REVERSIBLE DATA HIDING IN ENCRYPTED IMAGES WITH SECRET SHARING AND HYBRID CODING

Nadia Zaynab
CSE Department, Shadan women's College of Engineering and Technology,
Hyderabad, India
reach2nadia@gmail.com

Dr. Muhammed Pasha
CSE Department, Shadan women's College of Engineering and Technology,
Hyderabad, India
muhammed.pasha@gmail.com

Dr. K. Palani
CSE Department, Shadan women's College of Engineering and Technology,
Hyderabad, India
principalswcet2020@gmail.com

ABSTRACT

Data security methods such as reversible data hiding in encrypted pictures (RDHEI) are crucial. The majority of RDHEI techniques struggle to integrate security and capacity effectively. We suggest a novel RDHEI technique utilizing hybrid coding and Chinese remainder theorem-based secret sharing (CRTSS) in order to solve this problem. In particular, a hybrid coding is initially suggested for RDH in order to attain a high embedding capacity. A unique iterative encryption is intended to perform block-based encryption on the content owner's end, precisely maintaining the spatial correlation of the original blocks in their encrypted blocks. The spatial correlations of the encrypted blocks are then kept in the various encrypted picture shares that are produced by using the CRTSS with the limitations. Concurrently, the suggested approach benefits from strong security features offered by the CRTSS. Each share's blocks exhibit high geographical correlations, which the data-hider can take advantage of by performing data embedding to increase capacity using the suggested hybrid coding. As long as enough uncorrupted marked shares are retrieved, the original picture can be restored without any loss on the receiver side, even if some shares are corrupted or absent. The experiment's findings demonstrate that, in terms of embedding capacity, the suggested RDHEI method works better than a few cutting-edge techniques, including a few that rely on secret sharing (SS). This enhances the original image's security even more. We offer two coupled cases and two separable cases that is, four cases of the suggested model. We derive a separable, high-capacity RDH-EI approach from the suggested model. The success of the suggested approach is demonstrated by the experimental findings that are provided.

1. INTRODUCTION

The rapid expansion of cloud computing and storage has coincided with the advancement of information technology, particularly with the maturation of 5G communication and transmission. Owing to the benefits of cloud computing, an increasing number of users are storing their information on the cloud, particularly multimedia files like photos, audio snippets, and video files. Since user data is stored on distant or cloud servers or is transferred over public networks, it is not necessarily secure and dependable. As a result, both industry and academics now give considerable attention to the data privacy issue. Numerous strategies exist to safeguard data security, including hashing, data concealing, encryption, and secret sharing (SS). Digital images are important data manifestations that have a wide range of uses in fields like legal forensics, photography, medicine, and the military. One of the areas of research interest is visual data concealing for security protection.

The secret data and cover image are the two things that make up the image data concealment system, as is widely known. To create a stego-picture in an undetectable way, secret data is integrated into the cover image. The picture data hiding mechanism should take into account two circumstances. One is that the cover image is superfluous and the secret data is uniquely safeguarded. Another is that, for military and medical purposes, both the cover picture and the secret data are essential, and both need to be error-free on the decoder side. It is obvious that the latter is reversible data hiding (RDH) and the former is irreversible data hiding. Due to its reversibility, RDH is becoming more and more popular. RDH techniques to date have primarily relied on lossless compression, difference expansion (DE), histogram shifting (HS), and pixel error expansion (PEE). These RDH methods seek to achieve a favorable trade-off between modification distortion and embedding capability.

(1) For RDH, we suggest a hybrid coding. The suggested hybrid coding can surpass the payload restriction of a single coding and obtain a high payload since it combines the benefits of several coding methods.

(2) For data concealing, we create a block-based CRTSS with limitations and a new iterative image encryption. There is greater space for data embedding since repeated encryption can precisely retain spatial correlation. Multiple encrypted shares with good spatial correlations can be produced by the planned CRTSS.

Preprocessing is not necessary, and it won't cause data growth.
(3) To create a new RDHEI technique, we take advantage of the suggested hybrid coding, the iterative picture encryption, and the block-based CRTSS with limitations. The suggested approach works better in terms of embedding rate than several cutting-edge RDHEI techniques, according to experimental results.

2. RELATED WORK

Numerous RDHEI techniques have been developed by researchers to date. The current RDHEI methods can be broadly categorized into four groups: reserving room before encryption (RRBE) [26], [27], [28], [29], [30], [31], [32], [33]; vacating room by encryption (VRBE) [34], [35], [36], [37], [39], [40], [41], [42]; and SS based methods [43], [44], [45], [46], [47], [48], [49]. Together, these four categories comprise the majority of the current RDHEI methods.

