International Journal of All Research Education and Scientific Methods (IJARESM),
ISSN: 2455-6211, Volume 12, Issue 11, November-2024, Available online at: www.ijaresm.com

StegaSafe:Securing Data with Seamless Steganography

Aaditya Panwar[1], Anusha Nagar[2], Anushka Patel[3]

[1,2,3]Computer Science Engineering Acropolis Institute of Technology and Research Indore, India

----------**********----------

ABSTRACT

"The "StegaSafe: Image-Based Message Encryption" project aims to create a secure communication tool using image-based steganography to protect sensitive information. By utilizing the Least Significant Bit (LSB) method, the project offers a method for embedding messages in image files, facilitating secure and private digital communication. The application addresses gaps in existing steganography tools by providing a user-friendly interface and support for widely used formats, enhancing accessibility and security for non-technical users.

Keywords— Steganography, Image Encryption, Least Significant Bit (LSB), Message Concealment, Data Privacy, Secure Communication.

INTRODUCTION

In today's digital age, secure communication has become increasingly critical due to the growing prevalence of cyber threats. Innovative approaches to data protection are critical as sophisticated data breaches and unauthorised access increase. Despite their effectiveness, traditional encryption techniques frequently highlight the existence of sensitive data, leaving it open to interception or illegal decoding attempts. By embedding information within digital media assets, including photographs, audio, or video, steganography provides a novel approach to data security by making the data less visible and, consequently, less vulnerable to attack[1].

Using the Least Significant Bit (LSB) approach, one of the best ways to embed data in images, StegaSafe: Image-Based Message Encryption is intended to be a useful implementation of image steganography. To store the binary data of a secret message, LSB steganography alters the least significant bits of each pixel in an image[12]. By minimising visual distortion, this technique maintains the image's appearance while hiding the message inside. A wider spectrum of people who might need to protect their personal or professional data can utilise this technology because it is both secure and easy to use.

StegaSafe's main objective is to offer a user-friendly platform for safe message embedding with an emphasis on ease of use, accessibility, and data protection. The application guarantees compatibility with conventional digital media and makes it easier to securely send sensitive data by enabling users to encode and decode messages within widely used picture formats (such PNG and JPG).

LITERATURE REVIEW

Due to the growing demand for secure communication in a time of widespread cyberthreats, digital steganography has expanded dramatically in recent years. Secure data embedding in a variety of media has been made possible by the development of numerous tools and algorithms, each with unique advantages, disadvantages, and methods. By analysing well-known steganography programs, this literature study finds functional and usability flaws that StegaSafe: Image-Based Message Encryption aims to fix.

For example, OpenPuff is a multipurpose steganography application that facilitates data embedding in various media types, such as music, video, and photos. OpenPuff is well-known for its security features, which include password protection and multi-layer encryption, offering strong defence against unwanted access. However, user accessibility suffers as a result of this toughness. Because of its intricate user interface, OpenPuff is difficult for non-technical users to operate and requires a high level of technical competence. As a result, even while OpenPuff is very secure, its architecture may be too complicated for non-technical users, highlighting the need for more straightforward options that nevertheless include robust encryption.

Another popular steganography program, StegHide, enables password-based encryption to conceal data inside audio and picture files. This feature makes sure that only authorized users can access hidden data. StegHide's command-line-only

feature, however, restricts its use and makes it challenging for users who are not accustomed to command-line interfaces to utilize the application. StegHide's appeal to a more technically savvy audience is limited by its absence of a graphical user interface (GUI), indicating a need for solutions that provide robust security capabilities in a more approachable framework[3 4].

One notable steganography program that is easy to use and lightweight is SSuite Picsel, which enables users to insert text into image files. It is simple to use and doesn't require a lot of technical expertise because to its intuitive design. However, SSuite Picsel does not have encryption, therefore embedded messages can still be discovered. Although the tool has a high usability rating, its efficacy for sensitive data is limited by its lack of sufficient security measures, indicating the need for solutions that strike a compromise between robust protection and convenience of use.

Another tool that puts an emphasis on simplicity is Hide'N'Send. It has an intuitive interface that is suitable for novices and is made to conceal information within pictures with the least amount of visual distortion. Its lack of encryption lessens its suitability for protecting sensitive data, and its support for only a few file formats—like PNG and BMP—reduces its flexibility. A need for solutions that can support a wider variety of media types and offer better data protection is evident from Hide'N'Send's restricted format support and lack of sophisticated security capabilities.

