ديسمبر ٢٠٢٥م ISSN: 2958-7484
December 2025م Vol 46.pp.87-113
TURJ
العدد (46) مجلة بحوث جامعة تعز
Issue (46) Taiz University Research Journal

Image Steganography using image channels slicing technique and multiple random keys

Tahani Ali Esmaeel*
Faculty of Applied Science
Taiz University
Tahanialamri2015@gmail.com

Abdulmalek Alqobaty
Faculty of Applied Science
Taiz University
Qobaty@Taiz.edu.ye

ملخص - أصبح استخدام التكنولوجيا في الوقت الحاضر جزءًا لا يتجزأ من حياتنا اليومية، وخاصةً عند تبادل المعلومات عبر الإنترنت. يتطلب نقل البيانات الحساسة أساليب أكثر أمانًا لضمان حماية مناسبة للبيانات. الطريقة الأكثر شيوعًا هي إخفاء المعلومات، والتي تلعب دورًا رئيسيًا في حماية المعلومات من خلال إخفائها في غلاف آمن مثل الصور أو الصوت أو الفيديو أو أي وسيط رقمي لا يثير الشكوك. في هذا العمل، نقترح طريقة جديدة لإخفاء المعلومات في الصور الرقمية باستخدام البت الأقل أهمية(LSB) . تستخدم الطريقة المقترحة مفاتيح عشوائية يتم إنشاؤها عبر خوارزمية التجزئة SHA-256 والتعرف العشوائي على مواقع إخفاء البيانات بطريقة غير خطية. تعامل الطريقة قنوات ألوان RGB لصورة الغلاف كطبقة متصلة واحدة. لكل صف من الصورة، يتم ربط قيم البكسل بالترتيب الصارم التالي: أولاً قيم قناة اللون الأحمر للصف، ثم قيم قناة اللون الأخضر، وأخيرًا قيم قناة اللون الأزرق. تتكرر هذه العملية بالتتابع لجميع الصفوف عبر الصورة. تُقسّم الطبقة المتصلة بعد ذلك إلى كتل 32×32، والتي يتم تقسيمها بعد ذلك إلى كتل فرعية أصغر (16 × 16 ثم 8 × 8). يتم تحديد موقع واتجاه تضمين البيانات بناءً على مفاتيح التجزئة. تم تقييم أداء الطريقة المقترحة باستخدام معايير مثل متوسط مربع الخطأ (MSE)، ونسبة ذروة الإشارة إلى الضوضاء(PSNR) ، ومؤشر التشابه الهيكلي .(SSIM) أظهرت نتائج الأداء كفاءة عالية، وحافظت الصورة الناتجة على جودتها البصرية والهيكلية حتى مع وجود كميات كبيرة من البيانات المضمنة.

Abstract

Nowadays, the use of technology has become an integral part of our daily lives, especially when exchanging information over the Internet. The transmission of sensitive data requires more secure methods to ensure proper data protection. The common most widely used method is information hiding. which plays a major role in protecting information by hiding it in a secure cover such as images, audio, video, or any digital medium that does not raise suspicion. In this work, we propose a new method to hide information in digital images using the least significant bit (LSB). The proposed method uses random keys generated via the SHA-256 hashing algorithm and random identifying for data hiding locations in a nonlinear manner. The method treats the RGB color channels of the cover image as a single continuous layer. For each image row, the pixel values are concatenated in the following strict order: first the Red channel values of the row, then the Green channel values, and finally the Blue channel values. This process is repeated sequentially for all rows across the image. The continuous layer is then divided into 32×32 blocks, which are then divided into smaller subblocks (16×16 and then 8×8). The location and direction for data embedding are determined based on hash keys. The proposed method performance was evaluated using benchmarks such as mean square error (MSE), peak signal-to-noise ratio (PSNR), and structural similarity index (SSIM). The performance results demonstrated high efficiency, and the stego images maintained their visual and structural quality even with large amounts of embedded data.

Keywords: Steganography, LSB, Randomization, RGB, Blocks.

1. Introduction

Humans have recognized the importance of protecting information from hostile individuals, or organizations. Over time, various ideas, methods, and techniques have been developed to prevent information leaks or unauthorized disclosure. Before the advent of technology, simple methods were used that aligned with the limited knowledge and abilities of the time like (Wax Tablets, Shove Heads, Invisible Ink and Morse code) (Kumari, Pritam, 2013). The explosive growth of computers and information technology led to the emergence of communication system, some of which contain sensitive information. However, intruders attempt to access sensitive information especially during the transmission. Thus, many techniques have been proposed to ensure information security during transmission process. These techniques are still progressing toward becoming more secure and robust in terms of performance measures (Kadhim et al., 2019). Generally, information security systems are separated into two major categories: cryptography and steganography (Kadhim et al., 2019). Both fields try to protect transmitted message (K. P. and V. K. Sharma, 2014), but in deferent ways. While cryptography protects the content of message using encryption keys, the steganography hides the message into cover media (Haverkamp, Indy; Sarmah, 2024).

Cryptography is the science of converting secret messages into a different form to be exchanged over an insecure channel. Hence, no one can access the correct information unless she know the used key (K. P. and V. K. Sharma, 2014). Based on the selected key, there are two basic types of cryptography techniques: symmetric key and asymmetric key encryption techniques. In symmetric key encryption, a single key is used for both encryption and decryption. This technique, however, requires a secure way to share the encryption key between the sender and receiver. However, asymmetric key encryption involves a public key, which is used for encrypting and a private key which, is used for decrypting (K. P. and V. K. Sharma, 2014). Steganography uses effective techniques for securing transmitted messages by hiding the confidential information within a common media such as images, audio, video, or text. Steganography seeks to achieve three main goals: effectively hiding data, maintaining its integrity, and remaining undetectable. Sometimes, encryption may be combined with steganography to achieve an additional level of security (Ahmed et al., 2014).

Images are more widely used as a cover media to hide information. The spatial domain and transformation domain are the two primary approaches used in image steganography. In spatial domain techniques, secret data is directly embedded in the intensity values of individual pixels (Mohsin & Alameen, 2021), Least Significant Bit (LSB) steganography (Neeta et al., 2006), Most Significant Bit (MSB) steganography (A. Sharma et al., 2018), Pixel Value Differencing (PVD) (Wu & Tsai, 2003), Histogram shifting (Ni et al., 2006), and

Difference expansion(Tian, 2003). On the other hand, transformation domain techniques involve applying transformations on the image to hide data based on its frequency components. Methods like Discrete Cosine Transform (DCT)(Chu et al., 2004)(H. Patel & Dave, 2012), Discrete Wavelet Transform (DWT) (Tolba et al., 2004), and Discrete Fourier Transform (DFT) (Bhattacharyya & Kim, 2011) are commonly used. One of the main features of LSB is that it offers high data embedding capacity without increasing file size. However, embedding a large amount of data can significantly degrade the quality of the stego image (Swain & Saroj(MITS), 2012). This trade-off between capacity and imperceptibility is a common challenge in steganography (Hameed et al., 2022).

In this work, we propose a new secured method for hiding data within digital images using the least significant bit (LSB). It uses random keys generated via the SHA-256 hashing algorithm. Moreover, this method is proposed to enhance security and identify hiding locations in a nonlinear manner. The proposed method begins by concatenating the RGB color channels of a cover image into a single extended layer. This layer is then divided into 32×32-value blocks, which are further divided into smaller subblocks (16×16 and then 8×8). The embedding process uses the subblocks to hide the message. The proposed method is developed using C# programming language and is tested using some common images of 512×512 as a cover object. The experimental results show high imperceptibility and security has been achieved through proposed technique.

The rest of this paper will be organized as follows. The related work is presented in Section 2. In Section 3, we describe the proposed approach. Section 4 introduces the Experimental results and discussion of the proposed method. Finally, the conclusion is presented in Section 5.

