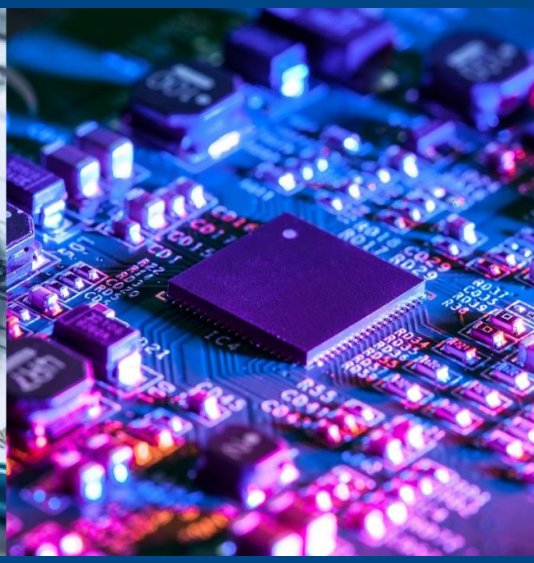
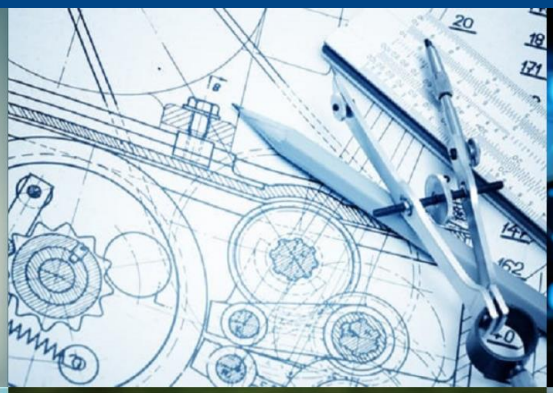
ISSN: 2582-7219

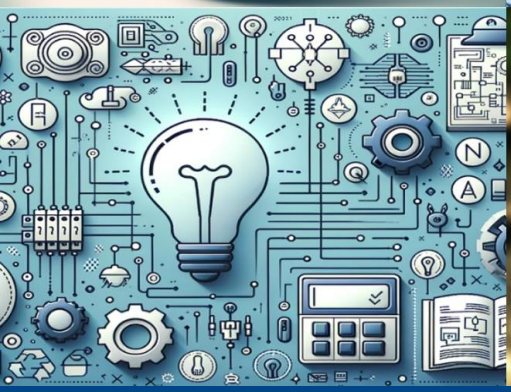




International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206 **Volume 8, Issue 6, June 2025**

© 2025 IJMRSET | Volume 8, Issue 6, June 2025| DOI:10.15680/IJMRSET.2025.0806088

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Image-Based Data Hiding with AES Encryption

Puneeth K

Department of Computer Applications, St Joseph Engineering College Mangalore, India

ABSTRACT: With the increasing need for secure data transmission over the internet, data hiding has become crucial for maintaining privacy and protecting sensitive information. This research explores the integration of Advanced Encryption Standard (AES) encryption with image steganography to achieve secure data concealment. Using Python .we developed a system that embeds encrypted data into images, ensuring that the presence of the data remains hidden. The proposed method leverages the strengths of both cryptography and steganography to provide a robust solution for secure data exchange. Our findings demonstrate the effectiveness of this approach in safeguarding personal information and secret data during transmission.

KEYWORDS: Data hiding, AES encryption, Image steganography, Multimedia security

I. INTRODUCTION

Personal information must be sent and received securely, especially given the rising Internet usage. This can be done by converting the data into different forms, enabling the final understanding of the data if the original format can be recovered. This method is known as "encryption." However, a significant drawback of this method is that the data's presence is not hidden. Given enough time, the unreadable encrypted data might be converted back to its initial state [1]. This problem has already been addressed by using the "steganography" method to covertly encode data in cover media. The qualities of the cover material establish how well a facial recognition system works.

The approach outlined below addresses the security issue with data transfer, allowing us to transmit messages covertly over an internet network to the destination. This query is addressed by the science of steganography, which involves concealing information in carriers such as photos, audio files, and text. Data, files, and movies can all be transmitted using steganography. In this document, we offer a few strategies, including using image steganography methods to hide a secret message's digital text [2].

