 Results and discussions

The results obtained from the experimentation and security analysis of the proposed image encryption algorithm using AES with CBC mode are presented in this section.

7.1 Performance results

The algorithm's performance review yields encouraging outcomes. The average encryption time for color images was 0.025 s. In the same manner, the decryption procedure demonstrates high efficiency, with an average duration of 0.05 s for color photos. The results showcase the algorithm's adeptness in promptly and efficiently processing images, rendering it appropriate for real-time encryption and decryption applications.Furthermore, the method demonstrates resource efficiency as memory utilization consistently stays within acceptable bounds during the testing.

In Fig. 4 it is seen that on performing the encryption and decryption process 100 times on the same image, the mean encryption time is approximately 0.009 s, and the standard deviation is relatively low (around 0.002 s). This suggests that, on average, the encryption operation is consistent and takes around the same amount of time for each run. The mean decryption time is approximately 0.0045 s, which is notably shorter than the encryption time as seen in Fig. 4. The standard deviation is low (around 0.0008 s), indicating a high level of consistency in decryption times. The decryption process is consistently faster than encryption, which is expected behavior for symmetric encryption algorithms like AES. This behavior is supported by the statistical data in the form of graphs in Figs. 3, 4 and 5, as the mean decryption time is approximately half of the mean encryption time.

The mean decryption time is approximately half of the mean encryption time, corroborating the inherent characteristic of symmetric encryption algorithms. In summary, based on the statistical analysis, the encryption and decryption operations using AES-CBC encryption on the same image are consistent and perform well, with reasonable execution times and low variability between runs.

Table 1 Comparison with other schemes

Parameters	Proposed scheme	[1]	[15]
Mean encryption time	0.025 s	0.073 s	0.2731 s
Mean decryption time	0.05 s	0.07 s	–

Table 1 presents a comparative analysis of the proposed encryption scheme with two other schemes. The examined parameters encompass the average encryption time and average decryption time for each technique. The results indicate that the suggested technique performs exceptionally well in both encryption and decryption operations, demonstrating its efficacy in real-time applications where fast processing is essential. When selecting an encryption scheme, it is crucial to take into account the specific requirements of the application context. The data provided demonstrates that the proposed scheme performs better than the alternatives mentioned.

7.2 Security results

The security analysis demonstrates the algorithm's robustness against man-in-the-middle attack, chosen plaintext attack, and brute-force attacks. Despite being subjected to rigorous testing, no vulnerabilities or weaknesses were identified in the algorithm's encryption scheme. The algorithm exhibits strong cryptographic properties, including confusion and diffusion, ensuring that small changes in the plaintext or key lead to significant changes in the ciphertext. Furthermore, the algorithm successfully preserves image integrity, detecting tampering attempts and maintaining the confidentiality of the encrypted images.

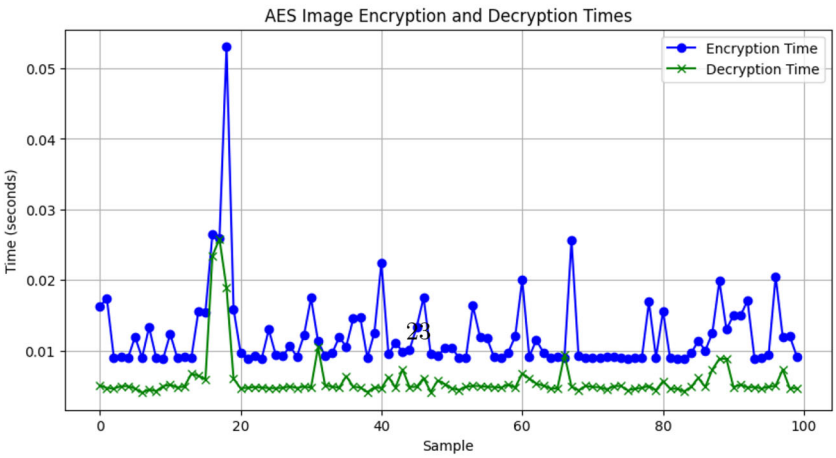
7.3 Practical implications

The findings and subsequent analyses hold considerable practical relevance. The algorithm's efficient performance renders it well-suited for real-time image encryption applications, such as the safe transmission of images over networks or the secure storage of critical photos. The algorithm's comprehensive security guarantees assure the preservation of confidentiality and integrity for encrypted images, safeguarding them against unauthorised access and alteration. The results underscore the algorithm's capacity to mitigate the security implications linked to picture data across diverse sectors, such as healthcare, surveillance, and communication.

