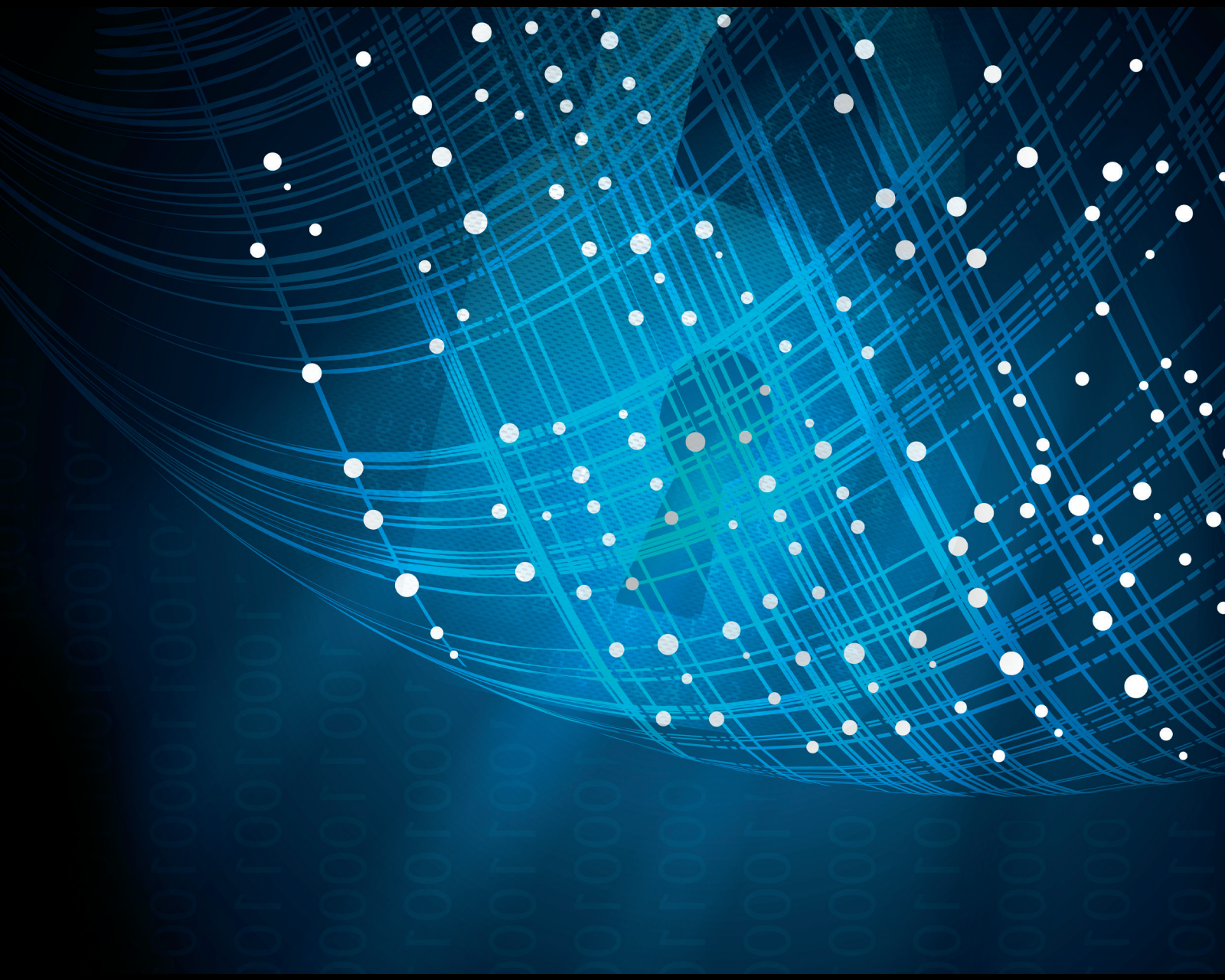
Security and Communication Networks

Multimedia Security: Novel Steganography and Privacy Preserving

Lead Guest Editor: Zhenxing Qian
Guest Editors: Kim-Kwang R. Choo, Rémi Cogranne, and Xinpeng Zhang



Multimedia Security: Novel Steganography and Privacy Preserving

Security and Communication Networks

Multimedia Security: Novel Steganography and Privacy Preserving

Lead Guest Editor: Zhenxing Qian
Guest Editors: Kim-Kwang R. Choo, Rémi Cogranne, and Xinpeng Zhang

Editorial Board

Contents

Hindawi
Security and Communication Networks
Volume 2018, Article ID 6390945, 2 pages
https://doi.org/10.1155/2018/6390945

Editorial

Multimedia Security: Novel Steganography and Privacy Preserving

Zhenxing Qian,[1] **Kim-Kwang Raymond Choo**,[2] **Rémi Cogranne**,[3] **and Xinpeng Zhang**[4]

[1]*Shanghai Institute for Advanced Communication and Data Science, School of Communication and Information Engineering, Shanghai University, Shanghai, China*
[2]*Department of Information Systems and Cyber Security, University of Texas, San Antonio, San Antonio, TX 78249, USA*
[3]*Laboratory of Systems Modeling and Dependability, Systems, Networks & Telecommunications, Troyes University of Technology, Troyes, France*
[4]*Shanghai Institute of Intelligent Electronics & Systems, School of Computer Science, Fudan University, Shanghai, China*

Correspondence should be addressed to Zhenxing Qian; zxqian@shu.edu.cn

Received 26 August 2018; Accepted 26 August 2018; Published 9 September 2018

Multimedia security is not a new research topic, as there have been a large body of work on various aspects of multimedia security. However, there remain a number of open research challenges, partly due to advances in multimedia and other related consumer technologies, for example, threats to conventional steganography due to emerging machine/deep learning-based steganalysis approaches. Thus, there is a need to design steganography approaches to mitigate such steganalysis efforts. Another ongoing challenge is how to ensure the privacy of multimedia data and the processing of such data, given constant advances in computational capabilities and big data analytics.

In this special issue, we aim to provide readers with a broad overview of multimedia security, such as novel steganography, privacy preserving on cloud computing, and multimedia tampering detection.

To achieve covert transmission, one could implement steganography in a cover. G. Swain (in "High Capacity Image Steganography using Modified LSB Substitution and PVD against Pixel Difference Histogram Analysis") demonstrated how we can expand the embedding capacity while mitigating steganalysis efforts. Another article on steganography was presented by J. He et al. ("A Novel AMR-WB Speech Steganography Based on Diameter-Neighbor Codebook Partition"). The authors used speech signals as covers to realize covert transmissions.

In addition to the use of covers, other approaches such as those based on images and audios can also be used to transmit secret data. For example, P. Cao et al. (in "A Wireless Covert Channel Based on Constellation Shaping Modulation") developed a system to transmit secret data over wireless channels, based on constellation shaping modulation. With constant advances in steganography solutions, X. ShuangKui et al. (in "A Modification-Free Steganography Method Based on Image Information Entropy") investigated the possibility of transmitting secret data in big data. Since secret data are directly mapped to the entropies of the covers, no modification is required when transmitting a single cover. For steganography, D. Hu et al. (in "Adaptive Steganalysis Based on Selection Region and Combined Convolutional Neural Networks") explained how to mitigate adaptive steganography by analyzing the selected regions of the covers using deep learning.

In terms of privacy preservation, a number of articles in this special issue focused on achieving privacy in the cloud computing environment. For example, during outsourcing of computations task from the user(s) to the cloud, Y. Ren et al. (in "Noninteractive Verifiable Outsourcing Algorithm for Bilinear Pairing with Improved Checkability") and H. Zhu et al. (in "Outsourcing Set Intersection Computation Based on Bloom Filter for Privacy Preservation in Multimedia Processing") presented two outsourcing protocols for bilinear

pairing and set intersection computations, respectively. The authors claimed that both protocols were more efficient and secure than prior work.

Multimedia indexing in encrypted domain is also another popular topic in privacy preservation. Hence, H. Liang et al. (in "Secure and Efficient Image Retrieval over Encrypted Cloud Data") proposed an efficient ciphertext retrieval algorithm, using balanced index tree and partial encryption. On the topic of labeling encrypted data in the cloud, D. Xu et al. (in "Separable Reversible Data Hiding in Encrypted Images Based on Two-Dimensional Histogram Modification") and X. Chen et al. (in "Improved Encrypted-Signals-Based Reversible Data Hiding Using Code Division Multiplexing and Value Expansion") proposed two reversible data hiding protocols for encrypted images, respectively. The use of watermarking to protect user privacy in the cloud environment was demonstrated by K. Liu et al. (in "A Cloud-User Protocol Based on Ciphertext Watermarking Technology").

In this special issue, multimedia forensics and network security were also discussed. For example, Y. Sun et al. (in "Nonoverlapping Blocks Based Copy-Move Forgery Detection"), H. Wang et al. (in "Perceptual Hashing-Based Image Copy-Move Forgery Detection"), and D. Niu et al. (in "Reference Sharing Mechanism-based Self-embedding Watermarking Scheme with Deterministic Content Reconstruction"), respectively, studied passive forensics, copy-move forgery detection, and watermarking. On the topics of network attack, R. Zhang et al. in ("Constructing APT Attack Scenarios Based on Intrusion Kill Chain and Fuzzy Clustering") presented an approach to simulate advanced persistent threat (APT) attack scenarios, and J. Chen (in "A Survey on Breaking Technique of Text-based CAPTCHA") surveyed existing approaches to circumventing CAPTCHA.

