



# RDHEI Method for Secure Data Embedding: Guaranteeing Error-Free Extraction and Lossless Restoration in Encrypted Images

**Neetha S. S\***  
*School of CS & IT  
Jain (Deemed-To-Be)  
University Bangalore,  
Karnataka, India*

**Dr. Bhuvana J**  
*School of CS & IT  
Jain (Deemed-To-Be)  
University  
Bangalore, Karnataka, India*

## ABSTRACT-

The RDHEI (Reversible Data Hiding in Encrypted Images) method is a novel technique for securely embedding data in encrypted images. This method addresses the critical requirements of error-free extraction and lossless restoration while preserving the confidentiality of the original image. The process begins by encrypting the cover image using a robust encryption algorithm, ensuring the protection of its content. The RDHEI method capitalizes on the residual values obtained after encryption, which represent the differences between the original pixel values and their encrypted counterparts.

During data embedding, the secret information is divided into message units, which are concealed within the residual values using a difference expansion technique. This technique modifies the differences between adjacent pixel values, allowing for the seamless integration of the embedded data. The RDHEI method guarantees error-free extraction by employing a reverse process of difference expansion on the decrypted image's residual values, thereby accurately recovering the embedded message units.

**Keywords:** Data embedding, Data hiding, Encryption, Visual cryptography

## INTRODUCTION

The RDHEI (Reversible Data Hiding in Encrypted Images) method is a cutting-edge approach for secure data embedding in encrypted images. Its primary objective is to provide error-free extraction and lossless restoration of embedded information while maintaining the confidentiality of the original image. This method is particularly valuable in scenarios where sensitive data needs to be concealed within images without compromising their privacy and integrity.

Traditional data embedding techniques typically operate on unencrypted images, which can be vulnerable to unauthorized access and compromise the confidentiality of the embedded data. [1] The RDHEI method overcomes this challenge by operating directly on encrypted images. It takes advantage of the residual values, which represent the differences between the original pixel values and their encrypted counterparts, to embed the data securely.

The RDHEI method utilizes a difference expansion technique to embed the secret information within the residual values. This technique modifies the differences between adjacent pixel values in a controlled manner, ensuring that the embedded data remains imperceptible to the human eye. By partitioning the secret data into message units

and concealing them within the modified differences, the RDHEI method ensures the seamless integration of the embedded information.

One of the key advantages of the RDHEI method is its ability to guarantee error-free extraction of the embedded data. When the recipient decrypts the image, the residual values are obtained by subtracting the decrypted image from the original encrypted image. By applying the reverse process of difference expansion to these residual values, the embedded message units can be accurately extracted without any errors or distortions.

Furthermore,[2] the RDHEI method ensures lossless restoration of the original image after extraction. By combining the decrypted image with the extracted message units, the method reconstructs the original image, preserving its visual quality and integrity. This lossless restoration is crucial to maintain the fidelity of the cover image even after the embedded data has been extracted. The security of the RDHEI method is also a significant focus of research. Extensive efforts have been made to enhance its robustness against various attacks, including statistical analysis and steganalysis techniques. These investigations aim to ensure that the embedded information remains confidential and undetectable, even under rigorous analysis.

In conclusion, the RDHEI method offers a powerful solution for secure data embedding in encrypted images. By leveraging the residual values and operating within the encrypted domain, it guarantees error-free extraction and lossless restoration while preserving the confidentiality of the original image. Ongoing research continues to advance the method's security, performance, and applicability in various domains where secure data hiding is of paramount importance.

## Literature Review

### OVERVIEW OF DATA HIDING

The overview of data hiding techniques refers to a broad understanding of various methods and approaches used to hide data within digital media, such as images, videos, audio, or text. Data hiding techniques aim to embed additional information, often called a "payload," into the host media in a way that is imperceptible to human observers. This hidden data can serve various purposes, including copyright protection, authentication, watermarking, steganography, and covert communication.

Some commonly used data hiding techniques include:

**Least Significant Bit (LSB) Substitution:** This technique replaces the least significant bits of the host media with the bits of the hidden data. As the least significant bits are less noticeable to the human eye, this method ensures a minimal impact on the visual quality of the host media.

**Transform Domain Techniques:** These techniques involve transforming the host media into a different domain (e.g., frequency domain using Fourier Transform or wavelet domain) and then modifying the transformed coefficients to embed the hidden data. The inverse transform is applied to restore the modified media.

