Kalpa Publications in Computing

Volume 19, 2024, Pages 394–409

Proceedings of 6th International Conference
on Smart Systems and Inventive Technology

SECURE IMAGE MESSAGING PLATFORM UTILIZING IMAGE STEGANOGRAPHY AND AES ENCRYPTION

S. Aaren Sherwin[1], E. Immanuel Simeon[2], C.A.Subasini[2], and Adlin Sheeba[4]

[1] St. Joseph's Institute of Technology, Chennai, India
aarensherwin007@gmail.com
[2] St. Joseph's Institute of Technology, Chennai, India
imma05simeon@gmail.com
[3] St. Joseph's Institute of Technology, Chennai, India
subasiniaji@gmail.com
[4] St. Joseph's Institute of Technology, Chennai, India
adlinsheeba78@gmail.com

Abstract

This paper introduces a robust web application for secure image transmission, integrating AES encryption and hashed data to safeguard both the confidentiality and integrity of shared images. Employing cyberblock chaining mode enhances the encryption process, fortifying data protection. The sender module employs AES encryption, while the receiver module decrypts and displays the images securely. User authentication ensures the integrity of data transfer, while a user-friendly inter- face streamlines the image transmission process. This application finds relevance across diverse sectors including healthcare, legal, and corporate environments where image confidentiality and data integrity are paramount.
Index Terms—Image encryption, Advanced encryption standard, hashed data, secure image transmission, web application, Steganog- raphy, security, confidentiality.

1 Introduction

In the era of digital communication, ensuring secure trans- mission of images is vital. This project introduces a web application module using the Advanced Encryption Standard (AES) algorithm for image encryption. Hashed data integra- tion ensures integrity and confidentiality. With sender and receiver modules, it offers a comprehensive and secure image transmission process. The solution, applicable in healthcare, legal, and business sectors, prioritizes user authentication for added security. The user-friendly interface allows easy uploading, encrypting, sending, receiving, and decrypting of images, catering to users from various backgrounds. This project addresses challenges in secure image transmission, emphasizing data security and user authentication through a practical and accessible web application

2 Related Works

Arun Kumar Rai et al. (2023)[1]: Innovative approach combining reversible data hiding and encryption. Blockwise encryption and a two-layer embedding scheme enhance data capacity. Three-phase process includes cover image encryp- tion, two-layer data embedding, and stego image deployment via blockchain.

Yin, Ji, and Luo (2020)[2]: Published in IEEE Transactions on Circuits and Systems for Video Technology (August 2020), explores reversible data hiding in JPEG images. Utilizes multi-objective optimization for concealing data while maintaining reversibility.

Hou, Ou, Tian, and Qin (2021)[3]: Featured in Signal Processing: Image Communication (March 2021), introduces reversible data hiding with neural networks. Combines mul- tiple histograms modification with deep neural networks for advancements in concealing data with a focus on reversibility.

Chen, Sun, Li, Chang, and Wang (2020)[4]: Published in the Journal of Information Security Applications (October 2020), proposes an efficient general data hiding scheme. Focuses on versatile approaches for data hiding methods, particularly centered on image interpolation.

Hassan and Gutub (2021)[5]: Presented in the Arabian Journal of Science and Engineering (March 2021), introduces an efficient image reversible data hiding technique. Empha- sizes interpolation optimization, representing a significant step forward in embedding data within images with an emphasis on optimization[6-15].

3 Proposed Methodology

3.1 Problem Definition

In today's digital era, secure transmission of sensitive images is a key concern. This project introduces a web application module for the secure exchange of images, relying on the Advanced Encryption Standard (AES) algorithm and hashed data for confidentiality and integrity. It comprises a sender module for AES encryption and a receiver module for decryption and data visualization.

The AES algorithm ensures image confidentiality, complemented by hashed data for data integrity. The web application prioritizes secure data transfer with user authentication. Its user-friendly interface streamlines image operations, facilitating easy upload, encryption, and transmission, as well as seamless decryption and data viewing.

3.2 Proposed System

The proposed system aims to establish a secure image trans- mission platform that incorporates the Advanced Encryption Standard (AES) for image encryption, safeguarding images during transmission. A distinctive feature of this system is the hidden embedding of text data within the images, which is encrypted alongside the image content. The system includes sender and receiver modules, with the sender responsible for encrypting images and embedding text data, while the receiver decrypts the images and extracts the hidden text which can run inside

the web applications. This innovative approach not only ensures image confidentiality and data integrity but also allows for the secure sharing of additional information. The system's versatility makes it applicable across various sectors, such as healthcare, legal, and corporate environments, where secure image transmission and hidden data are essential for maintaining data security and authenticity.

