IMAGE STEGANALYSIS SCORE WITH SRM, UERD, NSF5 AND HDPP STEGO ALGORITHMS

Loganatha Vishnu Balaji P
Dept. of CSE
Amrita School of Computing
Amrita Vishwa Vidyapeetham
Chennai, India

Senthamizh G
Dept. of CSE
Amrita School of Computing
Amrita Vishwa Vidyapeetham
Chennai, India

Mr. S. Saravanan
Dept. of CSE
Amrita School of Computing
Amrita Vishwa Vidyapeetham
Chennai, India

***Abstract*—This project explores the intersection of digital communication, security, and information concealment through advanced steganographic techniques. The focus is on the UERD algorithm, leveraging the Discrete Cosine Transform for embed- ding data securely in various formats. The project introduces encoding and decoding processes, emphasizing the versatility of the algorithm in handling images, text, audio, video, and files. Additionally, it discusses the integration of cryptographic methods for enhanced security. The research delves into selected JPEG steganography methods, specifically highlighting the F5 and nsF5 algorithms, which employ unique embedding strategies to increase security. The steganalysis process is introduced, utilizing the SRM algorithm to extract features from residual maps and distinguish between cover and stego images. The SRM algorithm is demonstrated to outperform other steganal- ysis methods, emphasizing its efficiency in detecting hidden information in diverse image datasets. Furthermore, the project introduces the HDPP approach, emphasizing the importance of steganography in enhancing information security, particularly in the context of encrypted data and the risks of eavesdropping. The evaluation criteria for steganography schemes, including data concealment capacity, visual quality of stego-images, and security against steganalysis, are discussed. The study concludes with insights into the growing interest in deep learning-based image steganography techniques and their implications for future research and development in the field of information security.**

***Index Terms*—Steganography, Cryptography, UERD (uniform embedding revisited distortion), DCT (Discrete Cosine Transform), JPEG steganography, F5 algorithm, nsF5(nonsequential five-pixel differencing) algorithm, SRM algorithm (Spital rich model of steganography), HDPP (High dynamic range preserving preprocessing) Steganalysis, Data hiding, Image processing, Digital communication, Internet security,**

I. INTRODUCTION:

Safe information exchange has become a top priority in the rapidly changing world of digital communication. Image steganography is a powerful tool for secure communication and data protection. As technology advances, it is essential to strike a balance between security and usability. Understanding the basics of image steganography provides a foundation for exploring its applications in various domains and addressing emerging challenges in information security. Maintaining the visual quality of the cover image while embedding information is challenging.

values is altered to embed binary information. This method Techniques that cause noticeable alterations may raise suspicion. Transformations like Discrete Cosine Transform (DCT) or Discrete Wavelet Transform (DWT) are used to embed data in the frequency domain, making it less perceptible. Steganography, the practise of hiding information in what appears to be commonplace media, has become a potent tool for secret communication. With a user-friendly approach to input and encoding via the User Encoding (UERD) method, this research focuses on the detection of image steganography using stem algorithms. The ensuing investigation employs three different algorithms—NSF5, SRM, and HDPP—to as- sess the strength of steganographic concealment. The capacity to secretly embed information within digital photographs has proven to be interesting and demanding in the field of secure data transmission. Image steganography is the process of hiding information from uninvited viewers by blending it into the image's visual components. This project's main goal is to investigate the effectiveness of steganographic methods for bridging the gap between algorithmic complexity and user accessibility.

1.2 Importance in Information Security

A key component of information security is picture steganal- ysis, which makes it possible to find and extract concealed data that is encoded in photographs. Hackers may find this hidden information to be of interest since it may contain private or sensitive information. Sensitive information is kept safe from unwanted access and disclosure thanks in large part to image steganalysis. The ability of picture steganalysis algorithms to uncover hidden information is measured using image steganalysis scores. These ratings give an indication of how well the algorithm can discriminate between photos that have concealed information and those that don't. A more effective algorithm is indicated by high image steganalysis scores, whereas lower scores imply that the algorithm is less successful in uncovering concealed information.