2. Related Work

The Least Significant Bit (LSB) technique has been enhanced by employing random distribution patterns to determine data hiding locations, with the aim of improving security, increasing hiding capacity, and maintaining image quality. Several studies have been published in this area, such as the researcher (Swain and Saroj (MITS) 2012) proposed a method based on dividing the image into 8 blocks and dividing the encrypted message using the RSA algorithm into 8 blocks. An index channel within each block was selected based on the highest sum of its values, and the other two channels were used to hide the data in 4 least significant bits (LSBs). The method demonstrated high security and excellent image quality with good data hiding ability. Another method to distribute data within an image pseudo-randomly using LSB and Knight's Tour algorithm was proposed by (Nie et al. 2019). The

method was divided image into small blocks of 4×4 pixels, where the starting point was determined using a secret key to move through the blocks based on the knight's movement, which ensures the data was distributed pseudo-randomly and makes the hidden data more difficult to detect. In the other hand, (Ehsan Ali et al. 2021) proposed an improvement to the

data hiding technique in 24-bit color images using a pseudo-random number generator (PRNG) and LSB technique. The PRNG was used twice: one to select a random pixel within the image to ensure non-repetition, and the second to determine the bit locations within the color channels of each pixel. The results showed greater data hiding ability and increased security.(Abdulraman et al., 2019) suggest a method for image steganography using spatially partitioning the image into 8×8 pixel blocks. The technique relies on representing each block in binary and comparing the bit values with the secret message data, with 6 secret pixels (48 bits) being inserted into each block. The smallest possible number of bits in the last row (LSB) are modified to indicate the location of the hidden data. Experiments have shown that the technique achieves a PSNR above 50 dB even with a 100% steganography ratio. However (Jyoti et al., 2014) proposed using a random pattern to embed the secret data inside the image in a way that makes it difficult for an attacker to decrypt. The image was divided into 4-pixel blocks and the RGB values of the first block are used to locate the key, then the blocks are randomly selected to hide the secret message. Also, (A. Patel & Vekariya, 2022) introduced to divide the image into 4×4 pixel blocks to hide the data inside them. The blocks are divided into input blocks (e-blocks) and index blocks (i-blocks), where the data of the index blocks was hidden in the input blocks. The method works well with different image formats and can hide large texts inside color images while maintaining image quality.

(S.Tamil Selvan, 2022), presented a method that provides three levels of security by using a pseudo-random number generator (PRNG) to generate random pixels, and then hiding the data using an inverted LSB algorithm. The method showed higher visual quality compared to conventional LSB technique. A data hiding technique based on calculating the image energy and cost matrix was proposed by (Khandelwal et al., 2016). The image was divided into 4×4 pixel blocks, and dynamic programming was used to determine the most efficient random path based on energy and cost. The data was hidden in the least significant bits (LSB) of randomly selected pixels, which enhances security and increases the level of difficulty in detection. Moreover, (Kareem et al., 2020) proposed a method for hiding text inside an image by encrypting the text using the 3DES algorithm, then dividing the image into 8×8 pixel blocks. The XOR operation was used between the cipher text bits and the image bits to hide the data, and then the modified image was encrypted again using 3DES. The results showed high security and excellent image quality, with effective resistance to attacks.

A new method based on circular shapes inside color images using the circular Hoff transform to identify circular regions was proposed by (Al-Kateeb et al., 2020). In this method, the text was encrypted using the Caesar algorithm and the ciphertext was distributed over the three-color channels (R, G, B) based on a distribution map that determines the location of the pixels within the circle. The method showed high efficiency in security and image quality with complete data retrieval without errors. While, (Kordov & Zhelezov, 2021), presented an algorithm for hiding texts in color images using a combination of encryption and LSB technique. A random number generator based on Duffing and Circle maps was used to generate random pixel locations and embed the ciphertext within the least significant bits (LSBs) of the color channels. The method showed high efficiency through PSNR tests and histogram analysis, with resistance to statistical attacks and high image quality.

(Saber et al., 2025)proposed a new method for hiding information within color images based on variable hiding centers and dynamic block sizes. The image was divided into four regions, and the block sizes (4×4 or 8×8) are dynamically selected based on a random value derived from the message. The embedding capacity depends on the number of blocks, with 4×4 blocks achieving a larger capacity of up to 2048 bytes compared to 512 bytes for 8×8 blocks. The methodology was evaluated using metrics such as PSNR, MSE, SNR, SSIM, and others, and the results demonstrated high quality of the steganographic image and effective protection against unauthorized retrieval.

(Rahman et al., 2025), They proposed an efficient LSB-based image steganography technique including Magic Matrix, Multi-Level Encryption Algorithm (MLEA), Secret Key, transposition, and flipping. The process starts by flipping and shifting the cover image, then splitting it into color channels (R, G and B), with a focus on the blue channel is divided into four blocks and reordered using a magic matrix. The difference values between the secret message and the red channel are computed, and the resulting data is encrypted using the MLEA algorithm with a secret key. The ciphertext is embedded into the blue channel subblocks via LSB, and the process is repeated until the embedding is complete. The channels are then re-encoded, and the operations are reversed to restore the original image structure. This methodology features a strong integration of encryption, transformation, and bit substitution techniques, achieving an effective balance between stealth capacity, resistance to analysis, and image quality preservation.

(Njoum et al., 2024), proposed a new methodology for hiding data in color images using the LSB algorithm, supported by AVL tree and queue data structures. The method is based on constructing an AVL tree from the green color channel, which is used only as an indicator of the locations of hidden bits in other channels. The hiding process is performed by randomly selecting non-consecutive pixels according to the tree order, without the need for a secret key.

The color channel in which data is embedded is randomly selected (R or B), enhancing the secrecy of the embedding and reducing the possibility of pattern detection. The method demonstrated high hiding capacity with minimal distortion of the cover image, facilitating subsequent retrieval.

(Yakoob, 2025), proposed a system that introduces a hybrid approach that combines image segmentation, LSB steganography, and zigzag steganography. The process begins by calculating the length of the secret message and converting it to a binary vector, then adjusting the dimensions of the cover image to a multiple of 8. The image is then divided into 8×8 pixel blocks, and each block is converted to a 64-element vector using zigzag scanning. Only one-color channel (R, G, or B) is used, from which the least significant bit (LSB) is extracted. Each LSB is replaced by a bit of the message, allowing 8 bytes to be hidden in each block. The process is repeated until the entire message is encapsulated, and the length of the message is hidden in a previously known pixel. Finally, the blocks are returned to their locations, and the carrier image is saved and transmitted to the receiver.

Finally, (Raiyan & Kabir, 2025) proposed a robust LSB-based image steganography framework that integrates randomized encryption and error correction to enhance data security and resilience. This method involves compressing and converting the secret payload into text, followed by pseudo-random shuffling using an SHA-256-derived seed. The shuffled text is then encrypted using the Fernet symmetric cipher. The resulting binary message is embedded into the least significant bits of RGB pixels, achieving a capacity of

3 bits per pixel. SCREEDSOLO demonstrated strong resistance to noise and passive steganalysis while maintaining high image quality.

Despite progress in digital steganography, existing methods still face challenges in balancing security, resistance to statistical attacks, and image quality. This thesis proposes a new approach that integrates RGB channel slicing, dynamic key generation, and pixel shuffling to distribute data uniformly, create non-linear embedding paths, and enhance randomness. This method improves resilience against statistical attacks while preserving high image quality.

Table 1. Comparison of image steganography methods in related works

Reference	Core Technique	Key Advantages	Disadvantages/Limitations
(Swain & Saroj(MITS), 2012)	LSB with RSA and block partitioning	High security and excellent image quality.	Still relies on traditional LSB.
(Nie et al., 2019)	LSB with Knight's Tour algorithm	Pseudo-random data distribution, making it difficult to detect.	Vulnerable if the fixed path or secret key is discovered.
(Ehsan Ali et al., 2021)	LSB with PRNG (Pseudo-Random Number Generator)	High hiding capacity and resistance to statistical analysis.	Relies on simple XOR and PRNG; may be vulnerable if randomization pattern is discovered.
(Abdulraman et al., 2019)	LSB with spatial partitioning	Excellent image quality (PSNR > 50 dB).	May be detectable if changes in the last row are analyzed.
(Jyoti et al., 2014)	LSB with a random pattern and key-based block selection	It makes it difficult for attackers to decrypt due to the random pattern.	May be vulnerable if the first block (a potential single point of failure) is compromised.
(A. Patel & Vekariya, 2022)	LSB with block partitioning (i-blocks, e-blocks)	Effective with different image formats.	Indexing may consume embedding capacity.
(S.Tamil Selvan, 2022)	LSB with PRNG and inverted LSB	Three levels of security and higher visual quality.	The method's complexity may introduce higher computational costs.
(Khandelwal et al., 2016)	LSB with energy/cost matrix and dynamic programming	Enhanced security and more difficult detection.	Higher computational cost due to the dynamic programming algorithm.

