Reversible Data Hiding in Encrypted JPEG Images with Polynomial Secret Sharing for IoT Security

Shuying Xu, Ji-Hwei Horng*, Ching-Chun Chang and Chin-Chen Chang*, *Fellow, IEEE*

***Abstract*—Crypto-space reversible data hiding has emerged as an effective technique for transmitting secret information over the internet. However, most existing schemes are designed for uncompressed images, while almost all images are processed and transmitted in compressed formats. There is an urgent need to develop methods for compressed images, such as JPEG (Joint Photographic Experts Group). In this paper, we propose a reversible data hiding in encrypted JPEG images, where the bitstreams of AC (alternating current) coefficients and the secret data are mapped to numbers over Galois field. The obtained numbers are then utilized to conduct a polynomial for secret sharing. By reproduction into secret shares, the AC coefficients and the secret data are secured. In addition, a block sorting strategy is used to reduce image distortion under low data payload. Experimental results demonstrate that the proposed scheme outperforms state-of-the-art methods in embedding capacity while preserving the file size and conforming to the JPEG format.**

***Index Terms*—Crypto-space steganography, Galois field, JPEG, reversible data hiding, secret sharing.**

I. INTRODUCTION

DATA hiding is an information technique that embeds secret message into cover media, such as images, texts, or videos. The media can then be transmitted through the public channel without drawing the attention of eavesdroppers. Based on restorability of the cover media, it can be classified into two categories, reversible data hiding (RDH) and irreversible data hiding. The irreversible data hiding causes permanent damage to the cover media, making it unsuitable for certain application scenarios where the distortion is intolerable. In contrast, RDH is considered a better option as it can recover the cover media without any distortion. Since digital images are the most popular form of media, various image RDH schemes have been developed in recent years. These include lossless compression approaches [1], difference expansion [2], [3], histogram shifting [4], [5], [6], [7], and prediction error expansion [8], [9].

However, in certain application scenarios such as medical imaging, the cover image may contain sensitive information, conventional RDH schemes might not offer sufficient privacy protection for cover images. Once the unprotected cover images are transmitted over public channel, it is vulnerable to malicious attacks. To solve this issue, the reversible data hiding in encrypted images (RDHEI) schemes has emerged. It encrypts the cover image before transmitting via the public channel to ensure security. In most RDHEI approaches, the cover image is compressed to free up a space for data hiding. According to the processing order of image encryption and space vacating, the existing RDHEI schemes can be divided into three categories, reserving room before image encryption (RRBE) [10], [11], [12], [13], vacating room by image encryption (VRBE) [14], [15], and vacating room after image encryption (VRAE) [16], [17], [18], [19].

Despite their effectiveness, these methods are designed for a single receiver, which may result in data loss in the event of receiver failure. To address this problem, various RDHEI schemes based on secret sharing techniques have been proposed. In 2021, Qin et al. [20] presented an RDHEI scheme using Shamir's secret sharing over Galois fields. Their method involves dividing the original image into blocks, scrambling the positions of all blocks and the pixels within each block, and applying Shamir's secret sharing for image encryption. Additional data is embedded into the generated encrypted shares based on the preserved pixel differences within blocks.

In 2023, Hua et al. [21] proposed a matrix-based secret sharing for RDHEI scheme, where the original image is divided into blocks that are subsequently encrypted using matrix-based secret sharing. Afterwards, the block error mixture encoding (BEME) method is used to embed additional data. Following that, Yu et al. [22] employed iterative encryption for block-based encryption and utilized Chinese remainder theorem-based secret sharing (CRTSS) to generate encrypted shares. Additional data is embedded into these shares using a hybrid coding method. By employing secret sharing techniques, these schemes ensure that even if some shares are lost or corrupted, the additional data can still be retrieved and the original image can be reconstructed.

The RDH-EI schemes introduced in [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22] are based on plaintext images, which are not suitable for applying to compressed formats. However, compressed image formats, like JPEG (Joint Photographic Experts Group), are more commonly used in application software. Therefore, research on RDHEI for compressed image formats holds significant value. Recently, several RDHEI schemes for JPEG images have been proposed [23], [24], [25], [26], [27], [28].

In 2014, Qian et al. [23] proposed a framework of RDH-EI for JPEG images. The JPEG bitstream is encrypted with the

Corresponding author: Ji-Hwei Horng and Chin-Chen Chang.

S. Xu and Chin-Chen Chang are with the Department of Information Engineering and Computer Science, Feng Chia University, Taichung, Taiwan. E-mail: shuying.xu.phd@gmail.com; alan3c@gmail.com.

J.-H. Horng is with the Department of Electrical Engineering, National Quemoy University, Kinmen, Taiwan. E-mail: horng@email.nqu.edu.tw.

Ching-Chun Chang is with the Information and Communication Security Research Center, Feng-Chia University, Taiwan. E-mail: ccc@fcu.edu.tw.

exclusive-or operation. Then, the encrypted bitstream is modified directly to embed secret data. In addition, the error correction code (ECC) is utilized to ensure both lossless data extraction and image recovery. Later, Chang et al. [24] introduced an RRBE-based scheme for JPEG images. In their method, the least significant bits (LSB) of the two-bit appended values (-3, -2, 2, 3) are collected as a biased bitstream. This bitstream is compressed with a new algorithm to reserve room for data embedding. After that, the compressed JEPG image is encrypted as the method utilized in [23], and the secret data is embedded into the reserved room directly.

In 2018, Qian et al. [25] split the entropy-encoded JPEG image blocks into two groups. The first block group is encrypted with exclusive-or operation and reshaped into a smaller JPEG image. The second block group is encrypted and put into the modified JPEG header together with the new file size information. In the data hider side, the appended bits of the encrypted AC (alternating current) coefficients in the header are compressed to vacate the room for data hiding. In the next year, Qian et al. [26] further proposed a new method that shares the same framework. This time, a combined data embedding method, including Huffman code mapping and an ordered histogram shifting, is employed to improve the data payload. However, using the exclusive-or operation for DC coefficients in [23], [24], [25], and [26] can lead to overflow or underflow issues during JPEG decoding.

In [27], He et al. proposed a method for encrypting plaintext JPEG images that involves scrambling and swapping the DC (direct current) coefficients, permuting AC coefficients within the same category, and exchanging AC coefficients block-wise. Secret data is embedded into the encrypted images using an invariant zero-run length approach. In 2022, a permutation-based RDH scheme for JPEG images is proposed by Hua et al. [28]. The DC coefficients are encrypted with the encryption method introduced in [29], while the AC coefficients are encrypted with position scrambling and advanced encryption standard (AES). After that, a permutation-based embedding is applied to embed the secret data. These methods suffer from low data payload, increased file size, and disruption of the file format. Moreover, they rely on a single receiver. In the event of receiver failure, it becomes impossible to recover either the original image or the secret data.

We propose a reversible data hiding in encrypted JPEG images with polynomial secret sharing. In our scheme, the DC coefficients are encrypted using a sketch-based method proposed by Minemura et al. [30], which will be introduced in Section III.B. The AC coefficients, along with the secret data, undergo manipulation using (k, n)-threshold polynomial secret sharing over the Galois field to generate n shares of encrypted AC coefficients. Each share of AC coefficients is combined with a single encrypted DC part, resulting in n shares of JPEG image shadows distributed among n participants. Only when k or more participants share their shadows, the original JPEG image together with the embedded secret data can be restored and extracted.

With the rapid proliferation of Internet of Things (IoT) applications, numerous devices connected to the internet are capable of capturing, transmitting, or manipulating digital images. Fig. 1 presents an example application of a smart home monitoring system, illustrating how our method can provide secure transmission of both digital images and secret data. In this scenario, IoT devices such as digital cameras continuously capture images within the household premises. A smart home gateway uploads the captured images, along with timestamps, to the cloud for storage and processing. By implementing our secret sharing scheme on the gateway computer, a JPEG image along with additional secret data can be distributed into n shadows and securely transmitted through different routes on the Internet. Access to the image and secret data requires the simultaneous operation of multiple authorized IoT devices such as smartwatches, cellular phones, or notebooks, as k shadow shares are needed to restore them. In this way, our approach provides secure communication in the IoT environment.

