Multi-Party Reversible Data Hiding in Ciphertext Binary Images Based on Visual Cryptography

Bing Chen, *Member, IEEE*, Jingkun Yu, Bingwen Feng, Wei Lu, *Member, IEEE*, and Jun Cai

***Abstract*—Existing methods for reversible data hiding in ciphertext binary images only involve one data hider to perform data embedding. When the data hider is attacked, the original binary image cannot be perfectly reconstructed. To this end, this letter proposes multi-party reversible data hiding in ciphertext binary images, where multiple data hiders are involved in data embedding. In this solution, we use visual cryptography technology to encrypt a binary image into multiple ciphertext binary images, and transmit the ciphertext binary images to different data hiders. Each data hider can embed data into a ciphertext binary image and generate a marked ciphertext binary image. The original binary image is perfectly reconstructed by collecting a portion of marked ciphertext binary images from the unattacked data hiders. Compared with existing solutions, the proposed solution enhances the recoverability of the original binary image. Besides, the proposed solution maintains a stable embedding capacity for different categories of images.**

***Index Terms*—Reversible data hiding, ciphertext binary image, visual cryptography, multiple data hiders.**

I. Introduction

DATA hiding, as a branch of information security, embeds secret data into the cover to produce the marked cover and extracts embedded secret data from the marked cover, which is widely applied in convert communication [1] and copyright protection [2]. In order to both extract the embedded data and reconstruct the original cover, reversible data hiding (RDH) is proposed. Existing RDH methods are mainly designed by exploiting histogram shifting [3], difference expansion [4] and prediction error expansion [5].

Most RDH methods are focused on grayscale images or color images. In fact, binary images play a critical role in our daily lives, such as digital signature and scanned text. As a result, reversible data hiding in binary images (RDHBI) has attracted considerable interests and made some achievements. In [6], a binary image is first segmented into multiple patterns, each of which is comprised of four pixels. The high-frequency patterns and low-frequency patterns are then obtained by counting the occurrence frequencies of the patterns, in which the locations of the low-frequency patterns are identified by a location map (LM). The secret data is embedded into the high-frequency patterns and low-frequency patterns via pattern replacement, respectively. To improve embedding capacity, Dong et al. [7] proposed that the low-frequency pattern is superseded by the second low-frequency pattern. The high-frequency patterns and the second low-frequency patterns are labeled as 0 and 1, which reduces the size of LM effectively. Another improved RDHBI method [8] was presented by adopting magnifying strategy. Specifically, the reference pattern is chosen by analyzing the cross pattern, and the original image is magnified by a factor of four based on the reference pattern. The redundant information of the magnified image is increased, thus more secret data can be embedded.

Inspired by the RDH technique for grayscale images, a histogram shifting-based RDHBI method [9] was proposed. In this method, the multiple pixels of a binary image are treated as one component, i.e., the binary image is treated as a multi-bit image and hence is processed using histogram shifting. By adjusting the number of pixels in the component, more peak points and fewer zero points are generated in the histogram, which results in a satisfying visual quality. In [10], Yin et al. divided a binary image into blocks with a size of 3×3 and generated symmetric flip degrees (SFDs) of the blocks via evaluating the distance between center pixel and adjacent pixels. The SFDs are divided into five groups, each of which contains five different SFDs. By modifying the big SFD with high bins and the small SFD with low bins, the secret data can be embedded in each group. To achieve the trade-off between visual quality and embedding capacity, an RDHBI method [11] by adopting an opposite center pixels mechanism was given. The secret data is embedded into the binary image by flipping an optimal pattern pair with opposite center pixels.

With the requirements of the outsourced storage and privacy-preserving, the image owner desires that the plaintext image does not exposed during transmission, so reversible data hiding in ciphertext binary image (RDHCBI) is presented. In [12], Ren et al. segmented binary image into pure blocks and non-pure blocks and labeled them as 0 and 1 with a LM. The secret data is embedded into pure blocks using bit replacement. To decrease the length of the LM, an RDHCBI method [13] based on halving

Received 1 December 2024; revised 24 March 2025; accepted 26 March 2025. Date of publication 2 April 2025; date of current version 16 April 2025. This work was supported by the National Natural Science Foundation of China under Grant 62102101, Grant 62261160653, Grant 62441237, and Grant 62472199. The associate editor coordinating the review of this article and approving it for publication was Prof. Zhongyun Hua. *(Corresponding author: Jun Cai.)*

Bing Chen, Jingkun Yu, and Jun Cai are with the School of Cyber Security, Guangdong Polytechnic Normal University, Guangzhou 510665, China (e-mail: chenbing@gpnu.edu.cn; yujingkun@gpnu.edu.cn; caijun@gpnu.edu.cn).