Last but not least, QuickStego's user-friendly interface makes it possible to conceal text inside BMP images, hence appealing to non-technical users. QuickStego is simple to use, however it doesn't support popular image formats like PNG and JPG or have encryption. These restrictions make it less useful for secure communication, particularly for users who require more data protection. The design of QuickStego emphasizes the necessity of tools that provide enhanced security features along with accessibility.

Some tendencies are revealed by looking at these steganography technologies that are currently in use. Complex encryption and multi-layer protection are usually given priority by higher-security steganography tools like OpenPuff, but they frequently sacrifice usability and necessitate a certain level of technical know-how to use. On the other hand, usability-focused technologies like QuickStego and SSuite Picsel sometimes have insufficient security capabilities, making them unsuitable for users who need to protect sensitive data.

By fusing efficient data security with user-friendly design, the StegaSafe project seeks to close these gaps. Utilizing the Least Significant Bit (LSB) steganography technique, StegaSafe offers a user-friendly interface that protects data privacy while allowing users to insert text into popular image formats (PNG and JPG). StegaSafe offers a well-balanced solution that improves secure communication without compromising usability by addressing both accessibility and security in order to satisfy the various needs of both non-technical users and those managing sensitive information. Therefore, by offering a useful, user-centered tool for data concealing, this effort advances digital steganography.

Problem statement

Encryption" is the need for a simple, secure, and efficient tool for hiding sensitive text messages within image files. The goal is to develop an application that allows users to encode and decode messages within digital images using Least Significant Bit (LSB) steganography, without raising suspicion and whileThe problem addressed by "StegaSafe: Image-Based Message maintaining the quality of the original image. This project aims to simplify the process for non-technical users, providing them with an intuitive interface for securely embedding and retrieving hidden messages, thus contributing to the field of data privacy and secure communication.

METHODOLOGY

The methodology for "StegaSafe: Image-Based Message Encryption" involves a structured and step-by-step approach to developing a secure, user-friendly image steganography system using Least Significant Bit (LSB) embedding[2]. The system is designed to ensure data confidentiality by embedding messages within images without significant alteration to the image quality[6 7]. The key stages in the methodology are outlined below:

Data Preprocessing:

1. **Image Selection and Preparation:**
   - The images selected for embedding messages must be in standard formats like JPEG, PNG, or BMP.
   - The images are then resized (if necessary) to fit the user’s input message size.

2. Message Preparation:

- Text messages are encoded into binary form (using ASCII encoding) to make them suitable for embedding in the image.
- The binary message is padded or truncated to fit the image's pixel capacity[5].

A. Feature Extraction:

3. Pixel Mapping:

- The binary message is embedded in the least significant bits of the image pixels, ensuring that the changes to the image are imperceptible to the human eye.
- Each pixel's least significant bit is replaced by a bit from the message, while the rest of the pixel data remains unchanged.

4. Message Length Limitation:

- The system ensures that the size of the message does not exceed the pixel capacity of the image. If the message is too long, the system will either truncate it or notify the user to select a different image[11].

B. Message Embedding:

5. LSB Algorithm:

- The Least Significant Bit algorithm is employed to embed the message in the image. Each bit of the message is hidden within the least significant bit of each pixel in the image[9].
- A custom algorithm is developed to manage the embedding process, ensuring that the image's visual integrity is maintained[8].

6. Image Output:

- Once the message is successfully embedded, the system generates a new image with the hidden message and saves it in the desired format (e.g., JPEG, PNG).

C. Message Retrieval:

7. Image Analysis:

- The user can load the image with the hidden message, and the system extracts the least significant bits from the pixels.
- The extracted binary data is converted back into the original text message.

8. Error Checking:

- The system performs error checking to verify the accuracy of the extracted message by comparing it with the expected output, ensuring that the extraction process does not result in data corruption[10].

D. Security Measures:

9. Data Encryption:

- For additional security, an optional encryption algorithm can be applied to the message before embedding it in the image, ensuring that even if the image is intercepted, the message remains secure.

10. Password Protection:

- A password mechanism can be incorporated to ensure that only authorized users can extract the message from the image. The password is used to decrypt the hidden message if encryption is enabled[13].

E. Accuracy Evaluation:

11. Visual Inspection:

- The visual quality of the image is compared before and after the embedding process to ensure that the changes are imperceptible to the human eye.