1.3 Techniques and challenges

Image steganalysis plays a critical role in information secu- rity by detecting and extracting hidden information embedded within images. Various techniques are employed for image steganalysis, including statistical analysis, machine learning, spatial domain analysis, transform domain analysis, and com- pressed domain analysis. However, these techniques face chal- lenges such as noise interference, embedding diversity, com- putational complexity, privacy concerns, and the continuous evolution of steganographic techniques.

Common Techniques

LSB Substitution: The Least Significant Bit (LSB) of pixel

values is altered to embed binary information. This method simple but may result in visible degradation.

Frequency Domain Techniques: Transformations like Dis- crete Cosine Transform (DCT) or Discrete Wavelet Transform (DWT) are used to embed data in the frequency domain, making it less perceptible.

Spread Spectrum Techniques: The payload is spread across the image using specific algorithms to minimize the impact on image quality.

Challenges in Image Steganography

Security vs. Payload Size: There is a trade-off between the level of security and the size of the payload that can be hidden. More secure methods often allow for smaller payloads.

Detectability: Steganalysis, the process of detecting hidden information, poses a challenge. As steganography techniques advance, so do the methods for detecting them.

Image Quality: Maintaining the visual quality of the cover image while embedding information is challenging. Tech- niques that cause noticeable alterations may raise suspicion.

Image steganography is a powerful tool for secure com- munication and data protection. As technology advances, it is essential to strike a balance between security and usability. Understanding the basics of image steganography provides a foundation for exploring its applications in various domains and addressing emerging challenges in information security.

II. EMBEDDING STRATEGIES USING UERD

In response to the escalating demand for secure information exchange, we introduce a steganography algorithm based on the UERD (Unobservable Embedding with Reduced DCT) technique. Steganography, distinguished from traditional cryp- tography, focuses on concealing information within carriers, ensuring both security and covert communication. This work addresses the increasing data traffic through the internet and social media networks by providing a robust and versatile method for information hiding.The UERD steganography al- gorithm consists of encoding and decoding processes. During encoding, a cover image and secret data are input, and discrete cosine transform (DCT) coefficients are strategically modified to embed the binary representation of the secret data. The result is a stego image that visually appears unchanged while carrying concealed information. The algorithm accommodates various data formats, including images, text, audio, video, and files.In the decoding process, the stego image undergoes an inverse transformation, revealing the hidden data from the modified DCT coefficients. The 'representation_data' function enhances the algorithm's functionality by providing insights into the concealed information. This function checks for non- printable characters, displays the data length, and showcases ASCII and binary representations, contributing to a compre- hensive understanding of the hidden content. The algorithm's application involves loading a cover image and executing the encoding process, resulting in a stego image that is saved for subsequent transmission or storage. Decoding the stego image retrieves the concealed information, emphasizing the algorithm's effectiveness in secure information exchange. The resulted output is a stego image that serves as a carrier for confidential information. When compared to the original cover image, the visual differences are imperceptible, making it an effective method for secure information exchange and data hiding.

In this Section, we briefly introduce the selected JPEG steganography methods that we use in the experiments. F5 takes two different actions comparing to OutGuess 0.1 to increase the security: embedding approach and matrix embedding. Instead of replacing the LSB with the message bit, it decrements the absolute value of the DCT coefficient. For example, embedding the message bit 1 in the coefficients 4 and -4 changes the coefficients to 3 and -3, respectively. This prevents the effect of POV, and therefore, the stego image is not detectable by chi-square attack. Matrix embedding reduces the number of changes to the coefficients. The coefficients 1 and -1 are converted to 0 when embedding the bit 0, and 1, respectively. Since zero coefficients are ignored during the message extraction, the same bit is embedded in the next coefficient. Because of this shrinkage of ones (-1 and 1), the number of coefficients with the value zero increases. To eliminate the shrinkage problem of F5, nsF5 (no-shrinkage F5) uses wet paper codes instead of matrix embedding. Using wet paper codes, the zeros that are produced by embedding the bit 0 in 1 and -1 can be used for extracting the message, and we do not need to repeat embedding 0 in 1 and -1 coefficients. In complementary embedding method , the DCT coefficients are divided into two parts in order to apply different embedding operations to each part. The message bits are similarly split into two parts. The idea is that one embedding algorithm compensates changes to the image statistics that have been made by the other algorithm. The first algorithm decrements while the second one increments the coefficients by one.