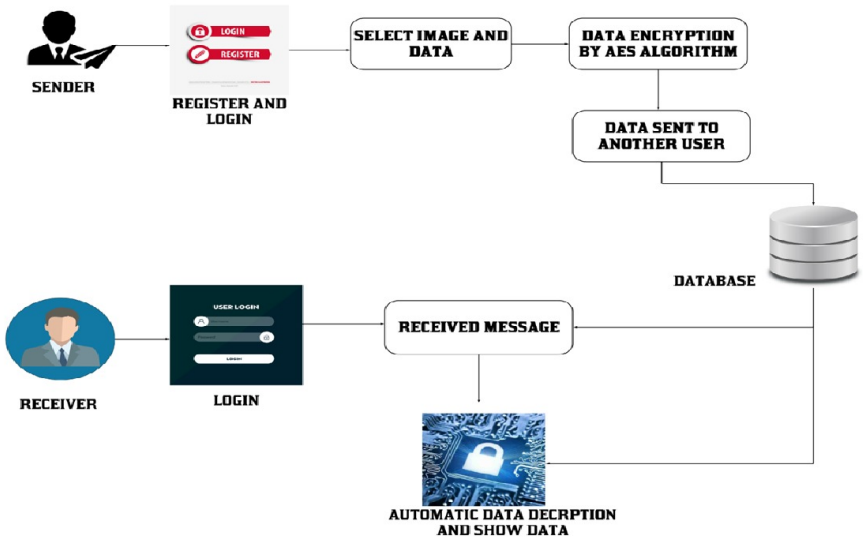

Figure 1: Working diagram of the Proposed System

Fig. 1. shows the working flow of the proposed system which includes collecting data from the user using front-end such as username, password, text data, image data,.etc. The data then is encrypted and embedded inside the image chosen by the user, then the data is stored in a database and retrieved by appropriate receiver by entering their user credentials.

In the domain of data security, the Advanced Encryption Standard (AES) algorithm marks a significant leap. Tailored for electronic data encryption, AES surpasses predecessors like DES and Triple DES. Operating on bytes, AES deploys operations like SubBytes, ShiftRows, MixColumns, and Add Round Key. Adaptable with key sizes of 128 or 192 or 256 bits, AES finds application in various sectors, ensuring secure data transmission across domains.

Basic Unit of AES: In the Advanced Encryption Standard (AES), the fundamental unit is the byte. AES operates on fixed-size blocks, specifically 128 bits or 16 bytes. Each block is treated as a 4x4 matrix of bytes, forming the foundational structure for encryption and decryption processes. The byte-level granularity allows AES to process data in a systematic manner, enhancing its effectiveness in securing electronic information.

Entry Mechanisms and Data Entry: AES utilizes a 4x4 matrix for data organization during encryption and decryption, undergoing transformations in rounds. Input data, usually blocks, undergoes byte-wise processing, exemplified in the code by treating each block as a 16-byte grid in a column-major arrangement.

Uniqueness: AES stands out for robustness and security, surpassing DES and triple-DES. Its unique ability lies in providing high security while efficiently processing fixed-size blocks. Variable key sizes of 128, 192, or 256 bits enhance adaptability and resilience against cryptography attacks.

Utilization and Flexibility: Extensively used for encrypting electronic data, AES's flexibility in the code accommodates key sizes of 128, 192, or 256 bits, catering to diverse security requirements. The negotiation-permutation network principle in the code showcases AES's flexibility in various applications, seen in sectors like healthcare, legal, and commercial environments.

Decryption: AES's decryption process reverses encryption stages using operations like Inverse MixColumns, ShiftRows, Inverse SubBytes, and AddRoundKey. This bidirectional capability ensures secure information exchange while maintaining confidentiality.

In conclusion, the code's integration of AES demonstrates its application in secure image transmission, emphasizing the algorithm's significance in protecting sensitive data across various domains.

The security issues that are considered here:In the existing system, security issues include computationally intensive encryption, susceptibility to cryptographic attacks, and compatibility challenges. These drawbacks may lead to processing delays, potential vulnerabilities, and inconvenience for users. The proposed system addresses these issues by leveraging the Advanced Encryption Standard (AES) for robust image encryption and introducing a unique approach of embedding encrypted text within images. This ensures strong encryption, data integrity, and compatibility across various industries. The advantages of the proposed system include enhanced security during image transmission, making it suitable for healthcare, legal, and corporate environments where data security is paramount. Overall, the proposed system significantly improves upon the security considerations of the existing system by adopting advanced encryption techniques and innovative approaches to secure image transmission.

3.3 Advanced Encryption Standard (AES):

The Advanced Encryption Standard (AES) stands as a cornerstone in contemporary cryptography algorithms, meticulously designed and endorsed by the U.S. National Institute of Standards and Technology (NIST) in 2001. As an internationally recognized encryption specification, AES serves as a linchpin for ensuring the confidentiality and integrity of electronic data across diverse applications. AES operates as a block cipher, transforming data in fixed-size 128-bit blocks to meet contemporary security demands. This block-wise encryption enhances security and computational efficiency. Key Size Diversity: AES supports key sizes of 128, 192, and 256 bits, empowering organizations to tailor encryption strategies based on specific security requirements for nuanced data protection. Block-wise Encryption: AES processes data in 16-byte grid blocks, enhancing efficiency for modern cryptography applications where secure and streamlined processing is crucial.

