# A Crypto-attacking Scheme for a (*t*, *n*)-threshold Image Encryption Method

*Chin-Chen Chang[1*], Shuying Xu[1], Jui-Chuan Liu[1], Ching-Chun Chang[2]*

*[1] Department of Information Engineering and Computer Science, Feng Chia University, Taiwan*
*[2] Information and Communication Security Research Center, Feng-Chia University, Taiwan*
*alan3c@gmail.com, shuying.xu.phd@gmail.com, p1200318@o365.fcu.edu.tw, ccc@fcu.edu.tw*

## Abstract

Recently, (*t*, *n*)-threshold secret sharing technology has found applications in reversible data hiding in encrypted images (RDHEI) for image encryption. It offers advantages such as independence from key management systems and resilience against $n - t$ points of failure. In [15], Hua et al. introduced the matrix secret sharing (MSS) method, which utilizes matrix multiplication for secret sharing. Building upon this foundation, MSS has been further applied to image encryption for RDHEI methods. In our study, we conduct a comprehensive analysis of the properties of the MSS-based image encryption scheme. Subsequently, we propose an innovative cryptanalysis approach based on vector quantization (VQ) attacks. Specifically, we introduce the concept of an element variance sequence to capture the changing patterns of the ciphertext image block-wise. We then employ VQ technology to estimate the plaintext content based on the captured changing patterns. Our experimental results provide evidence for the effectiveness of our proposed cryptanalysis approach. Based on this study, we conclude that the MSS scheme raises security concerns.

**Keywords:** Cryptanalysis, Image encryption, Matrix secret sharing, Vector quantization

## 1 Introduction

With the development of 5G technology, various forms of digital media, such as images, videos, and text, are being increasingly stored in cloud environments. Through cloud servers, users can efficiently store, manage, and access their data from anywhere with an internet connection. However, the growing reliance on cloud servers has also raised concerns regarding data security and privacy [1]. To safeguard data privacy in the cloud, a combination of cryptography and reversible data hiding (RDH) technology, known as the RDH-EI method, has emerged as one of an effective measure [2]. This approach not only protects the digital media content but also provides room for additional data.

In recent times, many RDH-EI schemes have been developed [3-14], which can be divided into two categories, namely, reserving room before encryption (RRBE) [3-7] and vacating room after encryption (VRAE) [8-14]. The RRBE-based methods free up room for data embedding before encryption, while VRAE-based methods vacate room after encryption. In most RRBE-based methods, the image owners fully utilize the correlations between neighboring pixels, leading to a relatively large data payload. Nevertheless, within typical user scenarios, image owners are often average users who may not possess the capability to handle such complex processes [8]. Therefore, many researchers primarily focus on the development of RRAE-based methods.

In the RRAE-based RDH-EI method, there are two common encryption algorithms: one is stream cipher [9-11] and the other one is block permutation and co-modulation (BPCM) [8, 12-14]. The stream cipher algorithm begins by generating a binary random keystream. It then converts the pixels in the image into 8-bit binary values. These values are subsequently subjected to an exclusive-or operation with a generated binary random keystream. The resulting bits are finally transformed into decimal values to yield the encrypted pixel values. In contrast, BPCM has the advantage of preserving inter-pixel correlations within the blocks to effectively increase the data payload of RDH-EI. The BPCM method first divides the digital image into non-overlapping blocks. These image blocks are then rearranged using a permutation method. After the block permutation, random integers are utilized to modulate all pixels within the same block. However, Hua et al. [15] have highlighted vulnerabilities in these RDH-EI schemes, specifically their reliance on a dependable key management system and susceptibility to a single point failure. To enhance security, recent schemes encrypt a digital image with secret sharing techniques [16-19]. These methods encrypt the digital image into multiple encrypted images. Only when a pre-set number of encrypted images are collected can the receiver recover the original digital image.

In 2023, Hua et al. introduced an efficient encryption scheme known as the matrix-based secret sharing (MSS) method [15]. Expanding on this approach, they proposed a VRAE-based RDH-EI scheme where the MSS method is employed to encrypt digital images. In this encryption process, the digital image is first partitioned into non-overlapping blocks. Subsequently, a $n \times t$ matrix is constructed for each image block. The pixels within

*Corresponding Author: Chin-Chen Chang; Email: alan3c@gmail.com
DOI: https://doi.org/10.70003/160792642025072604003

each block are then encrypted with a corresponding matrix. Consequently, the digital image is transformed into $n$ image shares. Only when $t$ such image shares are collected can the receiver reconstruct the original digital image. However, this approach carries inherent risks. In general, attackers cannot extract information about the plaintext image from the corresponding cyphertext image. The MSS-based image encryption method retains pixel correlations within the plaintext image in the shared ciphertext images, which could potentially be exploited to extract plaintext content. An attacker may attempt to reverse engineer or exploit discernible patterns in the cyphertext images to extract the plaintext image content based on these correlations.

In this paper, we propose a cryptanalysis for MSS-based image encryption via vector quantization (VQ) attack. With the proposed method, we can derive coarse-grained information about a plaintext image from a single image share. The contributions of this paper are detailed below:

(1) Our proposed scheme allows us to estimate coarse-grained information from ciphertext images regarding the plaintext image.

(2) In contrast to the established cryptanalysis algorithms, the cryptanalysis method presented in this paper does not depend on the availability of the plaintext image.

The paper is structured as follows: Section 2 introduces MSS and the RDHEI scheme on which our scheme is based. In Section 3, we delve into VQ technology. Detailed cryptanalysis using the VQ attack is presented in Section 4. Section 5 provides experimental results and analysis. Finally, Section 6 concludes the paper.

# 2 Related Works

In this section, we provide a concise introduction to the ($t$, $n$)-threshold MSS [19] method and then proceed to introduce an RDH-EI scheme based on the MSS approach.

## 2.1 Matrix-Based Secret Sharing

**Theorem 1.** For a $t \times t$ square matrix $M$ and a prime number $\gamma$, the matrix multiplication

$$\alpha = M \times \beta \bmod \gamma \tag{1}$$

is reversible. The corresponding inverse operation is

$$\beta = M^{-1} \times \alpha \bmod \gamma, \tag{2}$$

when $det(M)$ is coprime with the prime number $\gamma$.