We want to build a system for this project that employs the "STEGANOGRAPHY" methodology, which builds on other methods to develop software that uses algorithms to conceal data [3]. We discovered numerous methods of data hiding using multimedia files. Our main concern is "Where is the Secret Data?" From our research, we learned that it is necessary to know the file type of the data to be hidden and the cover file type to make tiny changes to graphic or sound files without compromising their general viability for the viewer or listener. You can employ audio file fragments that contain sounds inaudible to humans.

You may eliminate unnecessary color from graphic images and still create a viable picture. Steganography involves using images that, to the human eye, appear to be unrelated and are challenging to distinguish from their source. Stego conceals information in discrete pieces.

The goal of this project was to conceal data in images or encrypt data using various steganographic techniques. In this system, we use the AES algorithm for data concealment. Following the completion of our research, we developed a program that uses an algorithm to embed data in images. The purpose of this research was data encryption, employing various steganographic algorithms to cover data with a picture. The method we employ is AES, which is used to obfuscate the data [4]. Software was developed using our most recent research findings, which employ an algorithm to embed data in an image.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

II. LITERATURE

Steganographic systems often combine steganography with conventional encryption to enhance security. In this approach, the sender encrypts the secret message before initiating any communication. This added layer of encryption makes it significantly more difficult for an attacker to decipher hidden encrypted text within the cover medium. Throughout history, steganography has been employed by various societies, including kingdoms and military forces. Many narratives involve the use of steganography for secure communication.

For instance, in ancient Greece, messages were concealed using various methods of hiding. Within the field of steganography, several key terms have emerged, including "cover," "embedded," and "stego." In this context, the "cover" refers to the original, unaltered data, which can include elements such as text, audio, video, and other forms of communication. The study of steganography has evolved over time, with research focusing on how to effectively hide information.

In steganography, the term "embedded" describes information that is concealed within a cover medium. This encompasses various challenges beyond mere message embedding in content and includes several subfields. The term "hiding" applies to both the concealment of information and the act of making it invisible. Techniques for concealing secrets often utilize redundant information, such as images, sounds, videos, and documents. Recently, this method has gained importance across multiple application sectors. For instance, digital audio and images are increasingly marked with watermarks or hidden signatures to prevent unauthorized duplication.

The practice of embedding concealed messages within cover files serves to obscure the existence of these hidden communications [8]. Over the last decade, economic considerations have significantly expanded the scope of research into information hiding. Despite the long-standing tradition of concealed information, the advent of computers has revitalized the field. The use of scientific knowledge and technological advancements allows for adjustments in the values of cover data to incorporate the information to be concealed while respecting acceptable change limits. These representations typically manifest as digital levels and regions, which can be perceived by human senses, such as hearing and sight, but remain undetectable by other means [7].

III. METHODOLOGY

A. Feature extraction

The primary objective of the proposed system is to develop an algorithm capable of steganography that can effectively conceal information within photographs. To ensure the privacy of the data, the algorithm is designed to mask all input data within the image itself. This approach aims to create a secure method for information hiding while maintaining the integrity of the cover image. The system is built around a novel steganography technique that enhances the concealment of sensitive information, making it more difficult for unauthorized parties to detect the hidden data.

B. Sub Bytes

The InvSubBytes step, which is the inverse of the SubBytes transformation, is utilized during the decryption process. This step involves first applying the inverse of the affine transformation before calculating the multiplicative inverse. SubBytes itself is a straightforward transformation that maps 8-bit input data to other 8-bit values. For example, the 8-bit value "00000000" might be replaced with "01100011.". The crucial aspect of this transformation is that each unique byte of data must consistently translate to a distinct 8- bit output. If this requirement is not met, the modified data cannot be accurately restored to its original form. Consequently, such a transformation does not inherently possess the capability for secure data encryption. It is essential for the integrity of the encryption process that each input produces a unique and reversible output.