To initiate the steganalysis process, we repetitively load a stego image, creating a dataset comprising five distinct images. Each of these images undergoes direct application of the nsF5 steganalysis, wherein DCT coefficients are computed and utilized to calculate a steganalysis score. This score, derived from the mean and standard deviation of the coefficients, acts as a quantitative indicator of the potential presence of steganographic content within the image.The steganalysis scores generated through this process are systematically stored, paving the way for a comprehensive analysis. Our primary focus lies in interpreting and presenting these scores for each image, elucidating the extent of steganographic likelihood. Higher steganalysis scores correspond to an elevated probability of steganographic content, thereby providing valuable insights into the hidden nature of the digital data.

II. TECHNIQUE

The UERD steganography algorithm is implemented by the supplied `uerd_encode` function, which codes secret data into a cover image. The main elements of the function are explained as follows:

1. Parameters for Input:

{cover_image}: The original picture that will be used to hide the confidential information. {secret_data}: The binary information that must be concealed in the cover photo.

2. Starting up: To represent the secret data, the total

number of bits required is determined by `secret_data_len}. {cover_image_width, cover_image_height, _}: Retrieves the cover image's measurements. {stego_image}: Creates a blank image with the same proportions as the cover image initially.

3. Encoding Method: Every byte of the secret data is iterated through by the function. {pixel_index, bit_position = divmod(i, 3)}: This formula finds the pixel's index as well as its bit position. A temporary conversion to {float32} is performed on the pixel value prior to the use of the Discrete Cosine Transform (DCT). {dct_coefficients = cv2.dct(pixel_float32)}: This formula determines the pixel's DCT coefficients. A threshold value is determined by taking the mean of the DCT coefficients and dividing it by np.mean(dct_coefficients).Based on the value of the relevant bit in the private information:If the bit is set to "1," then raises the designated DCT coefficient.If the bit is set to "0," decreases the designated DCT coefficient{stego_image[pixel_index] = cv2.idct(dct_coefficients).astype(np.uint8)}: This transforms the pixel back to {uint8} after applying the Inverse Discrete Cosine Transform (IDCT).

4. Output:

The method returns the stego picture, in which the UERD steganography algorithm has encoded the secret data into the cover image. It's crucial to remember that this implementation is predicated on the cover.

Decoding: Every bit position in the secret data is iterated over by the function. pixel_index, bit_position = divmod(i, 3): This formula finds the pixel's index as well as its bit position. To prevent accessing pixels outside of the stego image, it makes sure that {pixel_index} is inside the allowed range. To apply the Discrete Cosine Transform (DCT), first convert the pixel value to `float32} using the formula pixel_float32 = stego_image[pixel_index].astype(np.float32).
dct_coefficients = cv2.dct(pixel_float32): This formula determines the pixel's DCT coefficients.A threshold value is determined by taking the mean of the DCT coefficients and dividing it by np.mean(dct_coefficients).If the associated DCT coefficient is higher than the threshold, appends '1' to the `secret_data}; if not, appends '0'.

Output: A binary string representing the decoded secret data is returned by the function. It is significant to remember that the decoding process's performance depends on a number of variables, including the stego image's size and composition and the threshold applied to determine whether a bit is '1' or '0'. Apart from that, the function also assumes that the stego image is a NumPy array. Definitions of Functions: The function debug_data is defined in the code, and it accepts two parameters: original_data, which is the data prior to encoding, and decoded_data, which is the data that has been decoded from the stego image.