A. VRAE Based Methods

The initial image, which is built on a VRAE framework, was immediately encrypted using conventional image encryption algorithms such stream cipher and the advanced encryption standard (AES) in the early RDHEI approaches [20], [21], [22], [23], [24], and [25]. The encrypted image in [20] is split up into many chunks. To make room for one hidden bit, the three least significant bits (LSBs) of each block's half pixels are flipped. The original image is simultaneously recovered and secret bits are extracted at the receiver side by using the fluctuation function after the marked image has been directly decrypted. Since then, some advancements in data extraction accuracy have been realized [21], [22]. Since then, some advancements in data extraction accuracy have been realized [21], [22]. Data extraction is not possible in [20], [21], and [22] when the encryption key is not provided. More adaptably, Zhang [23] and Qian and Zhang [24] suggested separable techniques that involve compressing some LSB planes of the stream cipher encrypted image to make more space for data embedding. In this way, data extraction becomes independent of picture recovery and the encryption key. To achieve data concealment, some randomly selected pixels from the stream cipher encrypted image are changed with secret bits in [25] in place of their high bit-planes. Since there are two possible outcomes for a single high bit-plane, "0" and "1," the prediction errors produced by these two outcomes are compared in order to recover the original image.

B. RRBE Based Methods

Despite the stream cipher's strong performance in picture encryption, the loss of pixel spatial correlation makes it challenging to remove an embedding room directly from the encrypted image. Consequently, a few RRBE-based strategies were put forth to help achieve large payloads [26], [27], [28], [29], [30], [31], [32], and [33]. In order to free up space before picture encryption, Ma et al. [26] integrated a few LSBs of the texture area into the smooth area using the conventional RDH technique. Secret data is included in the released room. The patch-level sparse representation method is employed in [27] to significantly reduce the amount of space in the original image. A binary-block embedding (BBE) technique was suggested by A binary-block embedding (BBE) technique was proposed by Yi and Zhou [28]. BBE is used to embed the original image's lower bit-planes into its upper bit-planes, freeing up the lower bit-planes for data hiding. Chen and Chang [29] to construct bitstreams, which were then effectively compressed to provide the embedding room, rearranged the most important bits (MSBs). In [30], the same high bit-planes that are successively labeled are compared between the original pixel and its predicted value. Secret bits are stored on the bit-planes with labels. In [31], secret bits are embedded into the embeddable bit-planes in accordance with the labels generated by hierarchically dividing the prediction errors into three magnitudes. In Yin et al. [32], To obtain a high embedding capacity, Yin et al. [32] employed pixel prediction and compressed the high bit-planes of prediction errors. In order to evacuate the huge room prior to encryption, an adaptive L predictor for preprocessing is constructed in Mohammadi's [33] general RRBE framework for RDHEI.

C. VRBE Based Methods

Although RRBE-based techniques provide outstanding embedding performance, their practical applicability may be limited because the content owner lacks the computational capacity to undertake pretreatment operations or is unaware that the following data is hidden. Some particular encryption-based techniques, especially VRBE-based techniques, were presented to overcome this problem. The encrypted image in [34], [35], and [36] is produced by block permutation and block-based bit-XOR, both of which are capable of maintaining the correlation inside each encrypted block. Next, data concealing is accomplished by using difference compression [36], adaptive block encoding [35], and difference histogram shifting (DHS) [34]. The original image is encrypted in [37] and [38] by using block permutation and disordering bit planes, which transfers the redundant space from the original image to the encrypted image. To remove space for data concealment, high bit-plane portions of the encrypted image are compressed using efficient sparse coding. In order to maintain spatial correlations among image blocks, Yi and Zhou [39] first encrypted the original image using block permutation and block-based modulation. They then used parametric binary tree labeling (PBTL) to insert secret data into the encrypted image. In [40], redundancy is preserved in the encrypted image by employing the CE technique, which encrypts the original image in chunks using the stream cipher. Subsequently, the redundancy matrix format is employed to free up space for data embedding. In order to obtain high embedding capacity, Yu et al. [41] presented an adaptive difference recovery (ADR) based data hiding technique and subsequently implemented this technique in RDHEI. A generalized methodology for high-capacity RDHEI utilizing pixel prediction and entropy encoding was presented by Qiu et al. [42] and is applicable to both the RRBE and VRBE scenarios.