By leveraging this theorem, Hua et al. proposed a ($t$, $n$)-threshold MSS method [19]. In their method, an $n \times t$ matrix $\mathcal{X}$ is first constructed, where any $t \times t$ sub-matrix within $\mathcal{X}$ is coprime with the prime number $\gamma$. The construction process of an $n \times t$ matrix is as follows:

Step 1：Generate $2n$ positive integers over Galois field $GF(\gamma)$ randomly, denoted as $\mu_1, \mu_2, \ldots, \mu_n$, and $v_1, v_2, \ldots, v_n$.

Step 2: Initialize an $n \times t$ matrix $\mathcal{X} \in \mathbb{R}^{n \times t}$.

Step 3: Set the 1-st column of matrix $\mathcal{X}$ with $\mu_1, \mu_2, \ldots, \mu_n$.

Step 4: Set the 2-nd to the $t$-th columns of matrix $\mathcal{X}$ with $\mathcal{X}(i, j) = \mathcal{X}(i, j-1) \times (v_i + j - 2)$, where $1 \le i \le n$ and $2 \le j \le t$.

With the pre-established matrix $\mathcal{X}$ and the prime number $\gamma$, the shared data $s$ can be secured using ($t$, $n$)-threshold MSS method. The secret sharing is conducted through matrix multiplication, which can be formulated as

$$\alpha = \mathcal{X} \times \beta \bmod \gamma, \tag{3}$$

where $\beta = [s, \beta_1, \beta_2, \ldots, \beta_{t-1}]^{-1}$ and $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n]^{-1}$. Note that $\beta_1, \beta_2, \ldots, \beta_{t-1}$ are random positive integers over Galois field $GF(\gamma)$. After the matrix multiplication, the $n$ pairs of identity number and share value $(i, \alpha_i)$ are dealt to $n$ participants, where $i$ is the identity number and $\alpha_i$ is the shared value.

When any $t$ participants gather their pairs, the shared data $s$ can be revealed. During this procedure, the $n \times t$ matrix $\mathcal{X}$ is first reconstructed. Subsequently, the $t$ rows in matrix $\mathcal{X}$ that correspond to the collected identity numbers are extracted to form a $t \times t$ square matrix $\mathcal{X}_t$. Afterwards, the secret data can be recovered by

$$\beta = \mathcal{X}_t^{-1} \times \hat{\alpha} \bmod \gamma, \tag{4}$$

where $\hat{\alpha} = [\widehat{\alpha_1}, \widehat{\alpha_2}, \ldots, \widehat{\alpha_t}]^{-1}$.

## 2.2 MSS-based RDHEI

Based on the ($t$, $n$)-threshold MSS method introduced above, an MSS-based RDHEI scheme is commenced. In the scheme, the content owner first encrypts the original image into $n$ share images using the ($t$, $n$)-threshold MSS method. These $n$ share images are then distributed among $n$ data hiders. After that, additional data are embedded into each shared image independently to obtain the corresponding marked image by the data hider. Only when $t$ or more marked images have been collected, the receiver can recover the original image and extract the secret data. In the following, we explain the details of MSS-based image encryption procedure.

In the scheme, the image encryption is performed block-wise. Initially, the original image $I$ with dimensions $W \times H$ is divided into blocks of size $b_1 \times b_2$. For each image block, an $n \times t$ matrix $\mathcal{X}$ is generated, with $\mu_1, \mu_2, \ldots, \mu_n$ are set to 1. Subsequently, the pixels within the block are considered as secret data and encrypted with Eq. (3) with $\gamma$ set to 257. As a result, the image block is transformed into $n$ share image blocks. After processing all the image blocks, $n$ encrypted images are generated and distributed among $n$ participants.

To illustrate the MSS-based image encryption approach more clearly, an example is provided in Figure 1. In this example, $n = 4$ and $t = 3$. To encrypt the $2 \times 2$ original image block, a $4 \times 3$ matrix is first constructed. $\mu_1, \mu_2, \ldots, \mu_4$ are set to 1 and $v_1, v_2, \ldots, v_4$ are set to 37, 121, 65,

210 respectively. We then encrypt the pixels within the image block using Eq. (3), where $\gamma$ is set to 257 and $\beta_1$, $\beta_2$ are set to 113 and 21, respectively. Consequently, four shared image blocks are produced with pixel values that are entirely distinct from those in the original image block. These shared image blocks are then distributed among four participants.

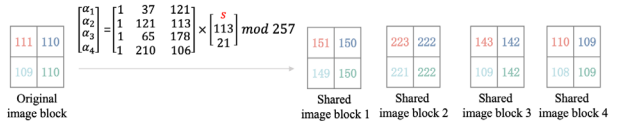



**Figure 1.** An example of the MSS-based image encryption method

## 3 Preliminary Works

Vector quantization (VQ) encoding is a fundamental technique in signal processing and data compression. It plays a crucial role in various applications, including image and speech compression, pattern recognition, and data transmission. This approach represents a continuous signal or data sequence using a limited set of specific vectors, thereby reducing data size and facilitating storage.

The VQ technique relies on the VQ codebook, which is trained through the following steps:

Step 1: Select 3 to 5 representative images to serve as training samples.

Step 2: Divide all training samples into non-overlapping blocks of size $b \times b$ to form a pool of potential candidates for centroid initialization.

Step 3: Randomly select $l$ blocks from the candidate pool to initialize the centroids for the clustering process.

Step 4: Assign each block in the training samples to the nearest centroid based on the Euclidean distance between the block and the centroids.

Step 5: Update the centroid for each cluster by calculating the mean of all blocks assigned to that cluster.

Step 6: Iterate through Steps 4 and 5 until the recalculated centroids stabilize, indicating convergence of the clustering process.

Thus, a VQ codebook is constructed, comprising $l$ codewords, with each codeword representing a $b \times b$ vector. Each codeword is then assigned a unique index from 1 to $l$. The codebook serves as a reference for encoding and decoding operations in the VQ technique.

