Volume – 1 Issue – I

Amitrakshar International Journal
of Interdisciplinary and Transdisciplinary Research (AIJITR)
(A Social Science, Science and Indian Knowledge Systems Perspective)
Open-Access, Peer-Reviewed, Refereed, Bi-Monthly, International E-Journal

Data Hiding Scheme Combining Coding Table With Enryption

Biswapati Jana[1] , Sharmistha Jana[2] and Tzu Chuen Lu[3]

Abstract. *We propose a novel data hiding method that combines irreversible LSB embedding with encryption for enhanced security. Our technique involves two phases: primary embedding using a coding table and secondary embedding in the most significant bits. This approach achieves an impressive embedding rate of 4.86 bpp, surpassing existing methods. Furthermore, it is resilient against adversarial attacks, making it a promising solution for data security in various applications.*

Keywords: *Data hiding, Encryption, Coding table, LSB embedding, Block permutation, Security, Efficiency, Adversarial attacks*

1. Introduction

The Information age has brought about the swift advancement of various technologies across interconnected networks. Consequently, in today's digital world, the importance of information security is more critical than ever. The broad use of technology and the internet has made it easier to access, transmit, and store large amounts of data. However, with this convenience comes an increase in cyber threats, which have become more frequent and sophisticated.

Cryptography [1] plays a crucial role in strengthening data protection methods. Essentially, cryptography is the practice of safeguarding communication and information by utilizing mathematical algorithms. It works by converting plaintext into ciphertext, making the data unreadable to anyone who lacks proper authorization.

Another method for securing data is information hiding, commonly referred to as steganography [2-3]. This technique involves embedding a message within a medium in a way that does not reveal the existence of the hidden content. Steganography can be applied to various types of media, including images, videos, and audio files, to conceal sensitive data. Among these, images are the most frequently used medium for data hiding. Recent studies have primarily focused on protecting medical data. Although there have been advancements in securing medical information, less emphasis has been placed on the protection of military data, which is equally critical.

Various Least Significant Bit (LSB) techniques have been developed over time [4-5], including 1-bit LSB, 2-bit LSB, 3-bit LSB, 4-bit LSB, N-bit LSB, and Hybrid LSB. In the 1-bit LSB method, only the least significant bit of each pixel is altered to embed the hidden message. This approach uses a single bit per pixel for embedding, which reduces the embedding capacity compared to higher bit LSB methods. The 1-bit LSB technique is preferred when maintaining image quality is critical, as modifying just the least significant bit has minimal visual impact. However, its data-hiding capacity is limited since only one bit per pixel is used for embedding.

Building upon the previously discussed methods, we propose a new data hiding technique inspired by their core concepts. Since the Most Significant Bits (MSBs) remain unaffected in LSB-based methods, it becomes clear that combining the LSB technique with an improved version of [6] can achieve a higher embedding capacity. The key contributions of this research are as follows:

[1] ***Department of Computer Science, Vidyasagar University, Midnapore West Bengal, Pin-721102, India,***
E-mail: biswapatijana@gmail.com
[2] ***Department of Mathematics, Midnapore College, Paschim Medinipur, West Bengal, Pin-721306, India,***
E-mail: sharmistha792010@gmail.com
[3]***Department of Information Management, Chaoyang University of Technology, Taichung 41349, Taiwan, R.O.C.***
E-mail: tclu@cyut.edu.tw

AIJITR, Volume-1, Issue-I, September- October, 2024, PP.58-68.
Revised and accepted on October 16, 2024

Volume – 1 Issue – I

Amitrakshar International Journal
of Interdisciplinary and Transdisciplinary Research (AIJITR)
(A Social Science, Science and Indian Knowledge Systems Perspective)
Open-Access, Peer-Reviewed, Refereed, Bi-Monthly, International E-Journal

i. Introduction of a novel LSB method that integrates a coding table.

ii. Development of a double embedding scheme, improving compatibility within the semi-reversible data hiding framework.

iii. Enhanced security through the application of permutation techniques.

2. Proposed Method

The proposed method involves two distinct embedding phases: Phase 1 and Phase 2. In Phase 1, a coding-table-based Least Significant Bit (CDLSB) technique is employed, while Phase 2 utilizes reversible data hiding within encrypted images. Both phases are designed to work with 512 x 512 images.

2.1 First phase embedding

The process of embedding follows a systematic sequence of actions, initiated by the establishment of a coding table agreed upon by both the sender and receiver. In Table 1, provided as a reference in this algorithm explanation, the left column denotes the secret to conceal in the first pixel (referred to as sfb), while the right column enumerates the number of secrets that can be concealed within the other three pixels. The table was generated by arranging three values summing up to 10 in various permutations, while ensuring adherence to the constraint that sfb comprises only three bits. Certain combinations, such as those containing the value 1 (e.g., 5, 4, 1), were deliberately omitted from consideration. This decision was made due to the potential impact on neighboring pixels within a range of 4 or 5 bits. Such combinations could result in less desirable outcomes, particularly during the subsequent (second)phase of embedding where the significance of the most significant bits (MSB) becomes paramount. The arrangement (2,4,4) was also excluded, resulting in the creation of distinct sfb configurations.