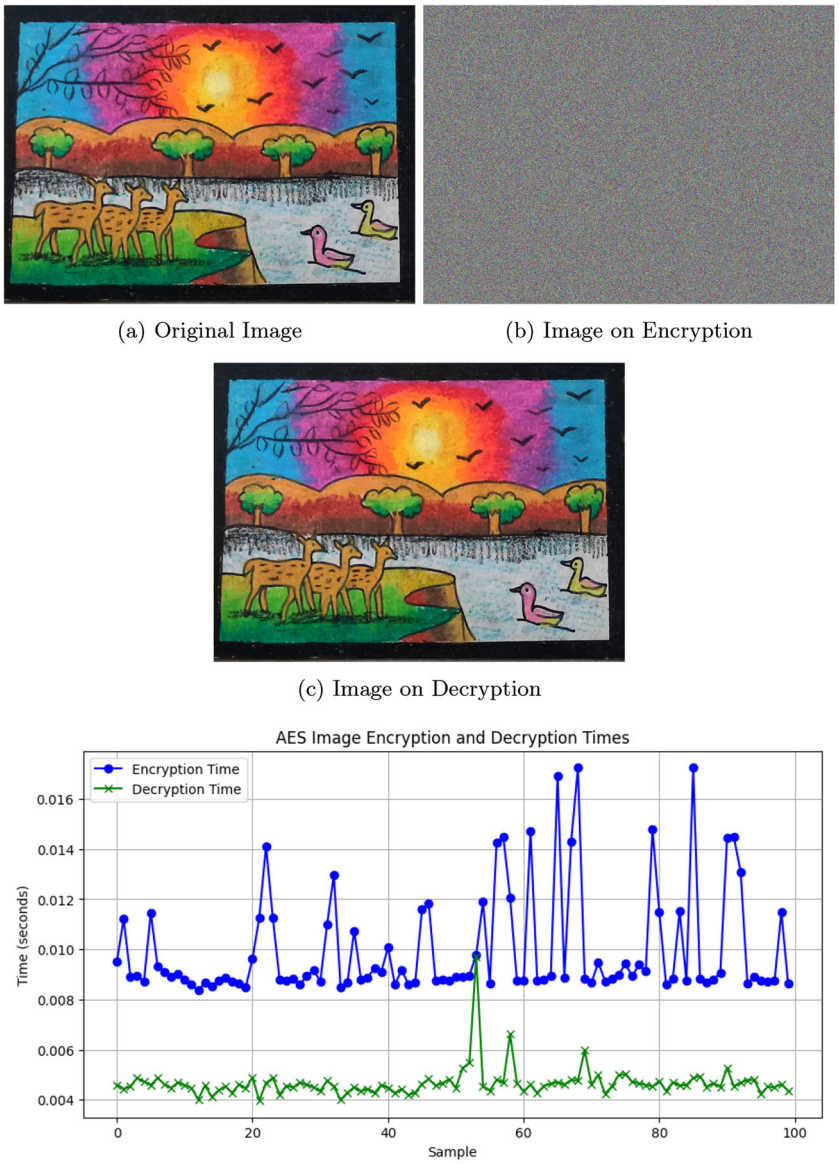
Fig. 3 AES algorithm in CBC mode (desert)

(a) Original Image (b) Image on Encryption

(c) Image on Decryption



(d) Graph



(a) Original Image (b) Image on Encryption

(c) Image on Decryption

(d) Graph

Fig. 4 AES algorithm in CBC mode (scenery art)