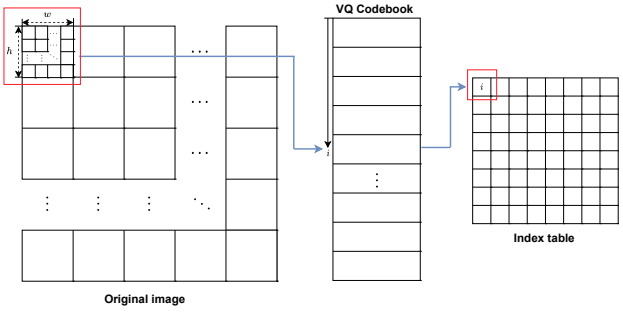



**Figure 2.** Illustration of the VQ encoding process

In the encoding process, the input image is divided into non-overlapping image blocks. For each image block, the distance to each codeword in the codebook is calculated, and the closest codeword is selected as the block encoding. Subsequently, the index corresponding to the selected codeword is recorded, which will be utilized in the decoding process. Figure 2 includes a legend that offers a more elucidating depiction of the encoding process. In the decoding process, the recorded indices are employed to retrieve the corresponding codewords from the codebook. These codewords are then assembled to reconstruct an image, resulting in an approximate reproduction of the original grayscale image.

## 4 Cryptanalysis via VQ Attack

In this section, we introduce a cryptanalysis of MSS-based image encryption via VQ Attack. We initiated our study with an analysis of the MSS-based image encryption scheme to gain insights of its properties. Additionally, we introduced an element variance sequence (EVS) technique to effectively capture the changing patterns within the ciphertext images block-wise. Finally, we applied the vector quantization (VQ) attack to estimate the content of plaintext images. The flowchart of the proposed scheme is illustrated in Figure 3.

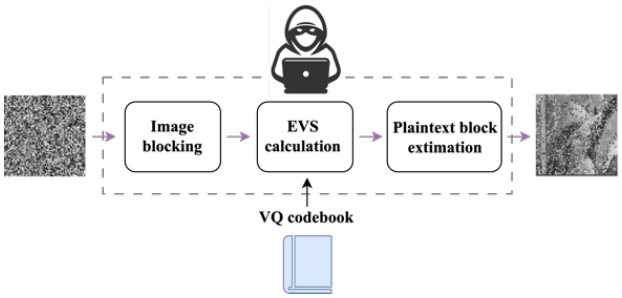



**Figure 3.** The flowchart of the proposed scheme

### 4.1 MSS-based Image Encryption Analysis

As introduced in Section 2.2, the MSS method is employed to encrypt images in the RDHEI scheme. Overall, the MSS-based image encryption method offers the following advantages:

(1) Big encryption space. In theory, the MSS-based image encryption can yield $256^{(W\times H)/(b_1\times b_2)}$ ciphertext versions. Taking a 512 × 512 image as an illustrative example, the MSS-based image encryption method would yield $256^{16384}$ ciphertext versions when the block size is set to 4 × 4. This results in a computational challenge even with the most advanced contemporary computer hardware when attempting to breach the MSS-based encryption through exhaustive brute force attacks.

(2) High tolerance. In their scheme, the original image is encrypted into $n$ share images, and only when $t$ or more share images are collected can the original image be recovered. As a result, the scheme can recover the original image even when there are $n - t$ of the shared ciphertext images are damaged or corrupted.

(3) Large data payload. In their method, pixels within the same image block are encrypted using the same pre-constructed $n \times t$ matrix and the same set of random positive integers $\beta_1, \beta_2, \ldots, \beta_{t-1}$. Consequently, most of the pixel correlation within the plaintext block are preserved in the ciphertext block. This results in a significant amount of redundant space within the shared images, which can be further utilized to achieve a large data payload in RDHEI methods.

Nevertheless, MSS-based image encryption methods raise security concerns:

(1) Preservation of Inter-block Correlation: Since this approach does not employ additional techniques, such as permutation, to mitigate inter-block correlations, they are fully preserved. This could potentially lead to the leakages of the image outlines.

(2) Preservation of Intra-block Correlation: The majority of pixel correlations from plaintext blocks are retained in the shared ciphertext blocks. This means that attackers can exploit the correlations within the ciphertext block to estimate the content of the corresponding plaintext block.

To clarify when intra-block correlations are preserved, we conduct the following derivation:

Suppose $p_1$ and $p_2$ represent two pixels within the same plaintext block. According to the introduction in Section 2.2, we represent Eq. (3) as

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1,t-1} \\ 1 & x_{21} & \cdots & x_{2,t-1} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & x_{n1} & \cdots & x_{n,t-1} \end{bmatrix} \times \begin{bmatrix} s \\ \beta_1 \\ \vdots \\ \beta_{t-1} \end{bmatrix} \bmod 257. \quad \textbf{(5)}$$

We encrypt these two pixels separately using Eq. (5) to obtain $q_1$ and $q_2$, the process can be expressed as

$$q_1 = (p_1 + x_{h1}\beta_1 + x_{h2}\beta_2 + \ldots + x_{h,t-1}\beta_{t-1}) \bmod 257, \quad \textbf{(6)}$$

and

$$q_2 = (p_2 + x_{h1}\beta_1 + x_{h2}\beta_2 + \ldots + x_{h,t-1}\beta_{t-1}) \bmod 257. \quad \textbf{(7)}$$

Here, $q_1$ and $q_2$ are two pixels within the same cyphertext block for the participant with the identity number $h$. Then, we further express Eqs. (6) and (7) as

$$k_1 \times 257 + q_1 = p_1 + x_{h1}\beta_1 + x_{h2}\beta_2 + \ldots + x_{h,t-1}\beta_{t-1} \quad \textbf{(8)}$$

and

$$k_2 \times 257 + q_2 = p_2 + x_{h1}\beta_1 + x_{h2}\beta_2 + \ldots + x_{h,t-1}\beta_{t-1}, \quad \textbf{(9)}$$

where

$$k_1 = floor((p_1 + x_{h1}\beta_1 + x_{h2}\beta_2 + \ldots + x_{h,t-1}\beta_{t-1})/257). \quad \textbf{(10)}$$

and

$$k_2 = floor((p_2 + x_{h1}\beta_1 + x_{h2}\beta_2 + \ldots + x_{h,t-1}\beta_{t-1})/257). \quad \textbf{(11)}$$