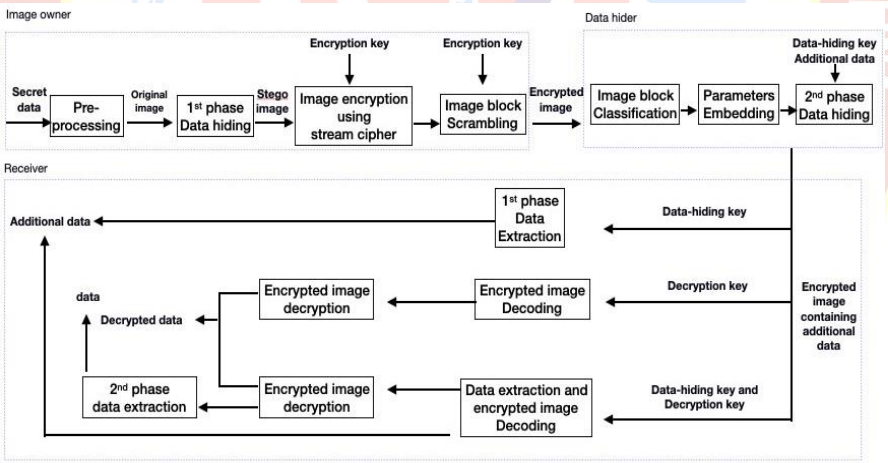


Figure 1. Proposed method Framework

Table 1. Coding Table

Secret to hide s_{fb}	Number of secrets to embed in three pixels
000	5 , 3 , 2
001	5 , 2 , 3
010	3 , 5 , 2
011	3 , 2 , 5
100	2 , 5 , 3
101	2 , 3 , 5
110	4 , 4 , 2
111	4 , 2 , 4

Published by: Amitrakshar Publishers – Amitrakshar International Journal of Interdisciplinary and Transdisciplinary Research (AIJITR)
https://www.amitrakshar.co.in/journal/ : editor@amitrakshar.co.in

Volume – 1 Issue – I

Amitrakshar International Journal
of Interdisciplinary and Transdisciplinary Research (AIJITR)
(A Social Science, Science and Indian Knowledge Systems Perspective)
Open-Access, Peer-Reviewed, Refereed, Bi-Monthly, International E-Journal

Step 1: Partition the image into non-overlapping 2 × 2 blocks, as illustrated in Figure 2.

P_f	e_1
e_2	e_3

Figure 2. Proposed method block structure.

Step 2: Organize the secret message (s) into groups, with each group comprising 13 bits. Utilize the first 3 bits from each group as a reference to the coding table (Table 1). Match these 3 bits with the corresponding sfb value from the table, representing a singular permutation dictating the number of secrets to be concealed within three pixels: e1, e2, and e3. Embed these 3 bits into the first pixel (pf) by substituting the binary values of its last 3 bits. Employ the identified permutation for subsequent steps.

Step 3: Conceal the secret bits according to the distribution outlined in the coding table (Table 1), utilizing the corresponding values obtained from Step 2. The initial value designates the starting position and the quantity of secrets to replace up to the least significant bit in e1. The second value dictates the same for e2, and the third value for e3. For example, if the value is 5, it signifies that the concealment process begins from the fifth position and extends to the least significant bit. Correspondingly, a permutation like 4,4,2 implies that 4 corresponds to e1, the subsequent 4 corresponds to e2, and 2 corresponds to e3.

Step 4: Iteratively repeat Steps 2 and 3 until the entire secret payload has been embedded, resulting in the formation of a stego image through concatenation of the manipulated pixels.

Step 5: The stego image is sent to the receiver without any auxiliary information.

2.2 Encryption and block permutation.

During the encryption process of images, the owner generates 8-bit binary keys for each block separately. These keys are utilized to encrypt each pixel within its respective block. The choice of 8-bit keys is deliberate, aiming to balance computational efficiency with cryptographic strength. The vast number of possible combinations (256) for an 8-bit key renders brute force attacks arduous. After key generation, each pixel within a block undergoes encryption via the XOR operation with the corresponding bits of the key. This ensures the pixel's security while maintaining computational efficiency, permitting straightforward decryption with the correct key.
Upon initial XOR encryption, block permutation is initiated. This involves rearranging the positions of the encrypted 2 × 2 blocks within the image based on a permutation key or algorithm. The permutation key dictates the new positions of the blocks, thereby introducing an additional layer of security. Even if an adversary decrypts individual blocks, the original spatial arrangement of these blocks remains undisclosed. This complication impedes unauthorized parties from reconstructing the original image accurately. The encrypted image post-permutation conceals the structural characteristics of the original image, highlighting the essential role of block permutation in obscuring the image's structure.

2.3 Second phase embedding.

In the second phase of embedding, the initial stage stego-image, post-encryption and permutation, is transmitted to the data hider. Following this, the data hider proceeds to embed secret data into the encrypted stego image, resulting in the generation of the final marked encrypted image, which is subsequently forwarded to the receiver.

Input: Initial Phase stego image (Encrypted image with applied permutation).
Output: Final marked encrypted image.