Bingwen Feng is with the College of Cyber Security, Jinan University, Guangzhou 510632, China (e-mail: bingwfeng@gmail.com).

Wei Lu is with the School of Computer Science and Engineering, Ministry of Education Key Laboratory of Information Technology, Guangdong Province Key Laboratory of Information Security Technology, Sun Yat-sen University, Guangzhou 510006, China (e-mail: luwei3@mail.sysu.edu.cn).

Digital Object Identifier 10.1109/LSP.2025.3557273

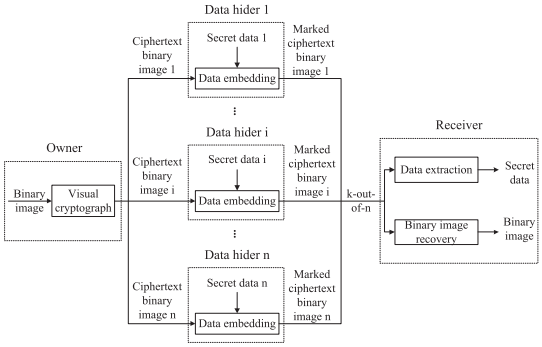

Fig. 1. Sketch of the presented approach.

compression was presented. In [12] and [13], to perfectly restore original binary image, each pure block needs to retain one pixel as an identifier, which takes up additional embedding space. To this end, Zhang et al. [14] segmented the image into white blocks, black blocks and mixed blocks and introduced Huffman coding to label auxiliary information. In addition, a weight prediction method is designed for mixed blocks to improve visual quality.

However, the RDHCBI methods only involve one data hider, and cannot reconstruct the original binary image on condition that the data hider is attacked. To eradicate the problem, this letter presents multi-party reversible data hiding in ciphertext binary images (MRDHCBI), where the binary image is encrypted into multiple ciphertext binary images via using visual cryptography [15]. Also, multiple data hiders are introduced to perform data embedding. The main contributions of this letter are listed as follows.

- Recoverability. The original binary image can also be losslessly recovered when parts of the data hiders are attacked.
- Stable embedding capacity. The embedding space is created by the visual cryptography algorithm.
- Slight fluctuation on encryption runtime. The binary image is encrypted into multiple ciphertext binary images pixel by pixel.

The rest of this letter is organized below. Section II elaborates the presented MRDHCBI approach in detail. Section III shows the experimental results and analysis. Finally, Section IV concludes this article.

II. Presented Approach

As shown in Fig. 1, the presented MRDHCBI approach based on visual cryptography consists of three procedures: binary image encryption procedure, data embedding procedure, and data extraction and binary image recovery procedure. Different from RDHCBI approach, the proposed MRDHCBI approach involves multiple data hiders to perform data embedding. Firstly, the owner encrypts a binary image to generate $n(n \geq 2)$ ciphertext binary images adopting visual cryptography and transmits them to multiple data hiders for data embedding. Secondly, each data hider embeds secret data into the ciphertext binary image to get a marked ciphertext binary image. Finally, when any k-out-of-n marked ciphertext binary images are licensed to a receiver, the receiver can recover the original binary image and extract the embedded secret data, where $2 \leq k \leq n$.

A. Binary Image Encryption

The binary image with size $X \times Y$ is denoted as BI. Let BI_{xy} be the pixel value at location (x, y), where $BI_{xy} \in \{0, 1\}$, $1 \leq x \leq X$, $1 \leq y \leq Y$. To realize binary image encryption, two basis matrices B^0 and B^1 are constructed, as represented by

$$B^t = \begin{pmatrix} a^t_{11} & a^t_{12} & \cdots & a^t_{1m} \\ a^t_{21} & a^t_{22} & \cdots & a^t_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ a^t_{n1} & a^t_{n2} & \cdots & a^t_{nm} \end{pmatrix}, t = 0, 1 \tag{1}$$

where $a^t_{ij} = 0$ or 1, $1 \leq i \leq n$, $1 \leq j \leq m$. After that, the set C^t is obtained by performing all possible column permutations of the basis matrix B^t.