(Kareem et al., 2020)	LSB with 3DES and XOR operation	High security and excellent image quality with effective resistance to attacks.	No specific disadvantages mentioned, but complexity can be a challenge.
(Al-Kateeb et al., 2020)	LSB with circular Hoff transform and Caesar cipher	High efficiency in security, image quality, and error-free data retrieval.	Limited to circular shapes within the image.
(Kordov & Zhelezov, 2021)	LSB with an advanced random number generator	Strong resistance to statistical attacks and high quality.	Complexity in key generation.
(Saber et al., 2025)	LSB with dynamic block sizes	Flexible capacity and effective security.	Dependent on a random value derived from the message.
(Rahman et al., 2025)	LSB with a Magic Matrix and MLEA	Strong integration of encryption and transformations.	Adds complexity to the process.
(Njoum et al., 2024)	LSB with AVL tree and queue data structures	High hiding capacity with minimal distortion.	Lack of a secret key could be security vulnerability in some contexts.
(Yakoob, 2025)	LSB with zigzag scanning and segmentation	Innovative hybrid methodology.	May be slow due to multiple transformations.
(Raiyan & Kabir, 2025)	LSB with randomized encryption and error correction	High resistance to noise and passive steganalysis.	The overall framework is complex.

3. Proposed Approach

In this work, we propose a new, robust, and secure enough method to embed message into a cover images. The proposed method considers the cover image as three channels: red, green and blue to embed message into. The main idea of the proposed approach is to treat the three-color channels as a single contiguous layer. This unified layer is then systematically

partitioned into N consecutive blocks of 32×32. Each block of 32×32 is then divided into 4 non-overlapping sub-blocks of 16×16, which in turn are divided into 4 sub-blocks of 8×8.

To embed the data in cover image block, first the embedding block and data embedding direction is selected randomly. To achieve that, each block needs to generate its own key. The key is generated using the SHA-256 algorithm of the 7 most significant bits of the bytes in each block pixel channels. The generated key bytes are combined with the previous block key bytes using XOR. The first block key is an exception. It is generated using a SHA-256 algorithm of the 7 most significant bits selected from the entire cover image.

Using chained keys enhances security and embedding unpredictability, as each key depends on the previous one. However, this dependency also means that if synchronization is lost during extraction or a block becomes corrupted, errors may propagate to subsequent blocks. To minimize such risks, the method integrates a hash-based integrity verification step, in which a SHA-256 hash of the secret message is generated before embedding and compared with the hash of the extracted message during retrieval. Matching hash values confirm that the message was recovered correctly without alteration, while a mismatch indicates potential corruption or synchronization loss.

The block and block keys and selecting the data embedding direction are selected for each block. Hence, the required keys and embedding directions are generated according to the following method:

- The cover image is divided into 32×32 blocks, which are orders sequentially.

$$\text{BlockKey} = \text{SHA256 (7 most significant bits of the inter image)} \quad (1)$$

- The first 32×32 block key is selected randomly from the number of the 32×32 blocks:

$$\text{BlockKey}_{\text{first}} = \text{BlockKey}$$

- The first 32×32 block is selected randomly using the following equation:

$$\text{Block}_{\text{first}} = (\sum(\text{ASCII } (\text{BlockKey}_{\text{first}}))\, mod\, (\text{block counts} + 1) \quad (2)$$

- Remove used block and reorder the others, each time the block count will be decreased by 1.
- The N_{th} 32×32 block is also now selected randomly from current count number of the blocks:

- The N_{th} 32×32 block key is selected randomly from the number of the 32×32 blocks:

$$\text{BlockKey}_{\text{N th}} = \text{SHA256 (7 most significant bits of the current } 32 \times 32 \text{ block)} \quad (3)$$

$$\text{BlockKey}_{\text{N th}} = \text{SHA256(Combine } (\text{BlockKey}_{N_{th}-1}, \text{BlockKey}_{N_{th}}))$$

$$\text{Block}_{N+1_{th}} = (\sum(\text{ASCII } (\text{BlockKey}_{\text{N th}}))\, mod\ (\text{block counts} + 1)$$

The embedding direction is selected from one of the sixteen possible directions illustrated in Figure 1. As explained previously, the first embedding block of 32×32 pixels is randomly selected. The embedding direction of the four 16×16 sub-blocks within each 32×32 block is determined by the following equation:

$$\text{Direction} = (\sum(\text{ASCII } (\text{BlockKey}))\, mod\ 16 \qquad (4)$$

The starting embedding position in 8×8 sub-block is determined by the following equation:

$$\text{StartPosition}_{(b1,b2,b3,b4)} = (\sum(\text{ASCII } (\text{BlockKey}))\, mod\ 64 \qquad (5)$$

Where $\boldsymbol{b1}, \boldsymbol{b2}, \boldsymbol{b3}, \boldsymbol{b4}$ represents four 8*8 sub-blocks.

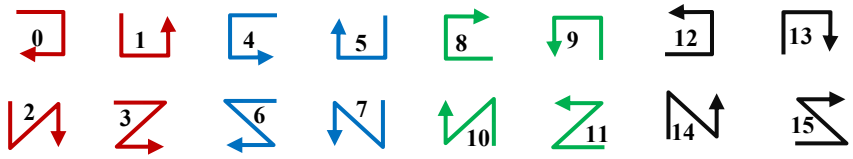


Figure 1: The expected embedding direction

The proposed approach is designed into two methods: embedding method to embed the secret message into the cover image and the other is extracting method to extract the message from the cover image.

Embedding Method

The main function of the embedding method is to embed the secret message into predefined blocks within the cover image. The method treats the RGB color channels of the cover image as a single continuous layer. The color values of pixels are concatenated in an order of red, green, and blue.

To embed the message into the cover image, we use the composed layer, which is divided into 32×32 blocks, which are selected randomly using the equations (2) and (3). The main key is, also, generated using equation (1). The key of each subsequent block is derived using equation (3). The first block key is an exception; it uses the main key that is calculated by equation (1). Each selected 32×32 block is divided into four non-overlapping 16×16 sub-blocks. The embedding process is performed on a 16×16 subblock, which interleaves

embedding on its four 8×8 sub-blocks using a selected direction as shown in Figure 2. The figure shows an example for embedding data into two pixels. In pixel 1 the embedding is start from first 8×8 subblock using the direction 2, while direction 3 is used with pixel 4.

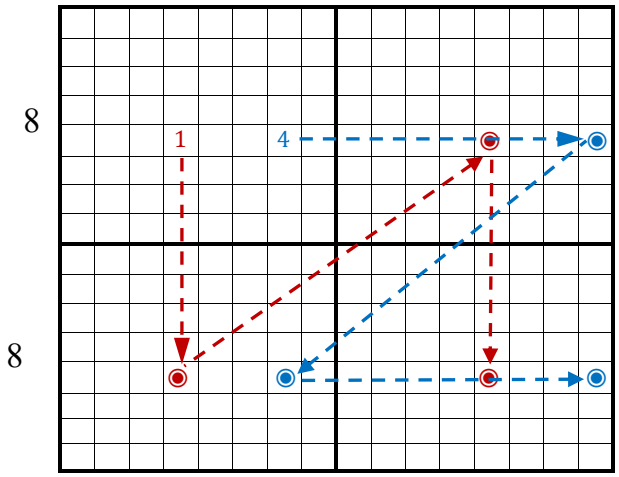


Figure 2: Example of Data Embedding in Four 16×16 Subblocks

Figure 3 shows the message embedding method to hide a message into the 32×32 blocks. In the step 2 the block is divided into four 16 ×16 subblocks. In the step 3 each subblock is further divided into four 8×8 subblocks. Moreover, the starting position of data embedding is determined in one of 8×8 subblocks using equation (5). In the step 4, the data embedding is performed starting from the stating position and in the opposite position in other 8×8 subblocks based in the selected direction that is determined by equation (4) as shown in Figure 2. The process is repeated until the data is embedded on all pixels of the 16 ×16 subblocks. Then move to the next 16 ×16 subblocks. The embedding process is stopped when the message is fully embedded into the cover image.