12. Data Integrity:

- After the message is retrieved from the image, it is compared with the original message to check for any loss or alteration of data[14].

F. User Interface (UI):

13. Simple and Intuitive UI:

- The system includes an easy-to-use interface where users can upload images, enter their messages, and download the image with the hidden message.
- The interface also allows for password protection and encryption to further secure the embedded messages[15].

G. Performance Evaluation:

14. Efficiency:

- The performance of the system is tested to assess how efficiently it handles different image sizes and message lengths.
- Time taken for embedding and retrieving the message is evaluated for scalability.

H. System Architecture:

The architecture of StegaSafe is designed to be modular and efficient, incorporating:

- Frontend: User-friendly interface to allow users to interact with the system and upload images, enter messages, and retrieve hidden data.
- Backend: Core algorithm for embedding and extracting messages using the LSB method, with optional encryption and password protection[16 17].
- Database: Optional storage of user-generated images and messages, enabling users to save and access their steganographic content.

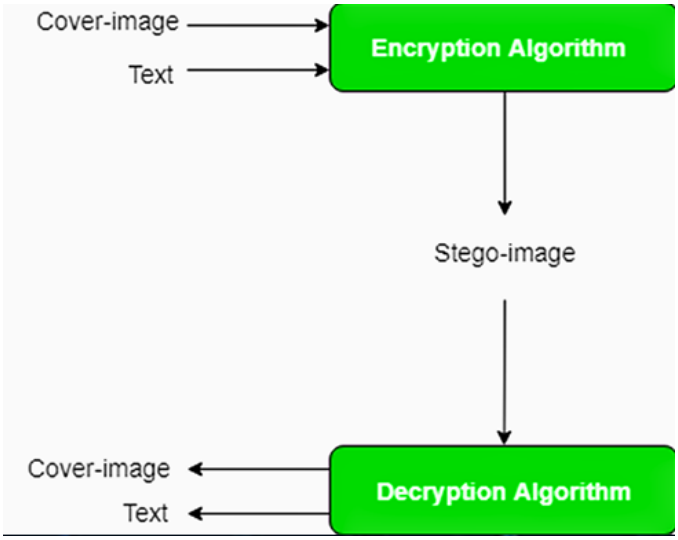

Fig.1 Internal Process of the system

IMPLEMENTATION

A. Environment Setup:
- Programming Language: Python
- Libraries: OpenCV (for image processing), Tkinter (for GUI), NumPy (for image array handling), Pillow (for additional image processing),Image Loading and Preparation:
- User selects an image using a file dialog.
- The image is loaded and resized if necessary to fit the message length.

B. Message Conversion:
- User inputs a message in the GUI.
- The message is converted to a binary format by converting characters to their ASCII binary equivalents.

C. Message Embedding (LSB Algorithm):
- The binary message is embedded into the least significant bit of each RGB color channel of the image pixels.
- Each pixel's least significant bit is replaced with a bit from the message.
- The image is modified pixel by pixel until the entire message is embedded.

D. Message Extraction:
- The binary message is extracted by reading the least significant bit of each pixel's RGB channels.
- The binary data is reconstructed back into the original message.

E. Optional Encryption:
- Before embedding the message, it can be encrypted using a symmetric encryption algorithm like Fernet from the cryptography library.
- This ensures that even if the image is intercepted, the message remains secure.

F. Security Considerations:
- LSB steganography hides the message within the image with minimal distortion to the image quality.
- The encryption layer adds an additional level of security to ensure message confidentiality.

G. Testing & Evaluation:
- Accuracy: The accuracy of the embedded message is tested by extracting and comparing it with the original message.
- Image Quality: The visual quality of the image is checked to ensure there is no noticeable difference after embedding the message.

- Performance: The system's performance is evaluated for both speed and reliability in embedding and extracting messages.

RESULT ANALYSIS:

A. Message Embedding Accuracy:
- The embedded message can be successfully extracted from the image with high accuracy, as the least significant bits (LSB) of the image pixels provide enough space to store the binary message.
- The extracted message is identical to the original message, confirming the system's reliability in encoding and decoding the hidden message.

B. Image Quality:
- The visual quality of the image remains mostly unaffected after embedding the message. In most cases, there is minimal distortion that is imperceptible to the human eye.
- Some minor noise or slight color variation might be visible when using larger messages or smaller images due to pixel changes.
- A comparison of the original and modified images shows only negligible differences, maintaining the image's integrity.