For the pixel $BI_{xy} = t$ in the binary image, a matrix is randomly chosen from the corresponding set C^t to generate n ciphertext pixel blocks. The generated n ciphertext pixel blocks are the row vectors of the chosen matrix and are denoted as

$$\begin{cases} CBI^1_{xy} = (a^t_{1r_1}\ a^t_{1r_2}\ \cdots\ a^t_{1r_m}) \\ CBI^2_{xy} = (a^t_{2r_1}\ a^t_{2r_2}\ \cdots\ a^t_{2r_m}) \\ \quad\vdots \\ CBI^n_{xy} = (a^t_{nr_1}\ a^t_{nr_2}\ \cdots\ a^t_{nr_m}) \end{cases} \tag{2}$$

where $(r_1, r_2, \ldots, r_m)$ is a permutation of $(1, 2, \ldots, m)$ and CBI^i_{xy} is the ith ciphertext pixel block, $1 \leq i \leq n$. Specifically, the binary image encryption is formulated by

$$(CBI^1_{xy}, CBI^2_{xy}, \ldots, CBI^n_{xy}) = Enc(BI_{xy}, B^{BI_{xy}}) \tag{3}$$

where $Enc(*)$ is the visual cryptography algorithm with the basis matrix B^t.

In the event that all the pixels in the binary image BI are encrypted, n ciphertext binary images are generated and transmitted to n data hiders. The ith ciphertext binary images CBI^i is generated by concatenating all ciphertext pixel blocks CBI^i_{xy}, as represented by

$$CBI^i = \begin{pmatrix} CBI^i_{11} & CBI^i_{12} & \cdots & CBI^i_{1Y} \\ CBI^i_{21} & CBI^i_{22} & \cdots & CBI^i_{2Y} \\ \vdots & \vdots & \vdots & \vdots \\ CBI^i_{X1} & CBI^i_{X2} & \cdots & CBI^i_{XY} \end{pmatrix}. \tag{4}$$

B. Data Embedding

In the data embedding procedure, each data hider embeds data into a ciphertext binary image to get a marked ciphertext binary image. To improve the security of the embedded data, the data is converted into secret data via a cryptographic algorithm with a data embedding key. For instance, a pseudo-random sequence is generated via a pseudo-random number generator using the data embedding key, and the pseudorandom sequence is used to encrypt the data into secret data via the exclusive-OR operation. Then, a bit of secret data is embedded into a ciphertext pixel block to generate a marked ciphertext pixel block. Let the ith secret data in the ith data hider be

$RD^i = (rd_1^i, rd_2^i, \ldots, rd_p^i, \ldots, rd_{X\times Y}^i)$, where $1 \le p \le X \times Y$, $rd_p^i = 0$ or 1 is a bit of secret data. Specifically, the data embedding is formulated by

$$MBI_{xy}^i[j] = \begin{cases} 1 - CBI_{xy}^i[j], & \text{if } rd_p^i = 1, j = 1 \\ CBI_{xy}^i[j], & \text{others} \end{cases} \quad (5)$$

where $CBI_{xy}^i[j]$ and $MBI_{xy}^i[j]$ are the jth pixel in the ciphertext pixel block CBI_{xy}^i and in the marked ciphertext pixel block MBI_{xy}^i, respectively, $1 \le j \le m$. In this way, the marked ciphertext binary image can be got by embedding secret data RD^i into ciphertext binary image CBI^i, denoted as $MBI^i, i = 1, 2, \ldots, n$.

C. Data Extraction and Binary Image Recovery

In this procedure, with any k marked ciphertext binary images, the data extraction and binary image recovery are performed by the receiver. It is assumed that k marked ciphertext binary images are licensed to the receiver, denoted as $MBI^{s_r}, 1 \le r \le k$, where $\{s_1, s_2, \ldots, s_k\}$ is a subset of $\{1, 2, \ldots, n\}$.

In the data embedding procedure, the ciphertext pixel blocks are modified to marked ciphertext pixel blocks, which results in different Hamming weights among them. Therefore, the embedded secret data in any k marked ciphertext binary images can be extracted by calculating the Hamming weights of the marked ciphertext pixel blocks. Specifically, the data extraction is formulated by

$$rd_p^{s_r} = \begin{cases} 0, & \text{if } H(MBI_{xy}^{s_r}) = W \\ 1, & \text{others} \end{cases} \quad (6)$$

where W is the Hamming weight of any row vector of the two basis matrices. The Hamming weight is calculated by the number of elements in a vector whose value is equal to 1. Sequentially, the secret data can be acquired and the original data is obtained via decoding the acquired secret data using data embedding key.