In conclusion, while the breadth and depth of the articles in this issue have contributed to the knowledge gap in multimedia security, many other challenges remain. It is hoped that the advances reported in this special issue will inspire new areas of research in the near future.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Zhenxing Qian
Kim-Kwang Raymond Choo
Rémi Cogranne
Xinpeng Zhang

Hindawi
Security and Communication Networks
Volume 2018, Article ID 1505896, 14 pages
https://doi.org/10.1155/2018/1505896

Research Article

High Capacity Image Steganography Using Modified LSB Substitution and PVD against Pixel Difference Histogram Analysis

Gandharba Swain

Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh 522502, India

Correspondence should be addressed to Gandharba Swain; gswain1234@gmail.com

Received 5 July 2017; Revised 28 August 2017; Accepted 7 September 2017; Published 4 September 2018

Academic Editor: Zhenxing Qian

The combination of pixel value differencing (PVD) and least significant bit (LSB) substitution gives higher capacity and lesser distortion. However, there are three issues to be taken into account: (i) fall off boundary problem (FOBP), (ii) pixel difference histogram (PDH) analysis, and (iii) RS analysis. This paper proposes a steganography technique in two variants using combination of modified LSB substitution and PVD by taking care of these three issues. The first variant operates on 2×3 pixel blocks and the second technique operates on 3×3 pixel blocks. In one of the pixels of a block, embedding is performed using modified LSB substitution. Based on the new value of this pixel, difference values with other neighboring pixels are calculated. Using these differences, PVD approach is applied. The edges in multiple directions are exploited, so PDH analysis cannot detect this steganography. The LSB substitution is performed in only one pixel of the block, so RS analysis also cannot detect this steganography. To address the FOBP, suitable equations are used during embedding procedure. The experimental results such as bit rate and distortion measure are satisfactory.

1. Introduction

Least significant bit (LSB) substitution is a very old image steganography approach, wherein the LSB bits (one, two, three, or four) of the pixels are substituted by secret data bits. This simplest technique is detected by RS analysis. Wu and Tsai [1] exposed the fact that the edge regions in an image can conceal more amounts of data as compared to the smooth regions. Based on this principle they proposed pixel value differencing (PVD) steganography. The image should be partitioned into different blocks, each of size 1×2 pixels. For a block the difference value between the two pixels is computed and changed to a new value by hiding data in it. The PVD technique with block size 2×2 has been proposed to enhance the embedding capacity [2, 3]. In blocks of size 2×2, edges in three directions are considered. Chang and Tseng [4] considered the values of 2, 3, and 4 neighboring pixels, to calculate the pixel value differences. However, they did not address the fall in error problem (FIEP). Yang et al. [5] calculated varieties of pixel value differences in 4 pixel blocks for data hiding. Hong et al. [6] used diamond encoding with pixel value differencing to achieve better peak signal-to-noise ratio (PSNR). LSB substitution techniques offer higher embedding capacity, but PVD techniques offer higher imperceptibility. Thus PVD and LSB substitution techniques have been combined to obtain larger hiding capacity and better imperceptibility [7, 8]. Chen [9] proposed a PVD steganography using two reference tables to randomize the data embedding. Based on pixel value differences, adaptive LSB substitution has been performed in [10]. Khodaei and Faez [11] proposed a hybrid approach by combining LSB substitution and PVD in 1×3 pixel block. It is extended to 2×2 size block in [12] and 2×3 size block in [13] to improve the performance.

The traditional PVD steganography techniques follow a static range table. Due to this, the pixel difference histogram of the stego-image becomes zig-zag in nature. This is called step effect. This step effect can be avoided by applying two tricks: (i) exploiting the edges in multiple directions and (ii) introducing adaptive range table. Luo et al. [14] also

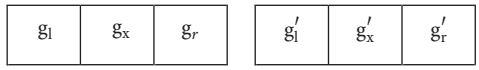


Figure 1: Original and stego-blocks of size 1 × 3.

proposed an adaptive PVD steganography with three pixel blocks, which does not suffer with the step effect. Swain [15] proposed two adaptive PVD steganography techniques using vertical, horizontal, and diagonal edges, which does not suffer with step effect. The first technique uses pixel blocks of size 2 × 2, and the second technique uses pixel blocks of size 3 × 3. In general, adaptive image steganography schemes possess lower embedding capacity. Anita et al. [16] optimized the performance of adaptive PVD by using 6 pixel blocks. The edges can be predicted by some prediction functions, and hiding capacity depends upon this prediction. If we hide bits of data in smooth regions, distortion will be more. Based on the level of complexity of the edge regions, adaptive embedding can be applied [17]. In this way capacity can be increased and chance of detection can be decreased. Balasubramanian et al. [18] proposed a PVD scheme with 3 × 3 size pixel blocks, to achieve higher hiding capacity. To prevent the detection from pixel difference histogram (PDH) analysis, multidirectional edges have been exploited in [19]. Darabkh et al. [20] also proposed PVD steganography using eight directional PVD which is an extension in principle from Wu and Tsai's original PVD. Any PVD technique which is an extension of Wu and Tsai's approach should qualify through PDH analysis.

Exploiting modification direction (EMD) steganography was initiated by Zhang and Wang [21], wherein the main idea was to convert a group of bits to a digit in (2n + 1)-ary notational system and hide it in a pixel of the block. The hiding capacity of this technique is very poor. It has been improved by Kim [22] using $(2^{n+x} - 1)$-ary notational system, where n and x are user defined values. Shen and Huang [23] combined PVD with EMD to achieve higher hiding capacity and better PSNR.

This paper proposes a combination of modified LSB substitution (M-LSB) and PVD. It is judiciously designed in such a manner that the fall off boundary problem (FOBP) does not arise and neither PDH analysis nor RS analysis can detect it. There are two main contributions in this paper: (i) discovering the FOBP that exists in Khodaei and Faez's [11] technique (ii) and addressing it by proposing an improved technique with larger block size.

2. Related Work

Khodaei and Faez [11] in 2012 proposed a steganography method using both PVD and LSB substitution. The embedding procedure of this technique is as described below. The image is divided into nonoverlapping blocks of size 1 × 3; i.e., a block comprises 3 pixels from one row and three adjacent columns of the image, as shown in Figure 1(a). Image is viewed as a two-dimensional matrix of pixels/bytes. On

Table 1: Range table for type 1.

Range	[0, 7]	[8, 15]	[16, 31]	[32, 63]	[64, 255]
capacity	3	3	3	4	4

Table 2: Range table for type 2.

Range	[0, 7]	[8, 15]	[16, 31]	[32, 63]	[64, 255]
capacity	3	3	4	5	6

middle pixel g_x, k bit LSB substitution is applied. The k can be chosen from 3, 4, or 5. An ideal value for k will be 3. After k bit LSB substitution, g_x becomes g'_x. Suppose the decimal value of k LSBs of g_x is L and the decimal value of k LSBs of g'_x is S. The deviation d is equal to L-S. Now g'_x is optimized by (1).

$$g'_x = \begin{cases} g'_x + 2^k, & \text{if } d > 2^{k-1} \text{ and } 0 \le g'_x + 2^k \le 255 \\ g'_x - 2^k, & \text{if } d < -2^{k-1} \text{ and } 0 \le g'_x - 2^k \le 255 \\ g'_x, & \text{otherwise} \end{cases} \quad (1)$$

This g'_x value is final. Two difference values $d_1 = |g'_x - g_l|$ and $d_2 = |g'_x - g_r|$ are calculated. Table 1 is the range table for type 1 and Table 2 is the range table for type 2.

The value d_1 belongs to one of the ranges in range table whose lower bound is l_1 and embedding capacity is t_1. From the secret data stream t_1 bits are taken and its decimal equivalent is s_1. To hide t_1 bits of data in g_l, the new value for d_1 is d'_1. Similarly, the value d_2 belongs to a range in the range table whose lower bound is l_2 and embedding capacity is t_2. From the secret data stream t_2 bits are taken and its decimal equivalent is s_2. To hide t_2 bits of data in g_2, the new value for d_2 is d'_2. These new difference values, d'_1 and d'_2, are calculated using (2).

$$\begin{aligned} d'_1 &= l_1 + s_1, \\ d'_2 &= l_2 + s_2 \end{aligned} \quad (2)$$

Two new values of g_l and two new values for g_r are calculated as below using (3).

$$\begin{aligned} g''_l &= g'_x - d'_1, \\ g'''_l &= g'_x + d'_1, \\ g''_r &= g'_x - d'_2, \\ g'''_r &= g'_x + d'_2 \end{aligned} \quad (3)$$

Now, g'_l, the final value of g_l, is calculated using (4). Similarly, the final value of g_r known as g'_r is calculated using (5).

$$g'_l = \begin{cases} g''_l, & \text{if } \left|g_l - g''_l\right| < \left|g_l - g'''_l\right| \text{ and } 0 \le g''_l \le 255 \\ g'''_l, & \text{otherwise} \end{cases} \quad (4)$$

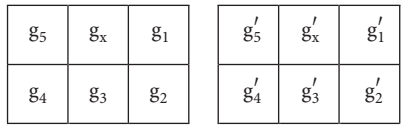


Figure 2: Original and stego-blocks of size 2 × 3.

$$g'_r = \begin{cases} g''_r, & \text{if } \left|g_r - g''_r\right| < \left|g_r - g'''_r\right| \text{ and } 0 \le g''_r \le 255 \\ g'''_r, & \text{otherwise} \end{cases} \quad (5)$$

Thus the block of Figure 1(a) is converted to its stego-block as shown in Figure 1(b).