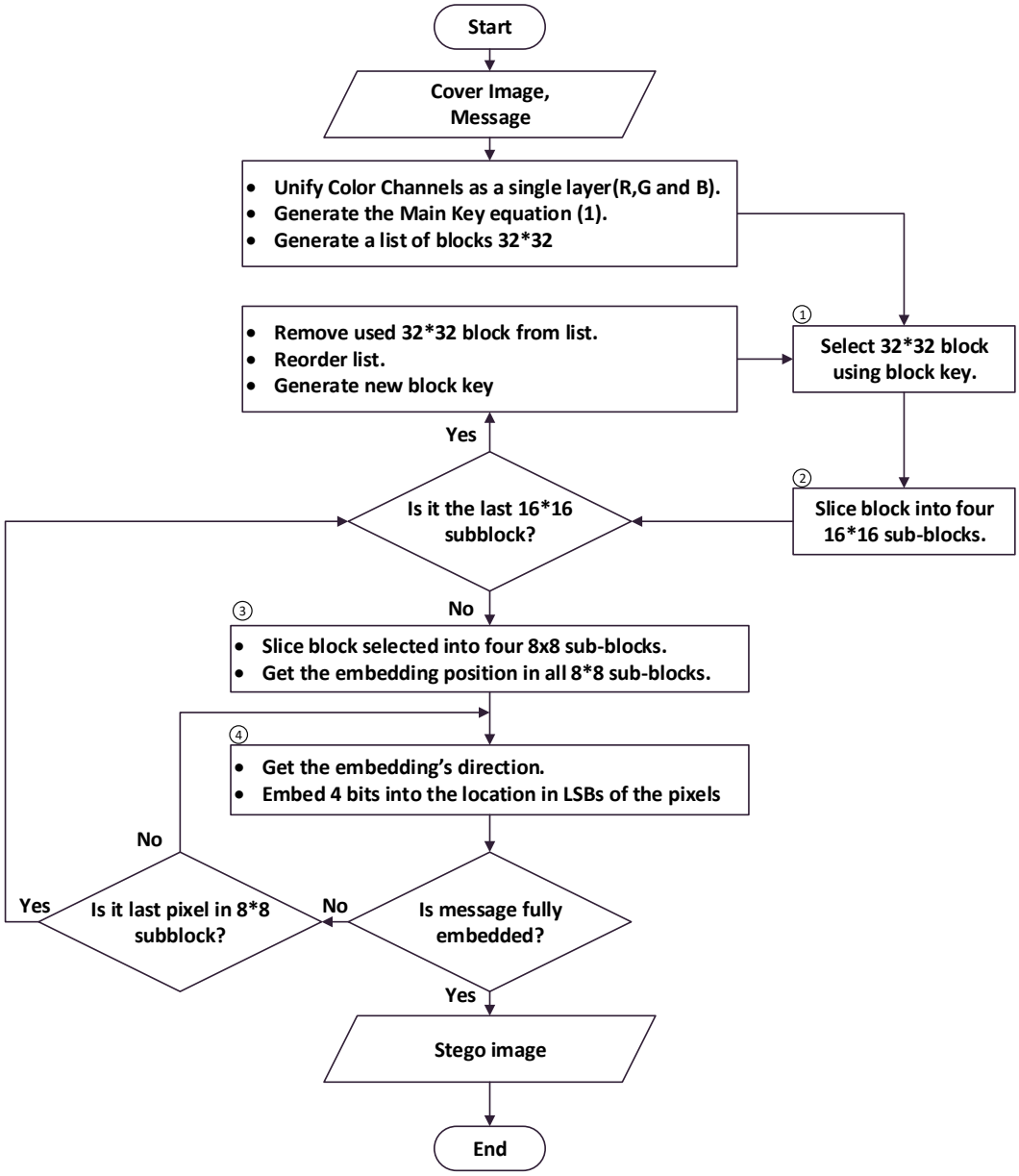


Figure 3: The embedding processes of proposed method

Extracting Method:

The extraction method is the reverse method of the embedding method and follows the same logic and structure to accurately retrieve the hidden data. It relies on regenerating the same block keys and navigating through the image using the same directional and positional rules. The extracting process is shown in Figure 4.

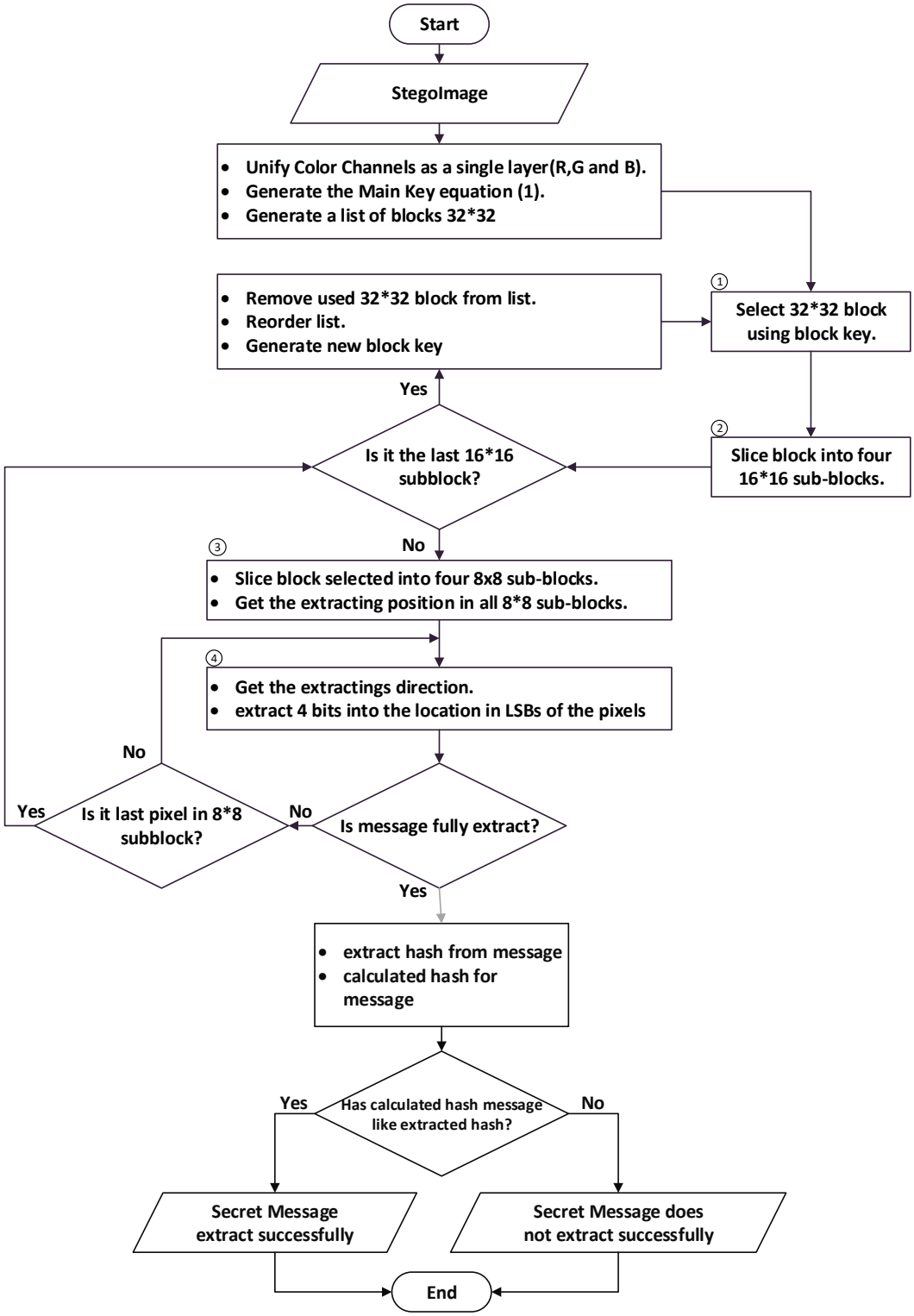


Figure 4: The extracting processes of proposed method

4. Experimental Results and Discussion

Digital steganography systems are evaluated based on four main characteristics: imperceptibility, security, robustness, and hiding capacity (Kunhoth et al., 2023).

Imperceptibility ensures that the hidden information is not revealed to the human eye or statistical analysis. Security aims to protect confidential data from discovery or removal. Steganalysis capacity refers to the amount of data that

can be included without affecting the quality of the medium. Finally, robustness expresses the system's ability to withstand digital modifications and manipulation while maintaining the integrity of the hidden data (Kadhim et al., 2019).

Experiment Environment

The proposed approach is implemented, and all experiments were performed using Visual Studio 2017 and C# programming language, on a laptop with a 2. 60GHz Core i7 processor, 8 GB RAM, and a 512 GB SSD with Windows 11 operating system. The approach was evaluated using the following performance metrics:

Mean Squared Error: MSE is a statistical measure used to determine the average squared difference between pixel values in two images. This measure is used to determine the extent of distortion or change that occurred during the masking process. The lower the MSE value, the less distortion in the image, which means that the cover image is more similar to the original image, which we want to achieve in the embedding process(Umme Sara1, Morium Akter2, 2019). The formula for calculating MSE is:

$$\text{MSE}= \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} (C(i,j) - S(i,j))^2$$

Where:

$C(i,j)$: The pixel value at position (i,j)in the cover image.