After that, let Eq. (8) and Eq. (9) be subtracted to obtain

$$\begin{aligned} &(k_1 - k_2) \times 257 + q_1 - q_2 \\ &= (p_1 + x_{h1}\beta_1 + x_{h2}\beta_2 + \cdots \\ &+ x_{h,t-1}\beta_{t-1}) \\ &-(p_2 + x_{h1}\beta_1 + x_{h2}\beta_2 + \cdots \\ &+ x_{h,t-1}\beta_{t-1}). \end{aligned} \quad \textbf{(12)}$$

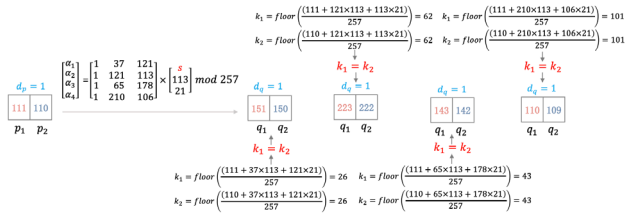
Further simplification gives us

$$(k_1 - k_2) \times 257 + q_1 - q_2 = p_1 - p_2. \quad \textbf{(13)}$$

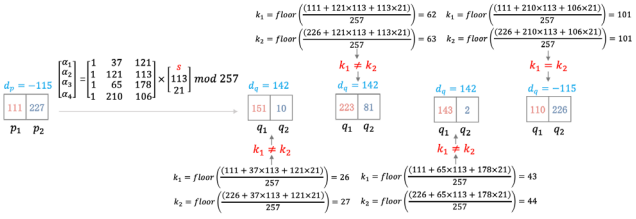
Obviously, we can further get

$$\begin{cases} q_1 - q_2 = p_1 - p_2 \ if \ k_1 = k_2 \\ q_1 - q_2 \neq p_1 - p_2 \ if \ k_1 \neq k_2 \end{cases}. \quad \textbf{(14)}$$

As a result, the pixel correlation is preserved when $k_1 = k_2$; otherwise, it is not. Building upon this, we further analyze Eqs. (10) and (11) and identify that the primary distinction between them lays in the values of $p_1$ and $p_2$. Given the inherent local smoothness of digital images, the difference between pixels within a local region tends to be small. Consequently, we can deduce that the probability of $k_1 = k_2$ is high when the difference $d$ between $p_1$ and $p_2$ is small.



(a)



(b)

**Figure 4.** Two examples to illustrate the MSS-based image encryption method

For the ease of understanding, we provide two examples in Figure 4. In these given examples, we set $n = 4$ and $t = 3$, which resulting in the construction of a 4 × 3 matrix. From the examples, we can observe that the difference $d_q$ between $q_1$ and $q_2$ is equal to the difference $d_p$ between $p_1$ and $p_2$ when $k_1 = k_2$, whereas they are different when $k_1 \neq k_2$. When we compare Figure 4(a) and Figure 4(b) furthermore, we observe that the pixel differences in Figure 4(a) are fully preserved, with small pixel differences between $p_1$ and $p_2$. In contrast, the pixel correlations in Figure 4(b) are not preserved, and the pixel differences between $p_1$ and $p_2$ are relatively large.

### 4.2 Element Variance Sequence

Before presenting the technology to obtain the plaintext content from the shared ciphertext images generated by the MSS-based image encryption method, we present the concept of the element variance sequence (EVS), which reflects the element changing pattern. The EVS for a sequence can be obtained using

$$v_i = e_i - e_{i+1},\ i = 1, 2, ..., n, \tag{15}$$

Where $e_1$, $e_2$, …, $e_n$ are the elements within the sequence and $v_1$, $v_2$, …, $v_{n-1}$ are the EVS results for the sequence.

### 4.3 Plaintext Content Estimation Based on VQ Attack

We can recall that the MSS-based encryption scheme preserves pixel correlations from the plaintext block in the ciphertext block when $k_1 = k_2$. As a result, the pixel change patterns within the plaintext block and the ciphertext block are also identical in this case. Based on this, we propose a plaintext content estimation method. We first capture the pixel change patterns within the ciphertext block by calculating its EVS. Subsequently, we compare the EVS of the ciphertext block with all EVS results of the codewords in its VQ codebook to identify the closest matching codeword. Finally, we substitute the ciphertext block with the selected codeword.

Assuming we have acquired a VQ codebook comprising $l$ codewords through the training process outlined in Section 3, and a shared ciphertext image $I_c$ obtained using the MSS-based image encryption method. The details of the plaintext content estimation method are as follows:

Step 1: Divide the ciphertext image $I_c$ into $b \times b$ blocks.

Step 2: Convert the ciphertext blocks into ciphertext sequences.

Step 3: Calculate the EVS for each codeword within the VQ codebook.

Step 4: Calculate the EVS for each ciphertext sequences.

Step 5: Compare the EVS of the ciphertext block with the EVS of each codeword in a VQ codebook to identify the closest matching codeword by

$$\arg\min_{c} = \sum_{i=1}^{b\times b-1} (v_i - v_i^c)^2, c = 1, 2, \cdots, l, \tag{16}$$

where $v_1$, $v_2$, …, $v_{b\times b-1}$ represent the EVS for a ciphertext sequence, and $v_1^c$, $v_2^c$, …, $v_{b\times b-1}^c$ represent the EVS for a codewords.

Step 6: Replace the ciphertext block with the codeword which has the closest EVS.

Step 7: Repeat Steps 5 and 6 until all ciphertext blocks have been processed.

Here, we assume that the vector length of a codeword is $b \times b$ in the proposed scheme.

## 5 Experimental Results and Analysis

In this section, we conduct experiments to assess the potential information leakage through a VQ attack. We first analyze the encryption performance of the MSS-based image encryption method. Subsequently, we present the estimation results obtained with the VQ attack. Finally, we provide the execution time of our scheme.