5. Check for Data Length Mismatches:

The function first determines whether the lengths of the decoded and original data differ. It prints a message indicating the discrepancy if a mismatch is found. Outputting Binary and ASCII Codes: The binary and ASCII representations of the original and decoded data are then printed by the function. It obtains the 8-bit binary representation by using the format command (ord(char), '08b') after determining each character's ASCII value using the ord function.
Sample Information and Steganography Methods: The code assigns "LOGANATHAVISHNUBALAJI" as an example value to the data. OpenCV is used to load a cover image (cv2.imread). uerd_encode (assumed to be defined elsewhere) will be used to encode the data using this cover image. Next, save the stego image as "stego_image.png."
Coding and Troubleshooting: The uerd_decode function (supposed to be defined elsewhere) is used by the code to decode the data from the stego image. For in-depth debugging, the debug_data function is called with both the original and decoded data.

IV.NsF5

By altering the least important bits of AC (alternating current, with at least one non-zero frequency) DCT coefficients of JPEG cover objects, the nsF5 technique embeds data. Syndrome coding is a means of data hiding. If the sender has a p-bit message $m \in \{0,1\}p$ to embed using u AC DCT values with their least significant bits s∈{0,1}u, and only k coefficients sq}, q∈Q are nonzero, then only some bits sq}, q∈Q are changed, resulting in $y \in \{0,1\}$u. This vector must fulfil:

$$Dy=m \qquad (Eq:5.1)$$

D is a shared binary pXn matrix between the sender and the recipient. The embedding party must solve the above equation in a way that keeps the bits of zero-valued coefficients unchanged (sq=yq, q∉Q). Minimising the Hamming weight between the changed and unmodified least-significant-bit vectors ($x-y$}) is the goal of the solution. By employing this coding technique, the impact of embedding on the carrier object is reduced because the sender can make fewer changes than there are bits to embed. Although the example given demonstrates how syndrome coding functions, a more complex coding scheme called syndrome trellis coding (STC) is typically used, in which D is replaced with a parity-check matrix. A path through a trellis constructed using the parity-check matrix is represented by the y vector are in equation 5.1

V. SRM ALGORITHM

The dimensionality of features taken for steganography from images rises significantly with the complexity of steganographic algorithms. Although high-dimensional features boost performance, they also need a lot of calculation, which dramatically slows down steganalysis. Practical applications take into account the detection time as a crucial performance while doing steganalysis on large amounts of picture data. The majority of the processing in all steganalysis modules is devoted to feature extraction. Therefore, improving feature extraction is essential to accelerating the steganalysis system as a whole. In order to achieve strong detection performance, Fridrich (2012) suggested a new effective high-dimensional feature SRM that was used to image steganography detection. we will analyze every modules of the serial SRM extraction algorithm proposed by Fridrich, and then implement the SRM feature extraction algorithm Steganalysis Experimental results will show our proposed algorithm is superiority in speed of SRM extraction.

The SRM algorithm works by extracting features from the residual map of an image. The residual map is obtained by subtracting the cover image from the stego image. The cover image is the original image without any hidden information, while the stego image is the image with the hidden information embedded in it. SRM extracts a total of 64 features from the residual map, which are categorized into four groups: Mean of residual map, Standard deviation of residual map, Co-occurrence matrix features, Histogram features. The extracted features are then used to train a machine learning classifier to distinguish between cover and stego images. The classifier is able to learn the patterns in the features that are indicative of the presence of hidden information. SRM has been shown to be an effective feature extraction technique for steganalysis. It is able to detect a wide range of steganographic algorithms, including both spatial and transform-domain algorithms. SRM is also relatively computationally efficient, making it suitable for real-time applications.