Fig. 2. Shows the processing of single data using Advanced Encryption Standard. As depicted in the picture a single data is processed through these levels to achieve a fully encrypted cipher text.

AES encryption of single data: AES's operational methodology emphasizes byte-wise data pro-

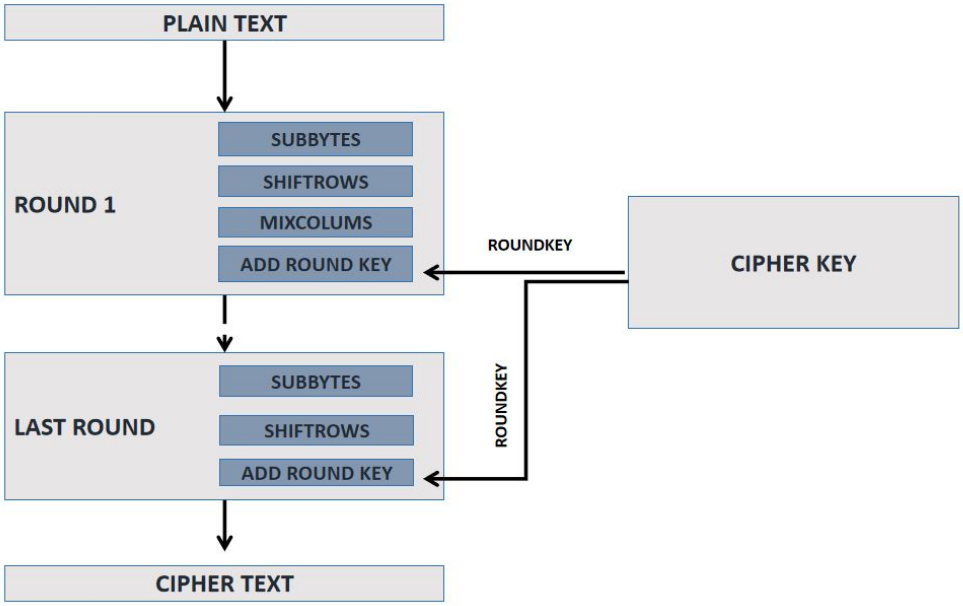

Figure 2: AES Encryption of a single data

cessing in each encryption round, ranging from 10 to 14 rounds based on key length, ensuring secure encryption with crucial steps:

- **SubBytes**: In the SubBytes step, each byte undergoes substitution using an S-box. This introduces non-linearity and complexity to the encryption process, fortifying against linear cryptography attacks. For example: Original Byte: 0xA4, Substituted Byte (from the S-box): 0x8C.

- **ShiftRows**: The ShiftRows step involves circular shifting of each row, contributing to confusion in the data structure. Rows are shifted by different offsets, adding an additional layer of permutation.

- **MixColumns**: In the MixColumns step, matrix multiplication is performed on each column, reshaping the byte positions. This step enhances diffusion and ensures that each byte in the output depends on multiple input bytes.

$$\text{ResultColumn} = \text{MixMatrix} \times \text{OriginalColumn}$$

  This particular step involves matrix multiplication. Each column undergoes multiplication with a designated matrix, leading to a change in the position of every byte within the column. It's important to note that this step is omitted in the final round.

- **AddRoundKey**: The AddRoundKey step involves XORing the state with a round key. Each round key is derived from the original encryption key. This step adds an additional layer of confusion and ensures that each bit in the state depends on the corresponding bit in the key.

The strength and resilience of AES lie in its ability to adapt to diverse key sizes, rendering it a stalwart defense against cryptography attacks and ensuring the confidentiality of sensitive information through a combination of substitution, permutation, and diffusion operations. The meticulous orchestration of these steps fortifies AES as a robust and versatile encryption standard. Following a series of iterations, the output consists of 128 bits of encrypted data. This repetitive process is carried out until the entire dataset undergoes the encryption procedure.

Reverse Operational Dynamics of AES (Decryption): The decryption process in AES follows a reverse order of the encryption steps. Similar to encryption, the decryption process consists of several rounds, and the number of rounds depends on the key length. The key steps in the decryption process are as follows:

- **AddRoundKey**: In AES decryption, the process begins by XORing the ciphertext with the last round key used in encryption. This operation cancels out the effect of the corresponding round key used during encryption. The round keys are derived from the original encryption key using a key schedule algorithm. By XORing with the last round key, the ciphertext is "unmixed" with the key, revealing intermediate decrypted data.
- **InverseMixColumns**: The InverseMixColumns step undoes the MixColumns operation performed during encryption. In encryption, each column of the state matrix undergoes a matrix multiplication operation with a fixed matrix. In decryption, this operation is reversed by applying the inverse matrix multiplication, effectively "unmixing" the columns and restoring them to their original state.
- **InverseShiftRows**: During encryption, the ShiftRows step cyclically shifts the rows of the state matrix to introduce diffusion and permutation. In decryption, the InverseShiftRows step reverses these shifts by moving the bytes of each row in the opposite direction. This step is crucial for restoring the original data structure before the final decryption step.
- **InverseSubBytes**: In encryption, the SubBytes step substitutes each byte of the state matrix using a fixed S-box. During decryption, the InverseSubBytes step applies the inverse of this substitution process. Each byte is mapped back to its original value using the inverse S-box, effectively reversing the byte-level substitutions performed during encryption.
- **Key Expansion (if applicable)**: If the original encryption key was expanded to generate round keys, the decryption process may involve using these round keys in reverse order. This step is necessary to cancel out the effects of the key expansion during encryption and to ensure proper decryption of each round.