**Spread Spectrum Techniques:** Inspired by the concept of spread spectrum communication, these techniques spread the hidden data across the entire host media using pseudo-random sequences. The hidden data is added to the host media by altering the amplitudes or phases of the media's samples.

**Statistical Methods:** These techniques exploit statistical properties of the host media to embed the hidden data. For example, modifications can be made to the histogram of pixel values or the correlation between neighboring pixels to hide the data.

**Steganography Techniques:** Steganography focuses on concealing the presence of hidden data rather than the data itself. It involves hiding data in such a way that it is undetectable to unauthorized observers. Common approaches include modifying the positioning of pixels, using invisible ink techniques, or utilizing specific patterns within the media to hide data.

It's important to note that each data hiding technique has its own advantages, disadvantages, and limitations. The choice of technique depends on factors such as the desired level of imperceptibility, robustness against attacks,

capacity for data embedding, and the specific characteristics of the host media. Researchers continuously explore and develop new techniques to enhance the security and efficiency of data hiding methods.

## OVERVIEW ON ENCRYPTION TECHNIQUES

The overview of encryption techniques involves understanding the fundamental concepts and methods used to transform plaintext data into ciphertext, ensuring its confidentiality and integrity. Encryption techniques play a crucial role in protecting sensitive information from unauthorized access or interception. Here are some key aspects of encryption techniques:

**Symmetric Encryption:** Symmetric encryption, also known as secret-key encryption, employs a single shared key for both encryption and decryption processes. The same key is used to scramble the plaintext into ciphertext and vice versa. Both the sender and the recipient must possess the key to encrypt and decrypt the data. Symmetric encryption algorithms include Advanced Encryption Standard (AES), Data Encryption Standard (DES), and Triple Data Encryption Standard (3DES).

**Asymmetric Encryption:** Asymmetric encryption, also known as public-key encryption, utilizes a pair of mathematically related keys: a public key and a private key. The public key is used for encryption, while the private key is used for decryption. The public key can be freely distributed, allowing anyone to encrypt data, while only the holder of the private key can decrypt it. Asymmetric encryption algorithms include RSA (Rivest-Shamir-Adleman), Diffie-Hellman, and Elliptic Curve Cryptography (ECC).

**Hybrid Encryption:** Hybrid encryption combines the strengths of symmetric and asymmetric encryption. In this approach, symmetric encryption is used for bulk data encryption, while asymmetric encryption is used for securely exchanging the symmetric encryption key. The sender generates a random symmetric key, encrypts the data using symmetric encryption, and then encrypts the symmetric key with the recipient's public key. The recipient uses their private key to decrypt the symmetric key and then uses it to decrypt the data.

**Hash Functions:** Hash functions are cryptographic algorithms that generate a fixed-size hash value or message digest from input data of any size. Hash functions are primarily used for data integrity verification rather than encryption. They ensure that the data remains unchanged during transmission or storage. Popular hash functions include Secure Hash Algorithm (SHA) and Message Digest Algorithm (MD5).

**Key Management:** Key management is a critical aspect of encryption techniques. It involves securely generating, storing, distributing, and revoking encryption keys. Proper key management ensures the confidentiality and integrity of encrypted data. Key management systems often employ key exchange protocols, key generation algorithms, and secure key storage mechanisms.

**Security Levels:** Encryption techniques are evaluated based on their security levels, which involve factors such as key length, resistance to brute-force attacks, vulnerability to cryptanalysis, and the availability of quantum-resistant algorithms. Strong encryption algorithms use longer key lengths and withstand extensive cryptographic analysis.

It's important to note that encryption is just one aspect of a comprehensive security framework. Additional measures, such as secure key exchange, secure storage, and proper implementation, are necessary to ensure the overall security of encrypted data.

## Challenges in combining data hiding and encryption:

Combining data hiding and encryption presents several challenges due to the inherent conflicts and complexities involved in reconciling the objectives and mechanisms of these two techniques. Some of the key challenges are as follows:

**Security:** The primary challenge lies in ensuring the security of both the hidden data and the encryption key. Combining data hiding and encryption should not weaken the security of either technique. Unauthorized access to the hidden data or the encryption key could compromise the confidentiality and integrity of the data. The integration should provide robust protection against attacks such as data extraction, key extraction, or unauthorized modifications.