Our experimental evaluations are conducted using two standard grayscale images, namely 'Airplane' and 'Lena', as depicted in Figure 5. In addition, we utilize two types of codebooks, one containing the target images as training samples, and the other without them, for the VQ attack. Specifically, the codebook containing the target images as training samples is trained using 'Lena', 'Airplane', 'Baboon', 'Peppers', and 'Boat', while the codebook without the target images as training samples is trained using 'Elaine', 'Barbara' 'Baboon', 'Peppers', and 'Boat'. Note that since the ciphertext images generated by the MSS-based image encryption method share nearly identical properties, our experiments provide estimation results based on a single shared ciphertext image for each specific configuration.

(a) Airplane (b) Lena

**Figure 5.** Two standard grayscale images

### 5.1 Analysis of the MSS-based Ciphertext Images

Figure 6 and Figure 7 depict the visual effect of MSS-based image encryption with various block sizes. As observed in the figures, the encryption effectiveness is the highest when the block size is set to 4 × 4, making it virtually impossible to discern any information from the ciphertext images with the naked eye. However, as the image block size increases, the encryption effectiveness diminishes. Notably, when dealing with a 16 × 16 image block, one can discern the contour lines of the original images in the encrypted version. Consequently, the MSS-based image encryption method is unsuitable for large image blocks. This conclusion finds further support in the corresponding pixel distribution histograms and pixel 3D views, where an increase in block size results in decrease in non-uniform patterns.

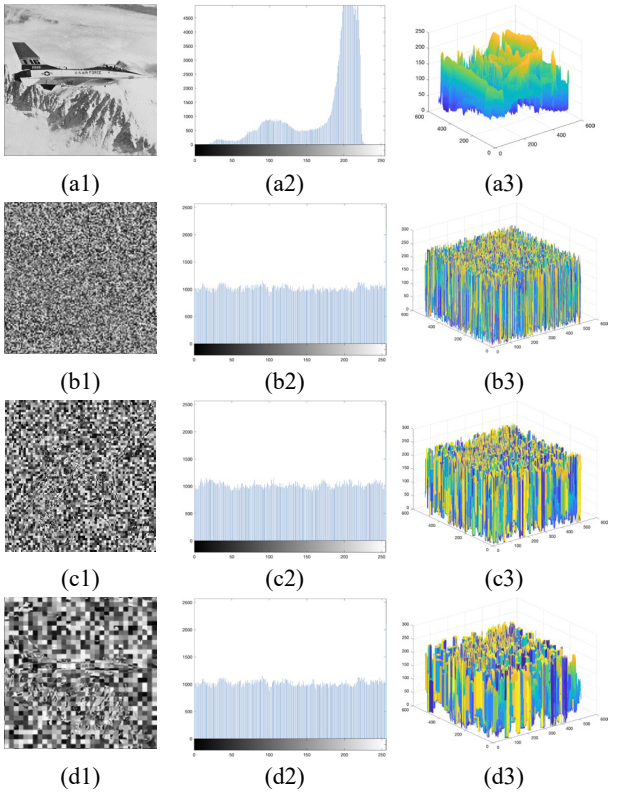



**Figure 6.** Visual effect of "Airplane" with MSS-based image encryption
((a1) shows the plaintext image, while (b1), (c1), and (d1) depict shared ciphertext images with block sizes of 4 × 4, 8 × 8, and 16 × 16, respectively. (a2) to (d2) are the corresponding pixel distribution histograms; (a3) to (d3) are the corresponding pixel 3D views.)

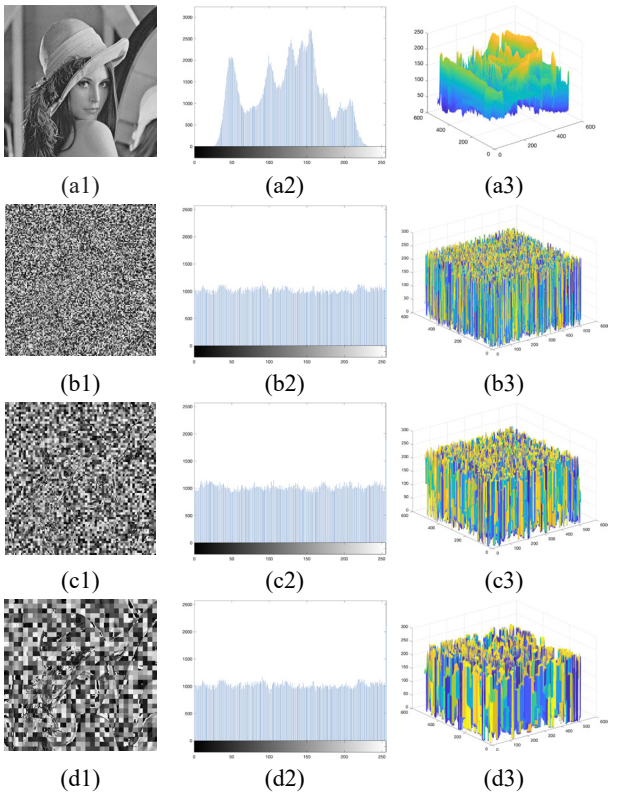



**Figure 7.** Visual effect of "Lena" with MSS-based image encryption
((a1) shows the plaintext image, while (b1), (c1), and (d1) depict shared ciphertext images with block sizes of 4 × 4, 8 × 8, and 16 × 16, respectively. (a2) to (d2) are the corresponding pixel distribution histograms; (a3) to (d3) are the corresponding pixel 3D views.)

### 5.2 Plaintext Image Estimation Performance

In the proposed VQ attack scheme, when an attacker obtains a shared ciphertext image, they can generate the corresponding estimated image. The estimated images obtained are presented in Figure 8 to Figure 11. Upon reviewing Figure 8 and Figure 9, it becomes apparent that when the block size is set to 16 × 16, the majority of the original image blocks are accurately estimated. As the block size decreases, the correctness in the estimated image blocks diminishes, but rough outlines of the plaintext images remain distinguishable. Moving on to Figure 10 and Figure 11, when the block size is 4 × 4 or 8 × 8, the general outlines of the plaintext images remain visible, especially with a 4 × 4 block size. However, in the case of a 16 × 16 block size, the estimation does not perform as well. Regardless of the scenario, it is worth noting that longer codebooks tend to result in more accurate estimates. Furthermore, upon comparing Figure 8 and Figure 9 with Figure 10 and Figure 11, we observe that the effectiveness of estimation using the codebook containing the target image surpasses that of the codebook without it. This is attributed to the inclusion of the target image in the training process, which may lead to the generation of codewords closely resembling the information present in the target images.