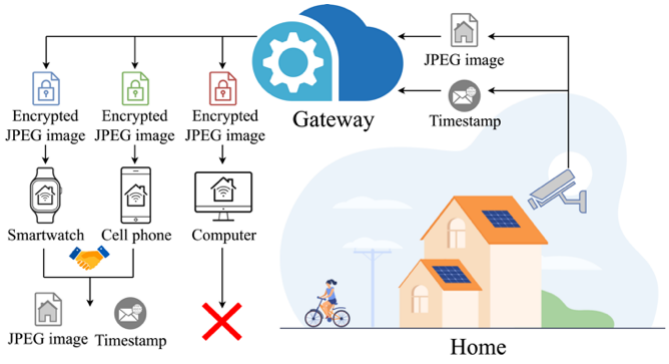


Fig. 1. A smart home monitoring system.

We propose a novel reversible data hiding technique in encrypted JPEG images, where polynomial secret sharing over the Galois field $GF(2^m)$ serves as the core technique for image encryption and data embedding. The contributions of the proposed scheme are outlined below:

(1) It can tolerate participant failures.

(2) It outperforms state-of-the-art methods in embedding capacity while preserving file size and format.

(3) A block sorting strategy is introduced to reduce image distortion under low data payload conditions.

(4) It is applicable to a wide range of IoT-based security applications.

The rest parts of this paper are as follows. In Section II, we give an overview of the JPEG image format and the Shamir's polynomial secret sharing. Technical details of the proposed scheme are presented in Section III. Section IV provides the experimental results and discussions. The conclusions are made in Section V.

II. Preliminary

Our data hiding scheme employs the Shamir's polynomial secret sharing to embed secret data into AC coefficients of the JPEG image shadows. Before introducing our approach, we first briefly describe the JPEG image file format and the polynomial secret sharing.

A. JPEG Image File Format

JPEG is a popular image compression format, which effectively compresses the data volume of digital images while maintaining high fidelity. The JPEG compression procedures can be summarized as follows.

1) Color space transformation and chroma subsampling: A color image is converted from the RGB, for red, green, and blue channels, color space to the YCbCr, Y for the luma component and CbCr for the chroma components, color space. The obtained chroma components are further subsampled to reduce their spatial resolution.
2) Partitioning and dynamic range shifting: All components are split into 8×8 blocks and entries in each block are subtracted by 128. Thus, the dynamic range is shifted to be centered around zero.
3) Discrete Cosine Transform (DCT): DCT is applied to each block of all components. For each 8×8 block, the coefficient at the top-left corner is the DC component and the remaining 63 coefficients are the AC components.
4) Quantization: The coefficients are quantized by dividing each block with the quantization matrix elementwise and rounding to integers. The image quality and compression ratio can be controlled by magnifying the quantization matrix.
5) Entropy coding: Each quantized coefficient matrix is first scanned in the "zigzag" order to form a sequence. Then, each sequence is encoded into a series of triplets using run length coding (RLC) [31]. Specifically, each triplet $c_j = [(r_j, s_j)/v_j]$ corresponds to a non-zero coefficient, where r_j is the length of zero run before the current coefficient, s_j is the length indicator of the coefficient value, and v_j records its value. The leading DC coefficient is differenced block-wise and expressed in doublet as $c_0 = [s_0/dv_0]$, where s_0 is length indicator of the differential DC value, and dv_0 records the differential value. Huffman coding is applied to encode all elements. An end-of-block (EOB) is appended to the end. The block encoding is illustrated in Fig. 2.

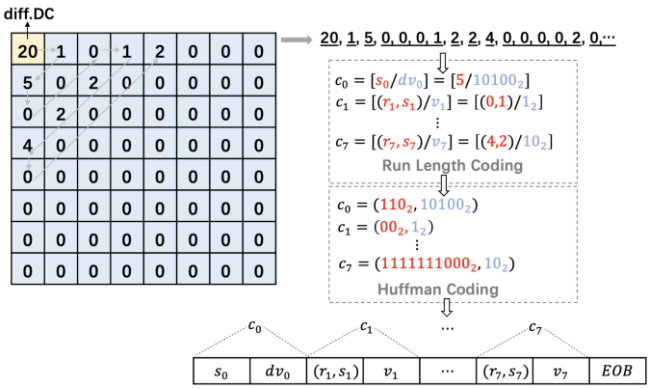


Fig. 2. An illustration of the entropy coding in JPEG.

B. Shamir's Polynomial Secret Sharing over Galois Field

Shamir's secret sharing is a typical (k, n) -threshold secret sharing that secures a secret data in distributed form [32]. In the method, the secret data is converted into n shadow shares and dealt to n participants. The secret data can be reconstructed by combining any $k (k \leq n)$ or more shares, while fewer than k shares provide no clue of the secret. The basic idea behind Shamir's secret sharing is to construct a polynomial over the Galois field, which is a field of finite elements. One common example is the Shamir's secret sharing over $GF(2^m)$. In which, the addition and subtraction are implemented by the exclusive-or operation denoted by $\oplus$, while multiplication and division are implemented by the Galois field multiplication and the extended Euclidean algorithm, respectively. Details of the Shamir's secret sharing over $GF(2^m)$ are given below.

To share a secret data by the Shamir's (k, n)-threshold secret sharing scheme over Galois field $GF(2^m)$, a polynomial of order $(k-1)$ is constructed as

$$f(x) = (a_0 \oplus a_1 x \oplus a_2 x^2 \oplus \cdots \oplus a_{k-1} x^{k-1}) \bmod p(x), \quad (1)$$

where a_0 is the secret data, $a_1, a_2, \cdots, a_{k-1}$ are $k-1$ randomly selected numbers over $GF(2^m)$, and polynomial $p(x)$ is a fixed irreducible polynomial. Then, the polynomial is evaluated at n distinct identity numbers $x_1, x_2, \cdots, x_n$ over $GF(2^m)$ to obtain their corresponding shadow values $f(x_1), f(x_2), \cdots, f(x_n)$. Here, m must satisfy the condition $m \geq \lceil \log_2(n+1) \rceil$. Otherwise, the number of distinct nonzero elements in $GF(2^m)$ are not enough to provide a distinct identity number for each participant. After that, the n pairs of identity number and shadow value $(x_i, f(x_i)), i = 1, 2, \ldots, n$ are dealt to the n participants.

When any k participants share their identities and shadow values, the polynomial $f(x)$ of order $(k-1)$ and the secret data a_0 can be recovered using the Lagrange interpolating polynomial

$$f(x) = \sum_{t=1}^{k} \left(f(x_t) \prod_{1 \leq w \leq k, w \neq t} \frac{x - x_w}{x_t - x_w} \right) \bmod p(x) \quad (2)$$

and

$$a_0 = \sum_{t=1}^{k} \left(f(x_t) \prod_{1 \leq w \leq k, w \neq t} \frac{-x_w}{x_t - x_w} \right) \bmod p(x). \quad (3)$$

Note that m merely satisfying the condition $m \geq \lceil \log_2(n+1) \rceil$ may not provide sufficient security in practical applications. Specifically, for the Galois field $GF(2^m)$, there are $C_n^{2^m-1} \times n!$ possible permutations of n distinct nonzero identity numbers. Choosing a smaller value for m may render the identity numbers more vulnerable to cracking. As a result, it is advisable to select a greater value for m, such as $m = 8$, to ensure a higher level of security.

III. Methodology

In this section, we present an RDH-EI scheme for JPEG images as illustrated in Fig. 3. The proposed scheme comprises two main phases: (1) JPEG encryption and data embedding, and (2) image decryption and data extraction. In the phase of JPEG encryption and data embedding, the dealer encrypts JPEG images using the sketch-based encryption [30] and the Shamir's secret sharing technique. The secret data is embedded during the secret sharing procedure. The dealer generates n shadow images and distributes

them to n participants. The participants can obtain the DC decrypted JPEG image, AC decrypted JPEG image, original JPEG image, or secret data based on their actual situation. Details are as follows.

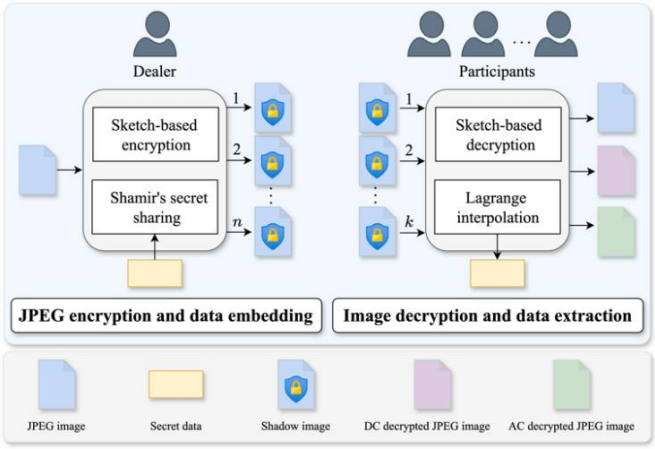


Fig. 3. An overview of the proposed scheme.

