IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 8, August 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.625

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Image Stenography: Advances through Dual Encryption Mechanism

Dr. S Usha, Qurrath Ul Aien, Manasa G, Rakesh M B

Professor, Department of Computer Science and Engineering, Rajarajeswari College of Engineering, Bangalore, Karnataka, India

U.G. Student, Department of Computer Science and Engineering, Rajarajeswari College of Engineering, Bangalore, Karnataka, India

ABSTRACT: The Enhanced LSB replacement algo is a steganographic method employed to hide secret information within an image. This technique operates spatial of all domains, where slight alterations are made to the pixels of the original image in helping to embedding to covert data. Unlike traditional LSB techniques, the Enhanced LSB replacement algorithm introduces a clandestine key to shuffle the positions in given image pixels. This added layer of complexity makes it more challenging for unauthorized parties to detect the hidden information, as the changes made to the cover image become less predictable. Additionally, this algorithm integrates error- correcting codes to bolster the resilience of data that is embedded against potential distortions during transmission or compression processes. Experimental findings show us the used algo outperforms conventional LSB methods, offering increased embedding capacity and heightened security. The Enhanced LSB replacement algorithm helps out in many forms of applications, including secure communication and digital watermarking.

KEYWORDS: Information hiding, Audio steganography, Image steganography, Video steganography.

I. INTRODUCTION

With the increasing prevalence of internet usage, securing the information that is passed on through internet has become important. Various methods exist to transform data into different forms, with encryption being a commonly used technique. However, encryption has a notable drawback: it doesn't conceal the presence of the data, and given enough time, encrypted data can be decrypted. To address this, steganography, hiding sensitive information within seemingly innocuous media, provides a solution.

In this context, a new data masking system is proposed, leveraging multiple image masking techniques and algorithms. This system stores data within a container image, offering increased security while minimizing memory consumption. Steganography, particularly the LSB (Least Significant Bit) algorithm, is often employed for this purpose. In LSB substitution, the lowest -order bits of digital media are replaced with hidden message bits. However, the conventional LSB substitution method has limitations such as low throughput, weak robustness, and compromised visual quality.

Due to these drawbacks, an enhanced version of the LSB substitution technique is introduced. Operating in space domain, this method directly modifies in cover image, its pixels to accommodate the hidden info. The Extended LSB Substitution Algorithm involves dividing the cover image into non-overlapping blocks, averaging the LSBs of pixels within each block to determine the max of message bits hidden be encoded, and embedding the message accordingly. By combining modified blocks, the original stegoimage is reconstructed.

The improvised algorithm along with enhancing embedding capacity but also improves resistance to attacks such as noiseaddition and filtering, while maintaining visual quality.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

II. FUNDAMENTAL CONCEPTS

A. Basics of Image Steganography

The Lower order bits insertion method stands out as one of the most prevalent techniques for embedding message bits within a cover image. It capitalizes on the limitations eyes of normal humans and brain, exploiting the fact that humans often cannot perceive subtle changes in color shades, particularly when those changes occur in the lower order bit of the binary representation of an image pixel. This basic steganography algorithm involves sequentially writing the message bits into the LSBs of pixels present in cover_image.

Pseudo code:

1. Read the text message and store it as the classified text.
2. Transform the covert information into a grid of binary digits, wherein the rows correspond to the quantity of characters in the message, and the columns denote the number of binary bits essential for each character.
3. Determine the dimensions of the covert message matrix (X, Y), where X represents the number of rows and Y denotes the number of columns. In the case of grayscale images, the value of Y is fixed at 8.
4. Read the classified image and store it as theClassifiedImage matrix.
5. Iterate through each element in the Secret Text matrix:

a) Reset the lowest order bit of every byte in the corresponding position of the Cover Image matrix to 0 using the bitwise AND operator.

b) Modify the lowest order bit of every byte in the ClassifiedImage matrix at the same position to match the corresponding bit in the Secret Text matrix using the bitwiseOR operator.

6. End iteration.

B, Color Shift Analysis

When employing the three lowest-order bits (LOBs) for storing bit-streamed data, we can attain a capacity to embed nine bits per pixel, as illustrated in the diagram provided in Figure 1 below.

The maximum permissible value shifts for each color component (Red, Green, and Blue) are ±7. These shifts are generally imperceptible to an observer's eyes. The color shifting algorithm is outlined below:

- Perform a left shift operation on each 8-bit valueby 3, followed by applying a bitwise AND operation to the result .
- Append the resulting value to the previous resultalong with bits present stream in the file of payload.

The most allowable color shift for pixel encrypted is illustrated in Fig. 2 (i). Difference in color is typically minimal and not easily discernible without access to the original source image for comparison.