With regard to binary image recovery, the receiver first transforms the marked ciphertext binary images to ciphertext binary images, and then transforms the ciphertext binary images to original binary image. According to the (5), the ciphertext pixel blocks are converted into marked ciphertext pixel blocks by modifying the first pixel of the ciphertext pixel blocks. Thus the ciphertext pixel blocks can be easily restored from the marked ciphertext pixel blocks, as represented by

$$CBI_{xy}^{s_r}[j] = \begin{cases} 1 - MBI_{xy}^{s_r}[j], & \text{if } H(MBI_{xy}^{s_r}) \ne W, j = 1 \\ MBI_{xy}^{s_r}[j], & \text{others} \end{cases} \quad (7)$$

When all marked ciphertext pixel blocks are restored to ciphertext pixel blocks, the ciphertext binary images are obtained. After that, the original binary image will be restored from the obtained ciphertext binary images. The receiver performs a Boolean OR operation on the ciphertext binary images to generate a combined binary image, denoted as OBI. Let OBI_{xy} be the pixel block of OBI. By judging the Hamming weights of the pixel blocks, the pixel blocks of the original binary image are restored, as represented by

$$BI_{xy} = \begin{cases} 0, & \text{if } H(OBI_{xy}) < d \\ 1, & \text{others} \end{cases} \quad (8)$$

where d is a threshold. d is chosen from $(u, v]$, where u and v are the Hamming weight of the vector obtained by performing a Boolean OR operation on any k row vectors of the basis matrices B^0 and B^1, respectively. At last, the original binary image is gained by concatenating the restored pixel blocks.

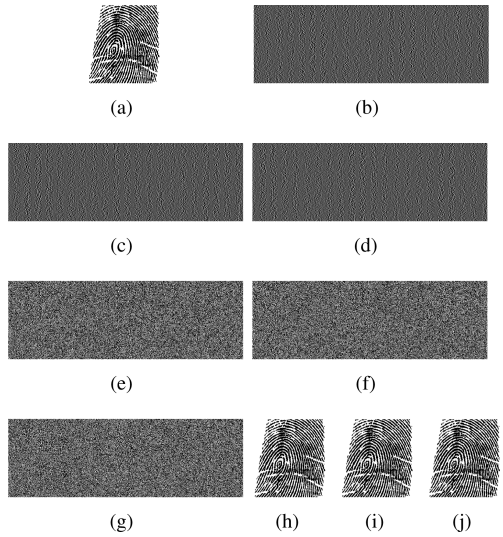

Fig. 2. Using (2, 3) threshold visual cryptography for the experiment. (a) Binary image Texture. (b)–(d) The three ciphertext binary images. (e)–(g) The three marked ciphertext binary images embedding 65536 bits. (h) Restored binary image from (e) and (f). (i) Restored binary image from (e) and (g). (j) Restored binary image from (f) and (g).

III. Experimental Results and Analysis

Six categories of binary images chosen from the dataset [16] are adopted for experiments, including "Cartoon(183.bmp)", "CAD(487.bmp)", "Texture(760.bmp)", "Mask(1001.bmp)", "Pattern(1704.bmp)", and "Document(3060.bmp)".

A. Feasibility of the Presented Approach

We conduct experiments on different binary images and encryption strategies to elucidate the feasibility of the presented approach. Firstly, the $(2, 3)$ threshold visual cryptography algorithm is used to encrypt binary image Texture. The experiment result is given in Fig. 2. Fig. 2(a) displays the original binary image Texture, and Fig. 2(b)–(d) display the corresponding three ciphertext binary images. Each ciphertext binary image can be embedded with 65536 bits of secret data to generate marked ciphertext binary image, as shown in Fig. 2(e)–(g). When any two marked ciphertext binary images are obtained, the original binary image is recovered losslessly, as shown in Fig. 2(h)–(j).

