IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, March 2026
Impact Factor: 8.2

StegoVault

Shubham Wagh, Shivam Ranwade, Aditya Munde, Hitesh Choudhary, Mrs. Sujata Sanap
Department of Computer Engineering
Rasiklal M. Dhariwal Institute of Technology, Chinchwad, Pune

Abstract: *This project focuses on developing a Secure Image Steganography Application named StegoVault, an intelligent Android application that allows users to hide sensitive information within digital images using password-based protection. The primary objective of this system is to enhance data security and privacy by embedding confidential text or files into images in a way that remains invisible to unauthorized users.*

The application utilizes image steganography techniques, specifically the Least Significant Bit (LSB) method, to embed secret data into image pixels without causing noticeable changes in visual quality. The system applies password-based encryption before embedding the data, ensuring that only users with the correct password can access and decode the hidden information.

The project is developed using modern Android technologies, mainly Kotlin, and incorporates efficient image processing and data handling techniques. The system follows a structured workflow that includes data input (text or file selection), encryption (securing data using a password), embedding (hiding data within image pixels), and decoding (extracting hidden data from the image). Additionally, the application includes an image analysis feature that provides details such as resolution, file size, and estimated data hiding capacity.

Keywords*: Secure Image Steganography*

I. INTRODUCTION

In today’s digital era, rapid advancements in mobile technology have led to the development of smart and secure applications that handle sensitive user data. With the increasing use of smartphones for communication, storage, and data sharing, ensuring data privacy and security has become a major concern. Traditional methods such as encryption protect the content of data but often reveal the presence of hidden information, which may attract unwanted attention.

To address this issue, steganography has emerged as an effective technique for secure communication. Steganography is the process of hiding secret information within a digital medium such as an image, audio, or video file in such a way that the existence of the data remains concealed. Unlike encryption, which only scrambles the data, steganography hides the data itself, making it difficult for unauthorized users to detect.

This project introduces StegoVault, a Secure Image Steganography Android application designed to hide and retrieve sensitive information within images using password-based protection. The application allows users to encode confidential text or files into an image and decode the hidden data only with the correct password. By combining steganography with encryption techniques, the system ensures both secrecy and security.

The application uses the Least Significant Bit (LSB) method to embed data into image pixels. This technique modifies only the least significant bits of pixel values, ensuring that the visual appearance of the image remains almost unchanged. As a result, the hidden data cannot be easily detected by human observation, providing an additional layer of security.

The system is developed using modern Android technologies, primarily Kotlin, and is designed to be efficient, user-friendly, and lightweight. It includes multiple functionalities such as data encoding, decoding, and image analysis, allowing users to understand the properties and capacity of the carrier image.

With increasing cyber threats, data leaks, and privacy concerns, there is a growing demand for secure and reliable communication methods. The StegoVault application provides a practical solution by enabling users to protect sensitive

DOI: 10.48175/568

information in a simple and effective way. It demonstrates how steganography can be integrated into everyday mobile applications to enhance data security.

Overall, this project represents a step toward the development of intelligent and secure mobile systems that protect user data while maintaining usability. By combining advanced techniques with a simple interface, StegoVault aims to provide a secure and efficient solution for hidden data communication.

II. LITERATURE REVIEW

Smart and secure data communication systems have gained significant importance in recent years due to the increasing need for privacy and protection of sensitive information. Researchers and developers have explored various techniques to ensure secure data transmission, among which steganography has emerged as a highly effective approach. Unlike traditional encryption methods, steganography focuses on concealing the existence of data, making it a valuable tool for covert communication.

Several studies have highlighted the use of digital media such as images, audio, and video files as carriers for hidden information. Among these, image steganography is widely preferred due to its simplicity, high data capacity, and minimal perceptual distortion. The Least Significant Bit (LSB) technique is one of the most commonly used methods for embedding data into images. It works by modifying the least significant bits of pixel values, allowing secret data to be stored without significantly affecting the image quality.