AES Encryption in (Cyber Block Chaining Mode): AES Encryption with Cipher Block Chaining Mode with respect to Proposed System:

- **Initialization Vector (IV) Generation:** Before encrypting the plaintext message, a random Initialization Vector (IV) is generated. This IV is a unique and random value that is crucial for introducing variability into the encryption process.
- **Padding:** The plaintext message is padded to ensure that its length aligns with the block size used by the AES algorithm. Padding is necessary to accommodate messages that are not an exact multiple of the block size.

- **XOR Operation with IV:** The padded plaintext is processed in blocks. In the first block, each byte is XORed with the corresponding byte of the IV. This XOR operation introduces an additional layer of randomness, making sure that even if the same message is encrypted multiple times with the same key, the results will differ due to the changing IV.

- **AES Encryption in CBC Mode:** The XORed result from the previous step is then subjected to AES encryption in Cipher Block Chaining (CBC) mode. In each subsequent block, the ciphertext from the previous block is XORed with the plaintext before encryption. This chaining mechanism ensures that each block's encryption is influenced by the ciphertext of the previous block, contributing to the overall security of the encryption.

- **Base64 Encoding:** The final encrypted result, consisting of the IV and the ciphertext, is Base64-encoded. This encoding transforms the binary data into a string of ASCII characters, making it suitable for transmission or storage.

- **Sending the Encrypted Result:** The Base64-encoded result is now ready for transmission or storage. The IV and the encrypted text together form a secure and unique representation of the original plaintext.

  In essence, Cipher Block Chaining (CBC) mode is applied during the AES encryption step, introducing chaining and dependency between blocks. This chaining mechanism, along with the Initialization Vector, significantly enhances the security and unpredictability of the encrypted output.

3.4 Implementation

Client Registration Module: The client registers by providing a title, individual ID, cellphone number, password, and confirming the password.

Client Login Module: The client logs in using their registered user ID and password. Authentication is performed by checking the entered credentials against the saved information in the database. If the credentials are correct, the user is redirected to the data page; otherwise, an error message is displayed.

Registration details are stored in the SQLite database, assuming the client ID is unique. If the individual ID is already registered, an error message is displayed.

Input Page for Sending a Secret Message: The client navigates to the input page to send a secret message. The client provides a message and uploads an image. The message is encrypted using the AES algorithm with a predetermined key. The encrypted message is embedded in the image using steganography. The resulting image, now containing the hidden encrypted message, is saved.

Encryption and Steganography module of the block diagram: The following occurs when a user moves to the input page and gives a message as well as an image:

Fig. 3. depicts the block diagram of the proposed system. It shows the flow of operation or function through each modules.

- **Step 1. Binary Conversion:** A binary conversion is done after the user inputs a value.

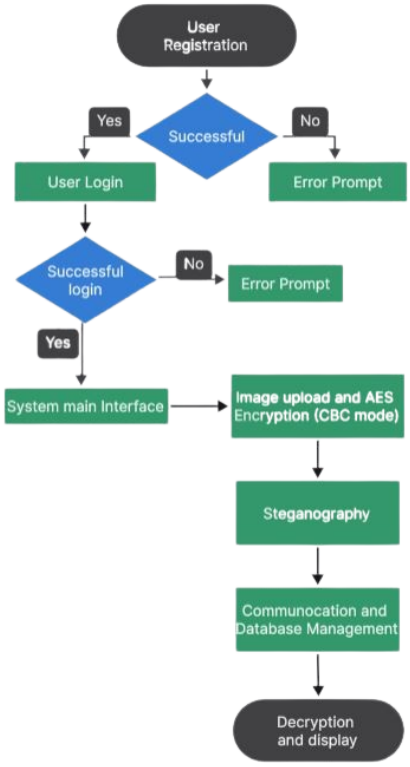

Figure 3: Block Diagram of the Proposed System

- **Step 2. Encryption (Advanced Encryption Standard Algorithm):** AES algorithm with a fixed key is used for encryption of the message. The encrypted message is then converted into binary format. Each character in the message is represented as a sequence of eight bits (binary digits).
- **Step 3. Steganography Image Upload:** The user uploads an image. The image can be in various formats, such as PNG, JPEG, and many more.
- **Step 4. Image Processing Binary Representation of Image Pixels:** Pixel values of the image are read by the steganography module. Each color channel (red, green, blue, and alpha) has an 8-bit value, making it a total of 32 bits for each pixel assuming there are 8 bits in one channel. **XOR Operation:** Each pixel's binary representation on the image undergoes XOR operation with its corresponding bits in the binary message. If it is zero in the binary message bit, then the bit in the pixel remains unchanged. If it is one in the binary message bit, then the pixel bit gets inverted (XORed with 1).
- **Step 5. Final Picture:** The XOR operation modifies the pixel values of a photo to embed a disguised message in it. The resulting picture now has data that cannot be seen.
- **Step 6. Image Saving:** This renamed image contains the encrypted information and is saved in a format inside the uploads folder.