B. Comparison of Embedding Capacity

Herein, the embedding capacity comparison of the presented approach and several state-of-the-art RDHCBI approaches [12],

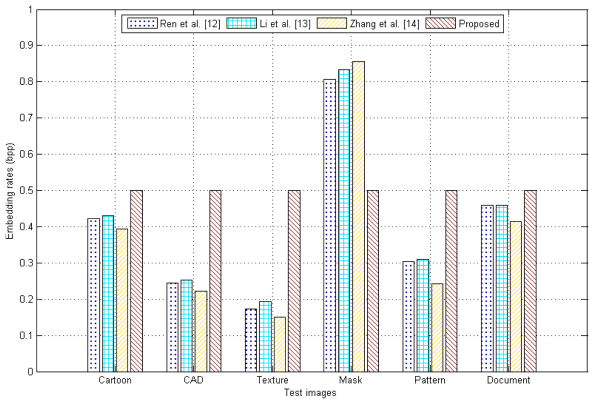

Fig. 3. Embedding rates of the presented approach and state-of-the-art RDHCBI approaches under the lossless recovery of the original binary image.

TABLE I
ENCRYPTION RUNTIME COMPARISON FOR DIFFERENT BINARY IMAGES

Images	Ren et al. [12]	Li et al. [13]	Zhang et al. [14]	Presented approach
Cartoon	71.1064	72.1823	54.8561	60.9352
CAD	73.7246	75.8646	59.7081	61.1015
Texture	79.0976	79.7218	62.9826	60.3014
Mask	92.4886	95.6055	66.3101	59.6784
Pattern	71.3982	73.0493	59.5149	60.6579
Document	71.1297	71.1876	55.3301	60.8692

[13], [14] is provided under the lossless restoration of the original binary image. Fig. 3 elucidates the embedding rate (bit per pixel, bpp) of different binary images. It can be seen that the presented approach outperforms those RDHCBI approaches in most of the test images. Also, the embedding rate of the presented approach is stable, that is, the embedding rate is not sensitive to the pixel features of binary images. For RDHCBI approaches, the embedding rate varies with different images. The reason is that they reserve embedding space before encryption by exploiting the correlation of adjacent pixels. Obviously, embedding rate is high for smooth textured images and low for complex textured images. In contrast, the presented approach adopts the visual cryptography algorithm to create the embedding space, which enables a stable embedding rate for binary images with various features.

C. Comparison of Encryption Runtime

In Table I, the runtime of the image encryption phase is given, in which (2,3) visual cryptography is considered in the presented approach. The codes of programs are compiled via Matlab 2021a and run on 64-bit Windows 10 with Intel Core i5-11600 CPU @2.8 GHz, 8 GB RAM, and 256 GB SSD. It is apparent from the table that the presented approach significantly outperforms the approaches in [12] and [13] at encryption runtime (in milliseconds). Compared with the approach in [14], the presented approach has shorter encryption runtime in image Mask and image Texture. In fact, the approach in [14] divides the binary image into blocks with larger size compared with the approaches of [12] and [13]. It means that fewer blocks are processed in image encryption phase and the corresponding encryption runtime can be reduced. On the other hand, we find that the encryption runtime of the presented approach is approximately stable, which is different from the approaches in [12], [13] and [14]. In other words, the encryption runtime of the presented approach is not affected by the pixel feature of the binary images. For the approaches [12], [13], [14], the encryption runtime is related to the pixel feature of the binary images, especially for the image Mask.

TABLE II
FUNCTIONAL COMPARISON

Functions	Ren et al. [12]	Li et al. [13]	Zhang et al. [14]	Presented approach
Embedding space	Correlation	Correlation	Correlation	Encryption
Preprocessing	Yes	Yes	Yes	No
Encryption	Stream cipher	Stream cipher	Stream cipher	Visual cryptography
Data hider	Single	Single	Single	Multiple

D. Functional Comparison

Table II gives functional comparison of the presented approach and state-of-the-art RDHCBI approaches [12], [13], [14]. Different from the RDHCBI approaches that generate the embedding space by the correlation of adjacent pixels, the presented approach generates the embedding space by the encryption algorithm. Because the embedding space is reserved before encryption, preprocessing is unavoidable in the approaches of [12], [13], [14]. All these three RDHCBI approaches encrypt the binary image by using stream cipher, in which a simple exclusive-OR operation is performed. However, to reinforce the security of the approach, the key of the stream cipher cannot be reused, which leads to difficulty in key management. In addition, the three RDHCBI approaches only involve one data hider, where the original binary image cannot be restored on condition that the data hider is attacked. The presented approach improves the recoverability of the original binary image via introducing multiple data hiders.