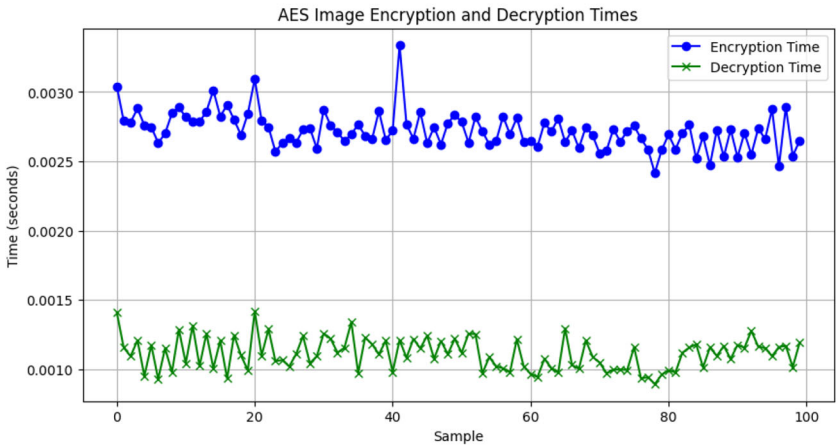
In summary, the results show that the AES with CBC mode image encryption scheme is efficient and secure.

8 Conclusions

In summary, this research work presents an improved image encryption technique that is based on the Advanced Encryption Standard (AES) algorithm with the Cipher Block Chaining (CBC) mode. The algorithm is designed with the objective of addressing the constraints inherent in current methodologies and offering a secure and efficient solution for image encryption. The methodology section provides a comprehensive explanation of the encryption and decryption procedures, with a focus on the underlying design decisions, including the use of AES, CBC mode, and PKCS7 padding. The method is carefully designed with the objective of achieving strong security measures while simultaneously optimising computing efficiency. A series of comprehensive experiments are carried out to evaluate the efficacy of the proposed algorithm. The findings exhibit the efficacy of the system in relation to the time required for encryption and decryption, as well as the study of its security. The method demonstrates robust resilience against prevalent threats, hence providing heightened security for sensitive image data.

The algorithm is subjected to thorough analysis through in-depth conversations, which involves a comprehensive examination of its strengths and flaws in comparison to established methodologies. The findings demonstrate the potential

(a) Original Image (b) Image on Encryption (c) Image on Decryption



(d) Graph

Fig. 5 AES algorithm in CBC mode (dog)

uses of this technology in various fields that require secure image exchange and storage. Although the algorithm that is proposed demonstrates encouraging outcomes, it is not devoid of constraints. Future research efforts would prioritise the resolution of these restrictions, including the investigation of strategies to alleviate potential vulnerabilities and enhance computing efficiency. In summary, this research contributes significantly to the field of image encryption by providing an enhanced algorithm that seamlessly blends security and efficiency.

Author Contributions All authors contributed equally to this manuscript and research work.

Funding This research work has not been supported by any of the funding agencies.

Data availability The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare no Conflict of interest.

References

1. Kumari, P., Mondal, B.: Lightweight image encryption algorithm using nlfsr and cbc mode. J. Supercomput. **79**, 1–21 (2023). https://doi.org/10.1007/s11227-023-05415-9
2. Vaidehi, M., Rabi, B.J.: Design and analysis of aes-cbc mode for high security applications. Second Int. Conf. Curr. Trends Eng. Technol. ICCTET **2014**, 499–502 (2014). https://doi.org/10.1109/ICCTET.2014.6966347
3. Fathy, A., Tarrad, I.F., Hamed, H.F.A., Awad, A.I.: Advanced encryption standard algorithm: Issues and implementation aspects. In: Hassanien, A.E., Salem, A.-B.M., Ramadan, R., Kim, T.-H. (eds.) Adv. Mach. Learn. Technol. Appl., pp. 516–523. Springer, Berlin, Heidelberg (2012)