- **Step 7. Sending Results:** The photograph altered with hidden encryption can now be transmitted to the recipient as planned.

3.5 Performance Analysis

Advanced Encryption Standard (AES): Below give the performance analysis of AES vs other encryption techniques by considering various aspects and attributes such as speed of the encryption process (in millisecond) and Versality comparison.

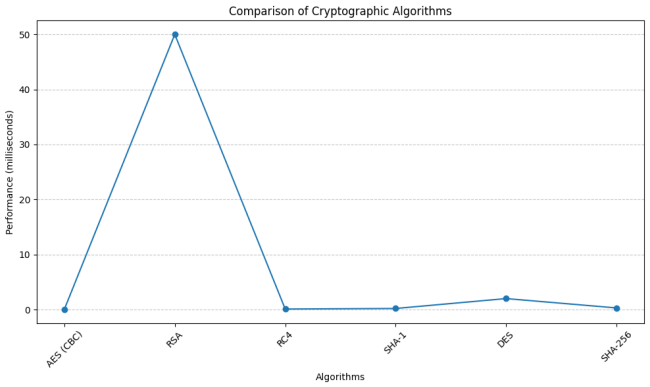

Figure 4: Performance graph

Fig. 4. depicts the comparison of AES vs other widely used encryption techniques in terms of performance (speed of execution).

- **Axis**:
  - The y-axis represents algorithm performance in microseconds, ranging from 0 to 50 microseconds.
  - The x-axis displays different cryptographic algorithms, including AES (CBC), RSA, RC4, SHA-1, DES, and SHA-256.
- **Efficiency**:
  - AES demonstrates exceptional efficiency, with a performance below 10 microseconds. This highlights AES's speed, crucial for real-time or near-real-time applications.
- **Consistency**:
  - Unlike RSA, which exhibits a significant peak, AES maintains consistent performance without major spikes. This predictability is advantageous for ensuring system reliability.
- **Comparison**:
  - **Versus RSA**: AES's performance is notably faster than RSA, operating under 10 microseconds compared to RSA's peak at 50 microseconds, making AES preferable in many applications.

- **Versus RC4**: Both AES and RC4 operate under 10 microseconds. However, AES is preferred for its robust security and widespread adoption.
- **Versus SHA-1**: AES's performance is comparable to SHA-1, another cryptographic hash function, both operating under 10 microseconds. However, they serve different purposes, with AES being an encryption algorithm.
- **Versus DES**: AES demonstrates similar performance to DES, an older encryption standard. However, AES is favored for its stronger security and capability to handle larger key sizes.
- **Versus SHA-256**: AES exhibits performance similar to SHA-256, another cryptographic hash function. Similar to SHA-1, SHA-256 serves a different purpose than AES.

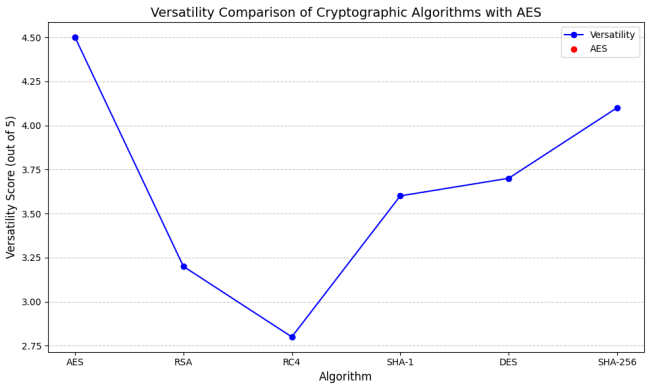

Figure 5: Versatility graph

Fig. 5. shows the difference between AES and other commonly used encryption techniques based on Versatility.

In the context of cryptography, versatility refers to the ability of a cryptographic algorithm to be used effectively in a wide range of applications1. A versatile algorithm can adapt to various cryptographic contexts and requirements, making it suitable for different purposes. In the graph, the versatility score is a measure of this adaptability. A higher score indicates that the algorithm is more adaptable and can be effectively used in a wider range of applications. In this case, AES shows a high versatility score, indicating its wide applicability in various cryptographic contexts.

- **Axes**:
  - The y-axis represents the versatility score of the algorithms, out of 5. It ranges from 2.75 to 4.50.
  - The x-axis represents different cryptographic algorithms. The algorithms included are AES, RSA, RC4, SHA-1, DES, and SHA-256.
- **Versatility Scores of Algorithms**:
  - **AES**: The versatility score for AES is the highest at 4.50, indicating its adaptability in various cryptographic contexts.