$S(i,j)$: The pixel value at the corresponding position (i,j) in the stego image.

M, N: The dimensions of the image (height and width, respectively).

Peak signal to noise ratio: PSNR the most prominent and widely used metrics to evaluate the quality of the resulting image. Higher values of PSNR show better quality of the output image and less influence of embedding on the cover image(Umme Sara1, Morium Akter2, 2019). For 8-bit depth images, the PSNR is calculated using the following equation:

$$PSNR = 10 \log_{10} (\frac{MAX^2}{MSE})$$

Where MAX Represents the highest possible value of pixel intensity in the images, MSE is the Mean Squared Error while comparing the stego and cover image

Structural similarity index measure: SSIM is a comparison metric used to assess the degree of similarity between two images based on structural properties, lighting, and contrast. It is used to evaluate the quality of an image from a visual perspective and gives a numerical score between 0 and 1 to show the degree of similarity. The closer the value is to 1, the more similar the images are(Umme Sara1, Morium Akter2, 2019). It is calculated as

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x{}^2 + \mu_y{}^2 + c_1)(\sigma_x{}^2 + \sigma_y{}^2 + c_2)}$$
$$c_1 = (k_1 L)^2$$
$$c_2 = (k_2 L)^2$$

where μx and μy are the mean intensity values of images x and y. σ_x^2 is the variance of x, σ_y^2 is the variance of y and $2\sigma_{xy}$ is the covariance of x and y. c_1 and c_2 are the two stabilizing parameters, L is the dynamic range of pixel values (2 # bits per pixel -1) and the contents k_1 = 0. 01 and k_2 = 0.03.

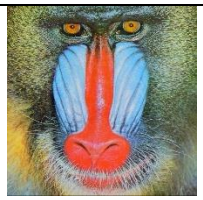
Quantitative and Visual Analysis

The performance of the proposed steganographic method was quantitatively evaluated using three standard test images: Baboon, Peppers, and Lena, with varying embedded message sizes ranging from 1 KB to 96 KB. Table 2 presents the results in terms of Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM). The results indicate that the proposed method maintains high image fidelity, even at maximum payload capacity. Notably, the SSIM values remain close to 1.0000 across all embedding sizes, suggesting that the perceptual quality of the images is virtually unaffected.

Figures 5, 6, and 7 further illustrate the impact of increasing payload size on MSE, PSNR, and SSIM, respectively. While MSE increases gradually with higher payloads, PSNR shows only a slight decrease, indicating that the embedded images retain a high level of visual quality. SSIM trends remain consistently high specially with low and medium hidden data,

reinforcing the robustness of the proposed method in preserving structural similarity as shown in Figure.

Table (2): The result of the different data size experiments of the proposed method

Image size 512×512	 Image name: Baboon.png			 Image name: Peppers.png			 Image name: Lena.png		
Message Size KB	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM
1	0.0052	70.9712	1.0000	0.0052	71.0107	1.0000	0.0052	70.9479	1.0000
2	0.0105	67.8993	1.0000	0.0103	67.9945	0.9999	0.0104	67.9493	0.9999
3	0.0157	66.1743	1.0000	0.0155	66.2313	0.9999	0.0156	66.2089	0.9999
4	0.0209	64.9289	1.0000	0.0206	64.9874	0.9999	0.0207	64.9650	0.9998
5	0.0261	63.9610	0.9999	0.0257	64.0260	0.9998	0.0260	63.9849	0.9998
10	0.0521	60.9665	0.9999	0.0516	61.0068	0.9996	0.0519	60.9761	0.9995
15	0.0780	59.2081	0.9998	0.0773	59.2484	0.9994	0.0782	59.1993	0.9993
20	0.1041	57.9557	0.9998	0.1033	57.9909	0.9993	0.1043	57.9493	0.9991
25	0.1300	56.9930	0.9997	0.1293	57.0159	0.9991	0.1304	56.9768	0.9989
30	0.1561	56.1964	0.9996	0.1552	56.2207	0.9989	0.1568	56.1777	0.9987
35	0.1820	55.5297	0.9996	0.1812	55.5496	0.9987	0.1830	55.5065	0.9985
40	0.2080	54.9503	0.9995	0.2070	54.9705	0.9985	0.2092	54.9261	0.9983
45	0.2342	54.4358	0.9995	0.2329	54.4597	0.9983	0.2351	54.4179	0.9980
50	0.2602	53.9771	0.9994	0.2586	54.0038	0.9981	0.2612	53.9618	0.9978
96	0.4997	51.1436	0.9989	0.4983	51.1557	0.9962	0.4994	51.1460	0.9959

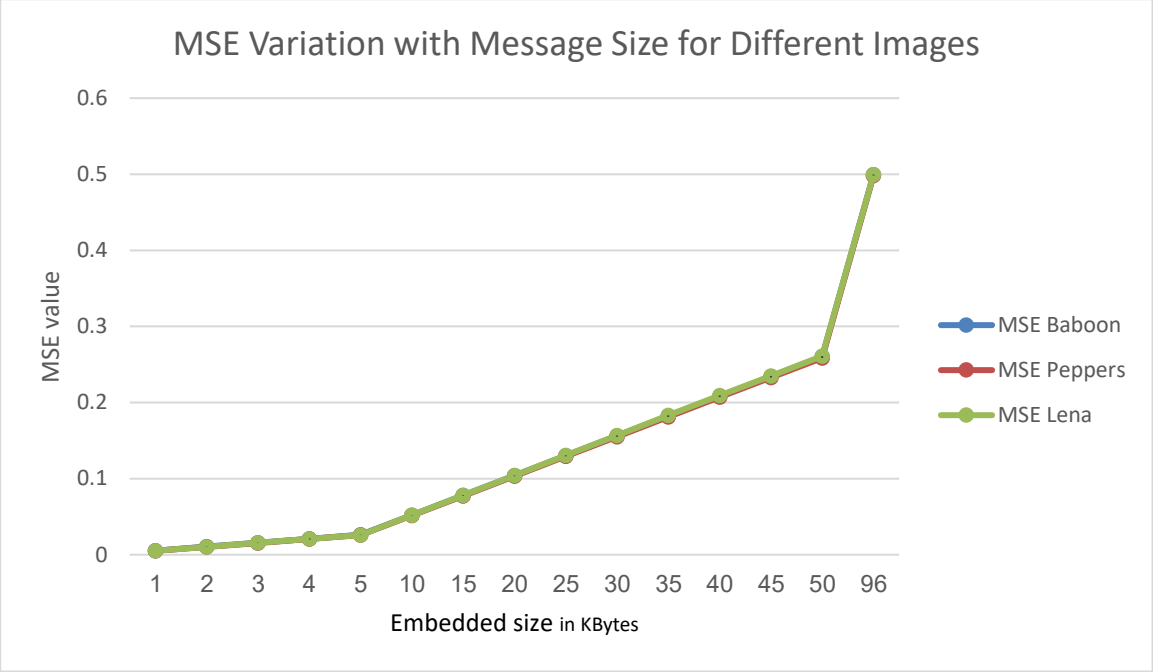


Figure 5: MSE Performance of Baboon, Peppers, and Lena Images with Increasing Embedded Message Size.

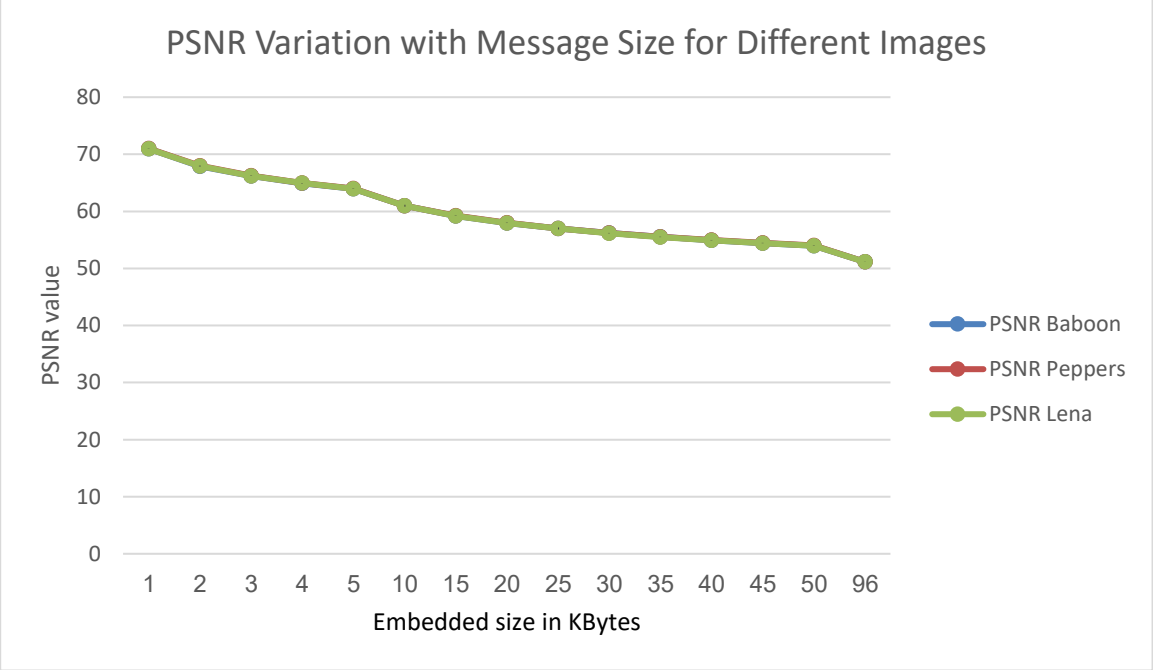


Figure 6: PSNR Performance of Baboon, Peppers, and Lena Images with Increasing Embedded Message Size.