**Compatibility:** Data hiding and encryption techniques often operate on different domains or require different data representations. Integrating these techniques should not result in data loss or distortion. The compatibility challenge involves maintaining the fidelity of the host media while embedding the hidden data and preserving the integrity of the encrypted data during the embedding process.

**Capacity:** Data hiding techniques require a certain amount of capacity in the host media to embed the hidden data. Encryption, on the other hand, typically expands the size of the data due to cryptographic overheads. Balancing the capacity requirements of both techniques can be challenging, especially when dealing with limited resources or small data carriers. Ensuring sufficient capacity for both encrypted data and hidden data without exceeding the limits of the host media is a significant challenge.

**Robustness:** The integration should ensure the robustness of both the hidden data and the encrypted data. Robustness refers to the ability of the hidden data to survive various attacks or transformations, such as lossy compression, cropping, filtering, or format conversions. Additionally, the encrypted data should remain resistant to cryptanalysis attacks even after the embedding of hidden data.

**Key Management:** Effective key management is crucial when combining data hiding and encryption. Both techniques require appropriate key management practices to maintain the security and integrity of the data. Generating, distributing, storing, and revoking keys for both encryption and data hiding can be complex. Ensuring synchronized key updates and protecting the keys from unauthorized access are key challenges in the integration process.

**Performance:** The integration of data hiding and encryption should not significantly impact the performance of the system. The computational complexity involved in both techniques can pose challenges, especially when embedding hidden data in real-time or resource-constrained environments. Achieving a balance between security requirements and computational efficiency is essential to ensure practicality and usability.

Addressing these challenges requires careful design and consideration of the specific requirements, trade-offs, and constraints of the application. Researchers continue to explore new methodologies and algorithms to overcome these challenges and enhance the integration of data hiding and encryption techniques.

### **Existing methods for secure data embedding in encrypted images**

There are several existing methods for secure data embedding in encrypted images. These methods aim to enable the embedding of additional data into encrypted images while preserving the security and integrity of both the hidden data and the encryption. Here are a few commonly used techniques:

**Reversible Data Hiding in Encrypted Images (RDH-EI):** RDH-EI methods focus on embedding data in encrypted images without decrypting them. These methods leverage the redundancy present in encrypted images and exploit the extra space available for data embedding. By carefully modifying the encryption process or the encryption keys, the hidden data can be embedded and extracted without affecting the decryption process or introducing noticeable artifacts.

**Homomorphic Encryption-Based Methods:** Homomorphic encryption allows performing computations on encrypted data without decryption. In the context of secure data embedding in encrypted images, homomorphic encryption can be used to modify the encrypted image data to accommodate the hidden data. The embedding process is performed on the encrypted domain, ensuring the privacy and integrity of both the image and the hidden data.

**Visual Cryptography-Based Methods:** [3,4] Visual cryptography is a technique that divides an image into shares, where each share individually reveals only a portion of the original image. By combining these shares, the original image can be reconstructed. Secure data embedding can be achieved by encoding the hidden data as additional shares and embedding them in the encrypted shares. The hidden data can be extracted by combining the shares.

**Steganography with Encryption:** Steganography involves hiding data within a cover media without raising suspicion. In the context of encrypted images, steganography can be combined with encryption to achieve secure data embedding. The encrypted image serves as the cover media, and the hidden data is embedded using

steganographic techniques that exploit the cover media's characteristics. The hidden data can be extracted by decrypting the image and recovering the embedded information.

**Key-based Embedding:** Key-based embedding methods utilize a separate encryption key specifically for embedding data. The encryption key is derived from the original encryption key or generated independently. The hidden data is embedded by modifying specific regions of the encrypted image using the derived or independent key. The extraction process involves using the same key to recover the embedded data.

These are just a few examples of existing methods for secure data embedding in encrypted images. Each method has its own advantages, limitations, and security considerations. The choice of method depends on the specific requirements of the application, the level of security desired, and the constraints of the system. Researchers continue to explore and develop new techniques to improve the security and efficiency of secure data embedding in encrypted images.

### Limitations of current techniques

While current techniques for secure data embedding in encrypted images have made significant progress, they still have certain limitations. Some of the limitations of these techniques include:

**Reduced Embedding Capacity:** One limitation is the reduced embedding capacity when compared to traditional data hiding techniques. Since the embedding process operates on encrypted data, the available space for embedding is limited by the size of the encrypted image and the constraints imposed by encryption algorithms. This limitation restricts the amount of data that can be securely embedded within the encrypted image.