- **RSA**: The score decreases to 3.00 for RSA, suggesting it may not be as versatile as AES.
- **RC4**: The score further drops to its lowest at 2.75 for RC4.
- **SHA-1**: Then it increases to 3.25 for SHA-1, showing a slight improvement in versatility.
- **DES**: The score continues rising to 3.75 for DES, indicating better versatility.
- **SHA-256**: The score peaks again at 4.25 for SHA-256, which is slightly lower than AES but still quite high.

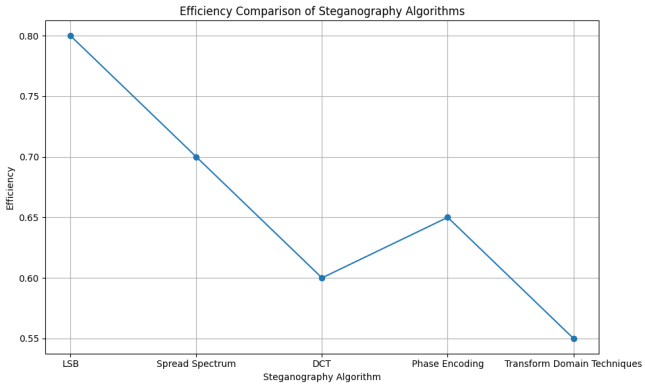

Figure 6: LSB Efficiency graph

Fig. 6. shows the comparison between Least Significant Bit (LSB) algorithm and other commonly used steganography algorithms.

This graph provides a clear comparison of the versatility of various cryptographic algorithms, highlighting the superior versatility of AES. LSB steganography is a common method of hiding secret data within digital images. The idea is that LSB embedding is that if we change the last bit value of a pixel, there won't be much visible change in the color. The process of LSB steganography involves replacing the least significant bit of each pixel in an image with a bit of the secret message. This method works best when the image file is larger than the message file. The upper hand of LSB is that its simple and has high data-hiding capacity. It's also challenging to find the presence of hidden information without knowing the exact technique used.

- **Axes**:
  - The y-axis represents the efficiency of the algorithms. It ranges from 0.55 to 0.80.
  - The x-axis represents different steganography algorithms. The algorithms included are LSB, Spread Spectrum, DCT, Phase Encoding, and Transform Domain Techniques.
- **Efficiency of Algorithms**:
  - **LSB**: The efficiency of LSB is highest at around 0.80, indicating its superior performance in terms of efficiency.

- **Spread Spectrum**: The efficiency decreases sharply to about 0.60 for Spread Spectrum, suggesting it may not be as efficient as LSB.
- **DCT**: The efficiency remains at about 0.60 for DCT, similar to Spread Spectrum.
- **Phase Encoding**: The efficiency increases slightly to approximately 0.65 for Phase Encoding, showing a slight improvement in efficiency.
- **Transform Domain Techniques**: The efficiency drops again to about 0.60 for Transform Domain Techniques, similar to Spread Spectrum and DCT.

Low susceptibility to detection refers to how difficult it is to detect the presence of hidden data within a file or a message.A steganography algorithm with low susceptibility to detection means that it's very difficult for an unauthorized person to even realize that there's hidden data. This is often achieved by embedding the secret data in such a way that the file or message doesn't appear to be altered significantly

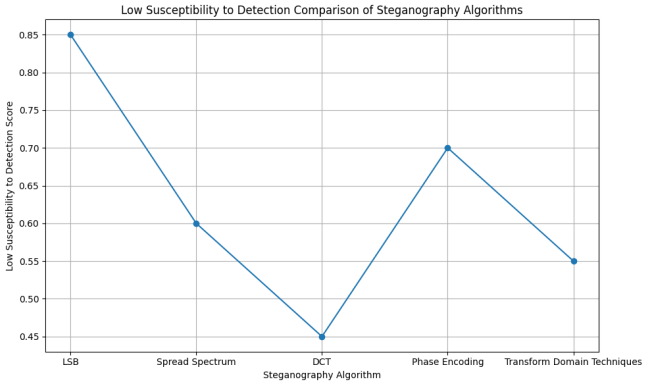

Figure 7: LSB Low susceptibility to detection graph

Fig. 7. shows the comparison between Least Significant Bit (LSB) algorithm and other commonly used steganography algorithm based on Low susceptibility detection aspect.

- **Axes**:
  - The y-axis represents the low susceptibility to detection score of the algorithms. It ranges from 0.45 to 0.85.
  - The x-axis represents different steganography algorithms. The algorithms included are LSB, Spread Spectrum, DCT, Phase Encoding, and Transform Domain Techniques.
- **Low Susceptibility to Detection Scores of Algorithms**:
  - **LSB**: The score for LSB is highest at around 0.85, indicating its superior performance in terms of low susceptibility to detection.
  - **Spread Spectrum**: The score decreases sharply to just above 0.5 for Spread Spectrum, suggesting it may have higher susceptibility to detection compared to LSB.
  - **DCT**: The score increases to around 0.7 for DCT, showing an improvement in low susceptibility to detection.