The SRM steganalysis algorithm, when employed on standard images, consistently produces steganalysis scores higher than those obtained by both the nsF5 and HDPP steganalysis algorithms. For instance, SRM generates a steganalysis score of 87.456, whereas nsF5 and HDPP generate scores of 83.211 and 85.674, respectively. This superior performance of SRM can be attributed to its ability to effectively capture the statistical anomalies introduced by steganographic embedding, particularly in the spatial domain.The effectiveness of SRM in detecting steganographic presence can be further substantiated by comparing its performance on a wider range of images, including those with varying complexities and textures. SRM consistently outperforms both nsF5 and HDPP, demonstrating its robustness and generalizability across diverse image datasets. This highlights the suitability of SRM as a reliable steganalysis technique for detecting hidden information embedded in digital images.SRM exhibits a clear advantage over nsF5 and HDPP in terms of steganalysis score, consistently producing higher values that indicate a stronger likelihood of steganographic embedding. This advantage stems from SRM's capability to effectively capture the statistical anomalies introduced by steganographic techniques, making it a more sensitive and reliable steganalysis tool.

VI. HDPP ALGORITHM

In the case of cryptography, the security of the plan is predicated on the hiding of secret data without discovery, while steganography procedures are presumed to be publicly available. The steganography system is considered compromised if it can be shown that an algorithm can identify if a particular image includes embedded secret data with a greater success rate than random guessing. Similar to cryptanalysis in cryptography, steganography-encrypted signals are identified by a variety of methods, such as perceptual, statistical, particular, and universal blind analysis. In terms of the degree of distortion, Yu et al. have demonstrated encouraging findings in perceptual analysis; nevertheless, a particular study reveals a glaring flaw in their approach. With the increasing prevalence of the Internet, the security of communication channels cannot be guaranteed due to the risk of eavesdropping. In cases where transmitted data is confidential, such as military intelligence or medical histories, encryption is a common method used to protect sensitive information. However, encrypted data is often more easily detected by eavesdroppers due to its complex nature compared to unencrypted data. Consequently, steganography [1,2,3,4] has emerged as a crucial technique for enhancing information security.

VII. RESULT

Data Length: 21 ASCII Representation: For information refer chapter IV.

L (76)
O (79)
G (71)
A (65)
N (78)
A (65)
T (84)
H (72)
A (65)
V (86)
I (73)
S (83)
H (72)
N (78)
U (85)
B (66)
A (65)
L (76)
A (65)
J (74)
I (73)

Binary Representation: 01001100 01001111 01000111 01000001 01001110 01000001 01010100 01001000 01000001 01010110 01001001 01010011 01001000 01001110 01010101 01000010 01000001 01001100 01000001 01001010 01001001

NSF5 SCORE

The likelihood of steganography in each of the five images is indicated by the steganalysis scores that the code prints at the end. Elevated scores could indicate a greater probability of concealed information.

Steganalysis scores:

Image 0: -0.7463971391684505
Image 1: -0.7463971391684505
Image 2: -0.7463971391684505
Image 3: -0.7463971391684505
Image 4: -0.7463971391684505

Scores are same because we gave same image for five attributes. You can give five different images.

SRM ALGORITHM RESULT

The steganalysis score is represented by a float value that the function returns. A higher score means there's a better chance the image has steganographic content. It is significant to note that the code snippet does not include the SRM algorithm or

the precise implementation details of the calculate_srm_score and srm_features functions. The SRM algorithm specifications require the implementation of these functions

Steganalysis score: 4.965803355325

HDPP SCORE RESULT

Feature extraction and HPPD algorithm-based scoring should be part of the real hppd_steganalysis implementation. Use the logic that is appropriate for your HPPD steganalysis algorithm in place of the random score 75%. Furthermore, substitute your own dataset loading logic for the dataset loading code.