A. JPEG Encryption and Data Embedding

To encrypt the JPEG image, the dealer employs a sketch-based encryption method [30] to encrypt the DC coefficients and utilizes the Shamir's (k, n)-threshold secret sharing over Galois field to encrypt the AC coefficients as shown in Fig. 4. The encryption process generates an encrypted version of the DC coefficients and n encrypted versions of the AC coefficients. By combining the same encrypted DC version with each of the n encrypted AC versions, n shadow images are generated and distributed to n participants. Additionally, the secret data is embedded during the AC coefficients encryption.

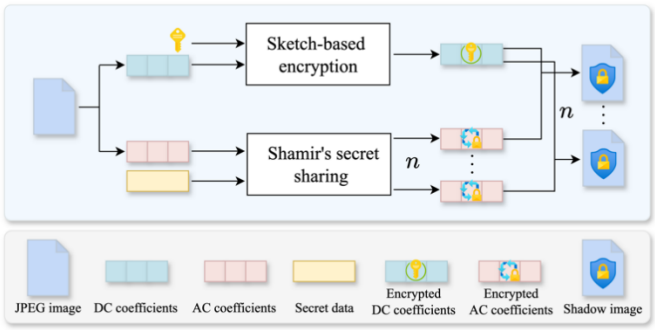


Fig. 4. The workflow of JPEG encryption.

1) DC Coefficients Encryption: Fig. 5 provides the schematic diagram of the DC encryption method proposed by Minemura et al. [30]. This method utilizes two inherent features of JPEG images. Firstly, the pixel values within an object or a connected region tend to have a low variance. Secondly, the DC coefficient of a block represents its average intensity value. In the method, a sketch of the original JPEG image is first obtained by

$$\text{Sketch}_{(\alpha,\beta)} = \text{round}(255 \times \frac{l_{(\alpha,\beta)}}{\max\{l_{(\alpha,\beta)}\}}), \qquad (4)$$

where $l_{(\alpha,\beta)}$ denotes the number of non-zero AC coefficients for the block indexed (α, β), and $\max\{l_{(\alpha,\beta)}\}$ denotes the maximum number of non-zero AC coefficients among all the image blocks. The obtained sketch image reveals the local complexity of the original image. That is, a higher $\text{Sketch}_{(\alpha,\beta)}$ value indicates a more complex texture in the image block.

Subsequently, the sketch image is converted into a binary image using Otsu's thresholding method [33]. Specifically, the histogram of the sketch image is first computed. Each potential threshold is applied to segment the sketch image into foreground and background, then the between-class variance is calculated. The threshold that maximizes this variance is chosen as the optimal threshold and then applied to binarize the sketch image. Next, morphological operations, including erosion and dilation, are applied to refine the binary image. Erosion erases small noise by shrinking foreground boundaries, while dilation expands boundaries to fill gaps and connect disjoint regions.

Then, the regions in the binary image are consecutively labeled. For each labeled region, the DC coefficients are extracted horizontally, i.e., left-to-right on odd rows and right-to-left on even rows. The extracted DC coefficients of each labeled region are sequentially added to a queue. These queued DC coefficients are then dequeued in order to generate an array of DC coefficients. Lastly, the DC array is combined with the AC coefficients to obtain the rearranged JPEG image. To improve security, the region labels are permuted, and the DC values are mapped to another value in the same category using the DC encryption key K_{DC}.

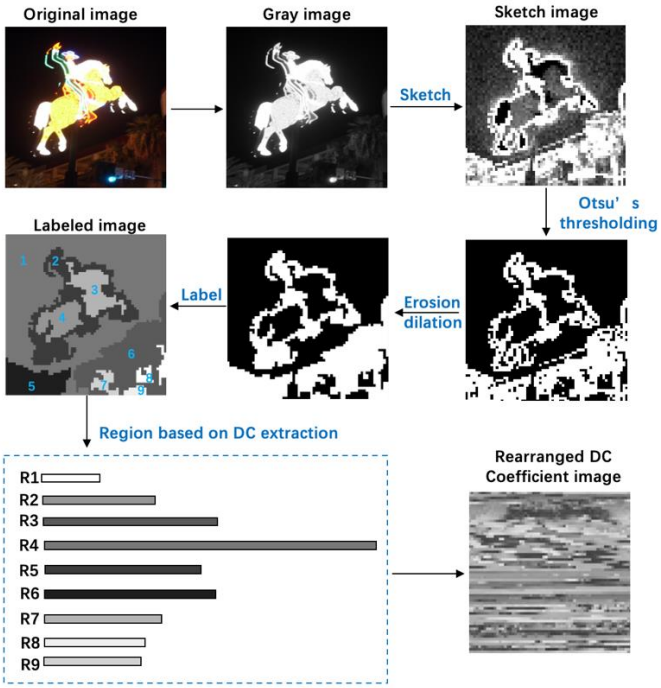


Fig. 5. The schematic diagram of DC encryption.

2) AC Coefficients Encryption and Data Embedding: The proposed scheme uses the (k, n)-threshold secret sharing over Galois field to accomplish both AC coefficients encryption and data embedding. Specifically, the AC coefficients of an image

block are combined to constitute the constant term of the polynomial in Eq. (1), while the remaining coefficients of the polynomial are constituted by secret bits. Thus, the AC coefficients of a block together with some secret bits are encrypted into n shares of AC coefficients.

Let us assume that the given image block contains l non-zero quantized AC coefficients. We first follow the entropy coding to encode the l non-zero AC coefficients into triplets $c_1, c_2, \cdots, c_l$, where $c_j = [(r_j, s_j)/v_j]$. Subsequently, $v_1, v_2, \cdots, v_l$ of these triplets are concatenated into a value bitstream $v = \{v_1||v_2||\cdots||v_l\}$ with length $s = s_1 + s_2 + \cdots + s_l$. Next, we extract $(k-1)$ segments of secret data with length s each and convert them into polynomial coefficients $a_1, a_2, \cdots, a_{k-1}$. Thus, a polynomial with order $(k-1)$ can be obtained as

$$v(x) = (v \oplus a_1 x \oplus a_2 x^2 \oplus \cdots \oplus a_{k-1} x^{k-1}) \bmod p(x), \qquad (5)$$

where $p(x)$ is an irreducible polynomial of order s randomly assigned by the dealer. The polynomial is then evaluated at n distinct identity numbers $x_1, x_2, \cdots, x_n$ over $GF(2^s)$, resulting in n corresponding shadow bitstreams $v(x_1), v(x_2), \cdots, v(x_n)$ with length s each. Following, each shadow bitstream $v(x_i)$ is treated as v and applied back to duplicate a shadow series of triples $c_1'(x_i), c_2'(x_i), \cdots, c_l'(x_i)$. Finally, each shadow series is combined with the encrypted DC coefficient of the block to obtain a complete block code. After processing all blocks in the JPEG image, n shadow images are generated. Finally, each shadow image is paired with its corresponding identity number and distributed to a participant.

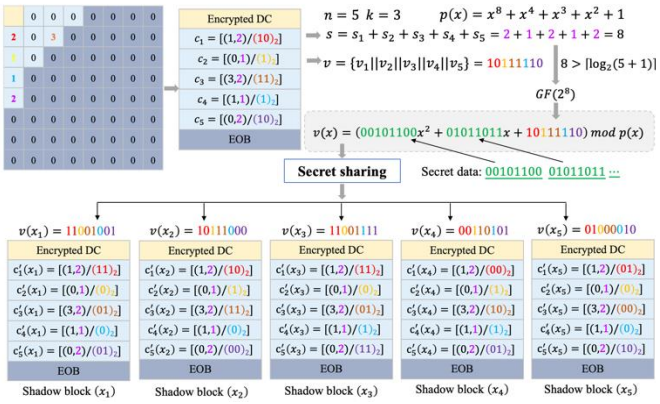


Fig. 6. Illustration of AC coefficients encryption together with secret data embedding.