The extraction of the secret embedded data is done by the procedure as described below. The image is partitioned into blocks as was done in embedding procedure. A stego-block is as shown in Figure 1(b). From the middle pixel g'_x, k LSBs are extracted. Two differences d'_1 and d'_2 are calculated using (6).

$$\begin{aligned} d'_1 &= \left|g'_l - g'_x\right|, \\ d'_2 &= \left|g'_r - g'_x\right| \end{aligned} \quad (6)$$

The difference d'_1 belongs to a range in the range table whose embedding capacity is t_1 and lower bound is l_1. Similarly, the difference d'_2 belongs to a range in the range table whose embedding capacity is t_2 and lower bound is l_2. Note that out of two range tables, the one that was used during embedding is the same one that should be used during extraction. The decimal equivalents of secret bit streams embedded in g'_l and g'_r are s_1 and s_2, respectively, as in (7).

$$\begin{aligned} s_1 &= d'_1 - l_1, \\ s_2 &= d'_2 - l_2 \end{aligned} \quad (7)$$

Finally, s_1 and s_2 are converted to t_1 and t_2 binary bits, respectively. These are the bits actually hidden.

3. Proposed Technique

3.1. Proposed Variant 1: Five Directional PVD with Modified LSB Substitution. The image is divided into nonoverlapping blocks of size 2 × 3; i.e., a block comprises 6 pixels from two adjacent rows and three adjacent columns of the image, as shown in Figure 2(a). The image is viewed as a two-dimensional matrix of pixels/bytes. On pixel g_x, k bit LSB substitution is applied. The k value can be chosen from 3, 4, or 5. An ideal value for k is 3. After k bit LSB substitution g_x becomes g'_x. Suppose the decimal values of k LSBs of g_x is L and the decimal value of k LSBs of g'_x is S. The deviation d is equal to L-S. Now g'_x is optimized using (1). This g'_x value is final.

For i = 1 to 5, five difference values $d_i = |g'_x - g_i|$ are calculated. Table 1 is the range table for type 1 and Table 2 is the range table for type 2. The value d_i belongs to a range in the range table whose lower bound is l_i and embedding capacity is t_i. For i = 1 to 5, take t_i binary bits from secret data stream and convert it to decimal value s_i. Now, for i = 1 to 5 calculate new difference values using (8).

$$d'_i = l_i + s_i \quad (8)$$

For i = 1 to 5, for each g_i, two new values, namely, g''_i and g'''_i, are calculated using (9).

$$\begin{aligned} g''_i &= g'_x - d'_i, \\ g'''_i &= g'_x + d'_i \end{aligned} \quad (9)$$

Now, for i = 1 to 5, the stego-value of g_i is known as g'_i. It is calculated using (10).

$$g'_i = \begin{cases} g'''_i, & \text{if } g''_i < 0 \\ g''_i, & \text{if } g'''_i > 255 \\ g''_i, & \text{if } \left|g_i - g''_i\right| < \left|g_i - g'''_i\right| \text{ and } g''_i \ge 0 \text{ and } g'''_i \le 255 \\ g'''_i, & \text{otherwise} \end{cases} \quad (10)$$

Thus the block of Figure 2(a) is converted to its stego-block as shown in Figure 2(b).

The extraction of the secret embedded data is done by the procedure as described below. The image is partitioned into blocks as was done in embedding procedure. A stego-block is shown in Figure 2(b). From the pixel g'_x, the k least significant bits are extracted. For i = 1 to 5, five difference values are calculated using (11).

$$d'_i = \left|g'_x - g'_i\right| \quad (11)$$

The difference d'_i belongs to a range in the range table whose embedding capacity is t_i and lower bound is l_i. Note that out of two range tables, the one that was used during embedding is the same one that should be used during extraction. The decimal equivalents of secret bit streams embedded in g'_i is s_i. It is calculated using (12).

$$s_i = d'_i - l_i \quad (12)$$

Finally, for i = 1 to 5, each s_i is converted to t_i binary bits. These bits are the embedded data.

3.2. Example of Embedding and Extraction for Proposed Technique-Variant 1 and FOBP. To understand the embedding procedure, let us consider the original block in Figure 3(a). Suppose the secret data to be embedded is 100 1111 101 000 001 001. Let us choose the k value as 3 and Table 1 as the range table. The value of g_x is 181. The values of g_1, g_2, g_3, g_4, and g_5 are 255, 200, 190, 192, and 182, respectively. The g_x in binary is $10110\mathbf{101}_2$. Take 3 bits of data, i.e., 100 from the secret data stream and embed in g_x using LSB substitution. The new binary value is $10110\mathbf{100}_2$ and its decimal equivalent

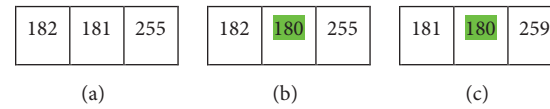
182	181	255
192	190	200

(a) Original block

181	180	101
189	188	201

(b) Stego-block

Figure 3: Examples of 2 × 3 original and stego-blocks.



(a) (b) (c)

Figure 4: Embedding example in 1 × 3 blocks.

is 180; thus g'_x is 180. The decimal value of 3 LSBs of g_x is 5 and the decimal value of 3 LSBs of g'_x is 4. The deviation d is equal to 5-4 = 1. Now g'_x is optimized using (1). Hence final value of g'_x after optimization is 180.

Now the five difference values are $d_1 = |180 - 255| = 75$, $d_2 = |180-200| = 20$, $d_3 = |180-190| = 10$, $d_4 = |180-192| = 12$, and $d_5 = |180-182| = 2$. The value d_1 belongs to the range [64, 255], so $t_1 = 4$ and $l_1 = 64$. The value d_2 belongs to the range [16, 31], so $t_2 = 3$ and $l_2 = 16$. Similarly, $t_3 = 3$, $l_3 = 8$, $t_4 = 3$, $l_4 = 8$, $t_5 = 3$, and $l_5 = 0$. Take t_1, t_2, t_3, t_4, and t_5 binary bits from secret data stream and convert them to decimal values s_1, s_2, s_3, s_4, and s_5, respectively. Hence $s_1 = 15$, $s_2 = 5$, $s_3 = 0$, $s_4 = 1$, and $s_5 = 1$.

Using (8), $d'_1 = 64 + 15 = 79$, $d'_2 = 16+5 = 21$, $d'_3 = 8 + 0 = 8$, $d'_4 = 8 + 1 = 9$, and $d'_5 = 0 + 1 = 1$. Using (9), $g''_1 = 180 - 79 = 101$, $g'''_1 = 180 + 79 = 259$, $g''_2 = 180 - 21 = 159$, $g'''_2 = 180+21 = 201$, $g''_3 = 180-8 = 172$, $g'''_3 = 180+8 = 188$, $g''_4 = 180 - 9 = 171$, $g'''_4 = 180 + 9 = 189$, $g''_5 = 180 - 1 = 179$, and $g'''_5 = 180 + 1 = 181$.

Using (10), $g'_1 = 101$, $g'_2 = 201$, $g'_3 = 188$, $g'_4 = 189$, and $g'_5 = 181$. Hence the stego-block is as shown in Figure 3(b).

To understand the extraction procedure, let us consider the stego-block in Figure 3(b). The value of g'_x is 180 and in binary is $10110\mathbf{100}_2$; the three LSBs, 100_2, are extracted. Using (11), $d'_1 = |180 - 101| = 79$, $d'_2 = |180 - 201| = 21$, $d'_3 = |180-188| = 8$, $d'_4 = |180-189| = 9$, and $d'_5 = |180-181| = 1$. The value d'_1 belongs to [64, 255], so $t_1 = 4$ and $l_1 = 64$. The value d'_2 belongs to [16, 31], so $t_2 = 3$, $l_2 = 16$. Similarly, $t_3 = 3$, $l_3 = 8$, $t_4 = 3$, $l_4 = 8$, $t_5 = 3$, and $l_5 = 0$.

Using (12), $s_1 = 15$, $s_2 = 5$, $s_3 = 0$, $s_4 = 1$, and $s_5 = 1$. Converting s_1, s_2, s_3, s_4, and s_5 to t_1, t_2, t_3, t_4, and t_5 binary bits, we get 1111 101 000 001 001. From g'_x, we had extracted 100_2. Thus the total data extracted is 100 1111 101 000 001 001.

Fall of Boundary Problem (FOBP). A pixel of a gray image is represented in 8 bits. The decimal value of a pixel falls in the range {0, 255}. If a stego-pixel value is less than zero or greater than 255, then we declare that a FOBP has arisen. Khodaei and Faez's LSB+PVD approach is a very fantastic one and has higher embedding capacity and PSNR, but it suffers with FOBP.

Khodaei and Faez's Technique Suffers with FOBP. It can be understood from the following discussion. Let us consider a 1 × 3 pixel block as shown in Figure 4(a). The secret data stream to be embedded is 100 001 1111 000, etc. The middle pixel, i.e., g_x, is 181; its binary value in 8 bits is $10110\mathbf{101}_2$. The values of g_l and g_r are 182 and 255, respectively. Let us choose k = 3 and Table 1 as range table. The 3 bits of data from the secret data stream is 100. After applying 3 bit LSB substitution, the binary value of g_x changes to $10110\mathbf{100}_2$, which is equal to 180 in decimal. Thus the value of g'_x is 180. The three least significant bits of g_x are 101; in decimal it is 5. The three least significant bits of g'_x are 100; in decimal it is 4. Thus d = L-S = 5-4 = 1. By applying (1), the optimized value of g'_x is 180, as shown in Figure 4(b).

Now we calculate two difference values $d_1 = |g'_x - g_l| = |180 - 182| = 2$ and $d_2 = |180 - 255| = 75$. The difference d_1 belongs to the range {0, 7}, so l_1 and t_1 value are 0 and 3, respectively. The next 3 bits of data from the secret data stream are 001; its decimal value is 1, i.e., $s_1 = 1$. The new difference d'_1 is 0 + 1 = 1, as per (2). The difference d_2 belongs to the range {64, 255}, so l_2 and t_2 value are 64 and 4, respectively. The next 4 bits of data from the secret data stream are 1111; its decimal value is 15, i.e., $s_2 = 15$. The new difference d'_2 is 64 + 15 = 79, as per (2).