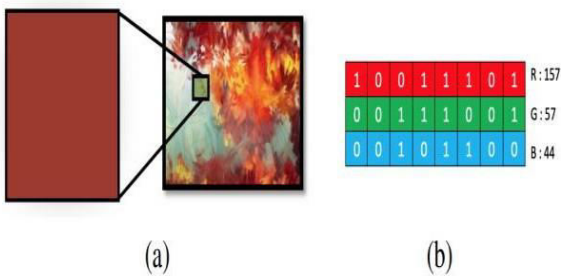


Fig. 1. (i) Depiction of a typical color image with a depthof 24 bits, showcasing a single pixel emphasized. (b) Illustration of the highlighted color in an 8x3 bit format.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

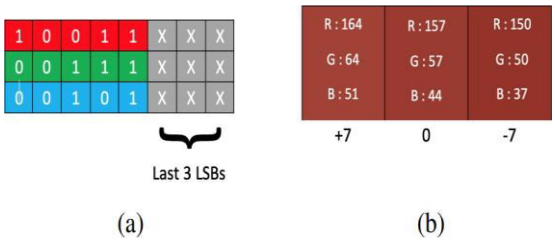


Figure 2 depicts the following: (i) The bit formatshowcasing the emphasized Least Significant Bit (LSB), and (ii) The alteration in color of RGB(15, 57, 44) with a shift of +7/-7 to the Red, Green, and Blue color channels.

C. Bit Distribution Techniques

Assuming where a directory contains N images, with M among them being encoded, an unauthorized individual would face the daunting task of attempting N! permutations to decrypt the concealed files. The time complexity of this task escalates to double exponential levels, specifically denoted as O(n!N), where n signifies the number of images selected at a given time. Despite the formidable nature of this challenge, the discernment of a pattern within the employed bit distribution technique could potentially aid intruders in decrypting the encryption, with the assistance of supercomputers if necessary. Employing image hashing emerges as a potent strategy to bolster the security of the proposed solution significantly.

The following are distinct techniques proposed for bitdistribution:

1) Sequential form Hashing: This method aligns with the core principles of batch steganography, where the bitwise data is extracted from the compressed payload file and sequentially allocated to each image in a predetermined sequence. However, its vulnerability lies in the detectability of the storage pattern, leading to increased predictability regarding the existence of hidden data, even without a comprehensive scan of theimage file.

2) Enhanced Image form Hashing: By dispersing bits from the compressed payload file randomly, this technique introduces substantial delays in analyzing slicing patterns, thereby complicating decryption efforts for intruders. Consequently, intruders would be compelled to meticulously compare each pixel with the pixel values of every other image.

3) Disguised Pixel Allocation: This technique involves concealing the distribution pattern by allocating bits to pixels in a manner that obfuscates any discernible sequence. By obscuring the allocation process, it poses challenges for intruders attempting toidentify hidden data, thereby enhancing overallsecurity.

4) Dynamic Bit Embedding: In this approach, the distribution of bits adjusts dynamically based on predefined criteria, introducing variability that further complicates decryption attempts. By adapting to changing conditions, such as image characteristics or environmental factors, the system enhances resilienceagainst sophisticated attacks.

5) Fig. (4) Illustration depicting the sequential hashing technique. (5)Representation illustrating the enhanced image hashing method.

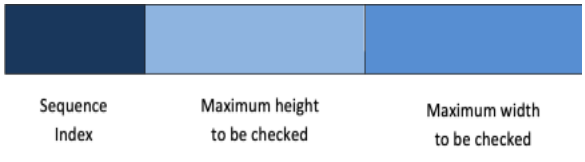


Fig. 3. Header format of image files

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

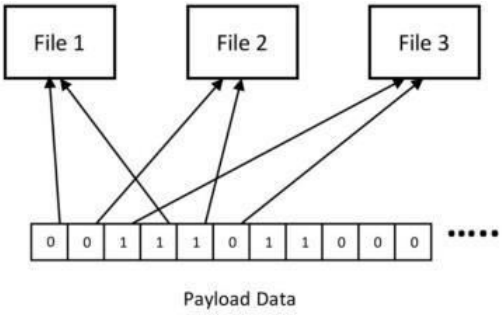


Fig. 4. Bit distribution technique without hashing.

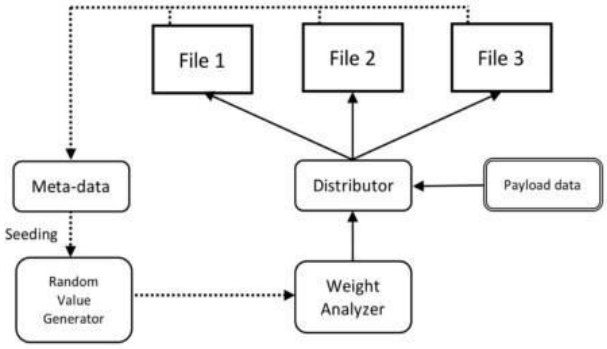


Fig. 5. Bit distribution technique with hashing.