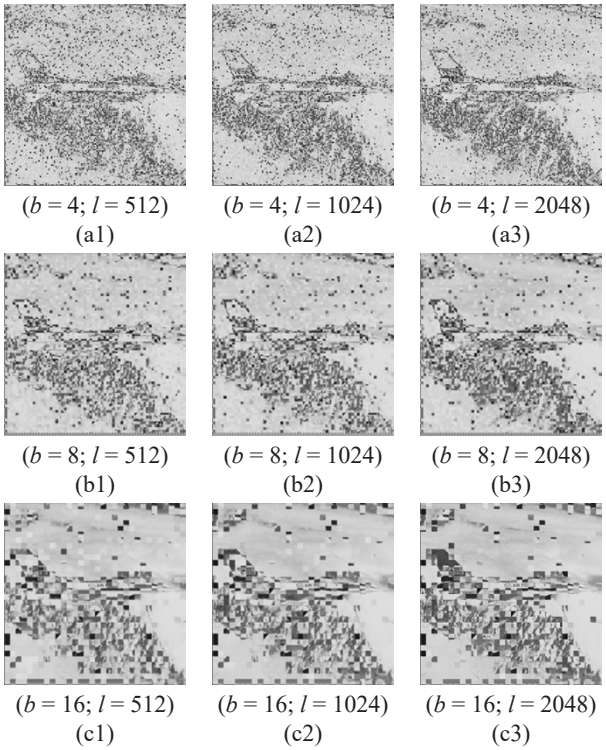



**Figure 8.** Estimated images of "Airplane" under different block sizes and VQ codebook lengths using the codebook trained with target images

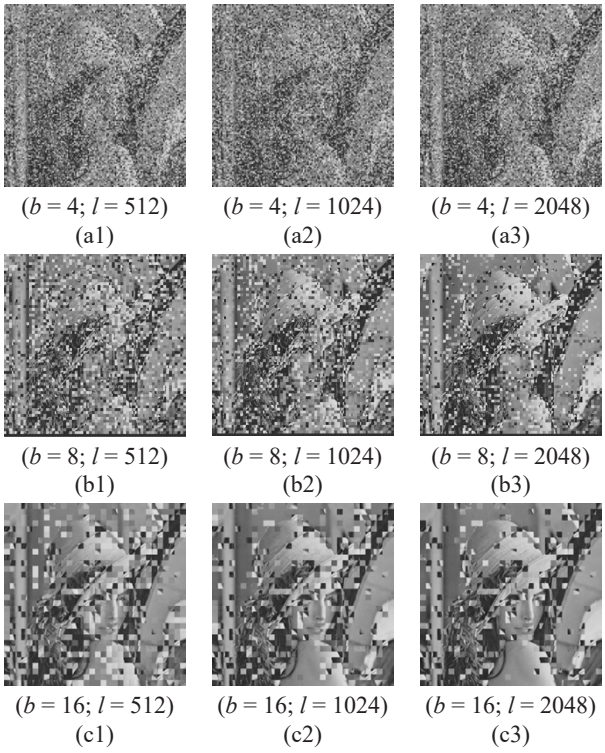



**Figure 9.** Estimated images of "Lena" under different block sizes and VQ codebook lengths using the codebook trained with target images

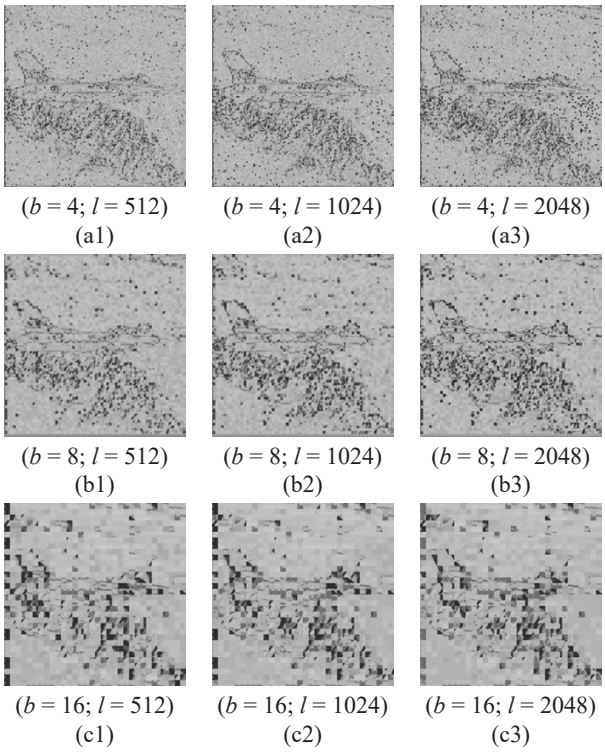



**Figure 10.** Estimated images of "Airplane" under different block sizes and VQ codebook lengths using the codebook trained without target images