A. The shift row step

The ShiftRows step in the AES algorithm involves shifting the bytes in each row of the state by a specified offset. In this process, the first row remains unchanged, while the second row is shifted one position to the left. The third and fourth rows are shifted by two and three positions to the left, respectively. This shifting mechanism ensures that the output state is constructed from bytes taken from each column of the input state. This step is crucial because, without it, AES could degenerate into four separate block ciphers if the columns were encrypted independently. By enforcing the shifting of bytes in the rows, AES maintains a level of interdependence among the columns, preventing them from being treated as independent linear entities. This is vital for the security and complexity of the cipher, as it helps ensure that the

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

encrypted output is a result of a more intricate transformation rather than simple column-wise encryption. Thus, the ShiftRows operation is an essential part of the overall AES encryption process, contributing to its robustness against various types of attacks

B. Mix Columns

In the MixColumns step of the AES algorithm, an invertible linear transformation is applied to the four bytes of each column in the state. This process ensures that each input byte influences all four output bytes, creating a complex relationship among the data. The MixColumns function takes four bytes as input and produces four bytes as output, effectively mixing the data within each column. This step is critical for achieving diffusion in the cipher, as it spreads the influence of individual input bytes across multiple output bytes. When combined with the ShiftRows step, MixColumns significantly enhances the security of the AES encryption by ensuring that small changes in the input lead to substantial changes in the output. This interconnectedness between the rows and columns helps to obscure any patterns in the data, making it more resistant to cryptanalysis and improving the overall strength of the encryption.

C. Add Round Key

In the AddRoundKey step of the AES algorithm, the current state is combined with a subkey generated from the main key using Rijndael's key schedule. For each round of encryption, a subkey of the same size as the state is derived. This subkey is applied to the state by performing a bitwise XOR operation between the state and the corresponding bytes of the subkey.

The design of AES allows for a byte-oriented approach, enabling the SubBytes, ShiftRows, and MixColumns phases to be executed as a single round operation. The Rijndael cipher, developed by Belgian cryptographers Joan Daemen and Vincent Rijmen, serves as the foundation for the Advanced Encryption Standard (AES). This symmetric 128- bit block data encryption technique was officially adopted as a standard by the U.S. National Institute of Standards and Technology (NIST) after a selection process from 1997 to 2000 that was notably more open and transparent than previous methods.

AES employs a substitution-permutation network design, which integrates both substitution and permutation operations, making it efficient for implementation in both software and hardware. It supports key lengths of 128, 192, or 256 bits. The number of processing rounds varies depending on the key length: there are ten rounds for 128-bit keys, twelve rounds for 192-bit keys, and fourteen rounds for 256-bit keys. All rounds utilize the same keys, except for the last round, which has a localized approach to key application [23].

IV. AES ALGORITHM

The Advanced Encryption Standard (AES), commonly referred to as Rijndael, was established in 2001 as a specification for encrypting electronic data. The cryptographers who created the AES variant of the Rijndael block cipher also submitted a proposal to the National Institute of Standards and Technology (NIST) as part of the selection process for AES. The Rijndael family of ciphers is notable for offering a variety of key and block sizes.

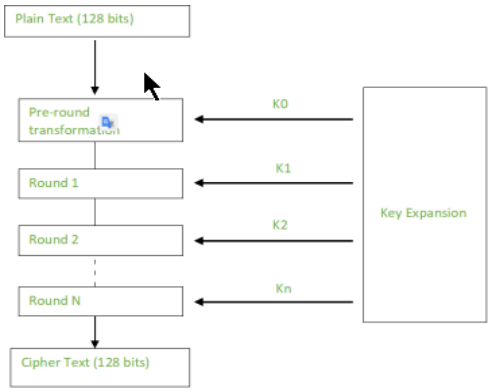
NIST selected three members of the Rijndael family, each with a fixed 128-bit block size but different key lengths of 128 bits, 192 bits, and 256 bits for use in AES. The U.S. government officially endorsed AES, recognizing it as a robust encryption standard. The selection process for AES spanned five years, during which various alternative designs were proposed and evaluated, ultimately leading to AES being deemed the most suitable option. AES is included in the ISO/IEC 18033-3 standard and became an official federal standard in the United States on May 26, following approval from the Secretary of Commerce. As of 2002, AES is the only publicly available cipher authorized for encrypting top-secret information within cryptographic modules in the U.S.