Published by: Amitrakshar Publishers – Amitrakshar International Journal of Interdisciplinary and Transdisciplinary Research (AIJITR)
https://www.amitrakshar.co.in/journal/ : editor@amitrakshar.co.in

Volume – 1 Issue – I

Amitrakshar International Journal
of Interdisciplinary and Transdisciplinary Research (AIJITR)
(A Social Science, Science and Indian Knowledge Systems Perspective)
Open-Access, Peer-Reviewed, Refereed, Bi-Monthly, International E-Journal

Step 1: Begin by partitioning the encrypted stego-images into blocks sized 2 × 2. The top-left encrypted pixel is selected as the fixed pixel, as depicted in Figure 3.

p_f^e	p_1^e
p_2^e	p_3^e

Figure 3. Proposed method 2 × 2 pixel block on Encrypted image.

Step 2: Within each block, the encrypted pixels undergo decomposition into a binary representation. Subsequently, Algorithm A is employed to calculate the number of shared bits.

Algorithm A: Shared bits N_{BS} calculation
Input: Encrypted pixels within a block in binary representation $p_f^{ebp}, p_1^{ebp}, p_2^{ebp}$ and p_3^{ebp}.
Output: Number of shared bits N_{BS}
Init $N_{BS} = 0$
Init $N_{BS_t} = 0$
for each encrypted pixel p_i^{eb} in the block do (t = 1 to 3)
  $N_{BS_t} = 0$
  for w = 1 to 8 do
    if $p_f^{ebp}[w]$ XOR $p_i^{ebp}[w]$!=1 then
      N_{BS_t}+= 1
    else:
      break;
    End if
  End for
End for
$N_{BS} = min(N_{BS_1}, N_{BS_2}, N_{BS_3})$
Return N_{BS}

Step 3: Determine the coding results α using the Bit planed sharing guide for the number of shared bits in each block N_{BS}.The colors in Figure 4 illustrate the N_{BS} values that correspond to each α binary value.

N_{BS}=0	N_{BS}=1	N_{BS}=2
N_{BS}=3	N_{BS}=4	N_{BS}=5
N_{BS}=6	N_{BS}=7	N_{BS}=8

(a)
Bit plane sharing condition based on Number of Bits shared NBS

α=N/A	α=000	α=001
α=010	α=011	α=100
α=101	α=110	α=111

(b)
Coding results α

Figure 4.Bit Plane sharing guide

Step 4: Record the non-shared bit of each pixel in the block, denoted as ns_1, ns_2, and ns_3. If a block doesn't share any bits, classify it as a non-embeddable block, keep the pixels unchanged, and note the Location map (LM) as 1. Otherwise,

Published by: Amitrakshar Publishers – Amitrakshar International Journal of Interdisciplinary and Transdisciplinary Research (AIJITR)
https://www.amitrakshar.co.in/journal/ : editor@amitrakshar.co.in

Volume – 1 Issue – I

Amitrakshar International Journal
of Interdisciplinary and Transdisciplinary Research (AIJITR)
(A Social Science, Science and Indian Knowledge Systems Perspective)
Open-Access, Peer-Reviewed, Refereed, Bi-Monthly, International E-Journal

classify it as an embeddable block, and note the Location map (LM) as 0. From the bit plane in Figure 4, α=0 at the first position of the coding result because there are no shared bits ($N_{BS} = 0$). This Location map can be compressed by using arithmetic code and the total size of this map is around 64 bits. The number of non-shared bits can be calculated using equation 3.

$$ns_i = 8 - N_{BS_i} \quad (3)$$

Where N_{BS} is the number of shared bits and i is a pixel position.

Step 5: If a block is classified as an embeddable block, the data hider reconstructs the 32-bit pixel bit plane. This bit plane comprises the fixed encrypted pixel p_f^{ebp}, coding results α, non-shared bits of ns_1, n_2, and ns_3, along with the hiding room HR, as illustrated in Figure 4. The colors in Figure 5 are used to clarify the difference between all the parts of the bit plane its structure.

p_f^{ebp}	α	ns_1	ns_2	ns_3	HR
8 (bits)	3 (bits)	$8 - N_{BS_i}$ (bits)	$8 - N_{BS_i}$ (bits)	$8 - N_{BS_i}$ (bits)	$32 - [3 + 8 + 3 \times (8 - N_{BS_i})]$ (bits)

Figure 5. Bit Plane structure.

Step 6: To utilize the available room for data embedding, the data hider initially encrypts the secret data using the data hiding key K_d. Subsequently, the secret data, comprising (32 - [8 + 3 × (8 –N_{BS_i})]) bits, is embedded into each block.

Step 7: Repeat the aforementioned steps iteratively until all the secret data is successfully embedded.

2.4 First phase embedding example

1. Suppose secret S=00010110001111110101011101
2. Divide the secret S into groups of 13 bits.
3. Secret S_1 =0001011000111
4. Secret S_2=1110101011101
5. Hide the first 3 bits of s_1 in p_f , at the same time look up these 3 bits from the encoding table as s_{fb}.
6. For s_1, after looking up 000, the permutation 5,3,2 is found and used to hide.