A demonstrative example is given in Fig. 6. In this example, we assume that $n = 5, k = 3$, and the irreducible polynomials $p(x)$ of different orders are given in advance. To accomplish both AC coefficient encryption and data embedding, we first encode all the five non-zero AC coefficients into triplets $c_1, c_2, \cdots, c_5$, and then concatenate $v_1, v_2, \cdots, v_l$ of the triplets into a bitstream $v = \{10111110\}$. Following this, we extract 2 segments of 8-bit secret data and convert them into polynomial coefficients $a_1 = \{10111110\}$ and $a_2 = \{01011011\}$. Next, we construct a second order polynomial with an irreducible polynomial $p(x) = x^8 + x^4 + x^3 + x^2 + 1$. Afterwards, the constructed polynomial is evaluated at the 5 distinct identity numbers $x_1 = 00000001$, $x_2 = 00000010$, $x_3 = 00000011$, $x_4 = 00000100$ and $x_5 = 00000101$ to obtain the bitstreams $v(x_1) = 11001001$, $v(x_2)$=10111000, $v(x_3)$=11001111, $v(x_4)$=00110101, and $v(x_5)$=01000010. Each of these shadow bitstreams is applied back to duplicate a shadow series of triplets. Finally, each shadow series is combined with the encrypted DC coefficient of the block to obtain a complete block code.

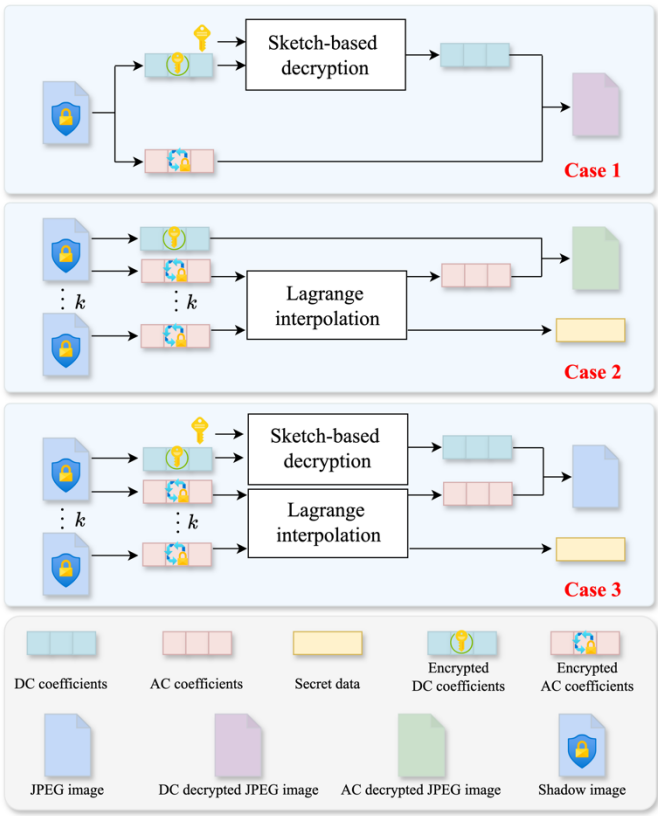


Fig. 7. Workflow of image decryption and data extraction.

B. Image Decryption and Data Extraction

The image decryption and data extraction in our scheme can be categorized into three cases as illustrated in Fig. 7. As shown, the decryption of AC coefficients and the extraction of secret data are joint processes, while the decryption of DC coefficients is conducted separately.

A participant who holds the DC encryption key K_{DC} can decrypt the encrypted DC coefficients with the DC decryption method provided in [30]. Specifically, the permuted region labels are restored, and the DC values are remapped to their original values using the encryption key K_{DC} first. Subsequently, the sketch image is calculated using Eq. (4) based on the AC coefficients. Next, the labeled image is obtained using a process similar to encryption. After that, the DC coefficients within the DC rearranged JPEG image are horizontally extracted to fill each labeled region. Lastly, the decrypted DC coefficient array is combined with the AC coefficients to obtain the original JPEG image. It is important to note that the number of AC coefficients

within each block must be preserved in the following processes; otherwise, the sketch image cannot be restored.

When k or more participants share their identity numbers and the shadow images, but none of whom have the DC encryption key K_{DC}, they can decrypt the encrypted AC coefficients and extract the secret data. In this case, the AC triplets $c_1'(x_i), c_2'(x_i), \cdots, c_l'(x_i)$ for each block are retrieved first. Then, $v_1(x_i), v_2(x_i), \cdots, v_l(x_i)$ are concatenated to generate the shadow bitstream $v(x_i) = \{v_1(x_i)||v_2(x_i)||\cdots||v_l(x_i)\}$ sized $s(x_i) = s_1(x_i) + s_2(x_i) + \cdots + s_l(x_i)$. As a result, k shadow bitstreams are generated for each corresponding image block. Afterwards, the k shadow bitstreams and the k identity numbers are combined to reconstruct the polynomial of order $(k-1)$ with Eq. (2) and obtain the decoded bitstream $v = \{v_1||v_2||\cdots||v_l\}$ with Eq. (3). Following, the secret data can be extracted from the polynomial coefficients $a_1, a_2, \cdots, a_{k-1}$. Last, the bitstream $v = \{v_1||v_2||\cdots||v_l\}$ is divided into $v_1, v_2, \cdots, v_l$ based on $s_1(x_i), s_2(x_i), \cdots, s_l(x_i)$, and then inserted into the corresponding position to reconstruct the AC coefficients of the block.

When k or more participants share their identity numbers and the shadow images, and at least one of them holds the DC encryption key K_{DC}, they can extract the secret data and recover the original JPEG image without loss.

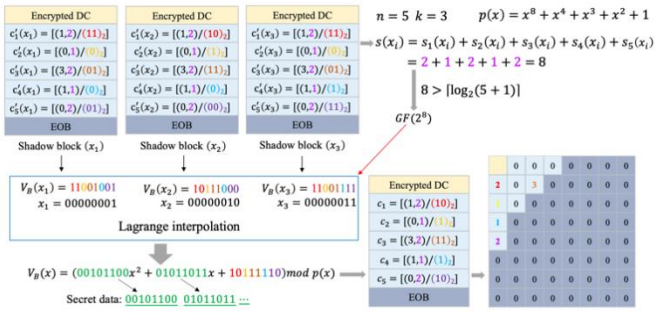


Fig. 8. Illustration of AC coefficient decryption together with secret data extraction.

Following from the example in Fig. 6, its corresponding AC coefficient decryption together with data extraction is illustrated in Fig. 8. Assume again that $n = 5, k = 3$, and we are given the irreducible polynomials $p(x)$ of different orders over Galois field. There are three participants who share their identity numbers and shadow images, but none of them have a DC encryption key K_{DC}. To decrypt the AC coefficients and extract the secret data, the AC triplets of three corresponding shadow blocks are first retrieved. Subsequently, the values within each shadow block are collected to obtain the bitstreams $v(x_1) = 11001001$, $v(x_2) = 10111000$, $v(x_3)$=11001111. These bitstreams and their corresponding identity numbers $x_1 = 00000001, x_2 = 00000010, x_3 = 00000011$ are combined to reconstruct the second order polynomial with Eq. (2). By cooperating with the irreducible polynomial $p(x) = x^8 + x^4 + x^3 + x^2 + 1$, the value bitstream $v = \{10111110\}$ can be solved with Eq. (3). After that, we can extract the secret data from the coefficients $a_1 = \{10111110\}$ and $a_2 = \{01011011\}$. Last, the value bitstream $v = \{10111110\}$ is decomposed and inserted back to reconstruct the original AC coefficients of the block.

C. Technical Considerations

Recall that we process the image block-wise, therefore, the order n for $GF(2^n)$ depends on the length of value bitstream for the non-zero AC coefficients. To ensure the security of identity numbers, we select the identity number for each participant from Galois field $GF(2^8)$, which provides $C_n^{2^8} \times n!$ possible permutations for n identity numbers. In implementation, when the length s of value bitstream for a block satisfies $s \geq 8$, we append $(s-8)$ zeros to the front of the identity numbers. For the cases that $s < 8$, the value bitstreams of two or more blocks are merged to meet the condition.

As a rule of thumb, the code length of an AC value increases with its absolute value in entropy coding. The secret sharing scheme may alter any bit in the bitstream. A coefficient with a greater absolute value may suffer from a severe distortion when its most significant bit is altered. To reduce distortion, we can sort the blocks according to their average length of coefficient values defined by

$$s_{av} = s/l. \tag{6}$$

When the image data payload is greater than the secret data to be embedded, we can process the blocks in the ascending order of average coefficient length. Thus, the blocks with greater coefficient values can be reserved to prevent severe distortion.