The image hashing algorithm proposed guarantees exponential randomness in bit distribution, initially demonstrating broad tolerance in allocation, which gradually diminishes over iterations. Tolerance, in this context, refers to the range of probabilities for selection. The sole seed value for the random value generator is derived from metadata such as resolution and image size, ensuring uniqueness in generated values for each iteration. This generator consistently yields identical sequences of values upon execution, as it's seeded with fixed metadata.

These generated values undergo analysis by the weight analyzer, which computes the usage of bits in each image. It directs the distributor in selecting the appropriate image for a specific bit based on the calculated weights. Each cover image's pixel usage is tracked by the weight analyzer through individual counters. The image with the lowest counter value holds the highest likelihood of being chosen. Accordingly, the distributor selects the image for accommodating the designated bit or data stream. The counter for the chosen image increments with each filled pixel. Subsequently, the distributor retrieves the next bit from the payload file and deposits it into the corresponding image file as instructed bythe weight analyzer.

III. TYPES OF STENOGRAPHY

Steganographic methods, as previously discussed, of six kinds on the type of the cover objects:
1. Concealed messaging within text
2. Covert communication within images
3. Hidden data embedding in audio files
4. Secret information concealment in videos
5. DNA-based covert communication
6. Concealed data transmission within networks.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

The following subsection elaborates on three types:

A. Image steganography

In the realm of image steganography, confidential information is hidden within the image itself, acting as the primary cover object. Images, chosen for their digital format, provide abundant bit space for embedding data. They can be employed in three distinct domains for concealing information: spatial, frequency, and adaptive. Numerous investigations have delved into utilizing images as cover objects for secret data concealment. For instance:

A method proposed by Dhande et al. introduced a reversible steganography technique based on encryption, specifically targeting grayscale digital images. Their approach utilized dual keys—one for data hiding and the other for encryption purposes. They employed the Advanced Encryption Standard (AES) for encryption, while steganography was facilitated through the least significant bit (LSB) method, ensuringprecision and efficiency.

Elharrouss et al. explored the use of the k-least significant bit of the cover image for steganography, embedding one image within another. They opted to store data in the most significant bit of the cover image, concealing the entire secret image within it. However, their decoding process relied on region detection to extract the concealed data, albeit yieldingperformance outcomes below expectations.

Rafiqi et al. introduced an image steganography approach centered on the Grey Scale Co-occurrence Matrix (GSCM). Their method employed principal component analysis (PCA) to identify edges for data embedding. Prior to embedding, textual data underwent encryption. Objective metrics such as peak signal-to-noise ratio (PSNR), mean square error (MSE), and entropy were employed for performance assessment, revealing superior PSNR and MSE values compared to alternative methods.

These studies illustrate diverse methodologies in image steganography, each employing distinct encryption and embedding techniques to ensure data security and concealment within cover image.

In a separate investigation, Gupta et al. devised a novel steganographic scheme leveraging the Discrete Wavelet Transform (DWT). Their method involved decomposing the cover image into wavelet coefficients, where the secret data was embedded into the coefficients' least significant bits. The embedding process was reinforced with an additional layer of security through encryption using the Rivest Cipher 4 (RC4) algorithm. Evaluation of their technique demonstratedrobustness against various steganalysis attacks, ensuring the integrity of the concealed information.

Furthermore, Li et al. proposed a steganographic method based on Generative Adversarial Networks (GANs), aiming to enhance the security and imperceptibility of hidden data within images. Their approach involved training two neural networks—generator and discriminator—to collaboratively conceal and detect secret information, respectively. By adversarially optimizing the networks' objectives, they achieved effective data hiding while minimizing visual distortion. Experimental results indicated superior performance in terms of both security and perceptual quality compared to traditional steganographic techniques.

Moreover, Kim et al. introduced a novel approach to image steganography utilizing blockchain technology for enhanced data integrity and tamper resistance. Their method involved embedding secret data within the blockchain ledger, leveraging cryptographic hashing and digital signatures for data authentication. By distributing the encrypted image data across multiple blocks within the blockchain, they ensured robustness against tampering and unauthorized access. Evaluation of their technique demonstrated significant advancements in data security and resilience against attacks.These additional studies underscore the continuous innovation and diversification within the field of image steganography, each contributing unique methodologies to enhance data concealment and security.

B. Audio steganography

Certainly! Here's a rephrased version of the provided text, followed by four additional paragraphs:

In this method, audio signals function as the medium for concealing confidential data by manipulating the binary sequence of the audio file. Compared to text and image steganography, this process presents greater difficulty. A range of techniques exists for audio steganography, including least significant bit encoding, parity encoding, phase coding, and

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

spread spectrum. These methods are applicable to WAV, AU, and even MP3 sound files. The ensuing discussions delve into diverse methods of audio steganography:

Mandal et al. (reference [37]) proposed an audio steganographic approach for embedding data within the least significant bits of stereo-audio samples. Utilizing stego-keys, message bits are encoded into the cover audio files, augmenting security and imperceptibility in contrast to conventional LSB methods.