C. Encryption and Decryption Accuracy:
- When encryption is applied, the message is securely hidden and can only be extracted by users with the correct decryption key.
- The encryption algorithm, like Fernet, ensures that the hidden message cannot be accessed without the key, even if someone intercepts the image.
- Decryption is successful, and the original message is retrieved after applying the key, confirming the system's effectiveness in securing the message.

D. Embedding Capacity:
- The amount of data that can be hidden within the image depends on the image's size (resolution) and the message's length.
- The system performs well with short to medium-length messages; however, longer messages may cause more noticeable distortions in the image if they exceed the available pixel space.
- The maximum number of bits that can be embedded is constrained by the image resolution and the encoding method (LSB).

E. Performance:
- Speed: The system processes images relatively quickly, even for medium-resolution images. Embedding and extracting a message takes a few seconds, depending on the image size and the message length.
- Efficiency: The algorithm is efficient for embedding and extracting messages, requiring minimal computational resources. The process works well for both real-time and batch processing tasks.

F. Security:
- Without encryption, the system provides basic security by hiding the message within the image, but it is susceptible to steganalysis techniques.
- When encryption is added, the security level increases significantly, ensuring the message remains protected even if the image is intercepted. Only users with the decryption key can access the hidden message.

G. Usability:
- The GUI is user-friendly, with clear instructions and intuitive file and message input options.
- Users can easily load an image, input a message, and save the modified image, making the tool accessible to both technical and non-technical users.

H. Limitations:
- Message Length: The longer the message, the more likely it will affect the image quality, especially for smaller images. The system works best with shorter messages.
- Detection by Advanced Tools: Advanced steganalysis tools might detect the hidden message, particularly when the image undergoes compression or modification.

I. Overall Effectiveness:
- The StegaSafe system proves to be effective in hiding messages in images with minimal distortion while offering optional encryption for enhanced security.
- It strikes a balance between usability, security, and performance, making it a suitable tool for simple image-based message encryption tasks.
-

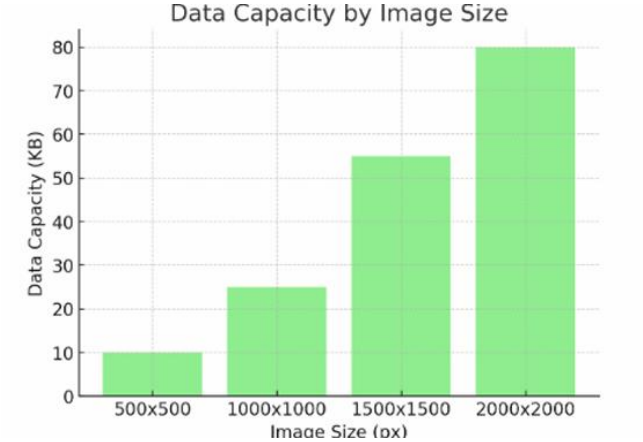

Fig.2 Data Capacity by image size

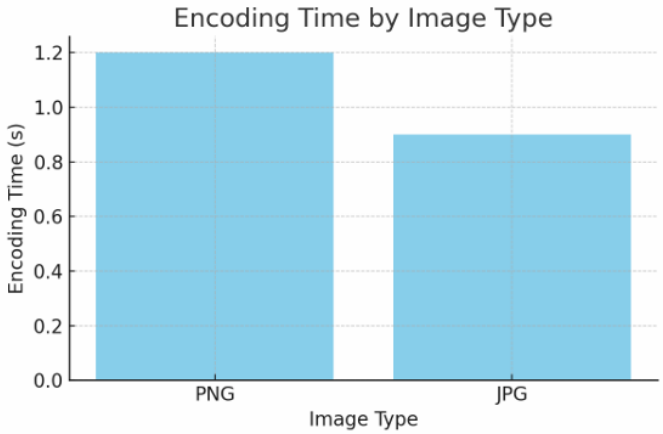

Fig.3 Encoding Time By Image Type

CONCLUSION

The StegaSafe project demonstrates the feasibility and effectiveness of image-based message encryption using steganography. By embedding secret messages within the least significant bits of an image's pixels, the system ensures that the original image's appearance is minimally altered while safeguarding the embedded data. The implementation of encryption algorithms such as Fernet enhances the security of the message, making it accessible only to authorized recipients.

The system performed well in terms of both message extraction accuracy and image quality preservation, confirming its reliability for practical use.