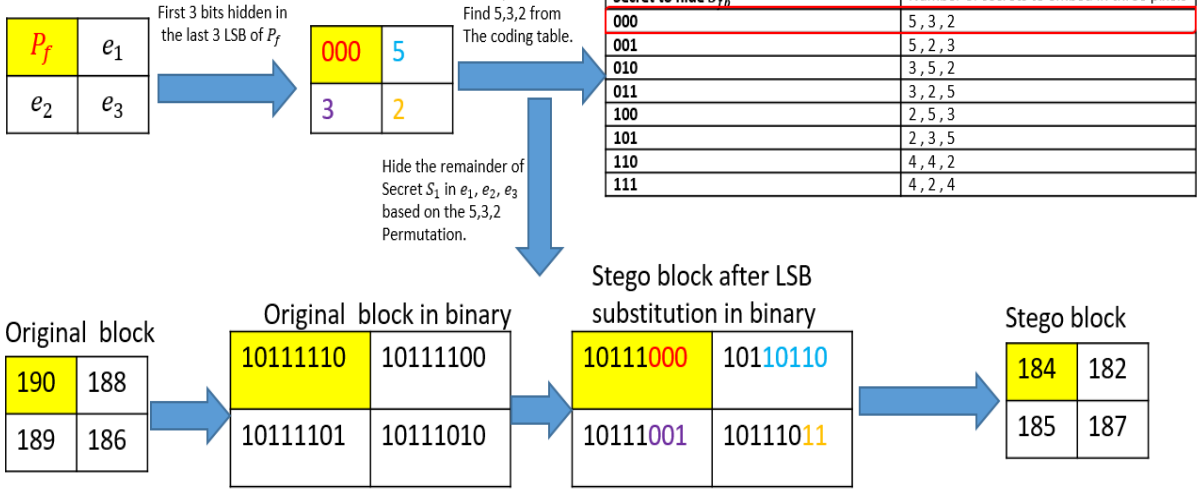


Table 1. Coding Table

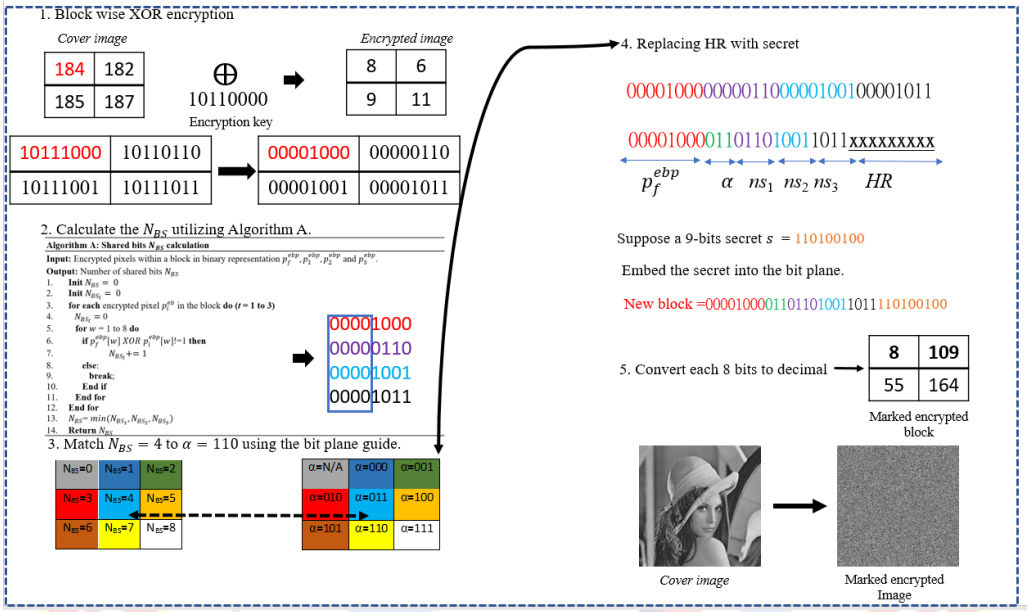
Secret to hide S_{fb}	Number of secrets to embed in three pixels
000	5,3,2
001	5,2,3
010	3,5,2
011	3,2,5
100	2,5,3
101	2,3,5
110	4,4,2
111	4,2,4

Published by: Amitrakshar Publishers – Amitrakshar International Journal of Interdisciplinary and Transdisciplinary Research (AIJITR)
https://www.amitrakshar.co.in/journal/ : editor@amitrakshar.co.in