Now by referring to (3), $g''_l = g'_x - d'_1 = 180 - 1 = 179$, $g'''_l = g'_x + d'_1 = 180 + 1 = 181$, $g''_r = g'_x - d'_2 = 180 - 79 = 101$, and $g'''_r = g'_x + d'_2 = 180 + 79 = 259$. By referring to (4) and (5) g'_l and g'_r are calculated. This is shown as (13) and (14), respectively.

$$g'_l = \begin{cases} 179, & \text{if } |182 - 179| < |182 - 181| \text{ and } 0 \le 179 \le 255 \\ 181, & \text{otherwise} \end{cases} \quad (13)$$

$$= 181$$

$$g'_r = \begin{cases} 101, & \text{if } |255 - 101| < |255 - 259| \text{ and } 0 \le 101 \le 255 \\ 259, & \text{otherwise} \end{cases} \quad (14)$$

$$= 259$$

Hence the $g'_l = 181$ and $g'_r = 259$. The stego-block is as shown in Figure 4(c). A FOBP occurred. Thus the embedding procedure of Khodaei and Faez's technique needs some modification.

3.3. Proposed Variant 2: Eight Directional PVD with Modified LSB Substitution. The image is divided into nonoverlapping blocks of size 3 × 3; i.e., a block comprises 9 pixels from three adjacent rows and three adjacent columns of the image, as shown in Figure 5(a). Image is viewed as a two-dimensional matrix of pixels/bytes. On pixel g_x, k bit LSB substitution is applied. The k value can be chosen from 3, 4, or 5. An ideal

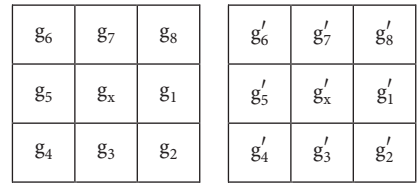


Figure 5: Original and stego-blocks of size 3 × 3.

value for k will be 3. After k bit LSB substitution g_x becomes g'_x. Suppose the decimal value of k LSBs of g_x is L and the decimal value of k LSBs of g'_x is S. The deviation d is equal to L-S. Now g'_x is optimized using (1). This g'_x value is final.

For i = 1 to 8, eight difference values $d_i = |g'_x - g_i|$ are calculated. Table 1 is the range table for type 1 and Table 2 is the range table for type 2. The value d_i belongs to a range in the range table whose lower bound is l_i and embedding capacity is t_i. For i = 1 to 8, take t_i binary bits from secret data stream and convert it to decimal value s_i. Now, for i = 1 to 8 calculate new difference values using (15).

$$d'_i = l_i + s_i \tag{15}$$

For i = 1 to 8, for each g_i, two new values, namely, g''_i and g'''_i, are calculated using (16).

$$\begin{aligned} g''_i &= g'_x - d'_i, \\ g'''_i &= g'_x + d'_i \end{aligned} \tag{16}$$

Now, for i = 1 to 8, g'_i, the stego-value of g_i is calculated using (17).

$$g'_i = \begin{cases} g'''_i, & \text{if } g''_i < 0 \\ g''_i, & \text{if } g'''_i > 255 \\ g''_i, & \text{if } \left|g_i - g''_i\right| < \left|g_i - g'''_i\right| \text{ and } g''_i \geq 0 \text{ and } g'''_i \leq 255 \\ g'''_i, & \text{otherwise} \end{cases} \tag{17}$$

Thus the block of Figure 5(a) is converted to its stego-block as shown in Figure 5(b).

The extraction of hidden data is done by the procedure described below. From the stego-image blocks are formed as in embedding procedure. A stego-block is as shown in Figure 5(b). From the pixel g'_x, k least significant bits are extracted. For i = 1 to 8, eight difference values are calculated using (18).

$$d'_i = \left|g'_x - g'_i\right| \tag{18}$$

The difference d'_i belongs to a range in the range table whose embedding capacity is t_i and lower bound is l_i. Note that out of two range tables, the one that was used during embedding is the same one that should be used during extraction. The decimal equivalent of secret bit stream embedded in g'_i is s_i. It is calculated using (19).

$$s_i = d'_i - l_i \tag{19}$$

Finally, for i = 1 to 8, each s_i is converted to t_i binary bits. These bits are the embedded data.

4. Results and Discussion

The proposed technique (variant 1 and 2) is experimented using MATLAB programming. The test images are taken from SIPI database [26]. The eight sample images are shown in Figure 6. These are RGB color images of size 512 × 512. Each pixel consists of 3 bytes. Each byte is treated like a pixel for computation point of view. Figure 7 represents the stego-images of Lena and Baboon for proposed variant 1. Figure 8 represents the stego-images of Lena and Baboon for proposed variant 2. It can be observed from the stego-images that there is no distortion observable to be suspected as stego-images; they are as good as original images. In each of these stego-images 840000 bits (eight lakhs and forty thousand bits) of data is hidden.

The proposed technique is compared with the Khodaei and Faez's technique in terms of PSNR, hiding capacity, quality index (Q), and bits per byte (BPB). The formulae to measure these parameters are referred to in [27]. The number of cases of fall off boundary arisen during execution is recorded for these techniques. From Table 3 it can be observed that in both type 1 and type 2 of Khodaei and Faez's technique FOBP arises. Further from Tables 4 and 5 it can be observed that both the variants of proposed technique do not suffer with FOBP. In proposed technique the hiding capacity and bit rate are higher than those of Khodaei and Faez's technique, but PSNR is slightly lower than that of Khodaei and Faez's technique.

Furthermore the PSNR and hiding capacity of three closely related techniques discussed in [12, 13, 19] are recorded in Table 6. Techniques in [12, 13] are extension of Khodaei and Faez's [11] technique, so they suffer with FOBP. Technique in [19] is extension of Wu and Tsai's [1] PVD technique. It also suffers with FOBP. Although the PSNR values of these three techniques are greater than that of the proposed technique, the hiding capacity of the proposed technique is higher than these techniques.

The proposed technique has been compared with the techniques in [16, 24, 25] as shown in Table 7. The technique in [16] is an adaptive PVD technique with two variants. The hiding capacity of the proposed technique (in all the four variants) is greater than this technique. As the hiding capacity of the adaptive PVD technique is very less, PSNR is comparatively higher than the proposed technique. The hybrid technique in [24] is a mixture of PVD, LSB substitution, and

Table 3: Results of Khodaei and Faez's PVD technique [11].

Images 512 × 512 × 3	Type 1					Type 2				
	PSNR	Capacity	Q	bit rate	FOBP	PSNR	Capacity	Q	bit rate	FOBP
Lena	41.74	2375248	0.999	3.02	98	40.46	2434603	0.999	3.09	121
Baboon	37.27	2443361	0.998	3.10	12	34.19	2662080	0.996	3.38	26
Tiffany	38.7	2372396	0.997	3.01	12045	39.27	2416944	0.998	3.07	11994
Peppers	38.57	2372858	0.999	3.01	0	36.91	2435223	0.998	3.09	2
Jet	39.98	2374048	0.998	3.01	0	40.13	2418419	0.998	3.07	0
Boat	39.18	2391994	0.999	3.04	0	36.9	2504613	0.996	3.18	7
House	38.58	2387183	0.998	3.03	4	37.97	2470824	0.998	3.14	16
Pot	40.33	2366001	0.999	3.00	0	37	2387494	0.999	3.03	0
Average	39.29	2385386	0.998	3.03	1520	37.85	2466275	0.998	3.13	1521

Table 4: Results of proposed variant 1: five directional PVD with M-LSB substitution.

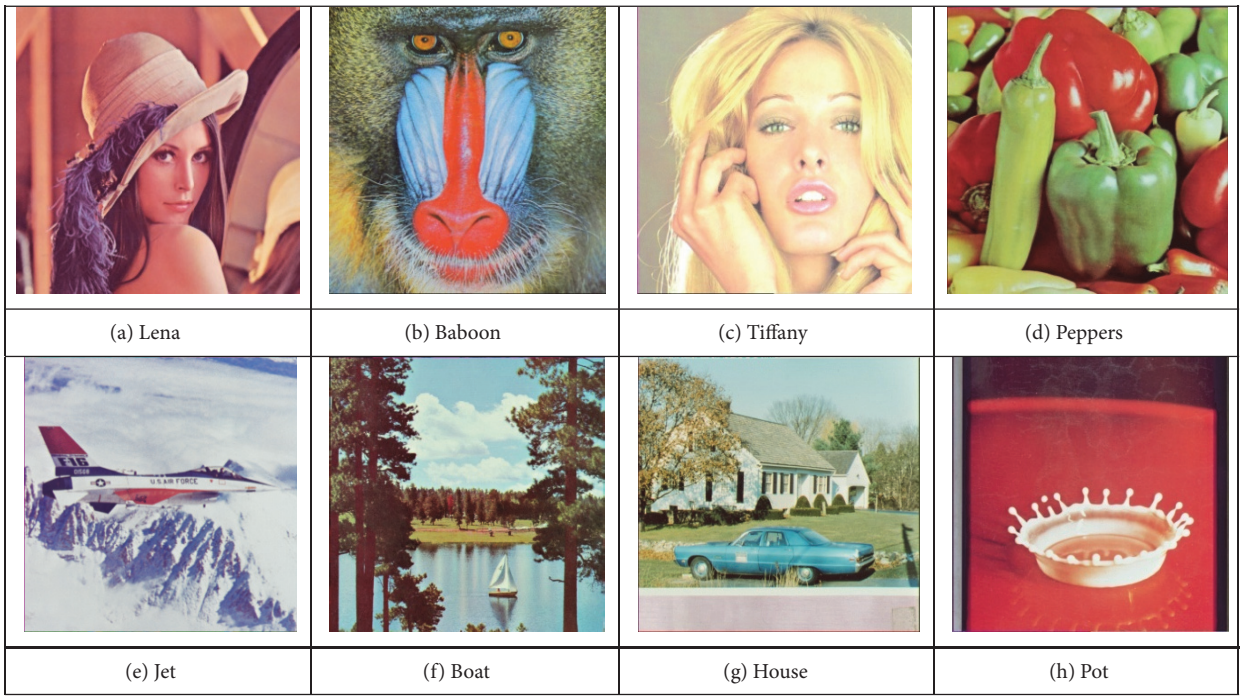
Images 512 × 512 × 3	Type 1					Type 2				
	PSNR	Capacity	Q	bit rate	FOBP	PSNR	Capacity	Q	bit rate	FOBP
Lena	41.14	2379222	0.999	3.03	0	39.58	2451046	0.999	3.12	0
Baboon	34.35	2496404	0.996	3.17	0	31.72	2816768	0.993	3.58	0
Tiffany	37.27	2376135	0.996	3.02	0	35.74	2432745	0.995	3.09	0
Peppers	35.56	2379292	0.998	3.03	0	35.45	2459762	0.998	3.13	0
Jet	39.57	2384676	0.998	3.03	0	39.73	2454060	0.998	3.12	0
Boat	35.84	2409235	0.998	3.06	0	34.16	2565007	0.997	3.26	0
House	36.9	2404582	0.998	3.06	0	36.46	2533362	0.997	3.22	0
Pot	40.35	2369274	0.999	3.01	0	36.96	2399556	0.999	3.05	0
Average	37.62	2399853	0.998	3.05	0	36.23	2514038	0.997	3.20	0