IV. Experimental Results

Experiments are conducted to evaluate the performance of the proposed reversible data hiding scheme in encrypted JPEG images. We begin by evaluating the visual effect of the images generated by our scheme. Then, we investigate the performance on data payload and file size preservation. After that, we conduct security analysis. Lastly, the effect of block sorting strategy is discussed. We employ a 512 × 512 grayscale image, 'Flowers,' shown in Fig. 9(a) as an example to demonstrate the performance of the proposed method. For color images, the luma and chroma components can be processed individually in the same manner as grayscale images. Two datasets, "Bossbase [35]" and "BOW-2 [36]", are utilized to demonstrate its generalizability. Each dataset contains 10,000 images, including both smooth and rough textures.

A. Visual Effect

Two metrics are used to evaluate visual similarity between images including Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM), which are defined by

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{255^2 \times W \times H}{\sum_{W \times H}(p_x - p_y)}\right), \tag{7}$$

and

$$\text{SSIM} = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{8}$$

where p_x and p_y are pixel values in the images x and y, respectively; μ_x and μ_y are the mean values, σ_x and σ_y are the standard deviations, and σ_{xy} is the covariance of the images x and y. In addition, C_1 and C_2 are constants added to avoid instability at low luminance or contrast values.

Fig. 9 displays the resulting images together with their visual index values generated by the proposed scheme with $(k, n) = (2,2)$. The JPEG image with quality factor QF=85, shown in Fig. 9(b), is generated from Fig. 9(a). The two shadow images given in Figs. 9(c) and (d) exhibit a highly disrupted appearance, making it challenging to visualize any meaningful information. The AC and DC decrypted images are given in Figs. 9(e) and (f). Some information can be observed from these partially decrypted images. Specifically, the AC decrypted image reveals the high frequency information such as the text inside the marked region, while the DC decrypted image exposes the low frequency information. The recovered images from the two shares are provided in Figs. 9(g) and (h).

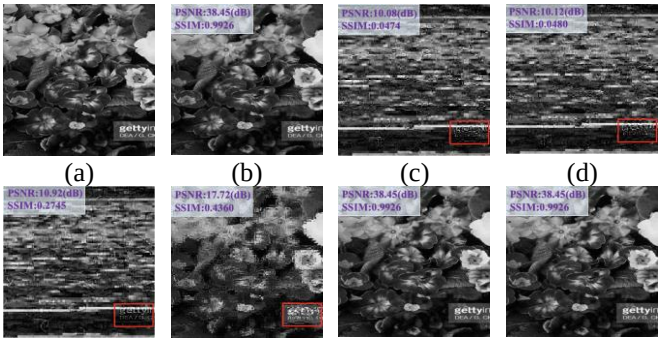


Fig. 9. Results with $(k, n) = (2,2)$: (a) original image; (b) JPEG image with QF=85; (c) share 1; (d) share 2; (e) AC decrypted image; (f) DC decrypted image; (g) recovered JPEG image from (c); (h) recovered JPEG image from (d).

To further evaluate visual performance, we randomly selected 1,000 sample images from both the 'BOSSbase' and 'BOW-2' datasets and assessed the PSNR and SSIM values of the generated shadow shares with QF=85. Figure 10 displays four sets of extreme cases. The first and second rows show the cases with the highest and lowest SSIM values from the 'BOSSbase' dataset, while the third and fourth rows display the extreme cases from the 'BOW-2' dataset. Sample images are shown in the first column, encrypted images in the second column, AC-decrypted images in the third column, and DC-decrypted images in the fourth column. It is evident that the encryption results for complex images are better than those for smooth images. This is due to the DC encryption method used, which scrambles DC coefficients based on connected regions. Nonetheless, our method still prevents visual leakage of image content. Regarding decryption results, DC-decrypted images reveal more information than AC-decrypted images.

B. Data Payload

To investigate data payload of the proposed scheme, the embedding capacity (EC), also known as total data payload, and the bits per non-zero AC coefficient (BPNZ) are utilized, where EC is the total amount of payload in bits and BPNZ is the average payload in bits per non-zero AC coefficient defined by

$$\mathrm{EC} = (k-1) \times \sum_{\alpha} \sum_{\beta} s_{(\alpha,\beta)}, \tag{9}$$

and

$$\mathrm{BPNZ} = \frac{\mathrm{EC}}{\sum_{\alpha} \sum_{\beta} l_{(\alpha,\beta)}}, \tag{10}$$

where $s_{(\alpha,\beta)}$ denotes the length of the value bitstreams for the block indexed (α, β), $l_{(\alpha,\beta)}$ denotes the number of non-zero AC coefficients, and $(k-1)$ is the order of the polynomial. In general, k is determined in advance, while $s_{(\alpha,\beta)}$ and $l_{(\alpha,\beta)}$ depend on the quality factor (QF) in the JPEG compression. Specifically, a higher QF value would result in a greater number of non-zero AC coefficients and a longer bitstream length. Thus, more secret data bits can be embedded.

EC and BPNZ for the "Flowers" and datasets under different settings of QF and k values, with $n = 5$, are listed in Tables I and II. As expected, EC and BPNZ increases with the increasing of QF and k values. Note that as QF increases, the file size of JPEG also increases to record more non-zero coefficients. Also, the increase of k value means additional costs are required to generate and manage more shadow shares.

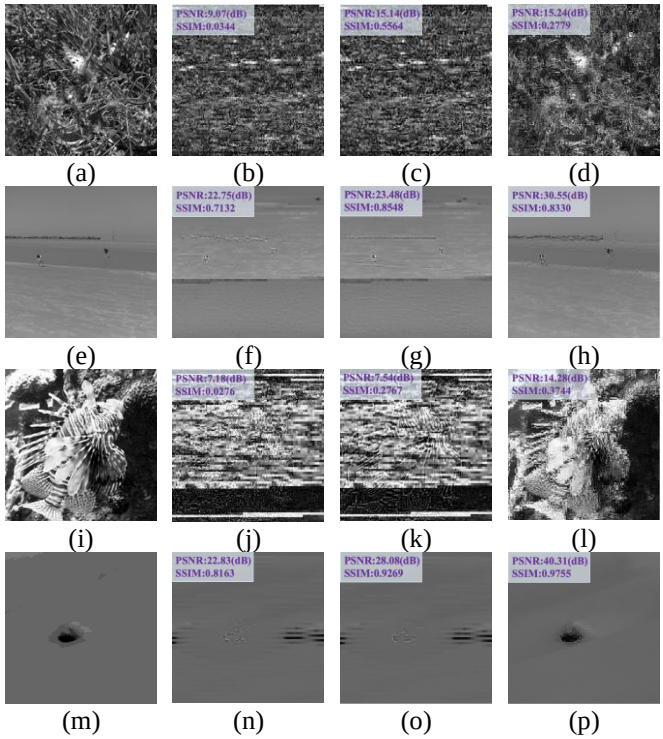


Fig. 10. Four sets of extreme cases with QF=85: the first and second rows show the cases with the highest and lowest SSIM values from the 'BOSSbase' dataset, while the third and fourth rows display the extreme cases from the 'BOW-2' dataset. Sample images are shown in the first column, encrypted images in the second column, AC-decrypted images in the third column, and DC-decrypted images in the fourth column.