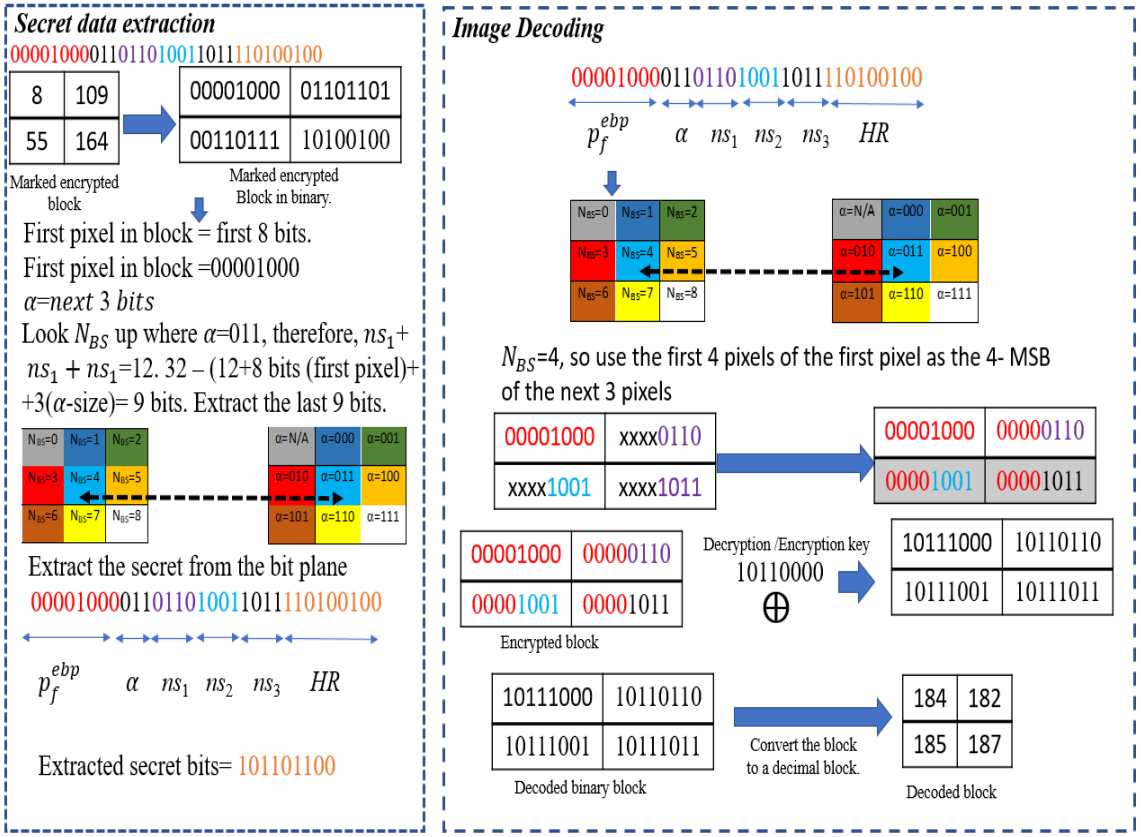
Volume – 1 Issue – I

Amitrakshar International Journal
of Interdisciplinary and Transdisciplinary Research (AIJITR)
(A Social Science, Science and Indian Knowledge Systems Perspective)
Open-Access, Peer-Reviewed, Refereed, Bi-Monthly, International E-Journal

2.5 Second phase embedding example



2.6 First phase extraction example.

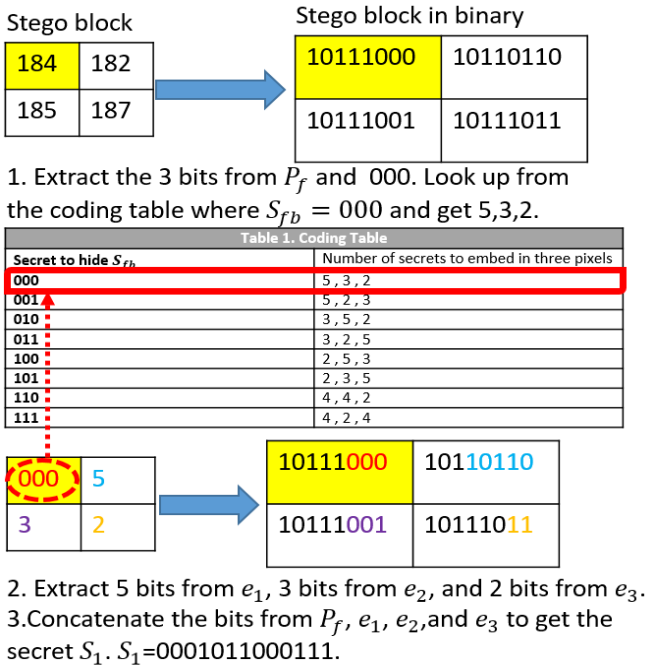


Published by: Amitrakshar Publishers – Amitrakshar International Journal of Interdisciplinary and Transdisciplinary Research (AIJITR)
https://www.amitrakshar.co.in/journal/ : editor@amitrakshar.co.in

Volume – 1 Issue – I

Amitrakshar International Journal
of Interdisciplinary and Transdisciplinary Research (AIJITR)
(A Social Science, Science and Indian Knowledge Systems Perspective)
Open-Access, Peer-Reviewed, Refereed, Bi-Monthly, International E-Journal

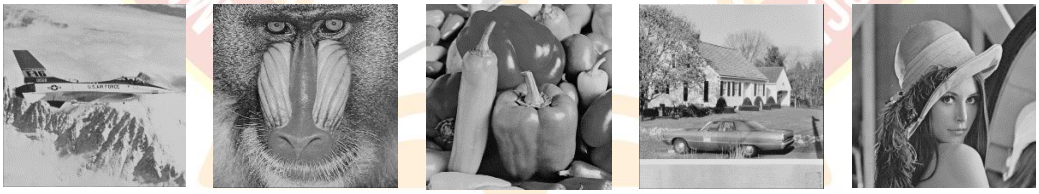
2.7 Second phase extraction example.