Table 5: Results of proposed variant 2: eight directional PVD with M-LSB substitution.

Images 512 × 512 × 3	Type 1					Type 2				
	PSNR	Capacity	Q	bit rate	FOBP	PSNR	Capacity	Q	bit rate	FOBP
Lena	41.04	2371255	0.999	3.02	0	39.38	2533551	0.999	3.22	0
Baboon	33.76	2502616	0.996	3.18	0	31.23	2939376	0.992	3.74	0
Tiffany	37.2	2366437	0.996	3.01	0	34.71	2511139	0.994	3.19	0
Peppers	35.15	2371837	0.998	3.02	0	34.1	2544392	0.997	3.24	0
Jet	39.52	2377031	0.998	3.02	0	39.2	2538801	0.998	3.23	0
Boat	35.32	2405009	0.998	3.06	0	33.17	2659795	0.997	3.38	0
House	37.58	2399229	0.998	3.05	0	36.66	2625804	0.998	3.34	0
Pot	40.34	2359271	0.999	3.00	0	36.33	2475977	0.999	3.15	0
Average	37.49	2394086	0.998	3.04	0	35.60	2603604	0.997	3.31	0

Table 6: Results of existing techniques [12, 13, 19].

Images 512 × 512 × 3	LSB + 3PVD technique [12]				LSB + 5PVD technique [13]				7 way PVD [19]	
	Type 1		Type 2		Type 1		Type 2			
	PSNR	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR	Capacity
Lena	42.22	2361875	40.83	2437700	41.84	2362944	39.98	2451046	40.81	1896662
Baboon	34.66	2393475	32.34	2772545	34.32	2396696	31.99	2816768	33.28	2226806
Tiffany	42.02	2363192	41.50	2425193	40.59	2363455	40.37	2432745	40.38	1400756
Peppers	39.27	2364428	38.04	2447737	38.16	2365366	36.50	2459762	39.63	1778072
Jet	41.81	2365839	41.90	2443492	41.26	2366486	40.62	2454060	40.36	1906254
Boat	38.04	2370147	36.13	2539530	37.10	2372127	34.76	2565007	37.14	1972086
House	39.66	2366686	38.73	2510373	38.59	2368303	37.51	2533362	37.83	1972223
Pot	42.39	2364360	41.17	2394782	40.26	2364549	36.95	2399556	41.81	1795551
Average	40.01	2368750	38.83	2496419	39.02	2369991	37.34	2514038	38.91	1868551



(a) Lena (b) Baboon (c) Tiffany (d) Peppers

(e) Jet (f) Boat (g) House (h) Pot

Figure 6: Original images.

Type 1 Type 2 Type 1 Type 2

Figure 7: Stego-images of proposed variant 1: five directional PVD with M-LSB substitution.

Type 1 Type 2 Type 1 Type 2

Figure 8: Stego-images of proposed variant 2: eight directional PVD with M-LSB substitution.

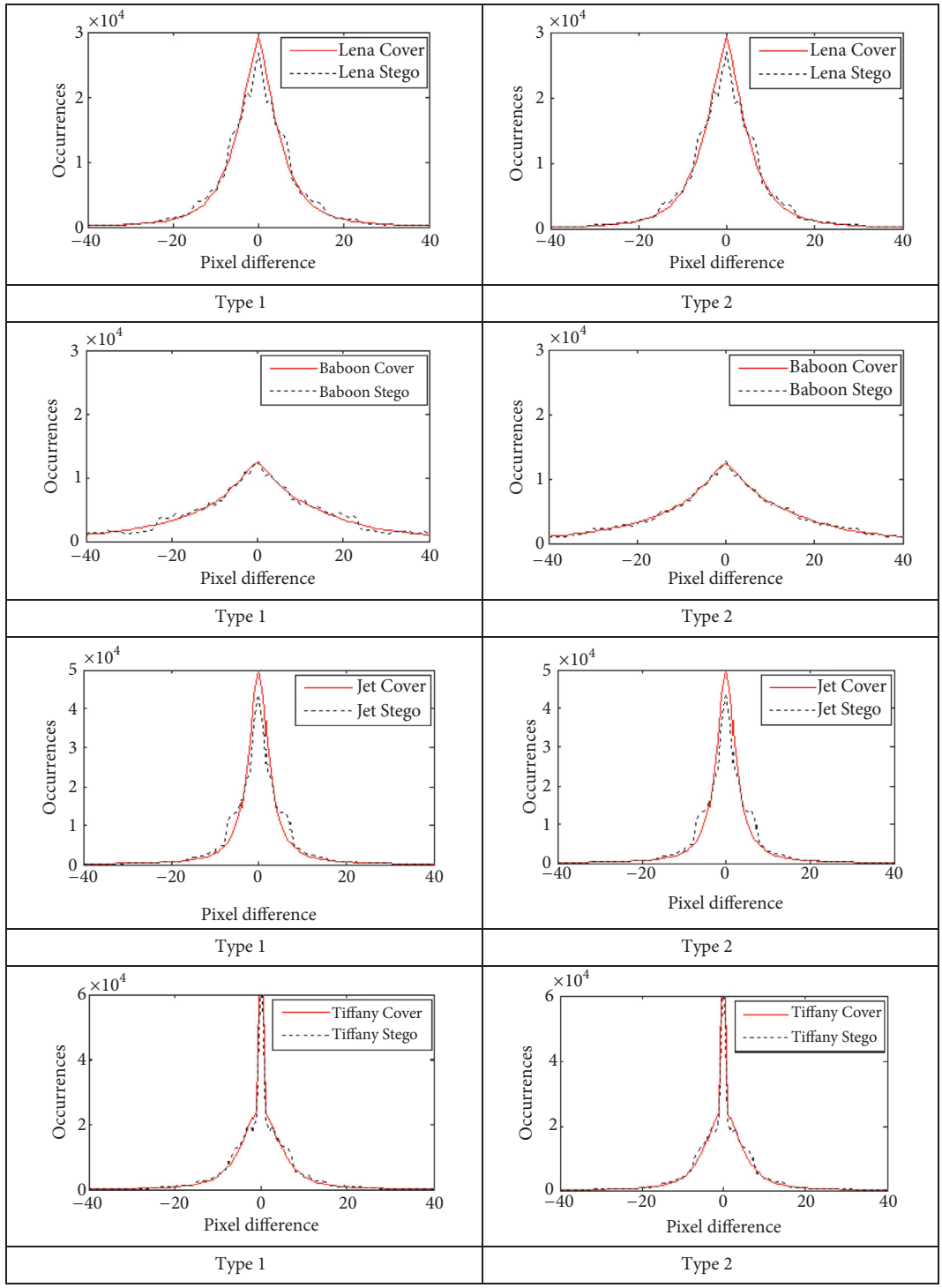


Figure 9: PDH analysis of Khodaei and Faez's PVD technique.

EMD. All the four variants of this technique possess lesser hiding capacity as compared to the proposed technique. In [25] the two variants of eight directional PVD technique have been discussed. The PSNR and hiding capacity of the proposed technique (type 1) are as good as those of the technique in [25] (type 1). Similarly, the PSNR and hiding capacity of the proposed technique (type 2) are as good as those of the technique in [25] (type 2).