**Increased Vulnerability to Attacks:** Secure data embedding techniques in encrypted images introduce additional complexities, which may make them more susceptible to attacks. Adversaries may exploit vulnerabilities in the integration process to extract or modify the hidden data, compromise the encryption, or reveal sensitive information. Ensuring robust security against such attacks is a challenge.

**Loss of Perfect Secrecy:** Many existing techniques do not provide perfect secrecy for the embedded data. In some cases, the extraction of the hidden data requires partial or complete decryption of the encrypted image, which exposes the embedded data to potential risks. Achieving both perfect secrecy and effective data embedding within encrypted images remains a significant challenge.

**Increased Computational Complexity:** The integration of data embedding and encryption can result in increased computational complexity. Additional processing steps are required for both embedding and extraction, which can impact the system's performance. Real-time applications or resource-constrained environments may face challenges in achieving efficient and fast processing.

**Limited Compatibility:** Current techniques may have compatibility limitations when it comes to different encryption algorithms, image formats, or applications. Some methods may be specific to certain encryption schemes or rely on particular image characteristics, making them less applicable in broader scenarios. Ensuring compatibility and interoperability across different systems and platforms is an ongoing challenge.

**Lack of Standardization:** The field of secure data embedding in encrypted images lacks standardized techniques, algorithms, or evaluation metrics. This lack of standardization hampers the comparability and reproducibility of different approaches, making it difficult to assess their effectiveness and security. Establishing standardized benchmarks and evaluation frameworks would facilitate advancements in the field.

Addressing these limitations requires ongoing research and development efforts. Future advancements may focus on improving embedding capacity, enhancing security against attacks, optimizing computational efficiency, ensuring compatibility across different systems, and establishing standardization guidelines for evaluating techniques in this domain.

### RDHEI Methodology

**Overview of the RDHEI method:** Methodology of the RDHEI (Reversible Data Hiding and Encryption Integration) method in detail.[1] The RDHEI method aims to enable secure data embedding in encrypted

**Encryption Phase:** The original image is encrypted using a symmetric or asymmetric encryption algorithm, generating the encrypted image (ciphertext). The encryption process ensures the confidentiality of the image content. The encryption key is securely stored or transmitted to the intended recipient.

**Data Embedding Phase:** The hidden data to be embedded (payload) is encrypted separately using a different encryption key (data hiding key). The payload can be any additional information that needs to be concealed within the encrypted image.

The encrypted payload is then embedded into the encrypted image. This embedding process takes advantage of the redundant or least significant bits (LSBs) in the encrypted image to hide the payload without affecting the encryption or introducing noticeable artifacts. The embedding process guarantees that the original image data and the encrypted image remain intact and indistinguishable.

**Data Extraction and Restoration Phase:** The recipient receives the encrypted image and has access to the encryption key used in the encryption phase. The recipient extracts the encrypted payload from the encrypted image using the data hiding key. The extracted encrypted payload is decrypted using the appropriate decryption algorithm and the corresponding decryption key to obtain the original hidden data. The original image can be reconstructed by decrypting the encrypted image using the encryption key.

The RDHEI method ensures error-free extraction and lossless restoration by carefully integrating the data embedding process with encryption, avoiding any modifications to critical information during the embedding and extraction phases.

### Encryption process

The encryption process involves transforming plaintext data into ciphertext to ensure its confidentiality and integrity

1. **Select an Encryption Algorithm:** Choose a suitable encryption algorithm based on security requirements, key length, and algorithm strength. Commonly used encryption algorithms include Advanced Encryption Standard (AES), Data Encryption Standard (DES), and RSA.
2. **Generate or Obtain Encryption Keys:** Encryption algorithms require one or more encryption keys to encrypt and decrypt data. These keys can be randomly generated or obtained through a key exchange protocol, depending on the encryption scheme used. Symmetric encryption uses a single shared key, while asymmetric encryption uses a key pair consisting of a public key and a private key.
3. **Preprocessing (Padding and Initialization):** Before encryption, the plaintext data may undergo preprocessing steps such as padding and initialization. Padding ensures that the data meets the encryption algorithms block size requirements. Initialization involves setting up the initial state or parameters for the encryption process.
4. **Key Expansion (Symmetric Encryption):** In symmetric encryption, the shared encryption key may undergo key expansion to generate a set of round keys. These round keys are used in each round of the encryption process to add complexity and enhance security.
5. **Encryption Rounds:** The encryption process typically involves multiple rounds, each of which applies a combination of substitution and permutation operations to the data. The exact operations and number of rounds depend on the encryption algorithm being used.
6. **Substitution:** Substitution involves replacing specific elements or bit patterns within the data with different values based on the encryption key or algorithm's rules. Substitution operations help to scramble the data and make it less recognizable.
7. **Permutation (Transposition):** Permutation involves rearranging the positions of the data elements or bits according to predetermined patterns or algorithms. Transposition operations further enhance the randomness and complexity of the encrypted data.
8. **Finalization and Output:** After completing the encryption rounds, any necessary finalization steps are performed, such as applying additional operations, truncating or padding the output, or performing checksum calculations. The resulting ciphertext is then generated as the encrypted representation of the original plaintext data.