3. Experimental Results

To demonstrate the practical application of the proposed method, a series of experiments were meticulously conducted. These experiments aimed to evaluate the embedding capacity of the method and underwent thorough security analysis. The experimental setup utilized a PC with 8 GB of RAM. All experiments were performed using Python 3 within the Spyder environment, ensuring a robust and controlled testing environment.

The experimentation phase involved a diverse range of images sourced from multiple databases, including Kaggle, and USC_SIPI. Figures 10 and 11 visually display a selection of the images prominently featured in these experiments.



(a) Airplane (b) Baboon (c)Peppers (d) House (e) Lena

Figure 10. Some of the images used during the experiments.

Embedding capacity, a critical metric in data hiding, quantifies the amount of confidential data that can be effectively concealed within each image. This metric facilitates the computation of the embedding rate (ER), which reflects how many bits can be discretely nestled within each pixel.

In the context of images, the Peak Signal-to-Noise Ratio (PSNR) [27] is commonly employed to assess the distortion introduced in image steganography. A lower PSNR value indicates a more distorted image. The PSNR value is contingent upon the Mean Squared Error (MSE), and it can be calculated as equation 4.

Published by: Amitrakshar Publishers – Amitrakshar International Journal of Interdisciplinary and Transdisciplinary Research (AIJITR)
https://www.amitrakshar.co.in/journal/ : editor@amitrakshar.co.in

Volume – 1 Issue – I

Amitrakshar International Journal
of Interdisciplinary and Transdisciplinary Research (AIJITR)
(A Social Science, Science and Indian Knowledge Systems Perspective)
Open-Access, Peer-Reviewed, Refereed, Bi-Monthly, International E-Journal

$$PSNR = 10 \cdot log_{10}\left(\frac{MAX^2}{MSE}\right) \quad (4)$$

The time and space complexity are both $O(n^2)$, where n is a dimension of the image.

3.1 Embedding capacity

The proposed method demonstrates a substantial capacity for data embedding, exemplified by its average embedding capacity of 851,968 bits for the 1st embedding phase and an even more noteworthy average embedding capacity of 423,380 bits for the 2nd embedding phase, as delineated in Table 2.

Table 2. Embedding capacity (EC) and Embedding Rate (ER) of the proposed method.

Image 512 × 512	1st phase embedding EC	2nd phase embedding EC	Total EC	Total ER (bpp)
Lena	851,968	390,042	1,242,010	4.74
Baboon	851,968	179,718	1,031,686	3.94
Girl	851,968	388,893	1,240,861	4.73
Airplane	851,968	385,593	1,237,561	4.72
Lake	851,968	316,917	1,168,885	4.46
Peppers	851,968	401,262	1,253,230	4.78
Boat	851,968	317,271	1,169,239	4.46
House	851,968	332,298	1,184,266	4.52
Tiffany	851,968	401,652	1,253,620	4.78
Jet.tiff	851,968	439,005	1,290,973	4.92
Bone 1	851,968	521,640	1,373,608	5.24
Bone 3	851,968	514,221	1,366,189	5.21
Bone 5	851,968	521,121	1,373,089	5.24
Bone 30	851,968	532,743	1,384,711	5.28
brain15	851,968	484,869	1,336,837	5.10
brain16	851,968	485,154	1,337,122	5.10
brain26	851,968	502,245	1,354,213	5.17
Average	**851,968**	**423,380**	**1,275,348**	**4.86**

References [8-12] encompass reversible data hiding schemes in encrypted images. The proposed method is a double embedding approach using LSB replacement and hiding room generation in encrypted images.

Table 3. Embedding Rate (ER) comparison of the proposed method with other methods in bits per pixel (bpp).

Image 512×512	Wang et al. [8]	Zhang et al. [9]	Yu et al. [10]	Yang et al. [11]	Yao et al. [12]	Proposed
Lena	2.23	3.81	3.02	3.4	3.11	4.74
Baboon	1.03	2.06	1.46	1.68	1.38	3.94
Peppers	2.28	3.47	N/A	3.11	N/A	4.78
Airplane	2.37	4.08	3.99	N/A	N/A	4.72
Lake	1.67	3.07	N/A	2.65	N/A	4.46

Published by: Amitrakshar Publishers – Amitrakshar International Journal of Interdisciplinary and Transdisciplinary Research (AIJITR)
https://www.amitrakshar.co.in/journal/ : editor@amitrakshar.co.in