TABLE I
EC AND BPNZ FOR IMAGE "FLOWERS"

		QF=50	QF=60	QF=70	QF=80	QF=90
EC	$k=2$	67806	79801	98028	125604	187395
	$k=3$	135612	159602	196056	251208	374790
	$k=4$	203418	239403	294084	376812	562185
BPNZ	$k=2$	1.3959	1.4872	1.6368	1.8054	2.7303
	$k=3$	2.7918	2.9744	3.2736	3.6108	5.4606
	$k=4$	4.1877	4.4616	4.9104	5.4162	8.1909

TABLE II
AVERAGE EC AND BPNZ FOR 1,000 IMAGES RANDOMLY SELECTED FROM TWO DATASETS

			QF=50	QF=60	QF=70	QF=80	QF=90
BOSS base	EC	$k=2$	39139	46921	59207	78304	122645
		$k=3$	78278	93842	118414	156608	225290
		$k=4$	117417	140763	177621	234912	337935
	BPNZ	$k=2$	1.5364	1.5899	1.6681	1.7575	1.9088
		$k=3$	3.0728	3.1798	3.3362	3.5150	3.8176
		$k=4$	4.6092	4.7697	5.0043	5.2725	5.7264
BOW-2	EC	$k=2$	49792	59174	73758	96396	148228
		$k=3$	99584	118348	147516	192792	296456
		$k=4$	149376	177522	221274	289188	444684
	BPNZ	$k=2$	1.5754	1.6339	1.7154	1.8101	1.9747
		$k=3$	3.1508	3.2678	3.4308	3.6202	3.9494
		$k=4$	4.7262	4.9017	5.1462	5.4303	5.9241

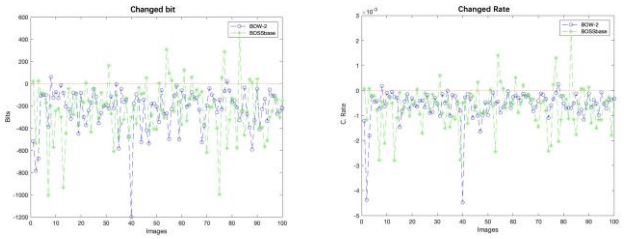
C. File Size Preservation

The objective of the proposed scheme is to increase data payload capacity while preserving the file size and format. Recall that the proposed AC secret sharing scheme modifies the value code of AC triplets. Such operation preserves the overall code length of AC coefficients. On the other hand, the region-based DC coefficient rearrangement makes close values consecutive. As a result, the differential DC values are smaller, allowing them to be encoded with shorter Huffman codewords. This often leads to a reduction in code length.

Table III lists the file size change in bits and file size change rate for image "Flowers" under different QFs. For most cases, the file size is slightly reduced due to DC encryption. At QF=90, the image is segmented into more regions and rearrangement does not result in the expected file size reduction effect. The results for 100 randomly selected images from the datasets "BOSSbase" and "BOW-2" are plotted in Fig.11. It is obvious that the proposed scheme reduces the file size for most cases.

TABLE III
THE FILE SIZE CHANGE FOR IMAGE "FLOWERS"

		QF=50	QF=60	QF=70	QF=80	QF=90
No payload	Bits	-92	-168	-125	-139	896
	C. Rate	-0.03%	-0.06%	-0.03%	-0.03%	0.1%
Half payload	Bits	-92	-168	-125	-139	896
	C. Rate	-0.03%	-0.06%	-0.03%	-0.03%	0.1%
Full payload	Bits	-92	-168	-125	-139	896
	C. Rate	-0.03%	-0.06%	-0.03%	-0.03%	0.1%



(a) (b)

Fig. 11. File size change for 100 sample images from two datasets with QF=85 and full payload.

D. Security Analysis

In our scheme, the DC coefficients are encrypted using a sketch-based method [30], which was developed based on the sketch attack approach. As a result, this method is inherently resistant to such attacks. The AC coefficients and secret data are secured using Shamir's secret sharing scheme. As discussed in [37], Shamir's secret sharing scheme provides information-theoretic security, ensuring that an attacker cannot recover the AC coefficients or extract the secret data without possessing the required number of shadow shares.

To assess the sensitivity of our encryption method to input changes, we conducted differential cryptanalysis using two metrics: number of pixels change rate (NPCR) and unified average changing intensity (UACI) [38]. We evaluated these metrics by altering a single bit of the DC encryption key or the identity number during the encryption of test image "Flowers." The values obtained for NPCR (Key), NPCR (Identity), UACI (Key), and UACI (Identity) are 99.78%, 99.66%, 30.03%, and 29.89%, respectively. The results demonstrate that even minor changes in the input produce high NPCR and UACI values, indicating significant differences between the encrypted images.

Shannon entropy analysis is used to evaluate the randomness of the encrypted images [39]. For the test image "Flowers," the entropy values for the original image, AC decrypted image, DC decrypted image, and marked image are 7.5578, 7.6359, 7.7682, and 7.7821, respectively. The results indicate that the proposed encryption method introduces a high level of randomness, with the entropy value of the marked image approaching the maximum of 8. Additionally, the AC and DC decrypted images exhibit varying levels of entropy reduction compared to the marked image.

Table IV presents the average NPCR (Key), UACI (Key), NPCR (Identity), UACI (Identity), and Shannon entropy values for the shadow images across 1,000 sample images from the two datasets, highlighting the overall security of our method. The results confirm that the method is highly sensitive to input changes, offering resistance to differential cryptanalysis. Additionally, the entropy analysis demonstrates that our encryption method introduces significant randomness, ensuring that the encrypted data remains unpredictable.

TABLE IV
SECURITY MEASURES FOR 1,000 IMAGES

Dataset	Entropy	NPCR (Key)	UACI (Key)	NPCR (Identity)	UACI (Identity)
BOSSbase	7.7863	99.49%	28.13%	99.44%	28.56%
BOW-2	7.7784	99.57%	27.74%	99.72%	28.65%

E. Block Sorting Strategy

In Section 3.E, we propose a block sorting strategy to reduce the image distortion under low payload. To verify this point, we sort the image blocks in two opposite ways and apply different payloads to both block sequences. The resulting DC decrypted images are displayed in Fig. 12, where similarity measures with respect to the original image are also marked. The image quality for each case reflects the image distortion caused by AC encryption since DC value has been decrypted. As theoretic expectation, the processing with ascending order of s_{av} produces a lower distortion image than its counterpart for all cases significantly.

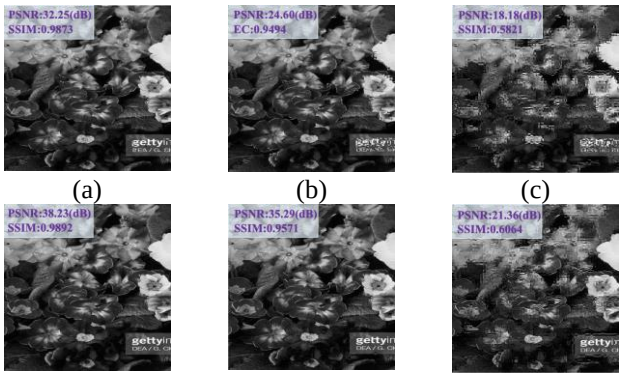


Fig. 12. DC decrypted images with different block sorting strategies. (a) - (c) Processed in descending order of s_{av} , (d) - (f) processed in ascending order of s_{av}. The payload for (a) and (d) are 1,000 bits, (b) and (e) are 10,000 bits, (c) and (f) are 100,000 bits.

F. Comparisons

We compare the proposed scheme with six state-of-the-art works [23], [24], [25], [26], [27], and [28] in terms of data payload, file size preservation, format compatibility, and security. In our comparison, two standard gray-scale images "Couple" and "Baboon" are used to provide comparative results.

Table V lists the comparison of data payload under different QF. To get the data, the number of shadow shares k in our scheme is set to 2. The total data payload of our scheme greatly outperforms the other methods. In addition, by selecting a larger value of k, we can even achieve a higher EC and BNPZ. Note that when more shadow shares are applied, the available space for embedding secret data is greatly increased since there is still only one cover image to be restored. Our approach successfully leverages this feature.

The comparison of the file size preservation under different EC are listed in Table VI. In the table, the results are obtained with QF=80 and the symbol "-" indicates data unavailable due to limitation in EC. The proposed method exhibits excellent performance in preserving file size compared to the others. It achieves the second-highest level of file size preservation when EC is set at 250 and 500 bits but outperforms all others when EC is set at 1,000, 1,500, and 2,500 bits.

For format compatibility, the stream cipher encryption is applied to encrypt the DC coefficients in [23], [24], [25], which may cause an overflow or underflow to the quantized DC coefficients [29]. In the proposed scheme, the encryption and data embedding processes do not alter the data structure of the JPEG format. As a result, the image generated by our scheme is compatible with existing JPEG decoders.

Table VII presents the NPCR, UACI, and Shannon entropy results for the encrypted versions of the 'Baboon' and 'Couple' images obtained using various methods. The six compared methods do not use secret sharing for image encryption, making "NPCR (Identity)" and "UACI (Identity)" metrics inapplicable to them. The results indicate that the security performance of our method is comparable to that of other methods, as evidenced by similar entropy, NPCR, and UACI values.