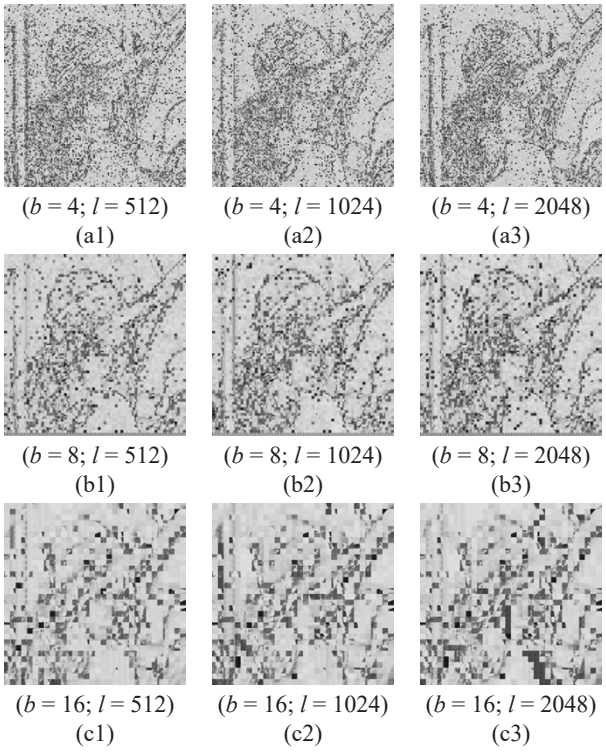



**Figure 11.** Estimated images of "Lena" under different block sizes and VQ codebook lengths using the codebook trained without target images

To quantify the estimation results, we extract the edges of the plaintext image and ciphertext image separately and match their similarities. The edges of the image are delineated by determining whether the pixel difference of each block exceeds the average pixel difference of the image. As a result, we can obtain the edge map $E$. Afterwards, we can calculate the similarity by

$$sim = (1 - ham(E(I) - E(I_c))) \times 100\%, \quad \textbf{(17)}$$

where $ham(*)$ is the Hamming distance, $E(I)$ and $E(I_c)$ represent the edge of plaintext image $I$ and cyphertext image $I_c$. We present quantitative results in Figure 12 and Figure 13. Based on the quantitative results, our VQ attack method demonstrates strong performance by achieving a similarity rate of 79.39% even in the worst case.

### 5.3 Execution Time

Table 1 displays the execution time of our scheme for both codebook training ($l$ = 512, $b$ = 4) and image estimation. The codebook training process takes 52.657s, while the image estimation process requires only 0.02s on average. The brief estimation time features the effectiveness of our method in decrypting MSS-based encrypted images.

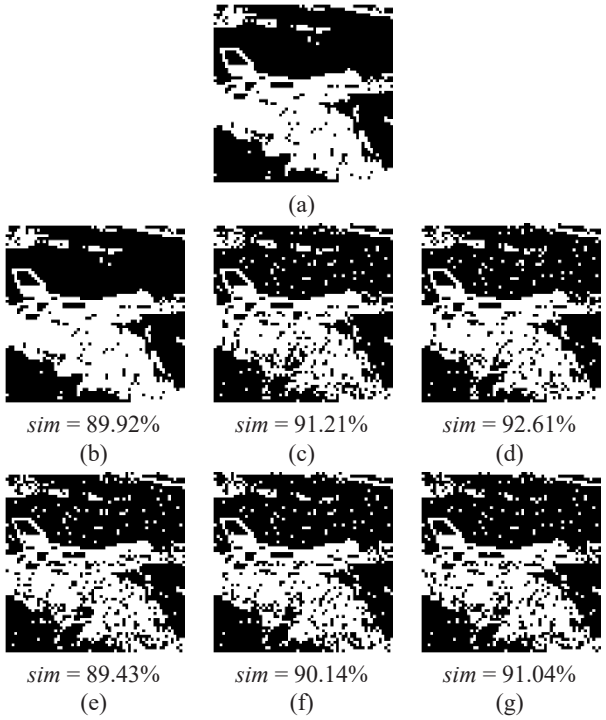



**Figure 12.** Quantitative results of "Airplane" under 8 × 8 block sizes and different VQ codebook lengths
((a) plaintext image, (b)-(d) edge maps using codebook trained with target image, and (e)-(f) edge maps using codebook trained without target images.)

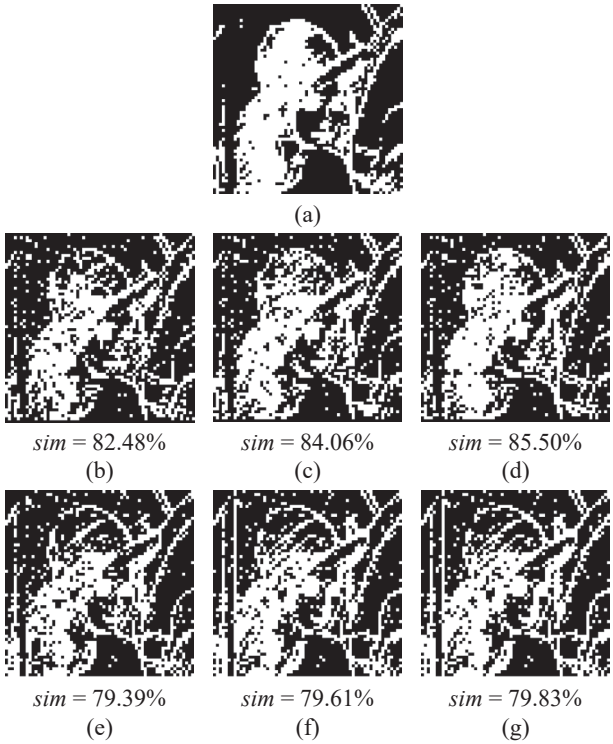



**Figure 13.** Quantitative results of "Lena" under 8 × 8 block sizes and different VQ codebook lengths
((a) plaintext image, (b)-(d) edge maps using codebook trained with target image, and (e)-(f) edge maps using codebook trained without target image.)

**Table 1.** The execution time of our scheme

| Codebook training | Image estimation (Average) |
|---|---|
| 52.657s | 0.02s |

# 6 Conclusions

In this paper, we conduct a security analysis of the MSS-based image encryption scheme and introduce an innovative cryptanalysis approach based on VQ attacks. Our objective is to estimate a plaintext content by analyzing the changing patterns of its ciphertext image using VQ technology. The experimental results demonstrate the effectiveness of our proposed cryptanalysis scheme in capturing the plaintext contents from the ciphertext images. These findings highlight a potential vulnerability within the MSS-based image encryption scheme and underscore the importance of considering security aspects in future MSS-related research. In addition, our proposed method exhibits high efficiency, with an average estimation time of only 0.02 seconds. The efficiency further reinforces the practicality and applicability of our approach in real-world scenarios. While the visual quality of the estimated images may not be optimal, it is essential to emphasize that our proposed method operates independently of the availability of plaintext images. Consequently, it can be deemed to exhibit relatively strong performance. In future research, we aim to integrate AI methodologies to further enhance the visual quality in images.