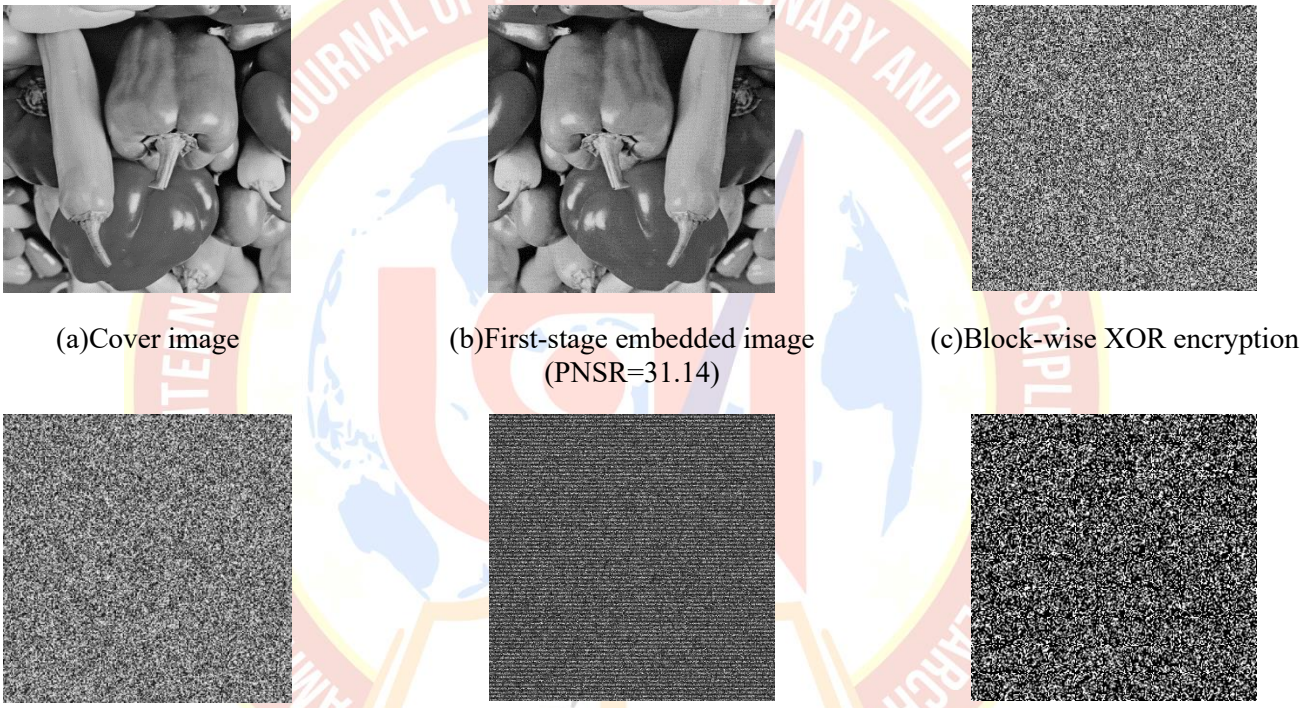
Volume – 1 Issue – I

Amitrakshar International Journal
of Interdisciplinary and Transdisciplinary Research (AIJITR)
(A Social Science, Science and Indian Knowledge Systems Perspective)
Open-Access, Peer-Reviewed, Refereed, Bi-Monthly, International E-Journal

Boat	1.89	N/A	N/A	3.32	N/A	4.46
Tiffany	N/A	N/A	3.09	N/A	3.23	4.78
Average	**1.91**	**3.30**	**2.89**	**2.83**	**2.57**	**4.55**

3.2 Security analysis

Complexity analysis [13] is a technique utilized to assess the intricate patterns and structures within an image. It involves scrutinizing various aspects of the image's pixel values and their interrelationships to identify regions or blocks of pixels that exhibit specific characteristics, such as high complexity or unpredictability. One aspect of this analysis involves recovering partial contents from the original image. Achieving this involves a complexity analysis procedure, outlined as follows:



(a)Cover image (b)First-stage embedded image (PNSR=31.14) (c)Block-wise XOR encryption

(d)Encrypted image after block-permutation (e)Encrypted marked Image (f) Encrypted marked Image after complexity analysis

Figure 12. An example of complexity analysis on the proposed method.

Step 1: The encrypted image is partitioned into non-overlapping blocks, each with dimensions of $w \times z$.

Step 2: For each image block, begin by calculating the average pixel value, denoted as Pval, using equation. 5. In this equation, pt,v represents the pixel located at coordinates [*t, v*] within the block. Subsequently, compute the average of the absolute differences between each pixel and *Pval* , denoted as *Fval*, using equation 6.

$$Pval = \frac{\sum_{t=1} m \sum_{j=1} nPt, v}{m \times n} \quad (5)$$

Published by: Amitrakshar Publishers – Amitrakshar International Journal of Interdisciplinary and Transdisciplinary Research (AIJITR)
https://www.amitrakshar.co.in/journal/ : editor@amitrakshar.co.in

Volume – 1 Issue – I

Amitrakshar International Journal
of Interdisciplinary and Transdisciplinary Research (AIJITR)
(A Social Science, Science and Indian Knowledge Systems Perspective)
Open-Access, Peer-Reviewed, Refereed, Bi-Monthly, International E-Journal

$$Fval = \frac{\sum_{t=1} m \sum_{v=1} n|Pt, v - Pval|}{w \times z} \quad (6)$$

Step 3: In the case of a complex block where *Fval* surpasses a predetermined threshold, all pixels within the block are set to 0. Conversely, for other blocks, all pixels are set to 255. Consequently, the complexity analysis culminates with the generation of an outline of the encrypted image. Should the need arise, iteratively increasing the dimensions of *w* and *z* by a factor of 2, before returning to Step 1 is done.