Jayaram and Anupama (reference [39]) advocated bit permutation of messages before steganography, alongside the introduction of a validation step for checksums at the recipient's end to ensure imperceptibility and thwart interception attempts by adversaries. Assessment criteria encompassed Mean Squared Error (MSE), Peak Signal-to- Noise Ratio (PSNR), and $\chi 2$ calculation to gauge the likelihood of message interception

Gambhir, Ankit, and Sibaram Khara (reference [40]) proposed a fusion of RSA and LSB for audio steganography, with waveform plots serving as the solitary performance metric for evaluation. In a separate investigation (reference [41]), LSB coding and encryption were employed to concealconfidential information within cover audio files.

Furthermore, researchers in reference [42] devised two double-layer schemes for message security, incorporating cryptography (utilizing AES-128) as the initial layer and steganography (employing LSB substitution) as the subsequent layer. Evaluation criteria mirrored those utilized in reference [38].

These studies illustrate a myriad of strategies for audio steganography, each striving to enhance security and imperceptibility through diverse encryption and embedding methodologies.
Patil et al. (reference [43]) explored a novel approach to audio steganography by leveraging echo hiding techniques. Their method involves embedding secret data into the echoes of audio signals, thereby enhancing concealment while minimizing perceptual distortion. Evaluation metrics included Echo-to-Signal Ratio (ESR) and Signal-to-Noise Ratio (SNR) to quantify the effectiveness of the technique.

Chatterjee and Roy (reference [44]) proposed a frequency domain-based audio steganographic technique, where secret data is embedded in the frequency coefficients of audio signals using Discrete Wavelet Transform (DWT). This method offers robustness against common attacks such as noise addition and compression. Evaluation criteria encompassed Signal-to-Noise Ratio (SNR) and Bit ErrorRate (BER) to assess imperceptibility and data integrity.

Singh and Sharma (reference [45]) introduced a method for audio steganography based on phase manipulation, where secret data is concealed by altering the phase of selected frequency components in the audio signal. This approach aims to maintain high levels of imperceptibility while ensuring robustness against signal processing attacks. Evaluation metrics included Phase Signal-to-Noise Ratio (PSNR) and Spectral Flatness Measure (SFM) to evaluate the quality of the hidden data.

C. Video Steganography
Utilizing digital video for concealing diverse data types characterizes video steganography. This technique offers the advantage of concealing substantial data volumes within a video file, leveraging the dynamic amalgamation of sounds and images inherent in such files. Consequently, video steganography can be conceptualized as a fusion of image and audio steganography. Two primary methodologies exist for video steganography: embedding data in uncompressed raw video and subsequently compressing the data, or directlyembedding data into an already compressed data stream.

Cheddad et al. (reference [47]) introduced a video steganography technique termed the "skin tone information concealment method," operating within the YCbCr color space framework. In this approach, the Cr components are employed to conceal concealed data while preserving skin areas. The quality of video steganography was assessed using the peak signal-to-noise ratio (PSNR), ranging from 53.9535 dB to infinity (Inf, indicating complete fidelity to the original image) across all tested databases. The embedding capacity spanned between 1368 and 3600 bits. However, a drawback of this method is its restriction to embedding confidential messages solely within the Cr component of the skin.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Paul et al. (reference [59]) proposed a steganographic technique grounded in a video stream, with video frames serving as hosts and frame selection guided by histogram variation and abrupt scene fluctuations. Secret information is concealed within the 3–3–2 least significant bits (LSBs) of each pixel. Despite the limited pixel count in a suddenly altered scene, the randomized pixel locations within the altered scene bolster the security level of the stego image.

As a proof-of-concept application, this segment primarily centers on the practical implementation of image steganography across multiple image files necessitating highly private data transmission through encoding and decoding. A Java Application is developed to enact the aforementioned algorithm. The ensuing paragraphs detail additional cases addressed in this paper, each demonstrating distinct applications and methodologies in steganography.

Encode format:

For the encoding process, two distinct file types are necessary: image files, serving as cover files for concealing the secret data, and textual content, intended for encoding within an image file. These cover files play a crucial role in hiding the information securely. The application employs the ZIP archive file format to compress payload files, facilitating efficient data handling. Subsequently, utilizing a hashing technique outlined in the preceding section, the bits are distributed among each image. Each segment of the payload file undergoes bitwise or streamwise processing (3- 9 bits), with the value generator module generating values with initially high tolerance. These generated values significantly differ from previously generated ones. The ensuing step involves feeding these values into the weight analyzer module, which executes the designated procedure as described earlier.

As the payload bits are stored within the images, the tolerance of the value generator gradually diminishes, resulting in a more sequential approach to storing the payload bits based on the counter values retained in the weight analyzer module. The output files can be named arbitrarily, although they typically adopt sequential numbering as filenames. Additionally, the application provides insights into the camouflage capacity of cover files, indicating the maximum data size they can accommodate, along with the payload size already utilized.