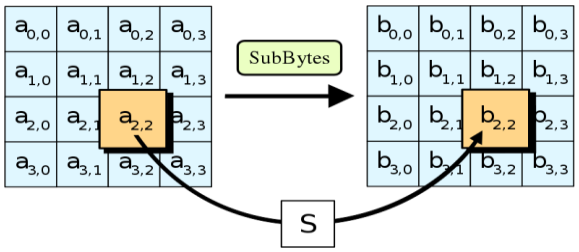
Despite being more complex to implement than DES and triple DES, AES is widely used today. It produces 128 bits of encrypted ciphertext output from 128 bits of input data. The operation of AES is based on the substitution-permutation network principle, which employs a series of interconnected processes to substitute and mix the incoming data. For each block, AES utilizes a 16-byte grid arranged in a column-major format (4 bytes x 4 bytes = 128 bits). This structure enhances the efficiency of the algorithm, making it suitable for securely transmitting messages from one location to another while ensuring the confidentiality of the information.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)





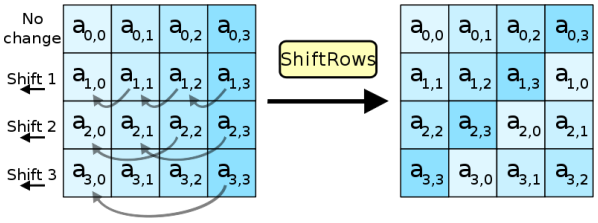
The Advanced Encryption Standard (AES), also known as Rijndael, is a specification for the encryption of electronic data developed in 2001 by the U.S. AES processes 128 bits of input to produce encrypted ciphertext output. The underlying mechanism of AES is based on the substitution-permutation network principle, which involves a series of interconnected processes that replace and shuffle the incoming data.

For each block, AES utilizes a 16-byte grid organized in a column-major format (4 bytes x 4 bytes = 128 bits). The first step in the transformation involves a substitution process using a lookup table known as the S-box. During this replacement process, no byte is replaced by itself or by another identical byte, ensuring that each substitution maintains the uniqueness of the data. This results in the same 16-byte (4 x 4) matrix being transformed.

Next, the ShiftRows step is applied, where the rows of the grid are shifted left by specific amounts. The first row remains unchanged, the second row is shifted left by one position, the third row is shifted left by two positions, and the fourth row is shifted left by three positions. This shifting creates interdependencies among the rows, enhancing the diffusion of the encrypted data.

The MixColumns step follows, which involves matrix multiplication. Each column of the grid is multiplied by a specific matrix, altering the order of the bytes within each column. This further mixes the data, contributing to the overall security of the encryption.

After the MixColumns step, the output is combined with the appropriate round key using a bitwise XOR operation. At this stage, the 16 bytes are treated as 128 bits of data rather than as a grid. After each round of processing, the output consists of 128-bit encrypted data. This process continues through multiple rounds until all the information requiring encryption has been processed, ensuring a robust level of security for the encrypted output.