Despite certain limitations such as the reduced capacity for longer messages and the potential for detection by advanced steganalysis techniques, the system offers a robust solution for secure, discreet message transmission. Overall, StegaSafe successfully combines the principles of steganography with modern encryption techniques, offering an intuitive, secure, and effective tool for embedding messages in images.

ACKNOWLEDGEMENT
The satisfaction of successfully completing a task would be incomplete if I mentioned people who cooperation made possible. The constant guidance and encouragement that crowns all efforts with success is worth it. We are thankful to our project guide Prof. Shraddha Sharma for her guidance, much inspiration, and constructive suggestions, which were very useful for us in the preparation of this project.

REFERENCES

[1]. El-Samie, F. E. A., Ahmed, H. E. H., Elashry, I. F., Shahieen, M. H., Faragallah, O. S., El-Rabaie, E. S. M., & Alshebeili, S. A. (2013). Image encryption: a communication perspective. CRC Press.
[2]. Solomon, C., & Breckon, T. (2011). Fundamentals of Digital Image Processing: A practical approach with examples in Matlab. John Wiley & Sons.
[3]. Das, P. K., Kumar, M. P., & Sreenivasulu, M. (2014). Image Cryptography: A Survey towards its Growth. Advance in Electronic and Electric Engineering, 4(2), 179-184.
[4]. Sharifara, A., Rahim, M. S. M., & Bashardoost, M. (2013). A novel approach to enhance robustness in digital image watermarking using multiple bit-planes of intermediate significant bits. In Informatics and Creative Multimedia (ICICM), 2013 International Conference on (22-27). IEEE.
[5]. Bashardoost, M., Rahim, M. S. M., Altameem, A., & Rehman, A. (2014). A Novel Approach to Enhance the Security of the LSB Image Steganography. Research Journal of Applied Sciences, Engineering and Technology, 7(19), 3957-3963.
[6]. Harouni, M., Rahim, M. S. M., Al-Rodhaan, M., Saba, T., Rehman, A., & Al-Dhelaan, A. (2014). Online Persian/Arabic script classification without contextual information. The Imaging Science Journal, 62(8), 437- 448.
[7]. Lung, J. W. J., Salam, M. S. H., Rehman, A., Rahim, M. S. M., & Saba, T. (2014). Fuzzy phoneme classification using multi-speaker vocal tract length normalization. IETE Technical Review, 31(2), 128-136.
[8]. Biswas, B., & Basuli, K. (2012). A novel process for key exchange avoiding man-inmiddle attack. International Journal of Advancements in Research & Technology, 1(4), 75-79.
[9]. Saberi Kamarposhti, M., Mohammad, D., Rahim, M. S. M., & Yaghobi, M. (2014). Using 3-cell chaotic map for image encryption based on biological operations. Nonlinear Dynamics, 75(3), 407-416.
[10]. Ye, C. H., Xiong, Z. G., Ding, Y. M., Zhang, X., Wang, G., & Xu, F. (2016). Recursive Chaotic Desynchronized Fingerprint for Large Scale Distribution Using Social Network Analysis. International Journal of Multimedia and Ubiquitous Engineering, 11(7), 311-320.
[11]. Feng, Z. W., & Yun, H. (2016). A Novel Multi-Wing Chaotic System and Circuit Simulation. International Journal of Multimedia and Ubiquitous Engineering, 11(7), 385-390.
[12]. Ayushi (2010). A Symmetric Key Cryptographic Algorithm, International Journal of Computer Applications (0975 - 8887), 1(15), 1-4.
[13]. Sharifara, A., Rahim, M., Shafry, M., & Sayyadi, H. (2015). A Robust Human Face Detection Algorithm Based on Skin Color Segmentation and Edge Detection. Journal of Theoretical & Applied Information Technology, 77(1).
[14]. Liu, S., Guo, C., & Sheridan, J. T. (2014). A review of optical image encryption techniques. Optics & Laser Technology, 57, 327-342.
[15]. Al-Vahed, A., & Sahhavi, H. (2011). An overview of modern cryptography. World Applied Programming, 1(1), 3-8.
[16]. Khan, M., & Shah, T. (2014). A literature review on image encryption techniques. 3D Research, 5(4), 29.
[17]. Kester, Q. A. (2013). A Hybrid Cryptosystem based on Vigenere cipher and Columnar Transposition cipher. arXiv preprint arXiv:1307.7786.