D. SS Based Methods

An original image is converted into an encrypted version and uploaded to the cloud using the RDHEI techniques mentioned above. Attacks by a third party could compromise the encrypted image and result in incorrect image recovery on the recipient's end. Some secret sharing (SS) based RDHEI techniques were proposed [43], [44], [45], [46], [47], [48], and [49] in an effort to increase the robustness of RDHEI. Wu et al. [43] used pairwise Shamir's SS [10] to encrypt the original image in order to create the encrypted shares, ensuring that each share's pixel pair difference is equal to that of the original pair. Secret data can be integrated into shares using the DE or DHS technique because of difference preservation. Another SS was proposed by Chen et al. [44]. Another SS-based RDHEI technique was proposed by Chen et al. [44]. This method encrypts a pair of pixels using a degree 3 polynomial after preprocessing them using the DE technique. The encrypted pixel pair may contain one secret bit inserted in it. One data-hider does the data concealing in [43] and [44].

The original image might not be recovered in the case that the data hider is an attacker for the original image since it might be an unreliable third party. Some multiple data-hiders based RDHEI approaches using SS were presented [45], [46], [47], [48], and [49] to address this problem. These methods distribute each share to a single data-hider and provide independent data hiding on each share. Two RDHEI techniques via SS over Galois fields GF(p) and GF(28) were proposed by Qin et al. [45]. These two techniques allow the embedding room to be abandoned in each share by preserving the pixel disparities within the 2x2 blocks of each share after SS. Multiple data-hiders get the encrypted shares in [46], which are produced using a particular Shamir's SS [10]. Through the bit-plane substitution approach, each data hider incorporates secret data into their share. Because Shamir's SS is often built on a finite field Fp with prime size 251, it is not possible to communicate pixels with values larger than 251. Accordingly, in these Shamir's SS based algorithms [43], [44], [45], and [46], the pixels with values more than 251 are prepossessed. CRTSS [8] is used in [47] to encrypt the original image and produce several shares. The additive homomorphism of CRT and the DE method are used in each share to hide data. Pixel enlargement arises from the use of CRTSS [8] and needs to be solved by compressing two MSBs of each share using this method. Because the DE approach is used, this method's embedding capacity is not high, hovering around 0.5 bpp. First presented by Hua et al. [48], cipher-feedback secret sharing (CFSS) is a technique that can be used to share images. An approach called multi-MSB prediction is used to hide data. Using matrix theory, Hua et al. [49] originally presented a matrix-based secret sharing (MSS). To attain a high payload, they subsequently suggested an MSS-RDHEI approach utilizing block error mixture encoding (BEME).

In essence, these high-payload RDHEI methods nearly take use of certain coding strategies to represent the image context with less information, freeing up space for secret data like BEME [49], PBTL [39], and entropy encoding [42]. These single-coding methods can yet be improved upon, even though they can yield a large payload. In order to introduce a novel RDHEI approach, we suggest in this study a hybrid coding with a bigger payload.

3. METHODOLOGIES

Proposed Hybrid Coding for RDH

Here, we provide a hybrid coding scheme for RDH. Entropy coding and hierarchical coding make up the hybrid coding method. Blocks are used to separate an image. Every block is encoded using either hierarchical or entropy coding. Additionally, the block is encoded using the coding technique that has more free space so that the embedding capacity is increased.

Assume that I is the original, 8-bit grayscale image with a size of H×W. First, the original image I is separated into n non-overlapping blocks, denoted by the numbers B_1, B_2,..., and B_Z, which are scanned in raster order. Every block has a t×t dimensions. The formula $z=\lfloor H/t\rfloor\times\lfloor W/t\rfloor$ is evident. The prediction value of a block's pixel $p_{i,j}$ is determined as

$$\tilde{p}_{i,j} = \begin{cases} a, \text{ if } i=1 \text{ and } j\geq 2 \\ b, \text{ if } i\geq 2 \text{ and } j=1 \\ \max(a,b), \text{ if } 2\leq i,j\leq t \text{ and } c\leq \min(a,b) \\ \min(a,b), \text{ if } 2\leq i,j\leq t \text{ and } c\geq \max(a,b) \\ b+a-c, \text{ Otherwise} \end{cases} \quad (1)$$

in where $b = p(i-1, j)$, $c = p(i-1, j-1)$, and $a = p(i, j-1)$.
The median edge detector (MED) predicts the remaining pixels, while the preorder pixels of the first row or column anticipate the remaining pixels. Keep in mind that the reference pixel, $p_{1,1}$, is not included in the data concealing. Following that, the prediction error (PE) is produced by