# References

[1] W. Zhang, H. Wang, D. Hou, N. Yu, Reversible Data Hiding in Encrypted Images by Reversible Image Transformation, *IEEE Transactions on Multimedia*, Vol. 18, No. 8, pp. 1469-1479, August, 2016.

[2] Y. Q. Shi, X. Li, X. Zhang, H.-T. Wu, B. Ma, Reversible Data Hiding: Advances in the Past Two Decades, *IEEE Access*, Vol. 4, pp. 3210-3237, May, 2016.

[3] Z. Yin, Y. Xiang, X. Zhang, Reversible Data Hiding in Encrypted Images Based on Multi-MSB Prediction and Huffman Coding, *IEEE Transactions on Multimedia*, Vol. 22, No. 4, pp. 874-884, April, 2020.

[4] P. Puteaux, W. Puech, A Recursive Reversible Data Hiding in Encrypted Images Method with a Very High payload, *IEEE Transactions on Multimedia*, Vol. 23, pp. 636-650, April, 2020.

[5] Y. Wu, Y. Xiang, Y. Guo, J. Tang, Z. Yin, An Improved Reversible Data Hiding in Encrypted Images Using Parametric Binary Tree Labeling, *IEEE Transactions on Multimedia*, Vol. 22, No. 8, pp. 1929-1938, August, 2020.

[6] F. Chen, Y. Yuan, H. He, M. Tian, H.-M. Tai, Multi-MSB Compression Based Reversible Data Hiding Scheme in Encrypted Images, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 31, No. 3, pp. 905-916, March, 2021.

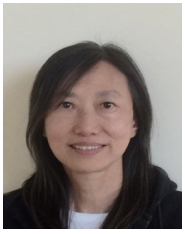
[7] S. Xu, J.-H. Horng, C.-C. Chang, C.-C. Chang, Reversible Data Hiding with Hierarchical Block Variable Length Coding for Cloud Security, *IEEE Transactions on Dependable and Secure Computing*, Vol. 20, No. 5, pp. 4199-4213, September-October, 2023.

[8] S. Yi, Y. Zhou, Separable and Reversible Data Hiding in Encrypted Images Using Parametric Binary Tree Labeling, *IEEE Transactions on Multimedia*, Vol. 21, No. 1, pp. 51-64, January, 2019.
[9] X. Zhang, Reversible Data Hiding in Encrypted Image, *IEEE Signal Processing Letters*, Vol. 18, No. 4, pp. 255-258, April, 2011.
[10] X. Zhang, Separable Reversible Data Hiding in Encrypted Image, *IEEE Transactions on Information Forensics and Security*, Vol. 7, No. 2, pp. 826-832, April, 2012.
[11] F. Huang, J. Huang, Y. Shi, New Framework for Reversible Data Hiding in Encrypted Domain, *IEEE Transactions on Information Forensics and Security*, Vol. 11, No. 12, pp. 2777-2789, December, 2016.
[12] D. Xu, R. Wang, Separable and Error-free Reversible Data Hiding in Encrypted Images, *Signal Processing*, Vol. 123, pp. 9-21, June, 2016.
[13] S. Yi, Y. Zhou, Parametric Reversible Data Hiding in Encrypted Images Using Adaptive Bit-level Data Embedding and Checkerboard based Prediction, *Signal Processing*, Vol. 150, pp. 171-182, September, 2018.
[14] Y. Qiu, Q. Ying, Y. Yang, H. Zeng, S. Li, Z. Qian, High-capacity Framework for Reversible Data Hiding in Encrypted Image Using Pixel Prediction and Entropy Encoding, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 32, No. 9, pp. 5874-5887, September, 2022.
[15] Z. Hua, Y. Wang, S. Yi, Y. Zheng, X. Liu, Y. Chen, X. Zhang, Matrix-Based Secret Sharing for Reversible Data Hiding in Encrypted Images, *IEEE Transactions on Dependable and Secure Computing*, Vol. 20, No. 5, pp. 3669-3686, September-October, 2023.
[16] X. Wu, J. Weng, W. Yan, Adopting Secret Sharing for Reversible Data Hiding in Encrypted Images, *Signal Processing*, Vol. 143, pp. 269-281, February, 2018.
[17] Y.-C. Chen, T.-H. Hung, S.-H. Hsieh, C.-W. Shiu, A New Reversible Data Hiding in Encrypted Image Based on Multi-secret Sharing and Lightweight Cryptographic Algorithms, *IEEE Transactions on Information Forensics and Security*, Vol. 14, No. 12, pp. 3332-3343, December, 2019.
[18] B. Chen, W. Lu, J. Huang, J. Weng, Y. Zhou, Secret Sharing Based Reversible Data Hiding in Encrypted Images with Multiple Data-hiders, *IEEE Transactions on Dependable and Secure Computing*, Vol. 19, No. 2, pp. 978-991, March-April, 2022.
[19] C. Qin, C. Jiang, Q. Mo, H. Yao, C.-C. Chang, Reversible Data Hiding in Encrypted Image via Secret Sharing Based on GF (p) and GF ($2^8$), *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 32, No. 4, pp. 1928-1941, April, 2022.

## Biographies

**Chin-Chen Chang** received a B.S. degree in Applied Mathematics, an M.S. degree in Computer and Decision Sciences from National Tsing Hua University, and a Ph.D. degree in Computer Engineering from National Chiao Tung University. He is currently the Chair Professor of Feng Chia University. His current research interests include database design, cryptography, and data structures.

**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, image processing, and computer vision.



**Jui-Chuan Liu** is pursuing her Ph.D. degree at Feng Chia University starting from 2023. She had worked in the EDA industry for almost 40 years after graduating from Florida Institute of Technology, Melbourne, Florida, USA. Her interests in academic research include cyber security, data hiding and machine learning.

**Ching-Chun Chang** received his PhD in Computer Science from the University of Warwick, UK, 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.