The encryption methodology ensures that the encryption key, the encryption algorithm, and the encryption process collectively provide confidentiality and integrity to the data. The resulting ciphertext can only be decrypted using the corresponding decryption key and the reverse process.

## DATA EMBEDDING PROCESS

The data embedding process involves hiding or embedding additional data within a carrier medium, such as an image, while maintaining the integrity and visual quality of the carrier

1. **Select a Data Embedding Technique:** Choose an appropriate data embedding technique based on the specific requirements and characteristics of the carrier medium. Different techniques, such as Least Significant Bit (LSB) substitution, transform domain techniques (e.g., Discrete Cosine Transform), or spread spectrum techniques, may be used.
2. **Prepare the Carrier Medium:** Preprocess the carrier medium to ensure it is suitable for data embedding. This may involve converting the carrier medium into a suitable format or applying any necessary modifications to accommodate the embedding technique.
3. **Convert the Data:** Convert the data to be embedded into a suitable format or representation compatible with the chosen embedding technique. This may involve converting the data into binary format or applying data transformations based on the embedding technique's requirements.
4. **Embedding Process:** The embedding process typically involves the following steps:
5. **Select Embedding Locations:** Determine the specific locations within the carrier medium where the data will be embedded. These locations can be predetermined or determined dynamically based on the embedding technique.
6. **Modify Carrier Data:** Modify the selected carrier data at the embedding locations according to the embedded data. This modification can be achieved through various techniques such as replacing least significant bits, modifying coefficients in transform domains, or using spread spectrum techniques to embed data within specific frequency ranges.
7. **Ensure Data Capacity and Quality:** Monitor the capacity of the carrier medium to ensure it can accommodate the entire data to be embedded. Additionally, ensure that the modifications made during embedding do not significantly degrade the quality or perceptual characteristics of the carrier medium.

**Data Extraction:** To extract the embedded data, the following steps are typically performed:

**Identify Extraction Locations:** Identify the locations within the carrier medium where the embedded data is located. These locations are determined based on the embedding technique used.

**Extract Data:** Retrieve the embedded data from the carrier medium by reversing the modifications made during the embedding process. This may involve extracting and assembling bits or performing inverse transformations, depending on the embedding technique used.

**Reconstruct the Original Data:** Reconstruct the original data by converting the extracted data from its embedded format back into its original representation.

The data embedding methodology aims to seamlessly integrate the additional data within the carrier medium while minimizing any noticeable changes or degradation in the carrier's quality. The specific steps and techniques used may vary depending on the chosen embedding method and the characteristics of the carrier medium.

### Error-free extraction and lossless restoration mechanism

The error-free extraction and lossless restoration mechanism in secure data embedding refers to the ability to accurately retrieve the hidden data from the carrier medium without introducing any errors or loss during the extraction and restoration process.

**Embedding Process:** [6] During the data embedding process, the hidden data is carefully integrated into the carrier medium using techniques that ensure the integrity and preservation of both the carrier and the hidden data. The embedding process should avoid modifying critical information or introducing irreversible changes that may result in data loss or corruption.

**Extraction Process:** The extraction process involves retrieving the embedded data from the carrier medium without any errors or loss. This process must accurately identify and extract the embedded data while preserving its original form and content. The extraction process should reverse the embedding operations and restore the hidden data without introducing any modifications or errors.

**Error Correction and Integrity Checking:** To ensure error-free extraction and lossless restoration, error correction and integrity checking mechanisms can be employed.