Research indicates that LSB-based steganography provides an efficient balance between data capacity and invisibility. However, basic implementations may be vulnerable to detection through statistical analysis or image processing attacks. To overcome these limitations, many advanced approaches combine steganography with encryption techniques, adding an extra layer of security and ensuring that even if the hidden data is detected, it remains unreadable without proper authorization.

The development of Android-based applications has further expanded the practical implementation of steganography systems. Modern mobile platforms support real-time image processing, secure data handling, and user-friendly interfaces, making it possible to develop efficient steganography applications for everyday use. Technologies such as Kotlin and Android Studio enable developers to build responsive and lightweight applications with features like encoding, decoding, and file analysis.

Existing steganography applications available on mobile platforms provide basic functionalities such as hiding text within images and retrieving it later. However, many of these applications lack strong security mechanisms, such as password protection or encryption, which limits their effectiveness in real-world scenarios. Additionally, some applications do not provide proper analysis of image capacity, leading to inefficient data embedding or data loss.

Recent research also emphasizes the importance of combining usability with security. A well-designed system should not only provide strong protection but also ensure ease of use for the user. Features such as simple interfaces, clear workflows, and real-time feedback contribute to better user experience and adoption of secure systems.

Despite the progress made in this field, there is still a need for improved mobile-based steganography solutions that offer enhanced security, better performance, and user-friendly design. This highlights the importance of developing applications like StegoVault, which integrate password-based protection, efficient data embedding techniques, and practical usability into a single platform.

III. OBJECTIVES

• The main objective of this project is to design and develop a secure and efficient Image Steganography Android application named StegoVault that enhances data privacy by hiding sensitive information within digital images using password-based protection.

• The specific objectives of the project are as follows:

• • To develop a system that allows users to securely embed confidential text or files into digital images using steganography techniques.

DOI: 10.48175/568

- To implement the Least Significant Bit (LSB) method for efficient and invisible data embedding without affecting the visual quality of the image.
- To incorporate password-based encryption to ensure that only authorized users can access and decode the hidden information.
- To design a reliable decoding mechanism that accurately extracts hidden data from the encoded image.
- To ensure the system maintains a balance between data security, embedding capacity, and image quality.
- To develop an image analysis feature that provides details such as resolution, file size, and estimated data hiding capacity.
- To create a user-friendly and intuitive interface that allows users to easily perform encoding, decoding, and analysis operations.
- To design a lightweight and efficient application that minimizes resource usage while maintaining smooth performance.
- To reduce the risk of data exposure by concealing the presence of sensitive information within images.
- To demonstrate the practical application of steganography and mobile security techniques in real-world scenarios.

IV. METHODOLOGY

The methodology describes the overall approach, system design, data handling process, and implementation techniques used to develop the StegoVault Image Steganography Application. This project follows a practical system-based approach where user data is securely embedded into images, processed, and retrieved when required.

4.1 Requirement Collection

The system requirements were identified based on the need for secure data hiding and retrieval. Key requirements include image selection, data input, password protection, accurate encoding and decoding, and a user-friendly interface.

4.2 System Analysis (Data Understanding)

The properties of digital images were studied, including pixel structure, color channels, and data capacity. This helped in understanding how data can be embedded without significantly affecting image quality.

4.3 Data Acquisition (Input Collection)

The system collects input data from the user, which may include text or small files. Additionally, the user selects an image that will act as the carrier for embedding the secret data.

4.4 Data Encryption

Before embedding, the input data is encrypted using a password provided by the user. This ensures that even if the hidden data is extracted, it cannot be understood without the correct password.

4.5 Data Embedding (Steganography Process)

The encrypted data is embedded into the image using the Least Significant Bit (LSB) technique. This method modifies the least significant bits of pixel values, ensuring that changes remain visually undetectable.