The proposed technique (variant 1 and variant 2) uses LSB substitution and PVD, so it needs to be analyzed by both RS analysis and PDH analysis. Figure 9 represents the PDH analysis using Lena, Baboon, Jet, and Tiffany images for type

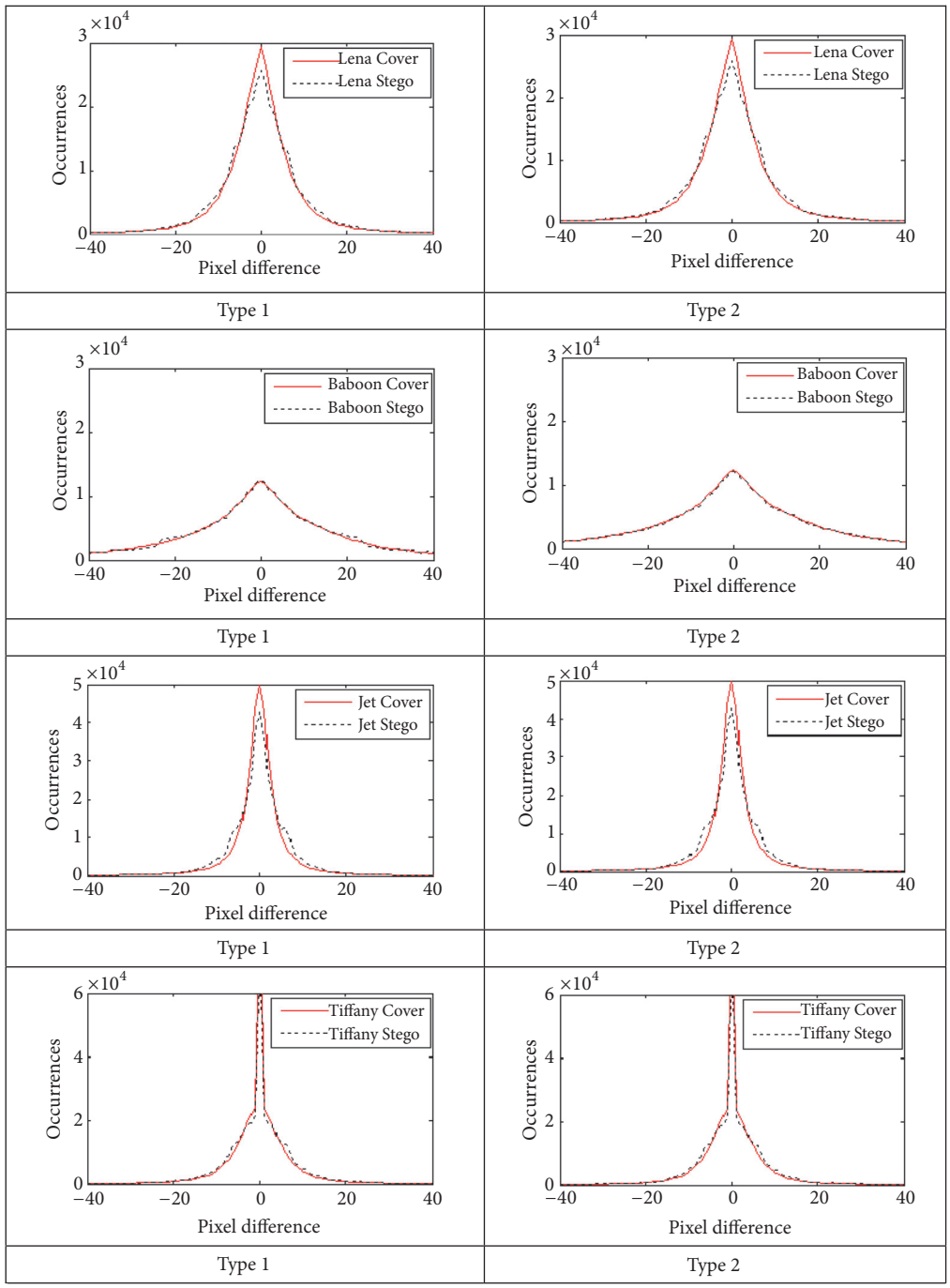


Figure 10: PDH analysis of proposed variant 1: five directional PVD with M-LSB substitution.

1 and type 2 of Khodaei and Faez's technique. There are eight subfigures in it. Each subfigure represents two curves. The solid line curve is for the original image and the dotted line curve is for the stego-image. The curve of original image will be smooth in nature, i.e., free from zig-zag appearance. This zig-zag appearance is called step effect. For Lena image the step effect is clearly identified, and for the other three images it is slightly identified. For the proposed variant 1 the PDH analysis is shown in Figure 10. The step effects are reduced as compared to that of Khodaei and Faez's technique. For the proposed variant 2 the PDH analysis is shown in Figure 11. The step effects do not exist at all.

The RS analysis of Khodaei and Faez's technique and the proposed technique is shown in Figures 12–14, using

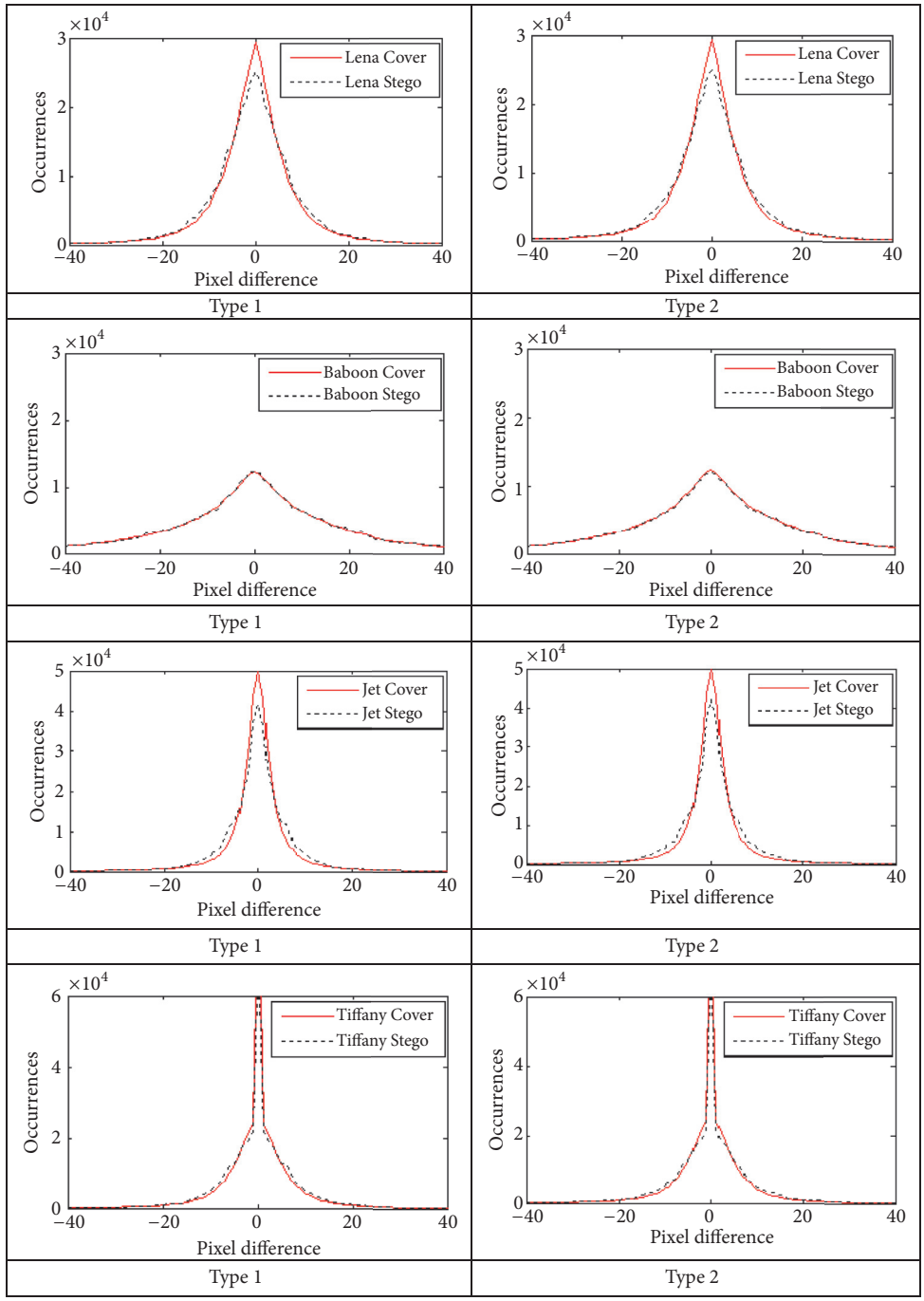


Figure 11: PDH analysis of proposed variant 2: eight directional PVD with LSB substitution.

Lena and Baboon images. It is carried out by computing 4 parameters such as R_{-m}, S_{-m}, R_m, and S_m [27]. If the condition $R_m \approx R_{-m} > S_m \approx S_{-m}$ is true, then RS analysis fails to detect the steganography technique. However, if the condition $R_{-m} - S_{-m} > R_m - S_m$ is true, then the RS analysis succeeds in detecting the steganography technique. From these figures it is evident that Khodaei and Faez's technique and the proposed technique both escape from RS analysis.