Error correction codes, such as Reed-Solomon codes or Hamming codes, can be used to detect and correct any errors that may have occurred during the embedding or transmission process. Integrity checks, such as checksums or cryptographic hashes, can be used to verify the integrity of the extracted data and ensure that it has not been modified or tampered with.

**Redundancy and Error Resilience:** Redundancy can be introduced during the embedding process to enhance error resilience and facilitate error-free extraction.

By adding redundant information or duplicating essential data, the extraction process can recover the embedded data even in the presence of errors or distortions in the carrier medium.

**Quality Evaluation and Metrics:** The error-free extraction and lossless restoration mechanism can be evaluated using appropriate metrics and quality measures.

Metrics such as Peak Signal-to-Noise Ratio (PSNR) or Structural Similarity Index (SSIM) can be used to assess the visual quality and fidelity of the restored data compared to the original hidden data. Other metrics specific to the nature of the embedded data, such as bit error rates or distortion measures, can also be employed to evaluate the accuracy and integrity of the extracted data.

### **Key management and security considerations**

Key management and security considerations are crucial aspects when dealing with secure data embedding in encrypted images.

**Key Generation:** The generation of encryption keys is a critical step. Keys should be generated using strong random number generators and should possess sufficient entropy. Key generation processes should adhere to industry best practices and cryptographic standards.

**Key Distribution:** Secure distribution of encryption keys is essential to ensure that only authorized parties have access to the keys. Secure channels, such as secure communication protocols or physical delivery mechanisms, should be employed to distribute keys securely. Key exchange protocols, such as Diffie-Hellman key exchange, can be used to establish secure communication channels.

**Key Storage:** [7] Encryption keys should be securely stored to prevent unauthorized access. Employing secure key storage mechanisms, such as hardware security modules (HSMs), smart cards, or secure key vaults, helps protect keys from physical and logical attacks.

**Key Revocation and Renewal:** In scenarios where key compromise or unauthorized access is suspected, key revocation and renewal mechanisms should be in place. Revoked keys should be invalidated to prevent their future use, and new keys should be generated to replace them.

**Key Length and Strength:** The strength of encryption keys is crucial to ensure the security of the encrypted data. Longer key lengths and stronger key algorithms provide increased resistance against brute-force attacks. Encryption algorithms and key lengths should adhere to current cryptographic standards and recommendations.

**Authentication and Authorization:** It is important to authenticate and authorize users or systems that have access to the encrypted images and the embedded data. Implementing robust authentication mechanisms, such as username-password combinations, biometrics, or multi-factor authentication, helps prevent unauthorized access.

**Secure Transmission and Storage:** When transmitting or storing encrypted images, secure protocols and mechanisms should be used to protect the confidentiality and integrity of the data. Secure network protocols, encryption of data in transit, and secure storage systems can mitigate the risk of data breaches.

**Adversarial Attacks:** Consideration should be given to potential adversarial attacks aimed at extracting or tampering with the hidden data or the encrypted images. Techniques such as watermarking, digital signatures, and integrity checks can be employed to detect and mitigate such attacks.

**Compliance and Legal Requirements:** Ensure compliance with relevant legal and regulatory requirements regarding data privacy, security, and encryption. Understand the specific obligations related to data protection and implement appropriate measures to adhere to these requirements.

By effectively managing encryption keys and considering key-related security aspects, organizations can strengthen the overall security posture of data embedding in encrypted images, protecting the confidentiality, integrity, and authenticity of the data.

## Implementation Details

### Software and tools used

**Programming Language:** Choose a suitable programming language, such as Python, MATLAB, or C++, for implementing the RDHEI method.

**Cryptographic Libraries:** Utilize cryptographic libraries or frameworks, such as OpenSSL or Cryptography.io in Python, to handle encryption and decryption operations.

**Image Processing Libraries:** Employ image processing libraries like OpenCV or PIL (Python Imaging Library) to manipulate the carrier images and perform data embedding/extraction operations.

**Additional Libraries:** Depending on specific requirements, additional libraries or tools may be used for specific functionalities, such as data compression, error correction coding, or statistical analysis.

**Experimental Setup: Image Dataset:** Select a suitable dataset of images to be used as carrier images for the experimental evaluation of the RDHEI method. The dataset should represent a diverse range of image types, sizes, and contents.

**Payload Data:** Prepare the payload data that will be embedded into the carrier images. The payload data can vary based on the specific experiment or scenario, such as text, images, audio, or other file formats.