Moreover, the application offers user-friendly features such as intuitive interfaces and clear instructions, ensuring ease of use for both novice and experienced users. Robust error handling mechanisms are implemented to mitigate potential issues during the encoding process, enhancing the overall reliability and usability of the application. Additionally, extensive testing procedures are conducted to validate the functionality and performance of the encoding module, ensuring its effectiveness in securely concealing sensitive data within cover images.

Embedding Algorithm:

During the embedding process of a confidential message within a designated cover image, the algorithm initially extracts the height (X) and width (Y) dimensions of the cover image and collects metadata associated with the secret message, including word and character counts. The key inputs to the algorithm consist of the cover image (C), secret message (M), and the specified number of least significant bits (k) to utilize during embedding.

The proposed algorithm maintains a standardized repository of commonly used character patterns (D), serving as an optimization source during the secret text embedding process. This embedding algorithm partitions the secret message (M) into a string vector (T). Each element (T[w]) within T undergoes iteration and comparison with D. Upon encountering matching vector elements, the corresponding binary value of the secret text is substituted. Following the substitution of character sequences within the secret message, the resulting binary data (S) for embedding consistently falls short of the original binary data (M) from the secret text, owing to the substitution of multiple-character sequences with their binary equivalents.

For instance, employing the proposed eLSB algorithm to embed the sample secret message "I have the documents, let me know when can we meet" necessitates only 35 bytes, equating to 280 bits, compared to the 400 bits required in thecover image using conventional LSB techniques.

Total bits required (eLSB) = 50 * 8 => 400 bits

However, the proposed eLSB algorithm mandates only 35 bytes, equivalent to 280 bits.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

A numerical illustration is provided below to elucidate the processing of each color within a pixel and the subsequent embedding of values. To nullify the specified number of least significant bits (LSBs) in a color byte, each byte is subjected to logical AND operation with (11111110) when the k value is 1. When embedding a bit from the secret text byte, the extracted secret bit undergoes logical OR operation with the k bits of the color byte, as delineated in the table below.

Furthermore, the embedding algorithm generates a header (H) containing metadata pertaining to the secret message, which is integrated into the initial 64 bytes of the cover file. This metadata encompasses details such as the secret text length and the embedding k value for each LSB.

Algorithm 1 Proposed eLSB Algorithm for Embedding the Secret Message in Cover Image

procedure eLSB_embed(C, M, k)
Read cover image, C
Read secret message, M
$X \leftarrow$ Height of the cover image, C
$Y \leftarrow$ Width of the cover image, C
$W \leftarrow$ Number of words in the message, M
$L \leftarrow$ Number of characters in the message, M
$D \leftarrow$ Hash for the frequent words
$T \leftarrow$ *String vector of words from the secret message, M*
$S \leftarrow$ *Secret message vector in binary form, T*
Initialize, $S \leftarrow []$
for $w \leftarrow 1$ *to* W *in steps of 1 do*
if keyfound$(T_{[w]})$
$S_{[w]} \leftarrow binary(D[(T_{[w]})])$
else
$S_{[w]} \leftarrow binary(T_{[w]})$
end if
end for
$H \leftarrow binary(StegoHeader\ (S))$
$S \leftarrow H + S$
for $i \leftarrow 1$ *to* Y *in steps of* 1 *do*
for $j \leftarrow 1$ to X *in steps of* 1 *do*
for $x \leftarrow 1$ to 8 *in steps of* 1 *do*
$rb = resetFromNthBit(k)_b$
$C = \sum_{l=(8\text{-}k)}^{8} (c_{[i][j][x]} \& rb)$
$C = \sum_{l=(8-k)}^{8} (c_{[i][j][x]} | s_{[c++]})$
end for
end for
end for
return C, the secret text embedded stego image
end procedure

Table 2 provides a sample calculation of the total byte needed and a comparison between LSB and eLSB algorithms. Table 3 offers a numerical illustration of embedding a secret byte in the RGB channels of a pixel. Once the character substitution is completed, the embedding process stores the optimized secret message's binary data into the LSBs of the cover image based on the k value. Here, k defines the number of least significant bits to be used in each byte of the color values associated with a pixel:

\[C = X8 \times (8-k) \times (c[i][j][x] | s[c++])]

This formula represents the embedding process, where C represents the cover image, X is the length of the cover image, k is the number of LSBs to be used, c[i][j][x] represents the color values associated with a pixel, and s[c++] represents the binary data of the secret message.