- **Phase Encoding**: The score remains just above 0.5 for Phase Encoding, similar to Spread Spectrum.
- **Transform Domain Techniques**: The score increases to around 0.7 for Transform Domain Techniques, similar to DCT.

This graph provides a clear comparison of the low susceptibility to detection of various steganography algorithms, highlighting the superior performance of LSB. *The parameters that are used to measure the security pheonmena:* The security phenomena in this research are evaluated through key parameters. First, the execution speed of the AES encryption algorithm is measured in microseconds, emphasizing its efficiency for real-time applications. Second, the versatility of cryptographic algorithms, particularly AES, is assessed for adaptability across diverse cryptographic contexts. Additionally, steganography algorithm efficiency, focusing on LSB, is considered for concealing information within digital images. The low susceptibility to detection in steganography, particularly demonstrated by LSB, is examined, highlighting its ability to evade unauthorized detection of hidden data. These parameters collectively provide insights into the security features and effectiveness of the proposed system within the constraints of the research space.

How to improve the performance and achieve: To boost system performance, strategies include optimizing cryptographic algorithms like AES through hardware acceleration and refining steganography techniques such as LSB for efficiency without compromising security. Ensuring reliability involves rigorous testing (unit, integration, functional), robust error handling, secure data transmission protocols, and redundancy measures. Regular maintenance and continuous monitoring for vulnerabilities contribute to a reliable and secure system.

4 Results and Discussion

In this section we discuss about the results of the proposed system and discuss about the functions of each functional results. Image Selection: Users initiate the process by selecting an image file to serve as the canvas for hidden data. The UI indicates whether a file has been designated for this purpose. Data Input: Within the designated UI field, users enter the confidential information they wish to conceal within the selected image. This step is crucial and demands thoughtful consideration of the data's nature and implications. Beneath the data input field, users find a condensed option to proceed with the embedding process. Instead of a standalone submission button, users are invited to confirm their data input, seamlessly integrating the initiation of the algorithmic process within the data entry interface.

Fig. 8. shows the message panel where the user selects an image of choice and enters their message.

Fig. 9 shows the inbox panel where a list of senders are mentioned with their details. In this case only one sender is mentioned

In this scenario, the recipient possesses comprehensive details about the sender, including identification information, along with an encoded image. The encoded image functions as a carrier for an encrypted message. To access the concealed information, the user is provided with a user-friendly option to select the pertinent image. This selection prompts the system to initiate an automated decoding process, obviating the need for manual decryption by the recipient. The decryption operation seamlessly transpires in the background, ensuring a streamlined and efficient experience for the user. It is noteworthy that the encoded message, embedded within the selected image, is intelligibly deciphered by the system, thereby revealing the original content. This approach not only enhances user convenience but also reinforces the

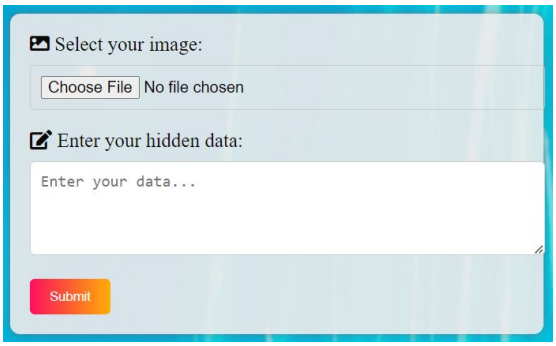

Figure 8: Image selection and message panel

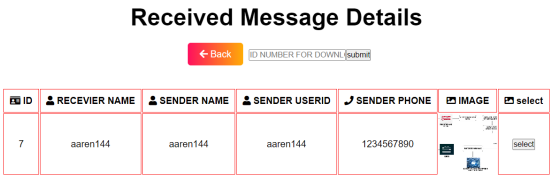

Figure 9: Message Inbox

security and confidentiality aspects of the communication process. Moreover, the adoption of image-based encoding introduces an additional layer of security to the communication process. By embedding the message within an image, it reduces the vulnerability to interception by malicious entities seeking to eavesdrop on the conversation. This method relies on steganography, the practice of concealing information within other data, to hide the message in plain sight. Consequently, even if unauthorized individuals manage to access the transmitted data, deciphering the concealed message without the appropriate decoding mechanism proves challenging. Thus, the integration of image-based encoding not only enhances user convenience but also reinforces the security and confidentiality of sensitive information exchanges across diverse digital platforms.

Figure 10: Decoded Message

Fig. 10. Shows the decrypted and decoded message upon selecting a sender from the inbox. After automated decryption, only the designated recipient with a unique ID gains access to the concealed message in the selected image. This strict access control enhances personalized security, aligning with contemporary standards for secure information exchange in digital environments.

5 Conclusion

The project presents a robust and secure solution for image transmission in web applications, integrating the Advanced Encryption Standard (AES) for image encryption and the embedding of hidden text data within images. It successfully addresses the critical needs of data security, confidentiality, and integrity, providing a versatile platform suitable for various sectors, including healthcare, legal, and corporate environments. By offering sender and receiver modules, the project ensures that image transmission is not only secure but also user-friendly, with easy image uploading, encryption, and decryption processes. The hidden text data feature adds an innovative dimension, allowing for the secure sharing of additional information alongside images, which can find applications in watermarking, data tracking, or context enrichment. Moreover, the incorporation of user authentication enhances security by preventing unauthorized access to the system. The project also emphasizes scalability, making it adaptable to the evolving requirements of a growing user base and increasing data volumes.