$$e_{i,j} = \tilde{p}_{i,j} - p_{i,j} \quad (2)$$

Each block's PEs are computed using Equation (2), and they are then converted into the one-dimensional PE sequence $e = [e_2, e_3,..., e_f]$ $(f = t\times t)$ in the order shown in Figure 1. The numbers in Fig. 1 represents the PEs' indexes in the one-dimensional sequence. The entire image can then be obtained as a PE sequence, $PE = \{e_1, e_2,....e_z\}$, with a dimension of $(t \times t - 1) \times z$. Next, each block is encoded using the hierarchical and entropy coding methods, respectively. The block is encoded using the coding scheme that has more free space. The following is an illustration of these two coding strategies.

$p_{1,1}$	0	1	2
3	6	7	8
4	9	10	11
5	12	13	14

Fig. 1: Scan Order

A. Hierarchical Coding

The hierarchical coding in this part is done in a single block. As an example, we consider a single block and its PEs. In the PE sequence $[\,|e_2|, |e_3|,..., |e_f|\,]$, let me be the maximum element. Me is first divided into three categories in [41]: me = 0, 0 < me < 64, and me ≥ 64. This block is designated l = 0 and all of the block's pixels, with the exception of $p_{1,1}$, can hold eight secret bits if me = 0. A secret bit cannot be inserted into this block if me is less than 64. This block is designated by l = 7. It is split hierarchically into a set of nodes when $0 < me < 2^6$. The root node $0 < me < 2^6$ is split into three nodes, $0 < me < 2^5$, $me = 2^5$, and $2^5 < me < 2^6$, as illustrated in Fig. 2. $0 < me < 2^{q-1}$, $me = 2^{q-1}$, and $2^{q-1} < me < 2^q$ make up the three nodes that make up the left node, $0 < me < 2^q$. Every layer is associated with a specific bit of the pixel; for example, the (q + 1)th bit corresponds to the (8 − q)th layer. Which layer mes is on is determined by the leaf node. The block label is considered to represent the layer index. In [41], the block label is $l = 8 - q - 1$, where $1 \le q \le 6$. If $2^{q-1} < me < 2^q$, me returns to the left node $0 < me < 2^q$ of the higher layer as the red arrow depicted in Fig. 2. With the exception of $p_{1,1}$, all pixels' (q + 2)th ∼ 8th bits can hold 8−q−1 secret bits. Consequently, the payload of this block is as follows when me is situated on the left node $0 < me < 2^q$ of the (8 − q − 1) th layer:

$$EC = (f-1) \times (8-q-1) \quad (3)$$

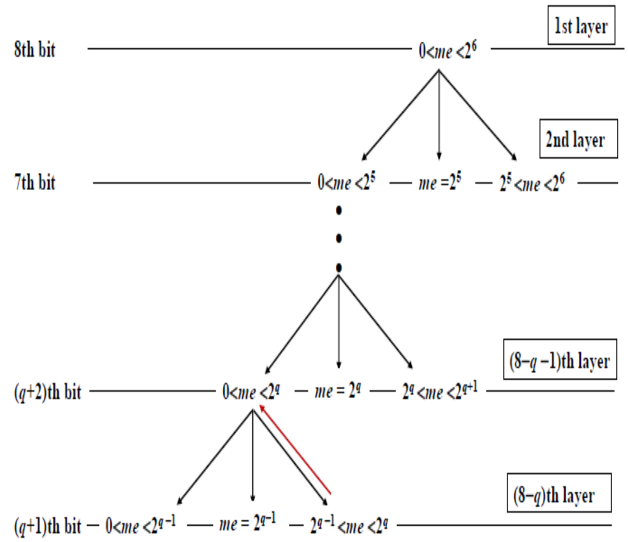

Fig. 1. Hierarchical Structure.

In this work, the payload gain determines whether me with $2^{q-1}<me<2^q$ returns the upper layer or not. For each $|e_i|(2\le i\le f)$ in the block where $2^{q-1}<me<2^q$, there are three possible outcomes: $|e_i|\in[0,2^{q-1}-1]$, $|e_i|=2^{q-1}$, and $|e_i|\in[2^{q-1}+1,2^q-1]$; these three situations can hold 8−q, 8−q−1, and 8−q secret bits, respectively, in accordance with the approach [31]. It is evident that l=8−q is the block label. To keep things simple, we will refer to the PEs with the numbers n_0, n_1, n_2, and n3 as C_0, C_1, C_2, and C_3, respectively, and the expressions $|e_i|=0$, $|e_i|\in[1,2^q-1]$, $|e_i|=2^{q-1}$, and $|e_i|\in[2^{q-1}+1,2^{q-1}]$. As illustrated in Fig. 2, when me with $2^{q-1}<me<2^q$ is situated on the (8−q)th layer, the block's payload is

$$ECp = (8-q) \times (n_0 + n_1 + n_3) + n_2 \times (8-q-1) \quad (4)$$

where $n_0+n_1+n_2+n_3 = f - 1$.