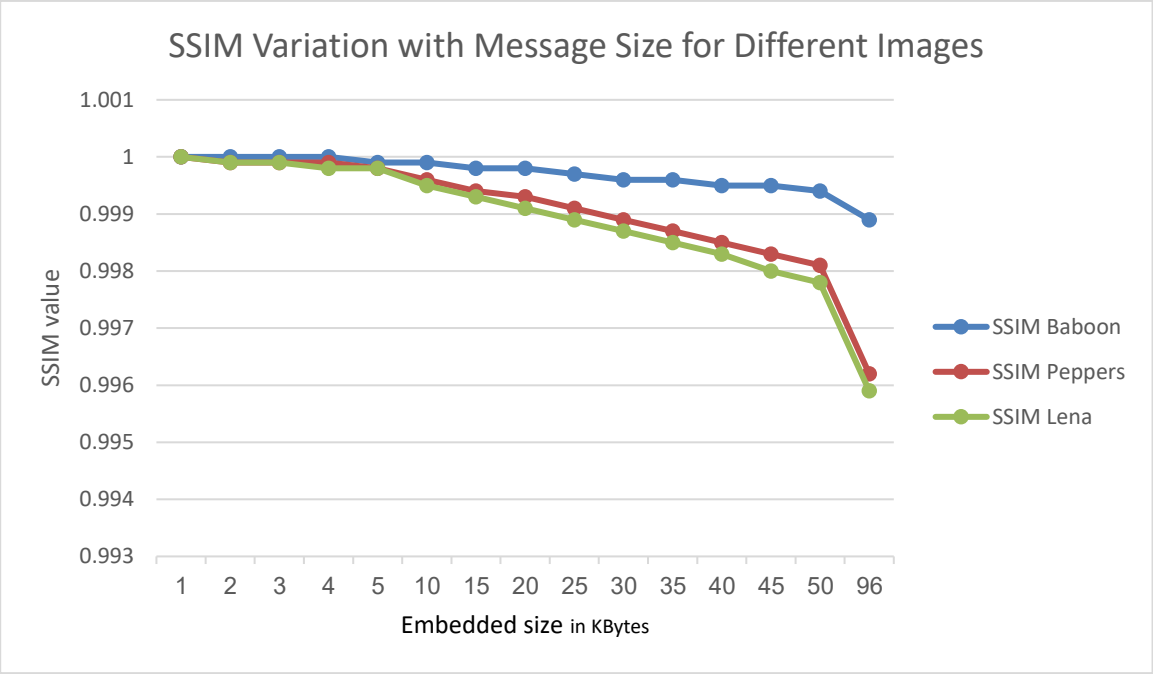
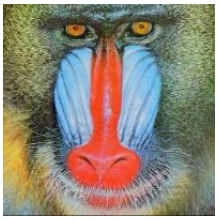
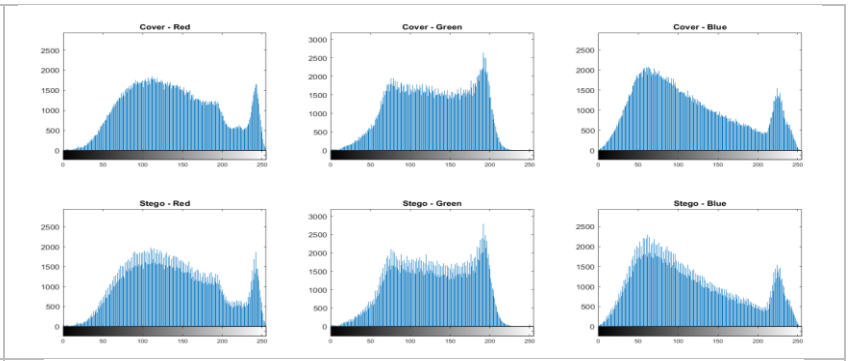
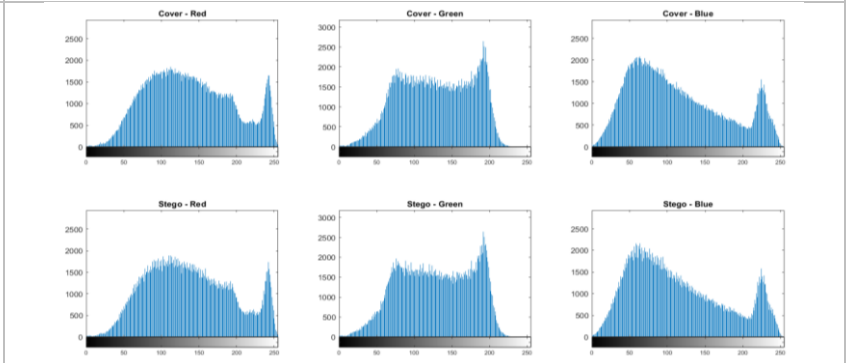
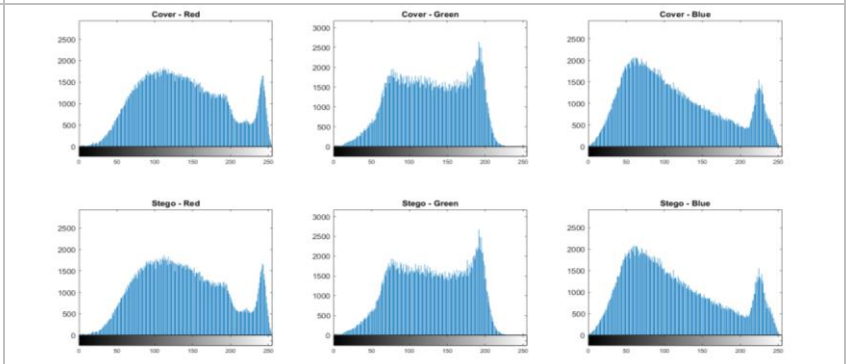
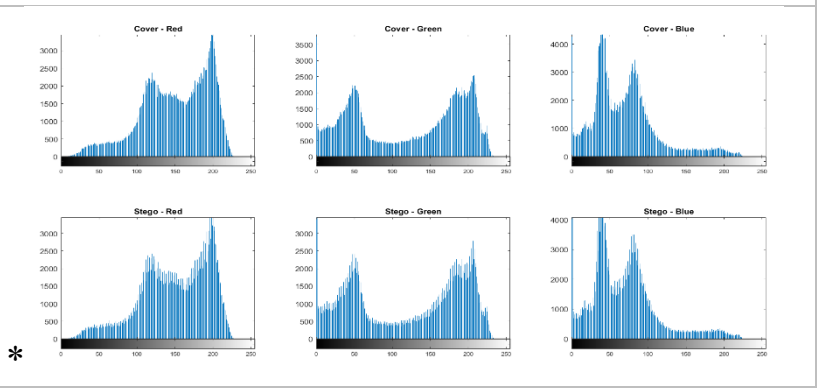