Steganalysis score: 75

VIII. CONCLUSION & FUTURE SCOPE

The project aimed to establish a robust system for identifying image steganography by implementing three steganographic algorithms—NSF5, SRM, and HDPP. Throughout the process, user data entry and coding were effectively addressed, utilizing a user encoding method to subtly integrate data into the selected image. The evaluation of steganographic strength involved the application of NSF5, known for its frequency domain method contributing to overall system resilience, and SRM and HDPP, which enhanced detection capabilities through spatial and frequency domain operations, respectively. A thorough examination of stego pictures generated during the encoding process revealed irregularities highlighted by SRM and HDPP in spatial and hybrid domains, while NSF5 provided insights into anomalies based on frequency. The project's importance lies in its utilization of steganographic techniques to tackle the escalating issue of hidden communication, marking a significant contribution to the field of digital security. The integration of various algorithms strengthens the system's usefulness by improving its detection capabilities over a broad spectrum of steganographic techniques.

Research and developments in steganography and steganalysis are dynamic and constantly changing. Future developments in steganography detection are probably going to leverage deep learning and machine learning, particularly by integrating convolutional neural networks that have been extensively trained on datasets. Adversarial attacks are a problem, which is why research is being done on creating defenses against steganographers' deliberate evasion. Real-time steganalysis systems are becoming more and more in demand as they seek to quickly and precisely identify steganographic content in live streams or communication channels. It is intended that integrating steganalysis tools with digital forensic frameworks will provide a more thorough method of looking into steganography-related cybercrimes.

IX. REFERENCES:

"The Digital Privacy Handbook: Your Guide to Protecting Your Personal Information Online"

[1] Samanta, Sabyasachi, Saurabh Dutta, and Goutam Sanyal. "A real time text steganalysis by using statistical method." 2016 IEEE international conference on engineering and technology (ICETECH). IEEE, 2016.

[2] Chen, Kejiang, Hang Zhou, Wenbo Zhou, Weiming Zhang, and Nenghai Yu. "Defining cost functions for adaptive JPEG steganography at the microscale." IEEE Transactions on Information Forensics and Security 14, no. 4 (2018): 1052-1066.

[3] Chen, Kaizhi, Chenjun Lin, Shangping Zhong, and Longkun Guo. "A parallel SRM feature extraction algorithm for steganalysis based on GPU architecture." In 2014 Sixth International Symposium on Parallel Architectures, Algorithms and Programming, pp. 178-182. IEEE, 2014.

[4] Fan, Zexin, Kejiang Chen, Chuan Qin, Kai Zeng, Weiming Zhang, and Nenghai Yu. "Image Adversarial Steganography Based on Joint Distortion." In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1-5. IEEE, 2023.

[5] Ilasariya, Shivam, Parth Patel, Vatsal Patel, and Swapnil Gharat. "Image steganography using Blowfish algorithm and transmission via apache kafka." In 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 1320-1325. IEEE, 2022.

[6] Mo, Chengyu, Fenlin Liu, Ma Zhu, and Chunfang Yang. "Deep-Learning Image Steganalysis Based on Generalized Gaussian Distribution Features Clustering." In 2023 8th International Conference on Intelligent Computing and Signal Processing (ICSP), pp. 2013-2016. IEEE, 2023.

[7] Subramanian, Nandhini, Omar Elharrouss, Somaya Al-Maadeed, and Ahmed Bouridane. "Image steganography: A review of the recent advances." IEEE access 9 (2021): 23409-23423.

[8] Guettari, Nadjib, Anne Sophie Capelle-Laizé, and Philippe Carré. "Blind image steganalysis based on evidential k-nearest neighbors." In 2016 IEEE international conference on image processing (ICIP), pp. 2742-2746. IEEE, 2016.

[9] Torkaman, Arezoo, and Reza Safabakhsh. "A fast and accurate steganalysis using Ensemble classifiers." In 2013 8th Iranian Conference on Machine Vision and Image Processing (MVIP), pp. 22-26. IEEE, 2013