It is important to note that C_2 payload is the same whether it is in the (8−q−1) th layer or the (8−q) th layer. Consequently, in contrast to the (8−q−1) th layer, the block's (8−q) th layer's payload gain is

$$EC_g = EC_p - EC = n_0 + n_1 + n_3 \quad (5)$$

when $2^{q-1}<me<2^q$.
When me with $2^{q-1}<me<2^q$ is found on the (8−q)th layer, the payload of this block can be enhanced; however, more information is needed to differentiate between C_1 and C_3 during data extraction and image recovery. It should be noted that during data extraction and picture recovery discussed in Section D C_0 and C_2 can be adaptively determined.

The layer on which me is located is determined by calculating the pure payload gain, which is used to improve the block's pure payload. As $2^{q-1}<me<2^q$ allows for the adaptive determination of C_0 and C_2, the only thing left to do is differentiate between C1 and C_3. Generally speaking, there are fewer C3s than C_1. To store the index of every C_3 in $[|e_2|, |e_3|,\ldots, |e_f|]$, more bits are needed. It is possible to determine the index of C_1 after determining the index of C_3. First, in order to record the value of n3, ⌈log2(t×t−1)⌉ bits are needed. Next, we record each C_3 index using the variable bit length. Assume that each C_3 has a location index value of $\{xi\}^{n3}i=1(0\le x_i\le f-2)$. Then, each C_3 index can be expressed in terms of r_i bits, which are computed as follows.

$$r_i = \begin{cases} \lceil \log_2(f-1) \rceil, & \text{if } i = 1 \\ \max\{\log_2(f-1-x_{i-1}), 1\}, & \text{if } 2 \le i \le n_3 \end{cases} \quad (6)$$

As a result, the index information length of C3 in a block is determined as

$$le \models \lceil \log_2(t \times t - 1) \rceil + \sum_{i=1}^{n_3} r_i \quad (7)$$

Le bits can be used to encode the index information of C_3 in the way described above. As such, the block label is produced by

$$l = \begin{cases} 8-q-1, & \text{if } ECg <= le \\ 8-q, & \text{if } ECg > le \end{cases} \quad (8)$$

In this case, $1 \le l \le 6$. When me with $2^{q-1} < me < 2^q$ is positioned on the $(8-q)$th layer, $EC_g \le le$ indicates that there is no yield on payload. And I ought to give back the top layer, which is the $(8-q-1)$th layer. This block's payload, EC, is determined by using Equation (3). In accordance with this, the block label is $8-q-1$. The block label is $8-q$ and the payload may be computed using Eq. (4) if $EC_g > le$. PEC = $EC_g - le > 0$ represents the pure payload benefit. Subsequently, since $0 \le l < 7$, the block label l is divided into three bits. Furthermore, an additional bit badd = 1 indicates that mes is subject to $2^{q-1} < me < 2^q$ when it is positioned on the $(8-q)$th layer. Furthermore, if me is on the $(8-q)$th layer, more bits badd = 1 mean that mes is subject to $2^{q-1} < me < 2^q$, and more bits badd = 0 mean that mes is susceptible to $0 < me < 2^{q-1}$ or $me = 2^{q-1}$. To capture the index information of C_3 for $0 < me < 2^{q-1}$ or $me = 2^{q-1}$, it is evident that no bits are needed.

The detailed encoding information for several scenarios when the original image is an 8-bit gray-level image is shown in Fig. 3. The block label bits include the encoding information for the block with l = 0, l = 7, or l = 1. The block label bits include the encoding information for the block with l = 0, l = 7, or l = 1. The block label bits, an extra bit badd = 1, and the index information of C_3 make up the encoding information for the block with $2 \le l \le 6$, or in this work, the block label bits and an additional bit badd = 0. Ultimately, the block's embedding capacity produced via hierarchical coding is determined by

$$EC_{hier} = \begin{cases} 8 \times (t \times t - 1) - 3, & \text{if } l = 0 \\ 0 - 3, & \text{if } l = 7 \\ t \times t - 1 - 3, & \text{if } l = 1 \\ ECp - 4, & \text{if } 2 \le l \le 6 \text{ and } b_{add} = 0 \\ ECp - le - 4, & \text{if } 2 \le l \le 6 \text{ and } b_{add} = 1 \end{cases} \quad (9)$$

where Eqs. (4) and (7) can be used to yield EC_p and le, respectively, for $l = 8 - q$. Given that there are no PEs with C_3, take note that $n_3 = 0$ if badd = 0 in Eq. (4).