4.6 Image Processing and Generation

After embedding the data, a new encoded image is generated and stored. The system ensures that the output image maintains nearly the same visual quality as the original image.

4.7 Data Extraction (Decoding Process)

During decoding, the system reads the modified pixel values and extracts the hidden data from the image. The extracted data is then processed for decryption.

4.8 Password Verification and Decryption

The system verifies the password entered by the user. If the password is correct, the encrypted data is decrypted and converted back into its original form.

4.9 System Implementation

The application is developed using Kotlin in Android Studio. It utilizes image processing techniques and Android APIs to perform encoding, decoding, and file handling operations efficiently.

4.10 User Interface Design

A simple and intuitive user interface is designed with separate modules for encoding, decoding, and image analysis. The interface ensures ease of use and smooth interaction for users.

4.11 Testing and Performance Evaluation

The application is tested with different images and data sizes to evaluate performance. Factors such as encoding accuracy, decoding reliability, processing time, and image quality are analyzed to ensure system effectiveness.

The proposed system follows a structured workflow where data is collected, encrypted, embedded into an image, and later extracted and decrypted when required. This ensures secure and reliable data hiding and retrieval.

The system operates in multiple stages, starting with user input and ending with secure data extraction. Each stage is designed to maintain data integrity, security, and efficiency while providing a seamless user experience.

Overall, the methodology ensures that the StegoVault application functions as a secure, efficient, and user-friendly solution for hiding sensitive data within images.

V. PROPOSED FRAMEWORK

The proposed framework explains how the StegoVault system is designed and how it operates to securely hide and retrieve data within digital images using steganography and password-based encryption.

Initially, the system begins with user input, where the user selects a carrier image and provides the data to be hidden, which can be text or a file. Along with the data, the user enters a password that will be used for securing the information during the embedding process.

After collecting the input, the system performs data encryption. The entered data is converted into an encrypted format using the provided password. This ensures that even if the hidden data is extracted by unauthorized means, it cannot be understood without the correct password.

Once the data is encrypted, the embedding process begins. The system uses the Least Significant Bit (LSB) technique to hide the encrypted data within the pixel values of the selected image. Only the least significant bits are modified, ensuring that the visual appearance of the image remains almost unchanged.

After embedding, a new encoded image is generated, which contains the hidden data. This image can be stored or shared without revealing the presence of any secret information.

For data retrieval, the user selects the encoded image and enters the correct password. The system then extracts the hidden data from the image by reading the modified pixel values. The extracted data is decrypted using the password to recover the original content.

5.1 Encoding Module

This module is responsible for embedding encrypted data into the selected image. It handles data input, encryption, and the steganography process using the LSB technique.

5.2 Decoding Module

This module extracts hidden data from the encoded image. It verifies the password and decrypts the extracted data to restore the original information.