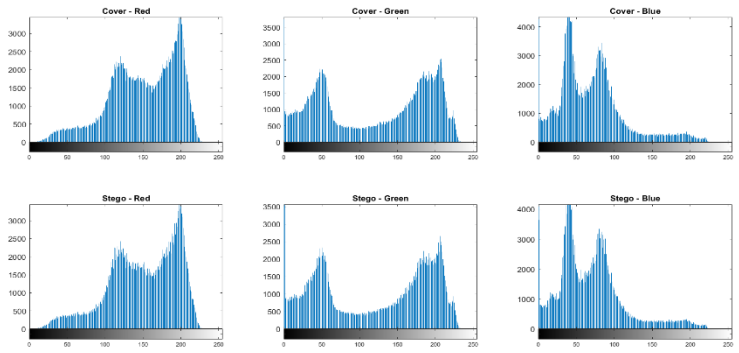
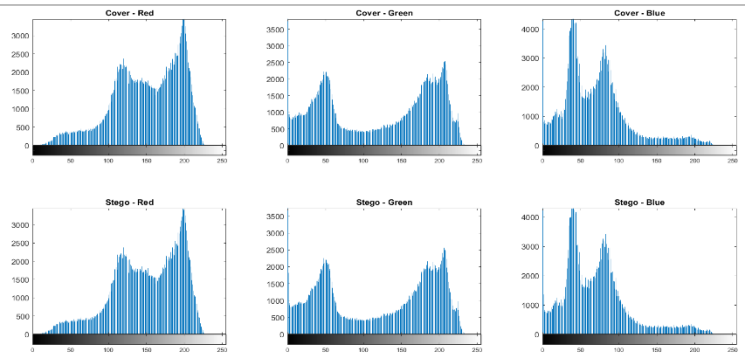
Figure 7: SSIM Performance of Baboon, Peppers, and Lena Images with Increasing Embedded Message Size.

From a visual perspective, Table 3 displays a side-by-side comparison of the cover and stego images at different payload sizes (10 KB, 45 KB, and 96 KB). Visual inspection reveals negligible differences between the original and stego images. In addition, the accompanying color histograms demonstrate a strong overlap between the original and modified images, indicating that the proposed embedding process introduces minimal perceptual or statistical artifacts. This further validates the method's effectiveness in concealing information without compromising image integrity.

Table 3: Visual and Color Histogram Analysis of Steganographic Embedding in Standard Test Images (Baboon and Pepper of size 512*512) with Different Data Payloads

 Cover image babbon	 Stego image 96 KB	
 Cover image	 Stego image 45 KB	
 Cover image	 Stego image 10 KB	
 Cover image	 Stego image 96 KB	

*

 Cover image	 Stego image 45 KB	
 Cover image	 Stego image 10 KB	

5. Comparison with Other Methods (with 3-bit per pixel embedding)

To ensure a fair and consistent evaluation of our method, we compared it with the methods of Patel et al. (A. Patel & Vekariya, 2022) and Raiyan and Kabir (Raiyan & Kabir, 2025). These methods share a fixed embedding rate of 3 bits per pixel (bpp), allowing us to objectively compare image quality metrics such as mean squared error (MSE), maximum signal-to-noise ratio (PSNR), and structural similarity index (SSIM) under the same conditions. Other studies using different embedding rates were excluded to avoid misleading conclusions.

The results in Table 4 clearly demonstrate the superiority of our method. Our method consistently achieved lower MSE values and higher PSNR than both other methods, demonstrating a better ability to preserve image quality.

Comparison with Patel et al.'s method (A. Patel & Vekariya, 2022): When embedding a data payload of 10,414 bytes, our method achieved a PSNR score of 60.8910 dB and an SSIM of 0.9999, significantly outperforming Patel et al.'s method. which achieved 51.39399 dB and 0.99795, respectively.

Comparison with the Raiyan and Kabir method(Raiyan & Kabir, 2025): When including a larger payload of 11,972,988 bytes in the Lena image, our method achieved a lower MSE (0.0607 vs. 1.003) and a higher PSNR (60.2980 dB vs. 52.888 dB). Our SSIM score (0.9995) also demonstrated better preservation of structural similarity.

These results confirm that our new method offers less visual distortion and better image fidelity preservation, even when including large amounts of data. Figure 8 also demonstrates the clear superiority of our method in terms of PSNR values compared to other methods.

Table 4. Comparison of results with other proposed methods

Method	MSE	PSNR	SSIM	Payload Capacity	Image / Size
Proposed Method	0.0530	60.8910	0.9999	10,414 bytes	Baboon.png (512×512)
(A. Patel & Vekariya, 2022)	–	51.39399	0.99795	10,414 bytes	W6.jpg (512×512)
(Raiyan & Kabir, 2025)	1.003	52.888	0.9996	11,972.988 bytes	Lena (512×512)
Proposed Method	0.0607	60.2980	0.9995	11,972,988 bytes	Lena (512×512)

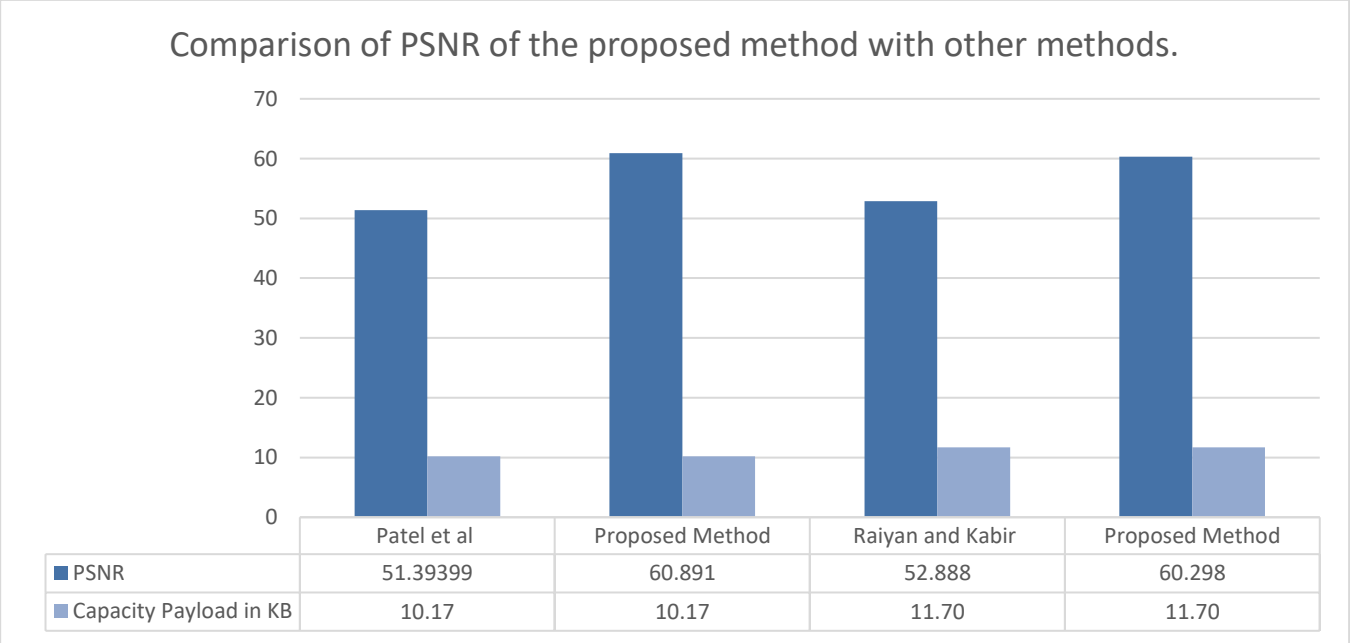


Figure 8: Comparison of PSNR Values for the Proposed Steganography Method Against Existing Methods.

6. CONCLUSIONS

A method for hiding data in images using Least Significant Bit and random keys is proposed in this work. The method relies on channel slicing and random number generation techniques to ensure a high level of security and high embedding capacity. Performance was evaluated using benchmarks such as PSNR and SSIM. The results demonstrate high quality of the modified images (stego image), while maintaining the capacity of the hidden data. It was also confirmed that the proposed

method achieves its goals of ensuring data anonymity and efficient data loading. Results reveal that the proposed method is considered an effective and secure solution to the modern challenges of hiding data in images.

Future Work:

In future work, the proposed method can be extended to other multimedia carriers, such as audio and video. This will allow a deeper evaluation of its adaptability and robustness under different transformations, including compression and transmission errors. Such an extension would demonstrate the method's potential applicability in broader real-world scenarios.

Data Availability

The Lena image used during the current study is available in image datasets in GitHub at the following link: https://github.com/mikolalysenko/lena/blob/master/lena.png. Additionally, the baboon and pepper image are available in image data sets in the USC-SIPI Image Database https://sipi.usc.edu/database.
The embedded texts were generated using the Lorem Ipsum generator available at:

https://www.blindtextgenerator.com/lorem-ipsum.