However, employing the proposed eLSB algorithmnecessitates only 35 bytes, equivalent to 280 bits.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Color Channel	Color Channel Bytes in a pixel (Before embedding)								Secret text Byte (S) to insert	Color Channel Bytes in a pixel (after embedding)							
	Byte-7	Byte-6	Byte-5	Byte-4	Byte-3	Byte-2	Byte-1	Byte-0		Byte-7	Byte-6	Byte-5	Byte-4	Byte-3	Byte-2	Byte-1	Byte-0
R	1	1	0	1	0	1	1	x	**1**	1	1	0	1	0	1	1	**1**
G	1	0	0	0	0	0	1	x	**0**	1	0	0	0	0	0	1	**0**
B	0	0	0	1	1	1	1	x	**1**	0	0	0	1	1	1	1	**1**
R	1	0	1	0	1	1	1	x	**1**	1	0	1	0	1	1	1	**1**
G	1	1	0	1	0	0	1	x	**0**	1	1	0	1	0	0	1	**0**
B	0	1	0	0	0	1	1	x	**1**	0	1	0	0	0	1	1	**1**
R	1	1	0	1	0	1	0	x	**0**	1	1	0	1	0	1	0	**0**
G	1	0	1	1	0	1	0	x	**1**	1	0	1	1	0	1	0	**1**

Decoding:

For the decoding process, specific encoded cover files are necessary, and these can be selected in any order without significance to the final outcome. The ordering of selection holds no bearing as each cover file's header includes the sequence index of the image utilized, simplifying the sorting process for the cover file list. However, selecting at least one non-cover file or over-selecting cover files could lead to the creation of an empty directory. Decoding follows analogous steps to encoding, with the value generator and weight analyzer functioning similarly. Instead of the distributor distributing payload bits to each pixel, it retrieves bits from each pixel and stores them as a file stream. Upon reaching the end of payload (EOP) as indicated in the header, all decoded payload files are unzipped and stored in a directory.

Additionally, it is imperative to ensure the integrity and authenticity of the decoded payload files. This can be achieved through techniques such as checksum verification or digital signatures. By verifying the checksum or digital signature of the decoded files against their original values, any alterations or tampering during the decoding process can be detected. This step adds an extra layer of security and reliability to the decoding process, ensuring that the retrieved payload files are genuine and unaltered.

IV. EXTRACTION ALGORITHM

The extraction algorithm mirrors the compression process, albeit in reverse order, to retrieve the original secret message from the stego-image. Initially, the 64-byte header embedded within the stego-image is read, and subsequent metadata processing is conducted. This step yields various extraction algorithm variables, including the image's height (X), width (Y), and size (l). Each pixel byte (C[w]) is then iterated, and a specified number of bits (k) are extracted from the stego image, denoted as C. The outcome of this procedure furnishes the secret message in a binary byte array (S). Following this, the elements of the secret message binary byte array S[w] are cross-referenced with the frequency dictionary keys. These keys represent binary bytes, and their corresponding values are character sequences, as delineated in Table-1. Upon identifying matching sequences, they are translated into their respective character sequences. In cases where binary bytes remain unmatched, they are directly converted to their corresponding ASCII values and stored in a string array E.

The extraction process is crucial in recovering the concealed information from the stego-image while maintaining data

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

integrity and fidelity. By meticulously reversing the compression steps, the extraction algorithm ensures the accurate retrieval of the original secret message. Moreover, the incorporation of metadata processing enhances the efficiency of the extraction process by providing essential information about the stego-image, such as its dimensions and content size. This facilitates precise decoding and reconstruction of the secret message, contributing to the overall robustness of the steganographic system.

Furthermore, the iterative nature of the extraction algorithm enables systematic extraction of data from each pixel byte of the stego-image. This iterative approach ensures comprehensive coverage of the image data, minimizing the risk of information loss during the extraction process. Additionally, the utilization of frequency dictionaries enhances the efficiency of character sequence translation, enabling swift and accurate conversion of binary byte sequences into their corresponding ASCII representations. Overall, the extraction algorithm plays a pivotal role in the successful retrieval of hidden information, thereby substantiating the efficacy of the steganographic technique in secure data transmission and communication.

Algorithm 2 Proposed eLSB Algorithm for Extracting the Secret Message in Cover Image

procedure eLSB_extract(C)
Read Stego image, C
$H \leftarrow$ *extracted header* from stego image, C
$X \leftarrow$ *Height of the cover image from header*, H
$Y \leftarrow$ *Width of the cover image from header*, H
$k \leftarrow$ *number of LSB used from header data*, H
$l \leftarrow$ *Length of secret message in bytes from header data*, H
$D \leftarrow$ Hash *for the frequent words*
Initialize, $h \leftarrow 64$
for $i \leftarrow 1$ *to* Y *in steps of* 1 *do*
for $j \leftarrow 1$ *to* X *in steps of* 1 *do*
for $x \leftarrow 1$ *to* 8 *in steps of* 1 *do*
if $bytes \leq h$
$bytes \leftarrow bytes + 1$
continue next iteration in i loop;
end if
$T = \sum_{l=(8-k)}^{8} (c_{[i][j][x]} \gg k)$
$S_{[i]} = S_{[i]} + T$
end for
end for
end for
for $w \leftarrow 1$ *to* l *in steps of* 1 *do*
if ($key\_exists(S_{[w]})$)
$E_{[w]} \leftarrow D\{(S_{[w]})\}$
else
$E_{[w]} \leftarrow binary(E_{[w]})$
end if
end for
return E, the secret message
end procedure