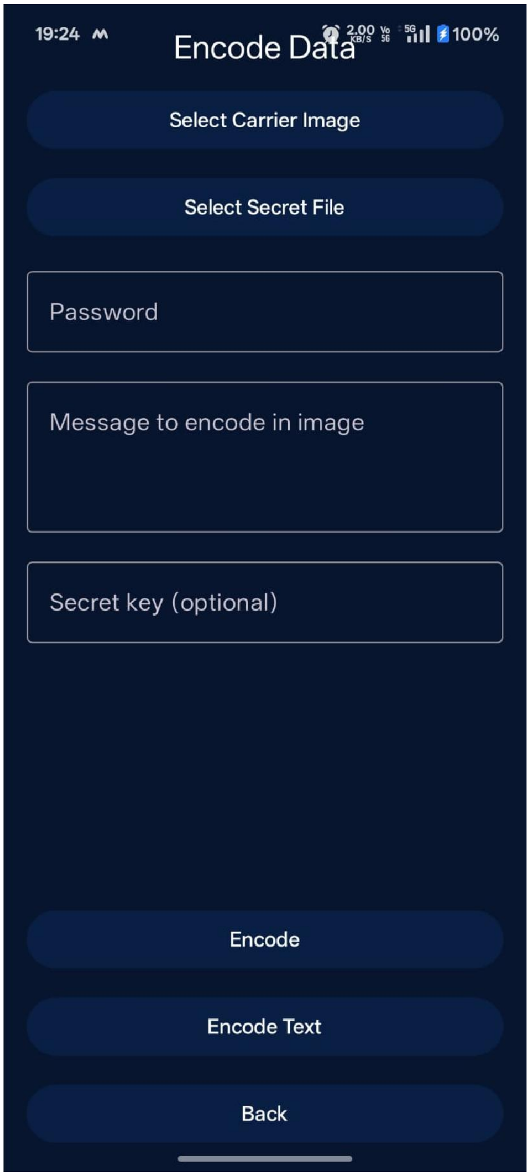
5.3 Encryption and Security Module

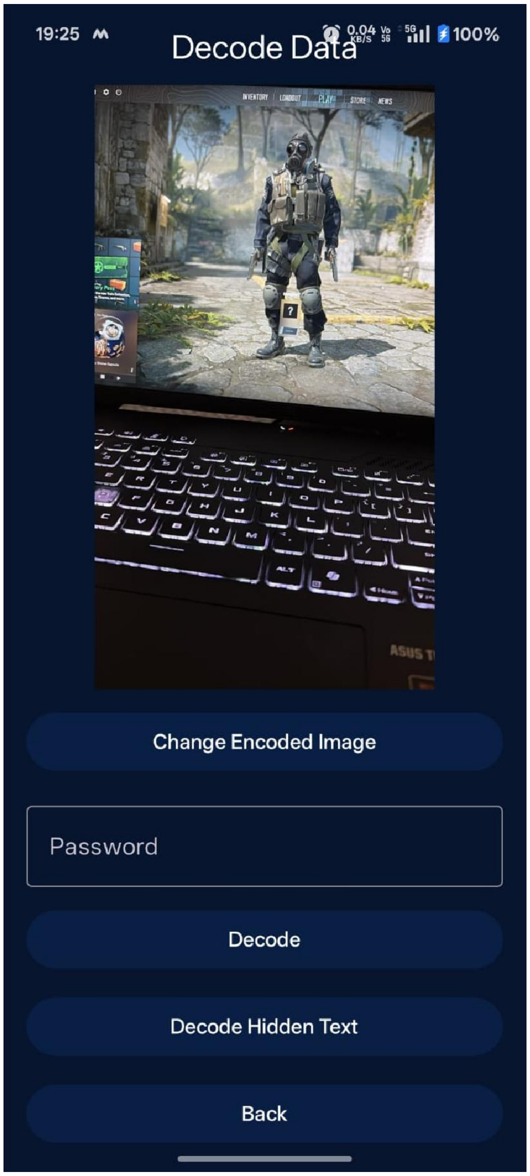
This module ensures that the data is securely encrypted before embedding. It protects the hidden information from unauthorized access and enhances overall system security.

5.4 Image Analyzer Module

This module provides information about the selected image, such as resolution, file size, and estimated data hiding capacity. It helps users choose suitable images for embedding.