IV. CONCLUSION

This letter presents a MRDHCBI approach based on visual cryptography, where multiple data hiders are introduced for data embedding. First, the original binary image is encrypted into multiple ciphertext binary images by using visual cryptography. Then these ciphertext binary images are transmitted to multiple data hiders for data embedding. Each data hider embeds secret data into a ciphertext binary image to generate a marked ciphertext binary image by modifying ciphertext pixel blocks. With any k-out-of-n marked ciphertext binary images, the original binary image is restored by adopting a Hamming weight mechanism. The simulation experiment is given to elucidate the superiority of the presented approach. However, data expansion occurs in the binary image encryption procedure, i.e., the generated ciphertext binary images are larger than the original binary image. In the future, the MRDHCBI approach without data expansion deserves investigation.

REFERENCES

[1] R. Cogranne, E. Giboulot, and P. Bas, "Efficient steganography in JPEG images by minimizing performance of optimal detector," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 1328–1343, 2022.

[2] B. Tondi, A. Costanzo, and M. Barni, "Robust and large-payload DNN watermarking via fixed, distribution-optimized, weights," *IEEE Trans. Dependable Secure Comput.*, vol. 22, no. 2, pp. 1082–1097, Mar./Apr. 2025.

[3] X. Liang and S. Xiang, "General distortion metric based histogram shifting for reversible data hiding," *IEEE Signal Process. Lett.*, vol. 31, pp. 2420–2424, 2024.

[4] A. K. Saini and S. Singh, "HSB based reversible data hiding using sorting and pairwise expansion," *J. Inf. Secur. Appl.*, vol. 80, 2024, Art. no. 103663.

[5] Y. Bai, G. Jiang, Z. Zhu, H. Xu, and Y. Song, "Reversible data hiding scheme for high dynamic range images based on multiple prediction error expansion," *Signal Process., Image Commun.*, vol. 91, 2021, Art. no. 116084.

[6] Y.-A. Ho, Y.-K. Chan, H.-C. Wu, and Y.-P. Chu, "High-capacity reversible data hiding in binary images using pattern substitution," *Comput. Standards Interfaces*, vol. 31, no. 4, pp. 787–794, 2009.

[7] K. Dong, H. J. Kim, Y. S. Choi, S. H. Joo, and B. H. Chung, "Reversible binary image watermarking method using overlapping pattern substitution," *ETRI J.*, vol. 37, no. 5, pp. 990–1000, 2015.

[8] F. Zhang, W. Lu, H. Liu, Y. Yeung, and Y. Xue, "Reversible data hiding in binary images based on image magnification," *Multimedia Tools Appl.*, vol. 78, no. 15, pp. 21891–21915, 2019.

[9] C. Kim, J. Baek, and P. S. Fisher, "Lossless data hiding for binary document images using n-pairs pattern," in *Proc. Int. Conf. Inf. Secur. Cryptol.–ICISC 2014*, 2015, pp. 317–327.

[10] X. Yin, W. Lu, J. Zhang, and W. Liu, "Reversible data hiding in halftone images based on minimizing the visual distortion of pixels flipping," *Signal Process.*, vol. 173, 2020, Art. no. 107605.

[11] X. Yin, W. Lu, J. Zhang, J. Chen, and W. Liu, "Reversible data hiding in binary images by flipping pattern pair with opposite center pixel," *J. Vis. Commun. Image Representation*, vol. 70, 2020, Art. no. 102816.

[12] H. Ren, W. Lu, and B. Chen, "Reversible data hiding in encrypted binary images by pixel prediction," *Signal Process.*, vol. 165, pp. 268–277, 2019.

[13] F. Li, L. Zhang, and W. Wei, "Reversible data hiding in encrypted binary image with shared pixel prediction and halving compression," *EURASIP J. Image Video Process.*, vol. 2020, no. 1, pp. 33–53, 2020.

[14] L. Zhang, F. Li, and C. Qin, "Efficient reversible data hiding in encrypted binary image with Huffman encoding and weight prediction," *Multimedia Tools Appl.*, vol. 81, no. 20, pp. 29347–29365, 2022.

[15] M. Naor and A. Shamir, "Visual cryptography," in *Proc. Adv. Cryptol.–EUROCRYPT'94*, 1995, pp. 1–12.

[16] B. Feng, W. Lu, and W. Sun, "Secure binary image steganography based on minimizing the distortion on the texture," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 2, pp. 243–255, Feb. 2015.