TABLE V
COMPARISON OF DATA PAYLOAD UNDER DIFFERENT QF

Quality Factor		QF=50		QF=60		QF=70		QF=80		QF=90	
Indicator		EC	BPNZ	EC	BPNZ	EC	BPNZ	EC	BPNZ	EC	BPNZ
Baboon	[23]	775	0.012	775	0.010	775	0.009	775	0.007	775	0.005
	[24]	1238	0.019	1277	0.017	1435	0.016	1723	0.016	2044	0.013
	[25]	700	0.011	705	0.009	705	0.008	705	0.006	695	0.005
	[26]	14719	0.224	15948	0.213	17796	0.201	21059	0.192	27630	0.181
	[27]	3775	0.058	4472	0.031	5311	0.060	6344	0.058	6520	0.043
	[28]	3022	0.100	7433	0.067	6148	0.100	11315	0.103	13477	0.088
	Ours	**103144**	**1.6518**	**122540**	**1.7184**	**152181**	**1.8090**	**198584**	**1.9122**	**302133**	**2.1087**
Couple	[23]	757	0.023	765	0.020	772	0.017	775	0.013	775	0.009
	[24]	582	0.018	718	0.019	680	0.015	680	0.012	690	0.008
	[25]	720	0.022	705	0.018	770	0.020	750	0.015	745	0.010
	[26]	8913	0.270	10074	0.262	11692	0.252	13917	0.238	18275	0.214
	[27]	1184	0.036	1601	0.042	2347	0.051	3107	0.053	4772	0.056
	[28]	1931	0.057	2740	0.071	3950	0.085	5737	0.099	9327	0.109
	Ours	**58163**	**1.6529**	**69208**	**1.7048**	**86298**	**1.7759**	**113116**	**1.8671**	**175196**	**2.0030**

TABLE VI
COMPARISON OF FILE SIZE CHANGE UNDER DIFFERENT EC

Quality Factor		EC = 250		EC = 500		EC = 1000		EC = 1500		EC = 2500	
Indicator		Bits	Ratio	Bits	Ratio	Bits	Ratio	Bits	Ratio	Bits	Ratio
Baboon	[23]	160	0.02%	160	0.02%	-	-	-	-	-	-
	[24]	392	0.06%	392	0.06%	1120	0.17%	1360	0.21%	1808	0.28%
	[25]	**-1224**	**-0.19**%	**-1224**	**-0.19**%	-	-	-	-	-	-
	[26]	2616	0.41%	2704	0.42%	2856	0.44%	3168	0.49%	3216	0.50%
	[27]	360	0.06%	360	0.06%	352	0.05%	384	0.06%	392	0.06%
	[28]	336	0.05%	336	0.05%	336	0.05%	368	0.06%	1808	0.08%
	Ours	-288	-0.04%	-288	-0.04%	**-288**	**-0.04%**	**-288**	**-0.04%**	**-288**	**-0.04%**
Couple	[23]	32	0.01%	80	0.07%	-	-	-	-	-	-
	[24]	160	0.04%	160	0.04%	160	0.04%	-	-	-	-
	[25]	**-2056**	**-0.56%**	**-2056**	**-0.56%**	-	-	-	-	-	-
	[26]	3312	0.90%	3240	0.88%	3496	0.95%	3752	1.02%	4184	1.13%
	[27]	208	0.06%	256	0.07%	248	0.07%	248	0.07%	240	0.07%
	[28]	240	0.07%	248	0.07%	248	0.08%	264	0.07%	296	0.08%
	Ours	-242	-0.07%	-242	-0.07%	**-242**	**-0.07%**	**-242**	**-0.07%**	**-242**	**-0.07%**

TABLE VII
COMPARISONS OF ENTROPY, NPCR AND UACI

Image	Method	Entropy	NPCR (Key)	UACI (Key)	NPCR (Identity)	UACI (Identity)
Baboon	[23]	7.7726	99.54%	25.08%	-	-
	[24]	7.8459	99.07%	26.57%	-	-
	[25]	7.6913	99.23%	26.76%	-	-
	[26]	7.7016	99.53%	25.77%	-	-
	[27]	7.7764	99.46%	30.24%	-	-
	[28]	7.7013	99.48%	27.03%	-	-
	Ours	7.7393	99.55%	27.34%	99.45%	30.24%
Couple	[23]	7.6241	99.73%	27.54%	-	-
	[24]	7.7362	99.64%	27.47%	-	-
	[25]	7.6527	99.83%	26.98%	-	-
	[26]	7.7728	99.57%	26.62%	-	-
	[27]	7.7597	98.46%	26.85%	-	-
	[28]	7.7384	99.71%	27.38%	-	-
	Ours	7.7802	99.51%	28.87%	99.67%	27.75%

G. Limitations

The proposed method utilizes Shamir's secret sharing scheme for encrypting AC coefficients and secret data. This scheme divides the data into n shares, which are distributed among participants. The original data can be reconstructed by combining any k or more of these shares. However, as n increases, managing and processing a larger number of shares results in higher computational overhead. Similarly, a larger k requires more shares for reconstruction, which complicates the encryption and decryption processes. Therefore, when selecting values for n and k, it is essential to consider these factors to balance security and computational efficiency.

V. CONCLUSIONS

We propose an RDH-EI scheme for JPEG images over Galois field. In our scheme, the bitstreams of AC coefficients and the secret data is first mapped to the numbers over Galois field. Then, these numbers are exploited to construct the polynomial for secret sharing. Through secret sharing, the bitstreams of AC coefficients and secret data are securely distributed into shares. Only when a sufficient number of participants share their shadows and identity numbers, they can recover the JPEG code and extract the secret data without loss.

Experimental results demonstrate that the proposed scheme provides an excellent data payload while preserving the file size. At QF=90, the "Baboon" and "Couple" images achieve EC values of 302,133 and 175,196 bits, with BPNZ values of 2.1087 and 2.0030, respectively. The file sizes of the two images are reduced by 288 bits and 242 bits, respectively. Importantly, the proposed scheme is format-compatible, ensuring that the encrypted image can be decoded using standard JPEG decoders. Additionally, it achieves security performance comparable to existing methods. Overall, our method shows significant improvements over current approaches. In future work, we aim to explore the potential of combining our method with blockchain technology to further enhance security.

REFERENCES

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

[2] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890-896, Aug. 2003.

[3] I. Dragoi and D. Coltuc, "Local-prediction-based difference expansion reversible watermarking," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1779-1790, 2014.

[4] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354-362, March 2006.

[5] X. Gao, L. An, Y. Yuan, D. Tao, and X. Li, "Lossless data embedding using generalized statistical quantity histogram," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 8, pp. 1061-1070, Aug. 2011.

[6] X. Li, B. Li, B. Yang, and T. Zeng, "General framework to histogram-shifting-based reversible data hiding," *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2181-2191, 2013.

[7] J. Wang, J. Ni, X. Zhang, and Y. Q. Shi, "Rate and Distortion Optimization for Reversible Data Hiding Using Multiple Histogram Shifting," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 315-326, Feb. 2017.

[8] D. M. Thodi and J. J. Rodriguez, "Prediction-error based reversible watermarking," in *Proc. IEEE Int. Conf. Inform. Process.*, Singapore, pp. 1549-1552, Oct. 2004.

[9] D. M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 721- 730, Mar. 2007.

[10] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, "Reversible data hiding in encrypted images by reserving room before encryption," *IEEE Trans. Inf. Forensics Secur.*, vol. 8, No. 3, pp. 553-562, 2013.

[11] P. Puteaux and W. Puech, "An efficient msb prediction-based method for high-capacity reversible data hiding in encrypted images," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, No. 7, pp. 1670- 1681, 2018.

[12] Z. Yin, Y. Xiang, and X. Zhang, "Reversible data hiding in encrypted images based on multi-msb prediction and huffman coding," *IEEE Trans. Multimedia*, vol. 22, No. 4, pp. 874-884, 2019.

[13] S. Xu, J. H. Horng, C.-C.Chang, and C.-C.Chang, "Reversible data hiding with hierarchical block variable length coding for cloud security," *IEEE Trans. Dependable Secure Comput.*, vol. 20, No. 5, pp. 4199-4213, 2022.

[14] X. Zhang, "Separable and error-free reversible data hiding in encrypted images," *Signal Process.*, vol. 123, pp. 9-21, 2016.

[15] F. Huang, J.Huang, and Y.-Q. Shi, "New framework for reversible data hiding in encrypted domain," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 12, pp. 27772789, 2016.

[16] X. Zhang, "Reversible data hiding in encrypted image," *IEEE Signal Process. Lett.*, vol. 18, no. 4, pp. 255-258, 2011.