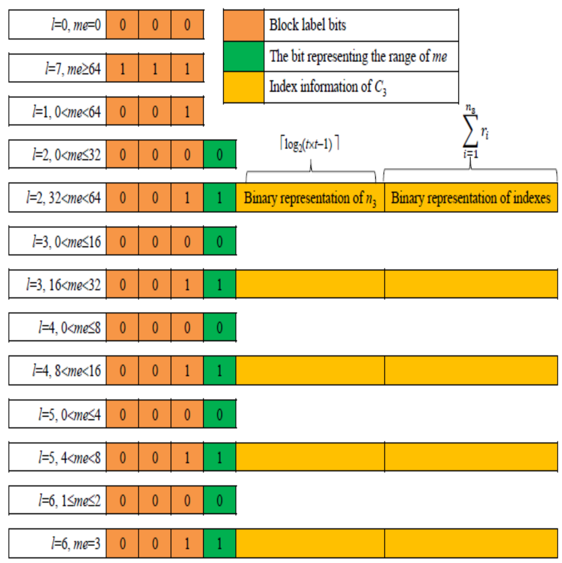

Fig. 3: Hierarchical encoding information

B. Entropy coding

Huffman and arithmetic coding are two popular lossless entropy coding methods. The codewords and symbols in Huffman coding correspond exactly to one another. Thus, for every block, we compute the embedding capacity using Huffman coding. First, we compress the image's PEs using Huffman coding. $\{e_1, e_2, \ldots e_z\}$ is the PE. The codewords c(−255), c(−254),..., c(254), c(255) formed using Huffman coding, whose length are len(−255), len(−254),..., len(254), len(255), are obtained since the PE falls within the range of [−255, 255]. Next, we may determine one block B's Huffman encoding information's length by

$$len_{\text{huff}} = \sum_{i=2}^{f} len(e_i) \quad (10)$$

The embedding capacity of one block generated by Huffman coding is derived by

$$EC_{\text{huff}} = 8 \times (t \times t - 1) - len_{\text{huff}} \quad (11)$$

C. Data embedding using hybrid coding

Blocks inside the designated image are separated in raster scanning sequence. In case the LSBs of the initial pixels \p1,1 are "0," the blocks are gathered. With the exception of the first pixel, each pixel in these blocks is converted into a binary sequence that contains embedded data and auxiliary information. It is evident that the substituted LSBs, entropy encoding information, and hierarchical encoding information make up the auxiliary information. The first pixel of each block can be retrieved as $p_{1,1}$ with the new LSBs. Decoding the entropy encoding information yields the blocks that start with "0." Subsequently, the blocks containing the number "1" can have their remaining embedded data retrieved and recovered using the hierarchical coding information.

IV. PROPOSED RDHEI METHOD WITH SECRET SHARING AND HYBRID CODING

The suggested RDHEI method's framework, which includes data concealing, data exaction and picture recovery, and iterative encryption and image sharing, is shown in Fig. 6. To further security, the content owner uses iterative encryption to create an encrypted image. This encrypted image is then shared via block-based CRTSS, and several encrypted shares are sent to multiple data hiders. He or she can use hybrid coding to separately perform RDH at each data-hider side. Using CRTSS, the original image can be retrieved without loss once the shareholders have provided enough marked encrypted shares.

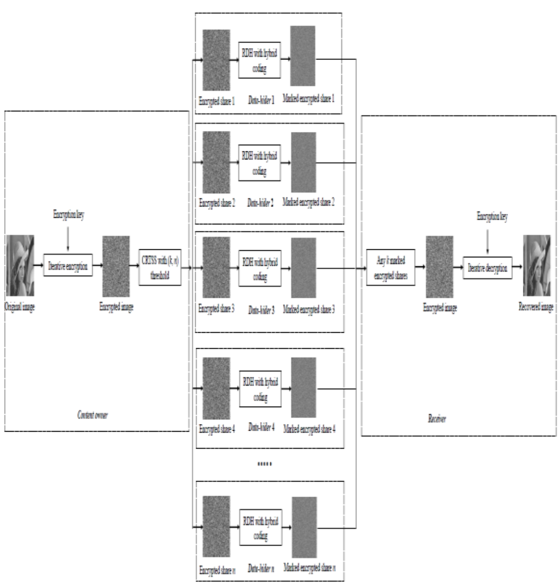

Fig. 6: Framework of the proposed RDHEI method

ABE System Algorithm:

AES – Encryption and Decryption Specification

The image can only be viewed by the receiver as the image is encrypted using AES and the key is only known to the sender and receiver Since the image is encrypted using AES, it is more secure than the DES and triple DES.