**Encryption Algorithms:** Choose appropriate encryption algorithms, such as AES, DES, or RSA, based on the security requirements and experimental objectives.

**Embedding Technique:** Implement the specific data embedding technique proposed in the RDHEI method, such as LSB substitution, transform domain embedding, or spread spectrum techniques.

**Evaluation Metrics:** Define performance metrics to assess the effectiveness of the RDHEI method, such as Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), Bit Error Rate (BER), or Capacity-Visual Fidelity trade-off measures.

### Performance Metrics:

**Capacity:** Measure the capacity of the carrier images to determine the maximum amount of hidden data that can be embedded while maintaining visual quality.

**Visual Quality:** Evaluate the visual quality of the carrier images after data embedding by calculating metrics like PSNR or SSIM. Higher values indicate better visual fidelity.

**Extraction Accuracy:** Assess the accuracy of the data extraction process by comparing the extracted payload data with the original hidden data. Calculate metrics like BER to quantify any extraction errors. **Robustness:** Evaluate the robustness of the RDHEI method against common attacks or distortions, such as compression, noise, or cropping, by measuring the extraction accuracy under these scenarios.

**Reversibility:** Measure the reversibility of the embedding process by evaluating the lossless restoration of the original carrier image after the extraction of hidden data.

**Error-Free Extraction:**

Evaluate the accuracy of the data extraction process by comparing the extracted payload data with the original hidden data. Calculate metrics such as Bit Error Rate (BER) or Hamming distance to quantify the number of extraction errors. Analyze the impact of various factors, such as embedding capacity, encryption algorithm, or noise levels, on the error-free extraction performance.

Compare the error rates achieved by the RDHEI method with other existing methods for secure data embedding in encrypted images.

**Lossless Restoration:**

Evaluate the quality of the restored carrier image after the extraction of hidden data. Measure the visual fidelity of the restored image using metrics like Peak Signal-to-Noise Ratio (PSNR) or Structural Similarity Index (SSIM). Analyze the impact of different embedding techniques, encryption algorithms, or image distortions on the restoration quality.

**Statistical Analysis:** Perform statistical analysis to assess the significance of the obtained results. Use appropriate statistical tests, such as t-tests or ANOVA, to compare the performance of the RDHEI method under different experimental conditions. Consider the confidence level and p-values to determine the statistical significance of the results.

**Discussion and Interpretation:** Analyze the obtained results in the context of the research objectives and hypotheses. Discuss the strengths and limitations of the RDHEI method in achieving error-free extraction and lossless restoration. Identify factors that may affect the performance, such as image characteristics, embedding capacity, or encryption key length. Discuss any trade-offs between embedding capacity, restoration quality, and computational complexity in the RDHEI method.

**Future Directions:**

Suggest potential improvements or optimizations to enhance error-free extraction and lossless restoration in future iterations of the RDHEI method.

Discuss possible research directions or areas of further investigation based on the insights gained from the analysis of the results.

**APPLICATIONS**

**Confidential Data Protection:**[1,8] RDHEI can be applied in scenarios where sensitive or confidential data needs to be embedded and transmitted securely. This includes applications in areas such as secure communication, digital rights management, or secure data storage.

**Forensics and Watermarking:** RDHEI can be utilized for digital forensics and image watermarking purposes. The reversible nature of the method allows for hidden data extraction without any loss, making it useful for authentication, copyright protection, or data provenance in digital images.

**Secure Multimedia Transmission:**[9] RDHEI can contribute to secure multimedia transmission over insecure channels. By embedding additional data in encrypted images, it enhances data security while allowing for error-free extraction and lossless restoration at the recipient's end.

**Steganography and Covert Communication:** RDHEI can be employed in steganography applications, where hidden information is concealed within encrypted images. It enables covert communication or secret message transmission while maintaining the confidentiality of the embedded data.

**Privacy Preservation:** The RDHEI method can assist in privacy preservation by embedding privacy-sensitive data within encrypted images. This can be valuable in applications such as medical imaging, where personal information needs to be protected while ensuring data integrity and accurate extraction.

**Improved Integration and Optimization:** Future research can focus on optimizing the RDHEI method, exploring new embedding techniques, or integrating it with advanced encryption algorithms. This can lead to enhanced performance, improved embedding capacity, and increased security for a broader range of applications.