[17] Z. Qian, X. Zhang, and G. Feng, "Reversible data hiding in encrypted images based on progressive recovery," *IEEE Signal Process. Lett.*, vol. 23, no. 11, pp. 1672-1676, 2016.

[18] C. Qin, W. Zhang, F. Cao, X. Zhang, and C.-C. Chang, "Separable eversible data hiding in encrypted images via adaptive embedding strategy with block selection," *Signal Process.*, vol. 153, pp. 109-122, 2018.

[19] K. Chen, Q. Guan, W. Zhang, and N. Yu, "Reversible data hiding in encrypted images based on binary symmetric channel model and polar code," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 6, pp. 4519-4535, 2023.

[20] C. Qin, C. Jiang, Q. Mo, H. Yao, and C. C. Chang, "Reversible data hiding in encrypted image via secret sharing based on GF(p) and GF (2^8)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 1928-1941, Jun. 2021.

[21] Z. Hua, Y. Wang, S. Yi, Y. Zheng, X. Liu, Y. Chen, and X. Zhang, "Matrix-based secret sharing for reversible data hiding in encrypted images," *IEEE Trans. Dependable Secure Comput*., vol. 20, no. 5, pp. 3669-3686, 2022.

[22] C. Yu, X. Zhang, C. Qin, and Z. Tang, "Reversible data hiding in encrypted images with secret sharing and hybrid coding," *IEEE Trans. Circuits Syst. Video Technol*., vol. 33, no. 11, pp. 6443-6458, 2023.

[23] Z. Qian, X. Zhang, and S. Wang, "Reversible data hiding in encrypted JPEG bitstream," *IEEE Trans. Multimedia*, vol. 16, no. 5, pp. 1486–1491, Aug. 2014.

[24] J.-C. Chang, Y.-Z. Lu, and H.-L. Wu, "A separable reversible data hiding scheme for encrypted JPEG bitstreams," *Signal Process.*, vol. 133, pp. 135–143, Apr. 2017.

[25] Z. Qian, H. Zhou, X. Zhang, and W. Zhang, "Separable reversible data hiding in encrypted JPEG bitstreams," *IEEE Trans. Dependable Secure Comput*., vol. 15, no. 6, pp. 1055–1067, Nov./Dec. 2018.

[26] Z. Qian, H. Xu, X. Luo, and X. Zhang, "New framework of reversible data hiding in encrypted JPEG bitstreams," *IEEE Trans. Circuits Syst. Video Technol*., vol. 29, no. 2, pp. 351-362, February 2019.

[27] J. He, J. Chen, W. Luo, S. Tang, and J. Huang, "A novel high-capacity reversible data hiding scheme for encrypted JPEG bitstreams," *IEEE Trans. Circuits Syst. Video Technol*., vol. 29, no. 12, pp. 3501-3515, 2018.

[28] Z. Hua, Z. Wang, Y. Zheng, Y. Chen, and Y. Li, "Enabling Large-Capacity Reversible Data Hiding over Encrypted JPEG Bitstreams," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 3, pp.1013-1018, 2023.

[29] J. He, S. Huang, S. Tang, and J. Huang, "JPEG image encryption with improved format compatibility and file size preservation," *IEEE Trans. Multimedia*, vol. 20, no. 10, pp. 2645-2658, 2018.

[30] K. Minemura, K. Wong, X. Qi, and K. Tanaka, "A scrambling framework for block transform compressed image," *Multimedia Tools Appl.*, vol. 76, pp. 6709-6729, 2017.

[31] G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM.*, vol. 34, no. 4, pp. 30–44, Apr. 1991.

[32] A. Shamir, "How to share a secret," *Commun. ACM.*, vol. 22, no. 11, pp. 612-613, 1979.

[33] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst. Man Cybern*., vol. 9, pp. 62–66, 1979.

[34] R. M. Haralick and L. G. Shapiro, "Computer and robot vision," 1st ed, vol. 1, pp. 28-48, Addison Wesley Longman Publ. Co. Inc., Boston, 1992.

[35] P. Bas, T. Filler, and T. Pevný, "Break our steganographic system—The ins and outs of organizing BOSS," in International Workshop on Information Hiding, Berlin, Heidelberg. [Online]. Available: http://dde.binghamton.edu/download/.

[36] P. Bas and T. Furon, "Image Database of BOWS-2." Accessed: Jun. 22, 2019. [Online]. Available: http://bows2.ec-lille.fr/".

[37] K. V. Rashmi, N. B. Shah, K. Ramchandran and P. V. Kumar, "Information-theoretically secure erasure codes for distributed storage," IEEE Trans. Inf. Theory, vol. 64, no. 3, pp. 1621-1646, Mar. 2018.

[38] L. Bao, S. Yi and Y. Zhou, "Combination of sharing matrix and image encryption for lossless (k, n)-secret image sharing," IEEE Trans. Image Process., vol. 26, no. 12, pp. 5618-5631, Dec. 2017.

[39] J. Lin, "Divergence measures based on the Shannon entropy," IEEE Trans. Inf. Theory, vol. 37, no. 1, pp. 145-151, Jan. 1991.

Shuying Xu received a B.S. degree in Software Engineering from Fujian Normal University, Fuzhou, China, in 2019. She is currently pursuing a Ph.D. degree with the Department of Information Engineering and Computer Science, Feng Chia University. Her current research interests include steganography, watermarking, biometrics, information security, image processing, and computer vision.

Ji-Hwei Horng received a B.S. degree from the Department of Electronic Engineering, Tamkang University, Taipei, Taiwan in 1990 and M.S. and Ph.D. degrees from the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan in 1992 and 1996, respectively. He was a professor and Chairman of the Department of Electronic Engineering from 2006 to 2009 and the Dean of the College of Science and Engineering from 2011 to 2014 at National Quemoy University (NQU), Kinmen, Taiwan. Currently, he is a distinguished professor and the Vice President of Academic Affairs in NQU. His research interests include image processing, pattern recognition, information security, and artificial intelligence.

Ching-Chun Chang received his PhD in Computer Science from the University of Warwick, UK, in 2019. He engaged in a short-term scientific mission supported by European Cooperation in Science and Technology Actions at the Faculty of Computer Science, Otto-von-Guericke-Universita ̈t Magdeburg, Germany, in 2016. He was granted the Marie-Curie fellowship and participated in a research and innovation staff exchange scheme supported by Marie Skłodowska-Curie Actions at the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, USA, in 2017. He was a Visiting Scholar with the School of Computer and Mathematics, Charles Sturt University, Australia, in 2018, and with the School of Information Technology, Deakin University, Australia, in 2019. He was a Research Fellow with the Department of Electronic Engineering, Tsinghua University, China, in 2020. His research interests include steganography, watermarking, forensics, biometrics, cyber- security, applied cryptography, image processing, computer vision, natural language processing, computational linguistics, machine learning and artificial intelligence.

Chin-Chen Chang has worked on many different topics in information security, cryptography, multimedia image processing and published several hundreds of papers in international conferences and journals and over 30 books. He was cited over 42,365 times and has an h-factor of 94 according to Google Scholar. Several well-known concepts and algorithms were adopted in textbooks. He also worked with the National Science Council, Ministry of Technology, Ministry of Education, Ministry of Transportation, Ministry of Economic Affairs and other Government agencies on more than 100 projects and holds 23 patents. He served as Honorary Professor, Consulting Professor, Distinguished Professor, Guest Professor at over 50 academic institutions and received Distinguished Alumni Award's from his Alma Master's. He also served as Editor or Chair of several international journals and conferences and had given almost a thousand invited talks at institutions including Chinese Academy of Sciences, Academia Sinica, Tokyo University, Kyoto University, National University of Singapore, Nanyang Technological University, The University of Hong Kong, National Taiwan University and Peking University. Professor Chang has mentored 7 postdoctoral, 66 PhD students and 200 master students, most of whom hold academic positions at major national or international universities. He has been the Editor-in-Chief of Information Education, a magazine that aims at providing educational materials for middle-school teachers in computer science. He is a leader in the field of information security of Taiwan. He founded the Chinese Cryptography and Information Security Association, accelerating information security the application and development and consulting on the government policy. He is also the recipient of several awards, including the Top Citation Award from Pattern Recognition Letters, Outstanding Scholar Award from Journal of Systems and Software, and Ten Outstanding Young Men Award of Taiwan. He was elected as a Fellow of IEEE in 1998, a Fellow of IET in 2000, a Fellow of CS in 2020, an AAIA Fellow in 2021.