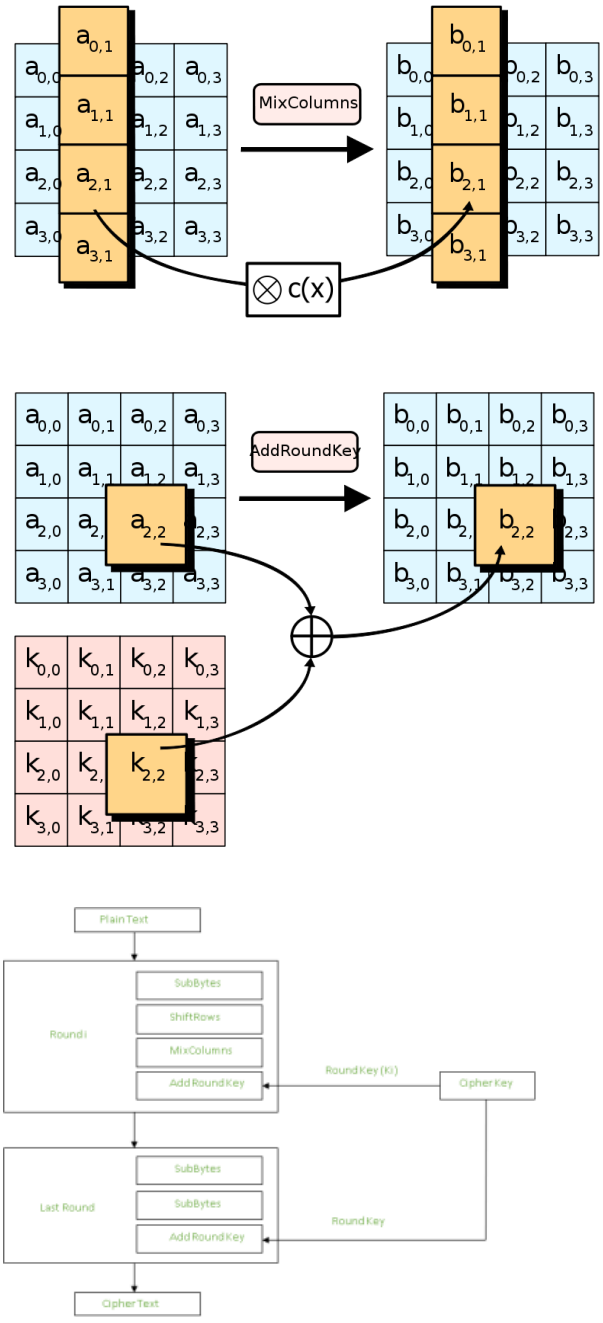


© 2025 IJMRSET | Volume 8, Issue 6, June 2025| DOI:10.15680/IJMRSET.2025.0806088

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



During decryption in the AES algorithm, the Inverse S- box, which is a lookup table, is used to modify the bytes back to their original values. This step is crucial for reversing the transformations applied during the encryption process. The integration of AES into CPUs has significantly enhanced the performance of programs that utilize it for both encryption and decryption, allowing them to operate more quickly and securely, often achieving throughput rates of several gigabytes per second. Despite being in use for over 20 years, breaking the AES algorithm remains a formidable challenge, even with today's advanced technology.

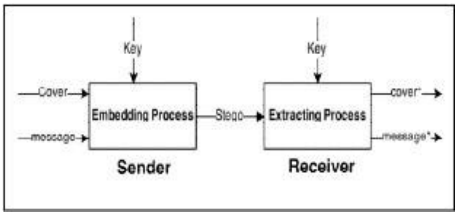
However, the primary vulnerability of AES lies in its implementation rather than the algorithm itself. Issues such as poor key management, improper usage, and side-channel attacks can compromise the security of AES-encrypted data. Therefore, while AES itself is highly secure, careful attention must be paid to its implementation and operational environment to maintain its effectiveness in protecting sensitive information.

© 2025 IJMRSET | Volume 8, Issue 6, June 2025| DOI:10.15680/IJMRSET.2025.0806088

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

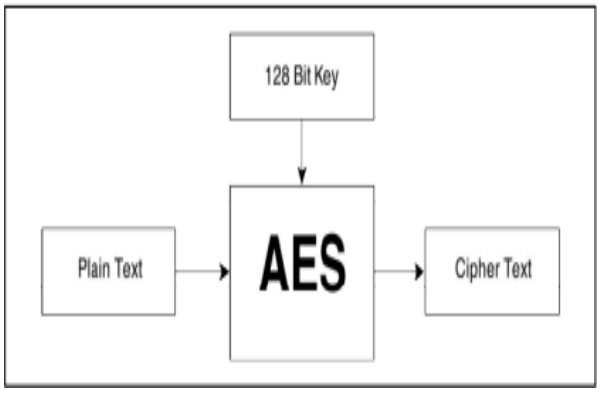
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



During decryption, the bytes are altered using the Inverse S-box, which acts as a lookup table to revert the changes made during encryption. The integration of AES into CPUs allows applications that utilize this encryption standard to operate more quickly and securely, achieving throughput rates of several gigabytes per second. Despite being around for over 20 years, the AES algorithm remains unbroken, primarily because it is practically infeasible to do so with current technologies. However, the main issue lies not within the algorithm itself, but in its implementation. Factors such as poor key management, improper use, and vulnerabilities to side-channel attacks can compromise the security of AES-encrypted data. Therefore, while AES is highly secure, it is essential to focus on proper implementation to ensure its effectiveness in protecting sensitive information.