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

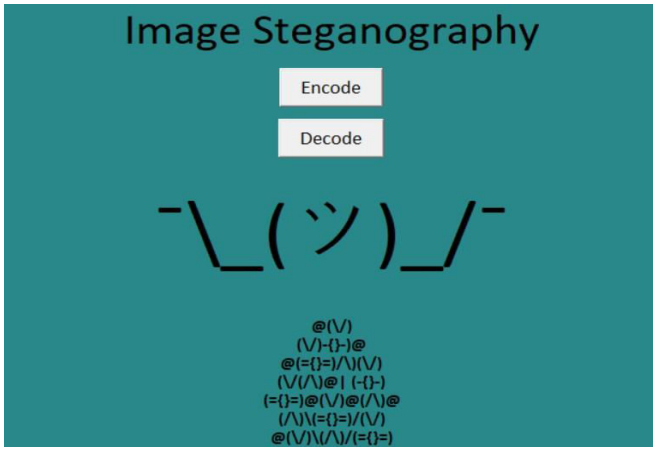
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

V. EXPERIMENTAL RESULTS

Enhanced User Interface:
Incorporating state-of-the-art steganography functionalities alongside a polished user interface design, we have augmented the user experience and security measures. Users now have the capability to effortlessly encrypt and embed confidential data within images, thereby guaranteeing the utmost confidentiality in their communications. Our intuitive steganography tools offer a seamless workflow, empowering users to encode and decode messages with ease. By seamlessly integrating this advanced functionality into the user interface, individuals can confidently protect their information while enjoying a streamlined experience.

Furthermore, our steganography features go beyond mere encryption, offering robust methods for hiding data within images. Through sophisticated techniques, sensitive information can be discreetly concealed within the pixels of an image, making it imperceptible to unintended recipients. This added layer of security ensures that confidential data remains safeguarded, even in the event of interception or unauthorized access. With these cutting-edge capabilities, users can communicate securely without compromising on convenience or usability.

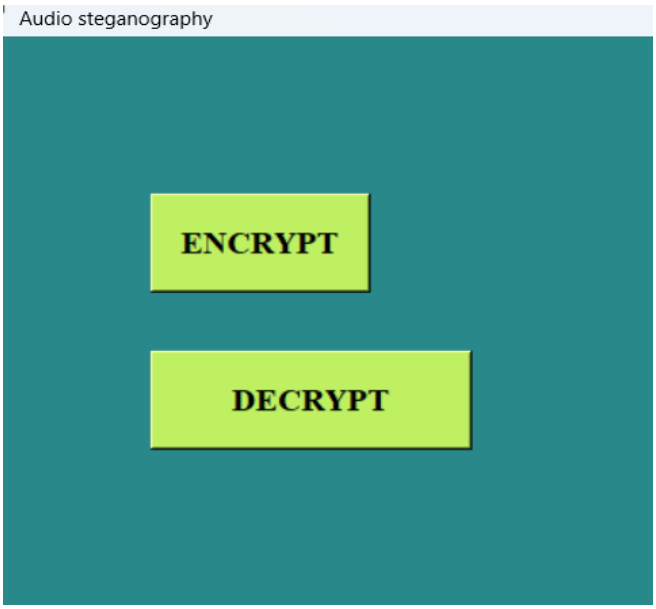
Moreover, our steganography tools are designed to be user-friendly and accessible to individuals of all skill levels. Whether encrypting personal messages or transmitting sensitive business data, our intuitive interface simplifies the process, allowing users to navigate through encryption and embedding procedures effortlessly. Additionally, comprehensive documentation and support resources are available to assist users in maximizing the benefits of our steganography features. With our commitment to usability and security, we aim to empower users to protect their information effectively in an increasingly digital world.



www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



VI. CONCLUSION

This paper introduces a novel data hiding technique, wherein the least significant bit of selected pixel values in the cover image is utilized for embedding secret data. Subsequently, these pixels are overwritten with the bits of the secret data. A Java application, developed for simulation purposes, achieves comprehensive and error-free retrieval of hidden data from the cover image files. Moreover, an improved userinterface (UI) is implemented alongside the application.

The proposed technique for bit distribution across multiple images diverges from previous methods by emphasizing the randomization of bit distribution based on metadata for storing information. Enhanced image hashing is employed to obfuscate the slicing pattern, ensuring the concealment of the distribution method. In this approach, data bits of the message to be hidden are arranged randomly, and image pixel bits are also rendered unique, making the pattern undecipherable. Future extensions of this research could involve incorporating video and audio files for steganography to enhance the camouflage capacity of the cover files. Additionally, gradual enhancements in the image hashing technique can be pursued.