6 References

References

[1] M. M. Mahmoud and H. T. Elshoush, "Enhancing LSB Using Binary Message Size Encoding for High Capacity, Transparent and Secure Audio Steganography–An Innovative Approach," *IEEE Access*, vol. 10, pp. 29954-29971, 2022. doi:10.1109/ACCESS.2022.3155146.

[2] S. Rahman, J. Uddin, H. U. Khan, H. Hussain, A. A. Khan, and M. Zakarya, "A Novel Steganography Technique for Digital Images Using the Least Significant Bit Substitution Method," *IEEE Access*, vol. 10, pp. 124053-124075, 2022. doi:10.1109/ACCESS.2022.3224745.

[3] Z. Yin, Y. Ji, and B. Luo, "Reversible Data Hiding in JPEG Images with Multi-objective Optimization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 8, pp. 2343–2352, August 2020. doi:10.1109/TCSVT.2020.2969463.

[4] J. Hou, B. Ou, H. Tian, and Z. Qin, "Reversible Data Hiding Based on Multiple Histograms Modification and Deep Neural Networks," *Signal Processing, Image Communication*, vol. 92, March 2021, Article no. 116118. doi:10.1016/j.image.2020.116118.

[5] A. Malik, G. Sikka, and H. K. Verma, "A Reversible Data Hiding Scheme for Interpolated Images Based on Pixel Intensity Range," *Multimedia Tools and Applications*, vol. 79, nos. 25–26, pp. 18005–18031, July 2020. doi:10.1007/s11042-020-08691-2.

[6] F. S. Hassan and A. Gutub, "Efficient Image Reversible Data Hiding Technique Based on Interpolation Optimization," *Arabian Journal of Science and Engineering*, vol. 46, no. 9, pp. 8441–8456, March 2021. doi:10.1007/s13369-021-05529-3.

[7] Y.-Q. Chen, W.-J. Sun, L.-Y. Li, C.-C. Chang, and X. Wang, "An Efficient General Data Hiding Scheme Based on Image Interpolation," *Journal of Information Security Applications*, vol. 54, October 2020, Article no. 102584. doi:10.1016/j.jisa.2020.102584.

[8] L. Yang, H. Deng, and X. Dang, "A Novel Coverless Information Hiding Method Based on the Most Significant Bit of the Cover Image," *IEEE Access*, vol. 8, 2020, pp. 108579-108591. doi:10.1109/ACCESS.2020.3000993.

[9] J. R. Jayapandiyan, C. Kavitha, and K. Sakthivel, "Enhanced Least Significant Bit Replacement Algorithm in Spatial Domain of Steganography Using Character Sequence Optimization," *IEEE Access*, vol. 8, 2020, pp. 136537-136545. doi:10.1109/ACCESS.2020.3009234.

[10] S. Bhattacharjee, L. B. A. Rahim, J. Watada, and A. Roy, "Unified GPU Technique to Boost Confidentiality, Integrity and Trim Data Loss in Big Data Transmission," *IEEE Access*, vol. 8, 2020, pp. 45477-45495. doi:10.1109/ACCESS.2020.2978297.

[11] T.-C. Lu, S.-R. Huang, and S.-W. Huang, "Reversible Hiding Method for Interpolation Images Featuring a Multilayer Center Folding Strategy," *Soft Computing*, vol. 25, no. 1, pp. 161–180, January 2021. doi:10.1007/s00500-020-05129-7.

[12] P. C. Mandal, I. Mukherjee, and B. N. Chatterji, "High Capacity Reversible and Secured Data Hiding in Images Using Interpolation and Difference Expansion Technique," *Multimedia Tools and Applications*, vol. 80, no. 3, pp. 3623–3644, January 2021. doi:10.1007/s11042-020-09341-3.

[13] B.-M. Zhou, L.-D. Lin, W. Wang, and Y. Liu, "Security Analysis of Particular Quantum Proxy Blind Signature Against the Forgery Attack," *International Journal of Theoretical Physics*, vol. 59, no. 2, pp. 465–473, February 2020. doi:10.1007/s10773-019-04340-z.

[14] A. Altigani, S. Hasan, B. Barry, S. Naserelden, M. A. Elsadig, and H. T. Elshoush, "A Polymorphic Advanced Encryption Standard – A Novel Approach," *IEEE Access*, vol. 9, pp. 20191-20207, 2021. doi:10.1109/ACCESS.2021.3051556.

[15] A. Menezes and D. Stebila, "The Advanced Encryption Standard: 20 Years Later," *IEEE Security & Privacy*, vol. 19, no. 6, pp. 98-102, Nov.-Dec. 2021. doi:10.1109/MSEC.2021.3107078.