DOI: 10.48175/568





5.5 User Interface Module

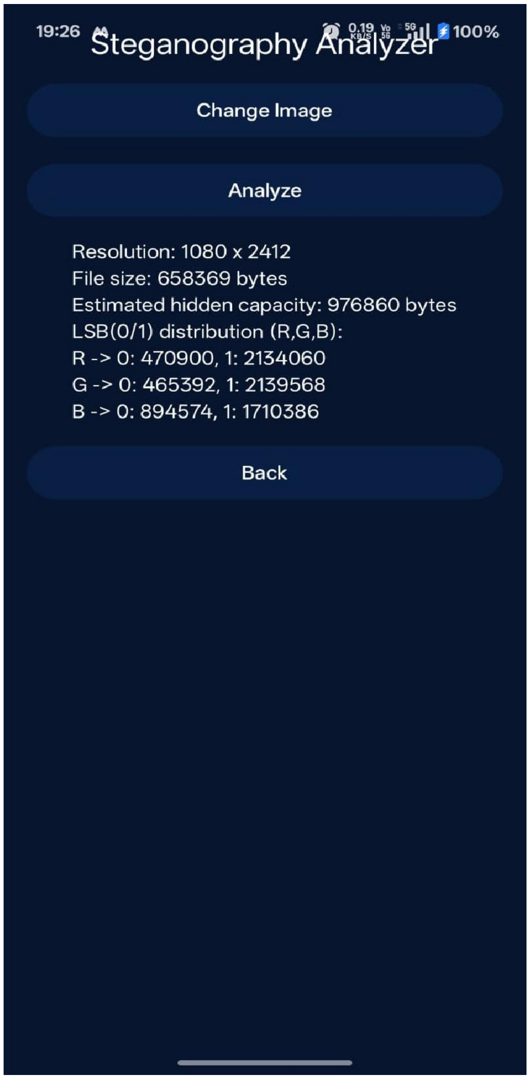
This module provides an interactive and user-friendly interface that allows users to perform encoding, decoding, and analysis operations easily.

5.6 System Workflow

The overall workflow of the system follows a structured sequence: data input is collected, encrypted, embedded into the image, and later extracted and decrypted when required. This ensures secure and efficient system operation.

5.7 Performance and Reliability

The system is designed to be efficient, secure, and reliable. It maintains data integrity, ensures accurate encoding and decoding, and provides consistent performance across different devices and image types.



```
Resolution: 1080 x 2412
File size: 658369 bytes
Estimated hidden capacity: 976860 bytes
LSB(0/1) distribution (R,G,B):
R -> 0: 470900, 1: 2134060
G -> 0: 465392, 1: 2139568
B -> 0: 894574, 1: 1710386
```

VI. RESULTS AND DISCUSSION

6.1 Successful Data Hiding and Retrieval

The application successfully embeds secret data into images and retrieves it accurately during decoding. The encoding and decoding processes work reliably when the correct password is provided, ensuring proper system functionality.

6.2 Effective Steganography Implementation

The use of the Least Significant Bit (LSB) technique allows data to be hidden within images without noticeable visual distortion. The encoded images appear almost identical to the original images, making it difficult for unauthorized users to detect the presence of hidden data.

6.3 Password-Based Security

The system ensures secure access to hidden data through password protection. Without the correct password, the extracted data remains encrypted and unreadable, providing an additional layer of security.

6.4 Real-Time Processing and Performance

The application demonstrates efficient performance during both encoding and decoding operations. The processing time is minimal, and the system responds quickly to user inputs, ensuring a smooth user experience.

6.5 Image Analysis Functionality

The image analyzer module provides useful information such as resolution, file size, and estimated data hiding capacity. This helps users select appropriate images for embedding data effectively.

6.6 System Efficiency and Stability

The application operates efficiently with minimal resource consumption. It maintains stable performance without causing noticeable lag or affecting device functionality.

6.7 Enhancement of Data Security

The combination of steganography and encryption improves overall data security. The system not only hides the data but also protects it from unauthorized access, making it suitable for secure communication.

VII. FUTURE SCOPE

The proposed StegoVault system can be further enhanced by integrating advanced features and improving its functionality to provide a more secure, efficient, and user-friendly experience.

Advanced Encryption Techniques:

The system can be improved by implementing stronger encryption algorithms such as AES or RSA before embedding the data. This will enhance security and make the hidden information more resistant to unauthorized access and attacks.

Improved Steganography Methods:

Future versions can incorporate advanced steganography techniques beyond the basic LSB method. Techniques such as adaptive steganography or frequency domain methods can be used to improve data security and reduce the chances of detection.