5. Conclusion

Steganography techniques, those using the principles of LSB substitution and PVD, need to address three issues:

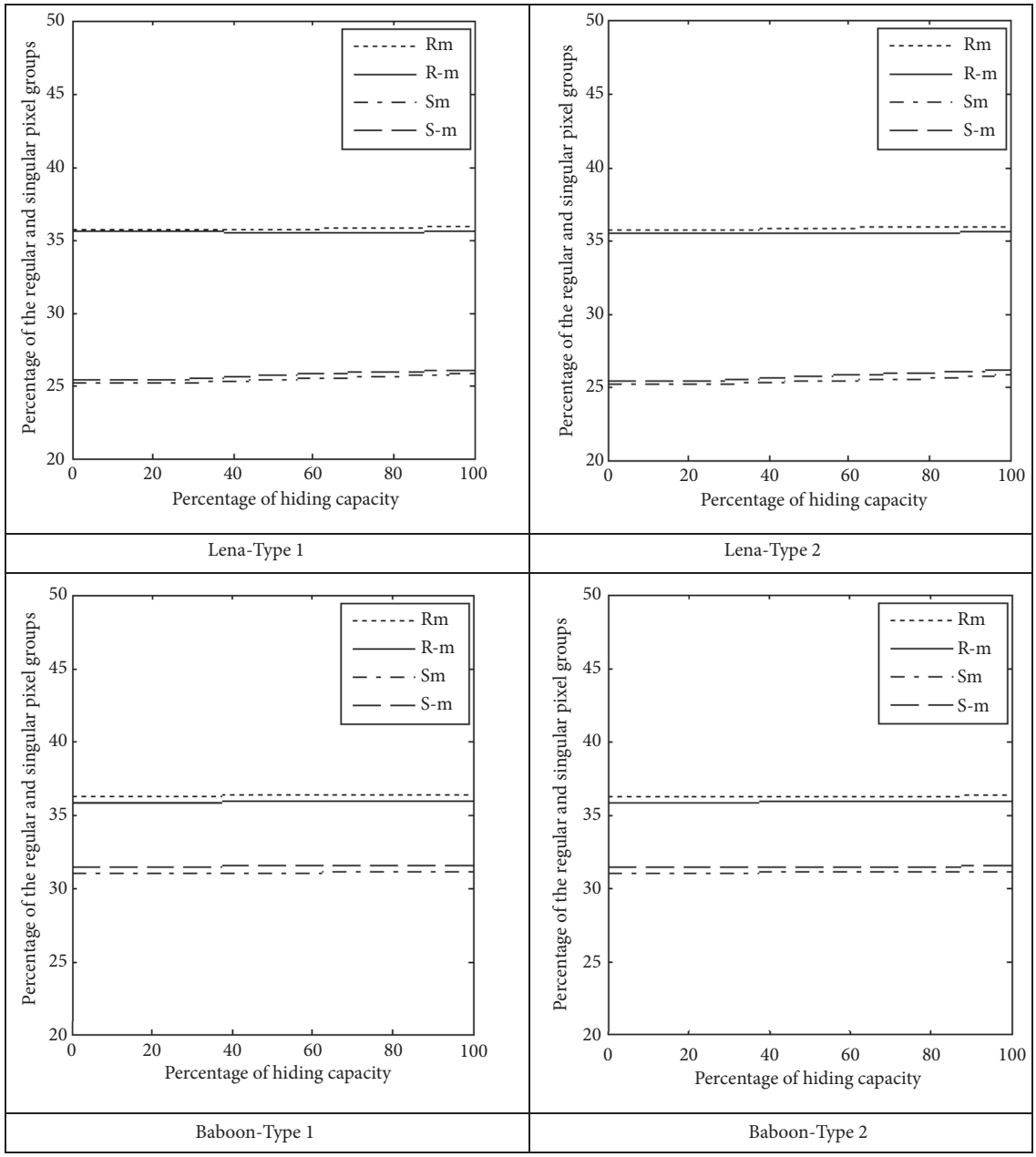


Figure 12: RS analysis of Khodaei and Faez's PVD technique over Lena and Baboon images.

(i) fall off boundary problem (FOBP), (ii) pixel difference histogram (PDH) analysis, and (iii) RS analysis. This paper proposes a steganography technique in two variants using LSB substitution and PVD. The first variant operates on 2×3 pixel blocks and the second variant operates on 3×3 pixel blocks. In one of the pixels of a block embedding is performed using modified LSB substitution. Based on the new value of this pixel, difference values with other neighboring pixels are calculated. Using these differences, PVD approach is applied. The edges in multiple directions are exploited, so PDH analysis cannot detect this steganography. The LSB substitution is performed in only one pixel of the block, so RS analysis also cannot detect this steganography. The FOBP is addressed by introducing suitable equations in the embedding procedure. The experimental results such as bit rate and PSNR are satisfactory.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

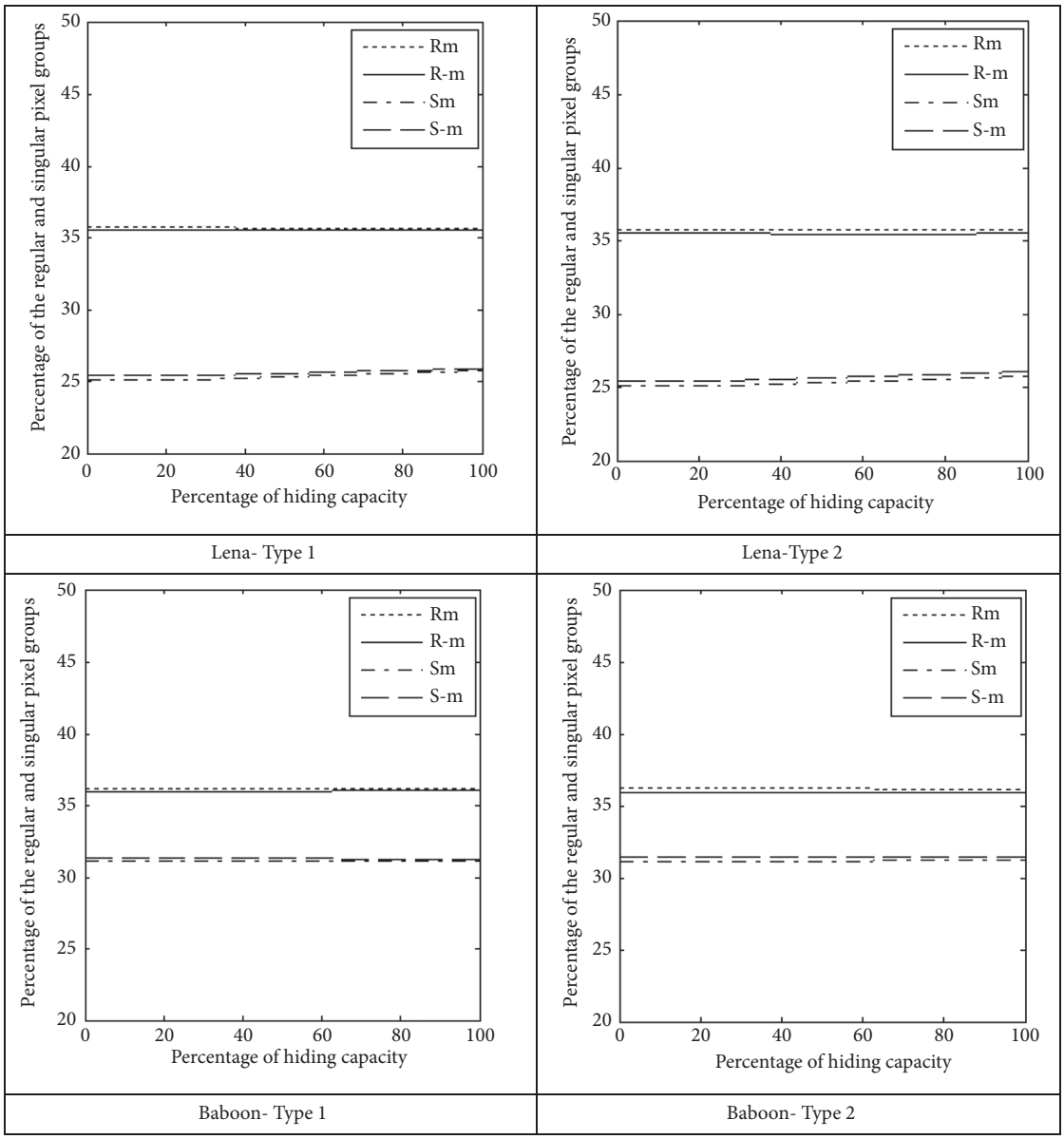


Figure 13: RS analysis of proposed variant 1: five directional PVD with M-LSB substitution over Lena and Baboon images.

Table 7: Comparison of techniques in [16, 24, 25].

Technique	PSNR	Capacity	Amount of hidden data in stego-image to compute PSNR
2 × 3 adaptive PVD [16]	50.93	1445645	One lakh and forty thousand (1,40,000) bits
3 × 2 adaptive PVD [16]	50.93	1435623	
3 PVD +3 LSB +EMD, type 1 [24]	41.07	1686041	Seven lakhs (7,00,000) bits
3 PVD +3 LSB +EMD, type 2 [24]	38.95	1798834	
7 PVD +3 LSB +EMD, type 1 [24]	40.28	1710888	
7 PVD +3 LSB +EMD, type 2 [24]	37.26	1823282	
Eight directional PVD, type 1 [25]	39.55	2361368	Seven lakhs (7,00,000) bits
Eight directional PVD, type 2 [25]	37.22	2603604	
Proposed five directional PVD+MLSB, type 1	37.62	2399853	Eight lakhs and forty thousand (8,40,000) bits
Proposed five directional PVD+MLSB, type 2	36.23	2514038	
Proposed eight directional PVD+MLSB, type 1	37.49	2394086	
Proposed eight directional PVD+MLSB, type 2	35.60	2603604	