Methods that can not withsatnd the complexity analysis will produce an outline of the original image, making it easy for attackers to detect which images were used. Such methods are considered not to be secure. Figure 12 (f) shows the proposed method under a complexity attack using the "peppers" image. Ase can be seen, there is no outline of the "peppers" image, therefore demonstrating that the proposed method passes the complexity analysis.

4. Conclusion

In this paper, a novel dual-phase semi-reversible data hiding method is proposed, utilizing a hybrid coding table-based least significant bits (LSB) and encryption approach. The method involves two distinct phases: firstly, employing a coding table to conceal data in the least significant bits of an image, followed by encryption and permutation processes. Subsequently, a second phase of data hiding is performed on the encrypted image by analyzing the most significant bits of pixels within a block to create hiding room. Notably, the second phase of data hiding is reversible, whereas the first phase is irreversible, resulting in the inability to fully recover the original image while allowing for lossless extraction of secret data,hence this scheme is semi-reversible.

The proposed method demonstrates a high embedding rate (ER) averaging at 4.86 bits per pixel (bpp). Moreover, it surpasses compared methods in terms of embedding rate, exhibiting a significant increase of nearly 62%. Furthermore, the method exhibits resilience against attacks such as complexity analysis and histogram analysis.

Given its high embedding rate and robustness against attacks, the proposed method outperforms state-of-the-art methods. In the future, efforts will be directed towards enhancing the reversibility of the first phase of data hiding, enabling the method to be applicable in fields where the full recovery of the original images, such as in the medical field, is necessary. Nonetheless, the proposed method can be effectively applied in domains prioritizing security and increased data capacity over full image recovery, such as in military applications.

5. References

[1] Abood, O. G., & Guirguis, S. K. (2018). A survey on cryptography algorithms. International Journal of Scientific and Research Publications, 8(7), 495-516.
[2] Morkel, T., Eloff, J. H., & Olivier, M. S. (2005, June). An overview of image steganography. In ISSA (Vol. 1, No. 2, pp. 1-11).
[3] Chowdhuri, P., Jana, B., & Giri, D. (2018). Secured steganographic scheme for highly compressed color image using weighted matrix through DCT. International Journal of Computers and Applications, 43(1), 38–49. https://doi.org/10.1080/1206212X.2018.1505024
[4] Arya, A., & Soni, S. (2018). Performance evaluation of secrete image steganography techniques using least significant bit (LSB) method. Int. J. Comput. Sci. Trends Technol, 6(2), 160-165.
[5] Bansal, K., Agrawal, A., & Bansal, N. (2020, June). A survey on steganography using least significant bit (lsb) embedding approach. In 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184) (pp. 64-69). IEEE.
[6] Y. Wang and W. He, "High Capacity Reversible Data Hiding in Encrypted Image Based on Adaptive MSB Prediction," IEEE Transactions on Multimedia, vol. 24, pp. 1288–1298, 2022.
[7] J. Lin, S. Weng, T. Zhang, B. Ou, and C.-C. Chang, "Two-Layer Reversible Data Hiding Based on AMBTC Image With (7, 4) Hamming Code," IEEE Access, vol. 8, pp. 21534–21548, 2020.
[8] P. Wang, B. Cai, S. Xu, and B. Chen, "Reversible Data Hiding Scheme Based on Adjusting Pixel Modulation and Block-Wise Compression for Encrypted Images," *IEEE Access*, vol. 8, pp. 28902–28914, 2020.

Volume – 1 Issue – I

Amitrakshar International Journal
of Interdisciplinary and Transdisciplinary Research (AIJITR)
(A Social Science, Science and Indian Knowledge Systems Perspective)
Open-Access, Peer-Reviewed, Refereed, Bi-Monthly, International E-Journal

[9] H. Zhang, L. Li, and Q. Li, "Reversible Data Hiding in Encrypted Images Based on Block-Wise Multi-Predictor," *IEEE Access*, vol. 9, pp. 61943–61954, 2021.

[10] C. Yu, X. Zhang, X. Zhang, G. Li, and Z. Tang, "Reversible Data Hiding With Hierarchical Embedding for Encrypted Images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 2, pp. 451–466, 2022.

[11] Y. Yang, H. He, F. Chen, Y. Yuan, and N. Mao, "Reversible Data Hiding in Encrypted Images Based on Time-Varying Huffman Coding Table," *IEEE Transactions on Multimedia*, vol. 25, pp. 8607–8619, 2023.

[12] Y. Yao, K. Wang, Q. Chang, and S. Weng, "Reversible Data Hiding in Encrypted Images Using Global Compression of Zero-Valued High Bit-Planes and Block Rearrangement,"*IEEE Transactions on Multimedia*, vol. 26, pp. 3701–3714, 2024.

[13] Y. Wang, Z. Cai, and W. He, "High Capacity Reversible Data Hiding in Encrypted Image Based on Intra-Block Lossless Compression," *IEEE Transactions on Multimedia*, vol. 23, pp. 1466–1473, 2021.