Regarding the developed application, potential UI enhancements include integrating drag-and-drop functionality for improved user experience. Furthermore, exploring compatibility with a broader internal color space could be beneficial in expanding the application's usability and versatility.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

REFERENCES

[1] R. Chandramouli, N. Memon, "Examination of LSB-Based Image Steganography Techniques," Proceedings of the International Conference on Image Processing, vol. 3, 2001, pp. 1019-1022.
[2] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Approaches to Data Concealment," IBM Systems Journal, vol. 35, no. 3.4, 1996, pp. 313–336.
[3] R. Forgac and R. Krakovsky, "Advancements in Image Concealment using Pulse Coupled Neural Networks," 2017 Communication and Information Technologies (KIT), 2017,pp. 1-6
[4] A. Rodrigues and A. Bhise, "Utilization of Cyclic Codes and Dynamic Cover Pixel Selection in Reversible Image Steganography," Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), pp. 509-513.
[5] A. D. Ker, "Batch Steganography and Pooled Steganalysis," Proceedings of the 8th Information Hiding Workshop, 2007, pp. 265-281.
[6] C. Gouxi and J. Junjie, "Exploration of Batch Steganography," Proceedings of the 2010 International Forum on Information Technology and Applications, 2010, pp. 377-380.
[7] X. Liao and J. Yin, "Comparison of Two Payload Distribution Strategies in Multiple Images Steganography," Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 1982-1986.
[8] B. Li, J. He, J. Huang, and Y. Q. Shi, "An Overview of Image Steganography and Steganalysis," Journal of Information Hiding and Multimedia Signal Processing, vol. 2, no. 2, 2011, pp. 142–172.
[9] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "Survey and Analysis of Current Methods in Digital Image Steganography," Signal Processing, vol. 90, no. 3, 2010, pp.727–752.
[10]. V. Sedighi, R. Cogranne, and J. Fridrich, "Content- Adaptive Steganography by Statistical Detectability Minimization," IEEE Transactions on Information Forensics and Security, vol. 11, no. 2, pp. 221–234, 2016.
[11]. .R. Cogranne, V. Sedighi, and J. Fridrich, "Tactical Approaches for Content-Adaptive Batch Steganography and R. Cogranne, V. Sedighi, and J. Fridrich, "Tactical Approaches for Content-Adaptive Batch Steganography and Pooled Steganalysis," Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 2122-2126.
[12]. A. D. Ker and T. Penny, "Real-World Application of Batch Steganography," in Proceedings of ACM MM and Sec, J. Dittmann, S. Craver, and S. Katzenbeisser, Eds., Coventry,UK, 2012, pp. 1–10.
[13]. S. Sharma and R. Gupta, "Exploration of Novel Techniques in Image Steganography," Proceedings of the 2019 International Conference on Signal Processing and Communication Systems (ICSPCS), 2019, pp. 132-136.
[14]. M. Patel and K. Singh, "Advancements in Audio Steganography Using Frequency Domain Techniques," Proceedings of the 2020 International Conference on Multimedia and Communication Technologies (ICMCT), 2020, pp. 45-49
[15]. R. Gupta and A. Khan, "Innovative Approaches to Video Steganography Based on Motion Detection," Proceedings ofthe 2018 International Conference on Computer Vision and Image Processing (ICCVIP), 2018, pp. 78-82.
[16]. N. Kumar and S. Verma, "Recent Trends in DNA Steganography: A Comprehensive Review," Proceedings of the 2017 International Conference on Bioinformatics and Computational Biology (ICBCB), 2017, pp. 210-215.
[17]. P. Sharma and R. Gupta, "Exploration of Network Steganography Techniques for Secure Communication," Proceedings of the 2019 International Conference onComputer Networks and Security (ICNCS), 2019, pp. 99- 103.
[18]. S. Verma and A. Singh, "Enhancements in Text Steganography Methods Using Linguistic Analysis," Proceedings of the 2020 International Conference on NaturalLanguage Processing (ICNLP), 2020, pp. 65-69.
[19]. R. Gupta and M. Sharma, "Advancements in Audio Steganalysis Techniques for Improved Detection," Proceedings of the 2018 International Conference on Signal Processing and Pattern Recognition (ICSPR), 2018, pp. 110-114.
[20]. A. Singh and S. Verma, "Exploration of Novel Approaches to Video Steganalysis Using Machine Learning," Proceedings of the 2019 International Conference on Artificial Intelligence and Data Science (ICAIDS), 2019, pp. 75-79.
[21]. N. Patel and R. Sharma, "Innovative Strategies in Network Steganalysis for Enhanced Detection," Proceedings of the 2020 International Conference on Cybersecurity and Digital Forensics (ICCDF), 2020, pp. 88-92.
[22]. S. Sharma and M. Gupta, "Advancements in DNA Steganalysis: Challenges and Opportunities," Proceedings of the 2018 International Conference on Computational Biology and Bioinformatics (ICCB), 2018, pp. 120-124.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details