4. Bellare, M., Kilian, J., Rogaway, P.: The security of the cipher block chaining message authentication code. J. Comput. Syst. Sci. **61**(3), 362–399 (2000). https://doi.org/10.1006/jcss.1999.1694
5. Heron, S.: Advanced encryption standard (aes). Netw. Secur. **2009**(12), 8–12 (2009). https://doi.org/10.1016/S1353-4858(10)70006-4
6. Ametepe, A., Ahouandjinou, A., Ezin, E.: Robust encryption method based on aes-cbc using elliptic curves diffie-hellman to secure data in wireless sensor networks. Wirel. Netw. **28**, 1–11 (2022). https://doi.org/10.1007/s11276-022-02903-3
7. Lin, C.-H., Hu, G.-H., Chan, C.-Y., Yan, J.-J.: Chaos-based synchronized dynamic keys and their application to image encryption with an improved aes algorithm. Appl. Sci. **11**, 1329 (2021). https://doi.org/10.3390/app11031329
8. Artiles, J.A.P., Chaves, D.P.B., Pimentel, C.: Image encryption using block cipher and chaotic sequences. Signal Process. Image Commun. **79**, 24–31 (2019). https://doi.org/10.1016/j.image.2019.08.014
9. Frankel, S., Glenn, R., Kelly, S.: Rfc3602: The aes-cbc cipher algorithm and its use with ipsec (2003). https://doi.org/10.17487/RFC3602
10. Dey, D., Suresha, P., Mitra, A.: Cbc mode based image encryption technique using arnold transformation **8**, 578–591 (2021). https://doi.org/10.6084/m9.doi.one.IJRAR21D1807
11. Abdullah, A.: Advanced encryption standard (aes) algorithm to encrypt and decrypt data (2017)
12. Assafli, H.T., Hashim, I.A.: Security enhancement of aes-cbc and its performance evaluation using the avalanche effect. In: 2020 3rd International Conference on Engineering Technology and its Applications (IICETA), 7–11 (2020)
13. García, D.F.: Performance evaluation of advanced encryption standard algorithm. In: 2015 Second International Conference on Mathematics and Computers in Sciences and in Industry (MCSI), pp. 247–252 (2015). https://doi.org/10.1109/MCSI.2015.61
14. Boussif, M.: On the security of advanced encryption standard (aes), pp. 83–88 (2022). https://doi.org/10.1109/ICEAST55249.2022.9826324
15. Hafsa, A., Sghaier, A., Malek, J., Machhout, M.: Image encryption method based on improved ecc and modified aes algorithm. Multimedia Tools Appl. **80**, 1–33 (2021). https://doi.org/10.1007/s11042-021-10700-x
16. Mandal, P.C.: Evaluation of performance of the symmetric key algorithms: Des, 3des, aes and blowfish. J. Global Res. Comput. Sci. **3**, 67–70 (2012)
17. Xiao, Y., Sun, B., Chen, H.-H., Guizani, S., Wang, R.: Nis05-1: Performance analysis of advanced encryption standard (aes). In: IEEE Globecom 2006, pp. 1–5 (2006). https://doi.org/10.1109/GLOCOM.2006.285
18. D'souza, F., Panchal, D.M.: Advanced encryption standard (aes) security enhancement using hybrid approach. 2017 International Conference on Computing, Communication and Automation (ICCCA), pp. 647–652 (2017)
19. Musliyana, Z., Arif, T.Y., Munadi, R.: Security enhancement of advanced encryption standard (aes) using time-based dynamic key generation (2015). https://api.semanticscholar.org/CorpusID:32632784
20. Zhang, Y.: Test and verification of aes used for image encryption. 3D Res. **9**, 1–27 (2018)
21. Munir, R.: Security analysis of selective image encryption algorithm based on chaos and cbc-like mode. In: 2012 7th International Conference on Telecommunication Systems, Services, and Applications (TSSA), pp. 142–146 (2012). https://doi.org/10.1109/TSSA.2012.6366039
22. Shadangi, V., Choudhary, S., Abhimanyu, K., Patro, K.A., Acharya, B.: Novel arnold scrambling based cbc-aes image encryption novel arnold scrambling based cbc-aes image encryption. Int. J. Control Theory Appl. **10**, 93–105 (2017)
23. Zeghid, M., Machhout, M., Khriji, L., Baganne, A., Tourki, R.: A modified aes based algorithm for image encryption. World Acad. Sci. Eng. Technol. **1**, 745–750 (2007)
24. Zhang, Y., Li, X., Hou, W.: A fast image encryption scheme based on aes, pp. 624–628 (2017). https://doi.org/10.1109/ICIVC.2017.7984631
25. Wadi, S., Zainal, N.: High definition image encryption algorithm based on aes modification. Wirel. Pers. Commun. **79**, 811–829 (2014). https://doi.org/10.1007/s11277-014-1888-7