AES is called AES-128, AES-192 and AES-256. This classification depends on the different key size used for cryptographic process. Those different key sizes are used to increase the security level. As, the key size increases the security level increases. Hence, key size is directly proportional to the security level. The input for AES process is a single block of 128 bits. The processing is carried out in several number of rounds where it depends on the key length: 16 byte key consists of 10 rounds, 24 byte key consists of 12 rounds, and 32 byte key consists of 14 rounds. The first round of encryption process consists of four distinct transformation functions:

- Substitution Bytes
- ShiftRows
- MixColumns
- AddRoundKey

The final round consists of only three transformation ignoring MixColumns. The Decryption method is the reverse of encryption and it consists of four transformations [4].

- Inverse Substitution Bytes
- Inverse ShiftRows
- Inverse MixColumns
- AddRoundKey

- **Setup (λ, U):** The setup algorithm takes as input a security parameter and attribute universe U, and outputs a master secret key MSK and the public parameters PP.
- **Encrypt (PP, A, M):** The encryption algorithm takes as input the public parameters PP, an access structure A and a message M, and outputs a ciphertext CT.
- **KeyGen (MSK, S):** The key generation algorithm takes as input the master secret key MSK and an attribute set S, and outputs a secret key SK.
- **Decrypt (PP, SK):** The decryption algorithm takes as input the public parameters PP, a secret key SK.

5. RESULT

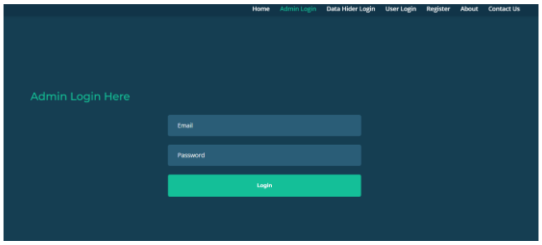

Fig 2. Admin Login Page

Fig 3. Admin Home Page

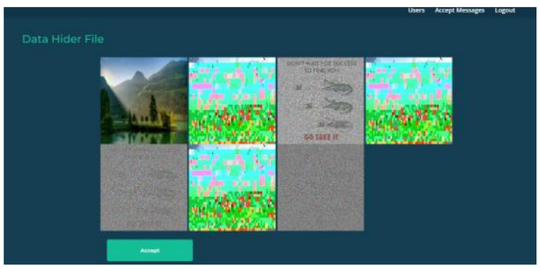

Fig 4: Data Hider Page

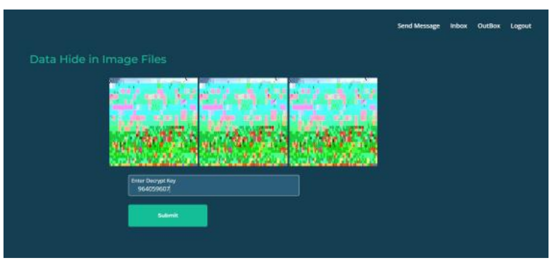

Fig 5. Receiver viewing messages with decryption key Page

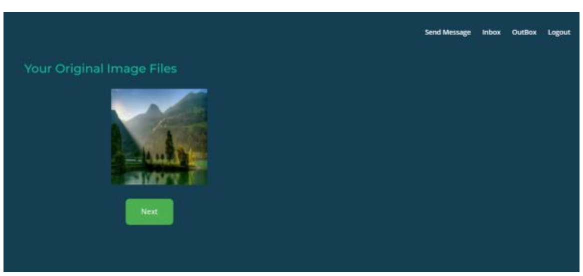

Fig 6. Receiver receiving the image

6. CONCLUSION AND FUTURE ENHANCEMENT

Our innovative RDHEI technique makes use of hybrid coding and secret sharing. A substantial embedding room for each encrypted share can be achieved by the suggested method's combination of block-based CRTSS and iterative encryption, which can effectively pervert correlations within the blocks. Hybrid coding is used in each encrypted exchange to hide data with a large payload. Furthermore, the suggested approach does not call either pre-processing or pixel extension. According to experimental findings, the suggested approach performs better in the payload than a few cutting-edge SS-based RDHEI techniques.