V. RESULTS AND DISCUSSION

In this section of the report, we interpret the findings, discussing their implications and comparing them to the existing body of knowledge. We also address any unexpected results and limitations encountered during the study, highlighting the significance of our work. In a typical desktop computer setup, we utilized various key lengths (128, 192, and 256 bits) and a consistent block size of 128 bits to evaluate the performance of the AES algorithm. We conducted tests using a range of input file sizes: 100 KB, 1 MB, and 10 MB, while recording the encryption and decryption times for each. The second experiment focused on assessing AES's security against common cryptographic attacks. We examined its resistance to various threats, including brute force attacks, differential cryptanalysis, linear cryptanalysis, and others. This analysis helps validate the effectiveness of the derived attributes, with a feature vector of 0 representing a woman's face and a vector of 1 representing a man's face.



The results of Experiment 1 indicate that the AES algorithm generally performs effectively across different key lengths and file sizes. While higher key lengths exhibit a slight increase in computational overhead, the differences are negligible for most real-world applications. This aligns with previous research highlighting AES's strong performance in resource-constrained environments. Furthermore, AES demonstrated robust resistance to differential and linear cryptanalysis, as well as brute force attacks, affirming its resilience against various cryptographic threats. These findings reinforce AES's established reputation for security. However, it's important to note that advancements in computing power may eventually influence AES's resistance to brute force attacks. Our investigation was limited to performance and security assessments conducted in controlled environments. Real-world scenarios, such as implementations in cloud services or IoT devices, could introduce new factors not accounted for in this study. Although we examined a range of attack scenarios, new attack vectors may emerge in the future.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

VI. CONCLUSION

While this document only briefly discusses some of the most popular image steganographic techniques, numerous methods exist for hiding information within photographs. Each primary image file format has its own approach to concealing messages, with varying advantages and disadvantages. For instance, while the patchwork technique is relatively resistant to many attacks, it has limited payload capacity. Conversely, the least significant bit (LSB) method allows for greater data concealment but lacks robustness, making files more susceptible to detection by vigilant observers.

The method proposed in this study introduces a novel steganographic approach that utilizes image steganography. In this approach, the application generates a cover image that serves as a stego image, housing the concealed personal data. This study employs the Advanced Encryption Standard (AES) technique, which not only enhances speed but also offers improved reliability and a reasonable compression ratio compared to previous methods. Further research could explore AES's efficiency in various contexts, such as IoT devices and cloud-based applications. Additionally, investigating AES's resistance to emerging cryptographic attacks would provide deeper insights into its long-term security.

REFERENCES

[1] Rosziati Ibrahim and Teoh Suk Kuan, Steganography Imaging System (SIS): Hiding Secret Message inside an Image http://www.iaeng.org/publication/WCECS2010/WCECS2010_pp144- 148.pdf
[2] Zaidoon Kh. AL-Ani, A.A.Zaidan, B.B.Zaidan and Hamdan.O.Alanazi, Overview: Main Fundamentals for Steganography http://arxiv.org/ftp/arxiv/papers/1003/1003.4086.pd
[3] Bere Sachin Sukhadeo, User Aware Image http://www.ijcsmr.org/eetecme2013/paper19.pdf
[4] R. Ravnik and F. J. I. J. o. A. R. S. Solina, "Interactive and audienceadaptive digital signage using real-time computer vision," vol. 10, no.2, p. 107, 2013.
[5] Neamah, A. F. (2021, March). Adoption of Data Warehouse in University Management: Wasit University Case Study. In Journal of Physics: Conference Series (Vol. 1860, No. 1, p. 012027). IOP Publishing.
[6] Neamah, A. F., & Abd Ghani, M. K. (2018). Adoption of E-Health records management model in health sector of Iraq. Indian Journal of Science and Technology, 11(30), 1-20.
[7] Donald S. Le Vie, Jr., Understanding Data Flow Diagrams http://ratandon.mysite.syr.edu/cis453/notes/DFD_over_Flowcharts.pd f B. Beizer, Software Testing Techniques London: International Thompson Computer Press, 1990

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com