References

1. Abdulraman, L. S., Salah, S. R. H., Maghdid, H. S., & Sabir, A. T. (2019). A robust way of steganography by using blocks of an image in spatial domain. *Innovaciencia*, *7*(1), 1–7. https://doi.org/10.15649/2346075X.516
2. Ahmed, E. A. E., Soliman, H. H., & Mostafa, H. E. (2014). Information Hiding in video files using frequency domain. *International Journal of Science and Research*, *3*(6), 2431–2437.
3. Al-Kateeb, Z. N., Al-Shamdeen, M. J., & Al-Mukhtar, F. S. (2020). Encryption and Steganography a secret data using circle shapes in colored images. *Journal of Physics: Conference Series*, *1591*(1), 1–11. https://doi.org/10.1088/1742-6596/1591/1/012019
4. Bhattacharyya, D., & Kim, T. H. (2011). Image data hiding technique using discrete Fourier transformation. *Communications in Computer and Information Science*, *151 CCIS*(PART 2), 315–323. https://doi.org/10.1007/978-3-642-20998-7_39
5. Chu, R., You, X., Kong, X., & Ba, X. (2004). A DCT-based image steganographic method resisting statistical attacks. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, *5*(20111029), 1–4. https://doi.org/10.1109/icassp.2004.1327270
6. Ehsan Ali, U. A. M., Ali, E., Sohrawordi, M., & Sultan, M. N. (2021). A LSB Based Image Steganography Using Random Pixel and Bit Selection for High Payload. *International Journal of Mathematical Sciences and Computing*, *7*(3), 24–31. https://doi.org/10.5815/ijmsc.2021.03.03
7. Hameed, R. S., Mokri, S. S., Sabah Taha, M., & Muneeb Taher, M. (2022). High Capacity Image Steganography System based on Multi-layer Security and LSB Exchanging Method. In *IJACSA) International Journal of Advanced Computer Science and Applications* (Vol. 13, Issue 8, p. 2022). www.ijacsa.thesai.org
8. Haverkamp, Indy; Sarmah, D. K. (2024). *Evaluating the merits and constraints of*

cryptography-steganography fusion: a systematic analysis. International Journal of Information Security.

9. Jyoti, A., Banerjee, S., & Gupta, G. (2014). High Capacity Image Steganography Using Block Randomization. *IJCSN -International Journal of Computer Science and Network ISSN*, *3*(6), 2277–5420.
10. Kadhim, I. J., Premaratne, P., Vial, P. J., & Halloran, B. (2019). Comprehensive survey of image steganography: Techniques, Evaluations, and trends in future research. *Neurocomputing*, *335*, 299–326. https://doi.org/10.1016/j.neucom.2018.06.075
11. Kareem, H. R., Madhi, H. H., & Mutlaq, K. A. A. (2020). Hiding encrypted text in image steganography. *Periodicals of Engineering and Natural Sciences*, *8*(2), 703–707. https://doi.org/10.21533/pen.v8i2.1302.g554
12. Khandelwal, P., Bisht, N., & Thanikaiselvan, V. (2016). Randomly hiding secret data using dynamic programming for image steganography. *2015 International Conference on Computing and Network Communications, CoCoNet 2015*, 777–783. https://doi.org/10.1109/CoCoNet.2015.7411278
13. Kordov, K., & Zhelezov, S. (2021). Steganography in color images with random order of pixel selection and encrypted text message embedding. *PeerJ Computer Science*, *7*, 1–21. https://doi.org/10.7717/PEERJ-CS.380
14. Kumari, Pritam, C. K. and J. B. P. (2013). *Data security using image steganography and weighing its techniques* (pp. 238–241). International Journal Of Scientific & Technology Research 2.11.
15. Kunhoth, J., Subramanian, N., Al-Maadeed, S., & Bouridane, A. (2023). Video steganography: recent advances and challenges. *Multimedia Tools and Applications*, *82*(27), 41943–41985. https://doi.org/10.1007/s11042-023-14844-w
16. Mohsin, N. A., & Alameen, H. A. (2021). A Hybrid Method for Payload Enhancement in Image Steganography Based on Edge Area Detection. *Cybernetics and Information Technologies*, *21*(3), 97–107. https://doi.org/10.2478/cait-2021-0032
17. Neeta, D., Snehal, K., & Jacobs, D. (2006). Implementation of LSB steganography and its

evaluation for various bits. *2006 1st International Conference on Digital Information Management, ICDIM*, 173–178. https://doi.org/10.1109/ICDIM.2007.369349

18. Ni, Z., Shi, Y. Q., Ansari, N., & Su, W. (2006). Reversible data hiding. *IEEE Transactions on Circuits and Systems for Video Technology*, *16*(3), 354–361. https://doi.org/10.1109/TCSVT.2006.869964
19. Nie, S. A., Sulong, G., Ali, R., & Abel, A. (2019). The use of least significant bit (LSB) and knight tour algorithm for image steganography of cover image. *International Journal of Electrical and Computer Engineering*, *9*(6), 5218–5226. https://doi.org/10.11591/ijece.v9i6.pp5218-5226
20. Njoum, M., Sulaiman, R., Shukur, Z., & Qamar, F. (2024). *High-Secured Image LSB Steganography Using AVL-Tree with Random RGB Channel Substitution*. https://doi.org/10.32604/cmc.2024.050090
21. Patel, A., & Vekariya, D. (2022). Randomly Hiding Secret Data Using I-Blocks and E-Blocks for Image Steganography. In P. K. Singh, S. T. Wierzchoń, J. K. Chhabra, & S. Tanwar (Eds.), *Futuristic Trends in Networks and Computing Technologies* (pp. 375–390). Springer Nature Singapore.
22. Patel, H., & Dave, P. (2012). Steganography Technique Based on DCT Coefficients. *International Journal of Engineering Research and Applications (IJERA)*, *2*(1), 713–717.
23. Rahman, S., uddin, J., Hussain, H., Shah, S., Salam, A., Amin, F., de la Torre Díez, I., Vargas, D. L. R., & Espinosa, J. C. M. (2025). A novel and efficient digital image steganography technique using least significant bit substitution. *Scientific Reports*, *15*(1), 1–16. https://doi.org/10.1038/s41598-024-83147-3
24. Raiyan, S. R., & Kabir, M. H. (2025). *SCReedSolo: A Secure and Robust LSB Image Steganography Framework with Randomized Symmetric Encryption and Reed-Solomon Coding*. http://arxiv.org/abs/2503.12368
25. S.Tamil Selvan, R. R. (2022). *Image Steganography using Complemented Random Inverted Least Significant Bit Substitution*. *21*(12), 1735–1743.
26. Saber, S. M., Tuieb, M. B., Jabbar, K. K., Ahmed, M. H., & Abbas, F. N. (2025). Utilizing

Variable Hiding Centers and Dynamic Block Sizes to Improve Image Steganography. *AIP Conference Proceedings*, *3264*(1). https://doi.org/10.1063/5.0260030

27. Sharma, A., Poriye, M., & Kumar, V. (2018). A Secure Steganography Technique Using MSB. *International Journal of Emerging Research in Management and Technology*, *6*(6), 208. https://doi.org/10.23956/ijermt.v6i6.270
28. Sharma, K. P. and V. K. (2014). *Information Security Based on Steganography & Cryptography Techniques: A Review*. International Journal.
29. Swain, G. S., & Saroj(MITS), L. (2012). A Novel Approach to RGB Channel Based Image Steganography Technique.pdf. In *International Arab Journal of e-Techhnology* (Vol. 2, Issue 4, pp. 181–186).
30. Tian, J. (2003). Reversible Data Embedding Using a Difference Expansion. *IEEE Transactions on Circuits and Systems for Video Technology*, *13*(8), 890–896. https://doi.org/10.1109/TCSVT.2003.815962
31. Tolba, M. F., Ghonemy, M. A., Taha, I. A., & Khalifa, A. S. (2004). *Using Integer Wavelet Transforms in Colored Image-Steganography*. *4*(2), 75–85.
32. Umme Sara1, Morium Akter2, M. S. U. (2019). *Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study*.
33. Wu, D., & Tsai, W. (2003). *A steganographic method for images by pixel-value differencing*. *24*, 1613–1626. https://doi.org/10.1016/S0167-8655(02)00402-6
34. Yakoob, Z. A. (2025). Embedding a Robust Secret Data Using Image Steganography with Segmentation and Zigzag. *Iraqi Journal of Computers, Communications, Control and Systems Engineering*, *25*(1), 68–82. https://doi.org/10.33103/uot.ijccce.25.1.6