Future Work: Future research will look into how to use the suggested model in additional scenarios, including a combined RDH-EI approach. Expanding the concept of secret sharing to include other forms of multimedia, including audio and video, is also an intriguing avenue to pursue.

REFERENCES

[1] Y. Du, Z. Yin, and X. P. Zhang, “High capacity lossless data hiding in jpeg bit stream based on general vlc mapping,” IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 2, pp. 1420–1433, 2022.

[2] Z. Wang, G. Feng, L. Shen, and X. Zhang, “Cover selection for steganography using image similarity,” IEEE Transactions on Dependable and Secure Computing, pp. 1–13, 2022.

[3] Y. Xian, X. Wang, and L. Teng, “Double parameters fractal sorting matrix and its application in image encryption,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 32, no. 6, pp. 4028– 4037, 2022.

[4] X. Wang and S. Gao, “Image encryption algorithm for synchronously updating boolean networks based on matrix semi-tensor product theory,” Information sciences, vol. 507, pp. 16–36, 2020.

[5] Z. Tang, L. Chen, X. Q. Zhang, and S. Zhang, “Robust image hashing with tensor decomposition,” IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 3, pp. 549–560, 2019.

[6] X. Liang, Z. Tang, J. Wu, Z. Li, and X. P. Zhang, “Robust image hashing with is map and saliency map for copy detection,” IEEE Transactions on Multimedia, pp. 1–1, 2021.

[7] X. Liang, Z. Tang, Z. Huang, X. Q. Zhang, and S. Zhang, “Efficient hashing method using 2D-2D PCA for image copy detection,” IEEE Transactions on Knowledge and Data Engineering, pp. 1–1, 2021.

[8] C. Asmuth and J. Bloom, “A modular approach to key safeguarding,” IEEE transactions on information theory, vol. 29, no. 2, pp. 208–210, 1983.

[9] X. Yan, Y. Lu, L. Liu, and X. Song, “Reversible image secret sharing,” IEEE Transactions on Information Forensics and Security, vol. 15, pp. 3848–3858, 2020.

[10] A. Shamir, “How to share a secret,” Communications of the ACM, vol. 22, no. 11, pp. 612–613, 1979.

[11] X. Yan, Y. Lu, C.-N. Yang, X. Zhang, and S. Wang, “A common method of share authentication in image secret sharing,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 31, no. 7, pp. 2896– 2908, 2021.

[12] J. Tian, “Reversible data embedding using a difference expansion,” IEEE transactions on circuits and systems for video technology, vol. 13, no. 8, pp. 890–896, 2003.

[13] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, “Reversible data hiding,” IEEE Transactions on circuits and systems for video technology, vol. 16, no. 3, pp. 354–362, 2006.

[14] W. Zhang, X. Hu, X. Li, and N. Yu, “Recursive histogram modification: establishing equivalency between reversible data hiding and lossless data compression,” IEEE transactions on image processing, vol. 22, no. 7, pp. 2775–2785, 2013.

[15] W. Zhang, X. Hu, X. Li, and Y. Nenghai, “Optimal transition probability of reversible data hiding for general distortion metrics and its applications,” IEEE Transactions on Image Processing, vol. 24, no. 1, pp. 294–304, 2014.

[16] C. Yu, X. Q. Zhang, D. Wang, and Z. Tang, “Reversible data hiding with pairwise PEE and 2D-PEH decomposition,” Signal Processing, vol. 196, p. 108527, 2022.

[17] D. Wang, X. Q. Zhang, C. Yu, and Z. Tang, “Reversible data hiding by using adaptive pixel value prediction and adaptive embedding bin selection,” IEEE Signal Processing Letters, vol. 26, no. 11, pp. 1713–1717, 2019.

[18] W. Puech, M. Chaumont, and O. Strauss, “A reversible data hiding method for encrypted images,” in Security, forensics, steganography, and watermarking of multimedia contents X, vol. 6819, p. 68191E, International Society for Optics and Photonics, 2008.

[19] P. Puteaux and W. Puech, “An efficient msb prediction-based method for high-capacity reversible data hiding in encrypted images,” IEEE Transactions on Information Forensics and Security, vol. 13, no. 7, pp. 1670–1681, 2018.

.