**Security Analysis and Robustness:** Further analysis and evaluation of the RDHEI method's security against various attacks and vulnerabilities would be valuable. This includes assessing its resilience to cryptographic attacks, statistical analysis, or content-preserving manipulations.

**Cross-Domain Integration:** The principles of RDHEI can be extended beyond image encryption and applied to other data types, such as video or audio, to achieve secure and reversible data hiding. This opens up possibilities for cross-domain applications and multimedia security.

**Standardization and Adoption:** As the RDHEI method matures and gains recognition, it could potentially be considered for standardization by organizations such as the International Organization for Standardization (ISO) or the National Institute of Standards and Technology (NIST). Standardization can promote widespread adoption and interoperability across different systems and platforms.

## CONCLUSION

**Encryption and Security:** The RDHEI method starts with encrypting the cover image to protect its confidentiality. Researchers have investigated the use of different encryption algorithms and their impact on the security of the embedded data. The selection of a robust encryption scheme is crucial to prevent unauthorized access to the image content.

**Difference Expansion Technique:** RDHEI utilizes a difference expansion technique to embed data in the encrypted image. Researchers have examined different approaches to difference expansion and their effects on the extraction process. The goal is to achieve error-free extraction of the embedded information while minimizing any distortions or artifacts in the image.

**Robustness and Attacks:** The research on RDHEI explores the robustness of the method against various attacks. This includes studying the resistance against statistical analysis, steganalysis techniques, and other common attacks that aim to detect or disrupt the embedded data. Enhancing the robustness of the RDHEI method is crucial to ensure the hidden information remains intact and undetectable.

**Extraction and Restoration:** One of the primary goals of the RDHEI method is to enable error-free extraction of the embedded information and lossless restoration of the original image. Researchers have focused on developing efficient algorithms and techniques to achieve accurate extraction and seamless restoration while preserving the integrity of the image.

Overall, the research on the RDHEI method for secure data embedding has explored different aspects such as encryption, difference expansion, robustness against attacks, and extraction/restoration processes. The goal is to develop a reliable and secure method that can embed sensitive information in encrypted images without compromising their confidentiality or introducing any errors or distortions during the extraction and restoration phases.

## REFERENCES:

- [1] Yu, C., Zhang, X., Li, G., Zhan, S., & Tang, Z. (2022). Reversible data hiding with adaptive difference recovery for encrypted images. *Information Sciences*, 584, 89-110.
- [2] Zhang, Y., & Luo, W. (2022). Vector-based Efficient Data Hiding in Encrypted Images via Multi-MSB Replacement. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(11), 7359-7372.
- [3] Yu, C., Zhang, X., Zhang, X., Li, G., & Tang, Z. (2021). Reversible data hiding with hierarchical embedding for encrypted images. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(2), 451-466.

- [4] Wang, P., Cai, B., Xu, S., & Chen, B. (2020). Reversible data hiding scheme based on adjusting pixel modulation and block-wise compression for encrypted images. *IEEE Access*, 8, 28902-28914.
- [5] Nasution, A. S., & Wibisono, G. (2020). An improved of joint reversible data hiding methods in encrypted remote sensing satellite images. In *Advances in Computational Collective Intelligence: 12th International Conference, ICCCI 2020, Da Nang, Vietnam, November 30–December 3, 2020, Proceedings 12* (pp. 252-263). Springer International Publishing.
- [6] Mostafa, G., & Alexan, W. (2019, June). A high capacity double-layer gray code based security scheme for secure data embedding. In *2019 International Symposium on Networks, Computers and Communications (ISNCC)* (pp. 1-6). IEEE.
- [7] Tseng, Y. C., Chen, Y. Y., & Pan, H. K. (2002). A secure data hiding scheme for binary images. *IEEE transactions on communications*, 50(8), 1227-1231.
- [8] Alanazi, A. S. (2021). A dual layer secure data encryption and hiding scheme for color images using the three-dimensional chaotic map and lah transformation. *IEEE Access*, 9, 26583-26592.
- [9] Haque, M. S., & Chowdhury, M. U. (2018). A new cyber security framework towards secure data communication for unmanned aerial vehicle (UAV). In *Security and Privacy in Communication Networks: SecureComm 2017 International Workshops, ATCS and SePrIoT, Niagara Falls, ON, Canada, October 22–25, 2017, Proceedings 13* (pp. 113-122). Springer International Publishing.