Multi-Format Data Support:

The application can be extended to support various types of data such as images, audio files, and documents. This will increase the usability of the system and allow users to hide different types of information within images.

Cloud Integration:

Cloud storage features can be added to securely store encoded images and retrieve them from anywhere. This will improve accessibility and provide backup options for users.

Enhanced Image Analysis:

The image analyzer module can be improved by providing more detailed insights such as optimal embedding capacity, compression impact, and security level indicators for different images.

User Customization Features:

Future versions can include customizable settings where users can control parameters such as encryption strength, embedding depth, and output image quality according to their preferences.

Cross-Platform Compatibility:

The application can be expanded to support multiple platforms such as iOS and web-based systems, ensuring wider accessibility and usability.

Integration with Secure Communication Systems:

The system can be integrated with messaging or file-sharing platforms to enable secure transmission of encoded images, making it more practical for real-world communication.

DOI: 10.48175/568

Artificial Intelligence Integration:

AI techniques can be used to optimize embedding strategies, detect suitable regions in images for hiding data, and improve resistance against steganalysis attacks.

Overall, the future scope of the StegoVault application lies in enhancing security, expanding functionality, and improving user experience to meet the growing demands of secure digital communication.

VIII. CONCLUSION

This project successfully demonstrates the design and development of a Secure Image Steganography Application, StegoVault, highlighting the practical use of data hiding techniques for secure communication. By combining steganography with password-based encryption, the system ensures both confidentiality and protection of sensitive information.

The application effectively embeds and retrieves hidden data within images while maintaining image quality and usability. Through the use of the Least Significant Bit (LSB) technique, the system achieves a balance between data capacity and invisibility, ensuring that the presence of hidden information remains undetectable.

The project also showcases the use of modern Android technologies such as Kotlin and efficient image processing methods to build a responsive and user-friendly application. The system performs reliably with minimal resource usage, making it suitable for real-world implementation.

Overall, StegoVault provides a practical and efficient solution for secure data communication. It reduces the risk of data exposure by concealing sensitive information within images and ensures that only authorized users can access it. This project highlights the importance of integrating security techniques into mobile applications and contributes to the advancement of secure and intelligent systems.

REFERENCES

[1] International Journal of Computer Applications, “A Study of Image Steganography Techniques Using LSB Method,” Vol. 182, Issue 10, 2023.

[2] International Journal of Engineering Research and Technology (IJERT), “Secure Data Hiding Using Image Steganography with Encryption,” Vol. 12, Issue 5, 2024.

[3] International Journal of Computer Science and Information Security (IJCSIS), “An Enhanced LSB Based Image Steganography Approach,” Vol. 21, Issue 2, 2024.

[4] IEEE International Conference on Communication and Signal Processing, “Secure Image Steganography Using Cryptography Techniques,” 2023.

[5] International Journal of Advanced Research in Computer Science (IJARCS), “A Review on Digital Image Steganography Methods,” Vol. 14, Issue 3, 2024.

[6] Android Developers Documentation, “Android Studio and Kotlin Development Guide.” Available at: https://developer.android.com

[7] Research articles on Digital Image Processing and Steganography Techniques, various online academic sources.

[8] Google Play Store, “Steganography Applications Analysis” – study of existing apps with limited security and basic encoding techniques.

[9] Tutorials and documentation on image processing and LSB algorithms from trusted technical platforms.

[7] “Volume Scheduler / Smart Volume Control Apps” on Google Play Store – study of applications providing rule-based volume control without real-time ambient noise analysis.

[8] “Sound Profile and Volume Manager Apps” – evaluation of existing mobile applications lacking continuous environmental monitoring and dynamic response.

[9] Research and reviews on Android-based sensor applications for real-time audio processing and adaptive system behaviour.

DOI: 10.48175/568