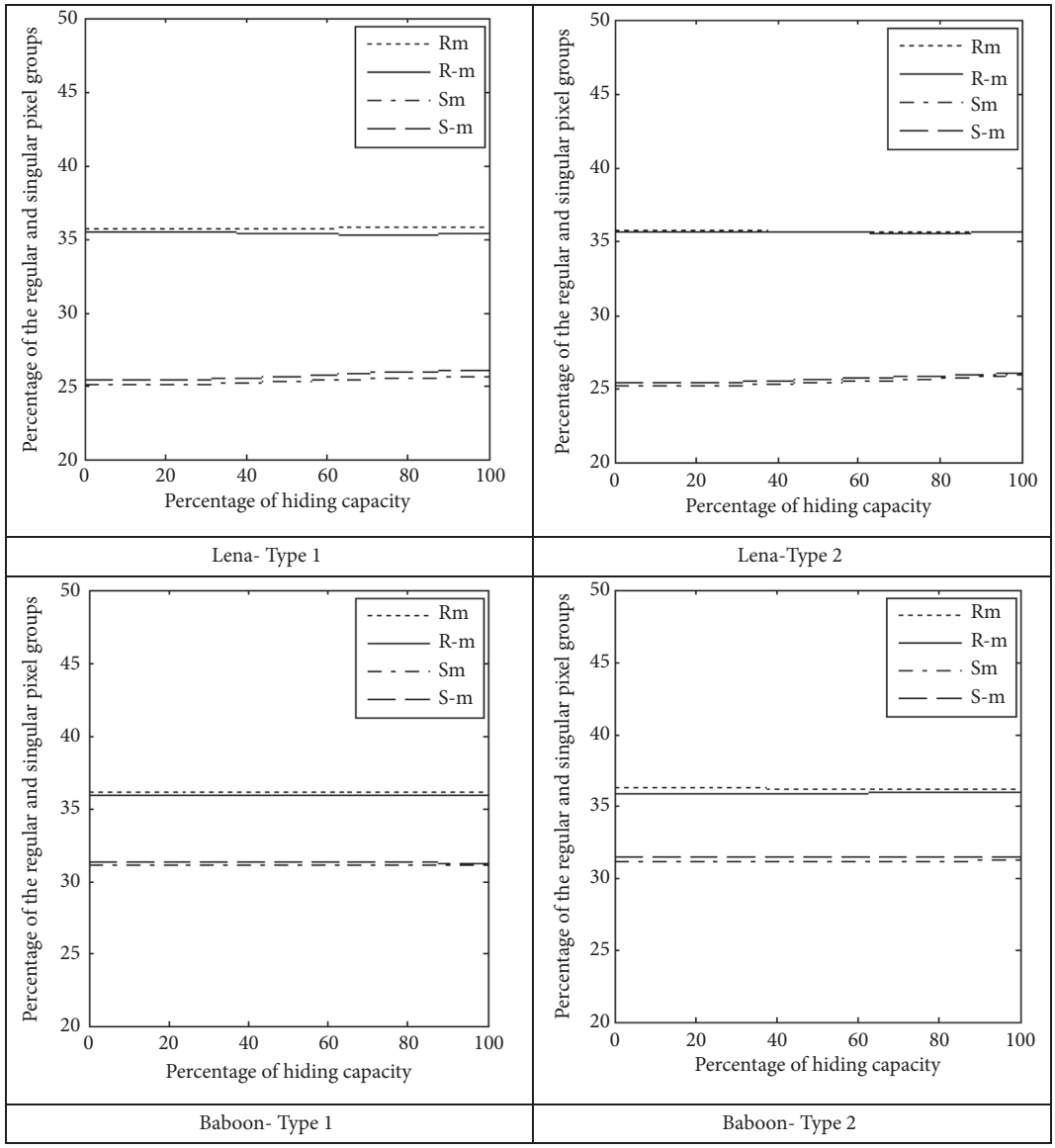


Figure 14: RS analysis of proposed variant 2: eight directional PVD with M-LSB substitution over Lena and Baboon images.

References

[1] D.-C. Wu and W.-H. Tsai, "A steganographic method for images by pixel-value differencing," *Pattern Recognition Letters*, vol. 24, no. 9-10, pp. 1613–1626, 2003.

[2] K. C. Chang, C. P. Chang, P. S. Huang, and T. M. Tu, "A novel image steganographic method using tri-way pixel-value differencing," *Journal of Multimedia*, vol. 3, no. 2, pp. 37–44, 2008.

[3] Y.-P. Lee, J.-C. Lee, W.-K. Chen, K.-C. Chang, I.-J. Su, and C.-P. Chang, "High-payload image hiding with quality recovery using tri-way pixel-value differencing," *Information Sciences*, vol. 191, pp. 214–225, 2012.

[4] C.-C. Chang and H.-W. Tseng, "A steganographic method for digital images using side match," *Pattern Recognition Letters*, vol. 25, no. 12, pp. 1431–1437, 2004.

[5] C.-H. Yang, C.-Y. Weng, H.-K. Tso, and S.-J. Wang, "A data hiding scheme using the varieties of pixel-value differencing in multimedia images," *The Journal of Systems and Software*, vol. 84, no. 4, pp. 669–678, 2011.

[6] W. Hong, T.-S. Chen, and C.-W. Luo, "Data embedding using pixel value differencing and diamond encoding with multiple-base notational system," *The Journal of Systems and Software*, vol. 85, no. 5, pp. 1166–1175, 2012.

[7] H.-C. Wu, N.-I. Wu, C.-S. Tsai, and M.-S. Hwang, "Image steganographic scheme based on pixel-value differencing and LSB replacement methods," *IEEE Proceedings Vision, Image and Signal Processing*, vol. 152, no. 5, pp. 611–615, 2005.

[8] C.-H. Yang, C.-Y. Weng, S.-J. Wang, and H.-M. Sun, "Varied PVD + LSB evading detection programs to spatial domain in data embedding systems," *The Journal of Systems and Software*, vol. 83, no. 10, pp. 1635–1643, 2010.

[9] J. Chen, "A PVD-based data hiding method with histogram preserving using pixel pair matching," *Signal Processing: Image Communication*, vol. 29, no. 3, pp. 375–384, 2014.

[10] X. Liao, Q. Wen, and J. Zhang, "A steganographic method for digital images with four-pixel differencing and modified LSB substitution," *Journal of Visual Communication and Image Representation*, vol. 22, no. 1, pp. 1–8, 2011.

[11] M. Khodaei and K. Faez, "New adaptive steganographic method using least-significant-bit substitution and pixel-value differencing," *IET Image Processing*, vol. 6, no. 6, pp. 677–686, 2012.

[12] G. Swain, "A Steganographic Method Combining LSB Substitution and PVD in a Block," in *Proceedings of the International Conference on Computational Modelling and Security, CMS 2016*, pp. 39–44, India, February 2016.

[13] A. Pradhan, K. Raja Sekhar, and G. Swain, "Digital image steganography combining lsb substitution with five way PVD in 2×3 pixel blocks," *International Journal of Pharmacy and Technology*, vol. 8, no. 4, pp. 22051–22061, 2016.

[14] W. Luo, F. Huang, and J. Huang, "A more secure steganography based on adaptive pixel-value differencing scheme," *Multimedia Tools and Applications*, vol. 52, no. 2-3, pp. 407–430, 2011.

[15] G. Swain, "Adaptive pixel value differencing steganography using both vertical and horizontal edges," *Multimedia Tools and Applications*, vol. 75, no. 21, pp. 13541–13556, 2016.

[16] A. Pradhan, K. R. Sekhar, and G. Swain, "Adaptive PVD steganography using horizontal, vertical, and diagonal edges in six-pixel blocks," *Security and Communication Networks*, vol. 2017, 13 pages, 2017.

[17] S. Chakraborty, A. S. Jalal, and C. Bhatnagar, "LSB based non blind predictive edge adaptive image steganography," *Multimedia Tools and Applications*, pp. 1–15, 2016.

[18] C. Balasubramanian, S. Selvakumar, and S. Geetha, "High payload image steganography with reduced distortion using octonary pixel pairing scheme," *Multimedia Tools and Applications*, vol. 73, no. 3, pp. 2223–2245, 2014.

[19] A. Pradhan, K. Raja Sekhar, and G. Swain, "Digital image steganography based on seven way pixel value differencing," *Indian Journal of Science and Technology*, vol. 9, no. 37, pp. 1–11, 2016.

[20] K. A. Darabkh, A. K. Al-Dhamari, and I. F. Jafar, "A new steganographic algorithm based on multi directional PVD and modified LSB," *Information Technology and Control*, vol. 46, no. 1, pp. 16–36, 2017.

[21] X. Zhang and S. Wang, "Efficient steganographic embedding by exploiting modification direction," *IEEE Communications Letters*, vol. 10, no. 11, pp. 781–783, 2006.

[22] C. Kim, "Data hiding by an improved exploiting modification direction," *Multimedia Tools and Applications*, vol. 69, no. 3, pp. 569–584, 2014.

[23] S.-Y. Shen and L.-H. Huang, "A data hiding scheme using pixel value differencing and improving exploiting modification directions," *Computers & Security*, vol. 48, pp. 131–141, 2015.

[24] A. Pradhan, K. R. Sekhar, and G. Swain, "Digital image steganography using LSB substitution, PVD, and EMD," *Mathematical Problems in Engineering*.

[25] G. Swain, "Digital image steganography using eight directional PVD against RS analysis and PDH analysis," *Advances in Multimedia*.

[26] USC-SIPI Image Database, http://sipi.usc.edu/database/database.php?volume=misc.

[27] A. Pradhan, A. K. Sahu, G. Swain, and K. R. Sekhar, "Performance evaluation parameters of image steganography techniques," in *Proceedings of the International Conference on Research Advances in Integrated Navigation Systems (RAINS '16)*, pp. 1–8, Bangalore, India, May 2016.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 7915393, 14 pages
https://doi.org/10.1155/2018/7915393

Research Article

Secure and Efficient Image Retrieval over Encrypted Cloud Data

Haihua Liang,[1,2] **Xinpeng Zhang**,[1,3] **Hang Cheng**,[4] **and Qiuhan Wei**[1]

[1]*School of Communication and Information Engineering, Shanghai University, Shanghai, China*
[2]*Department of Computer Science and Engineering, Changshu Institute of Technology, Suzhou, China*
[3]*Key Laboratory of Specialty Fiber Optics and Optical Access Networks, Joint International Research Laboratory of Specialty Fiber Optics and Advanced Communication, Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai, China*
[4]*College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China*

Correspondence should be addressed to Haihua Liang; lhh@cslg.edu.cn

Received 28 September 2017; Revised 7 April 2018; Accepted 6 June 2018; Published 8 July 2018

Academic Editor: Zheng Yan

This paper proposes a novel image retrieval scheme over encrypted cloud data, which achieves high efficiency and confidentiality. For the purpose of improving search efficiency, an index tree is often deployed in the image retrieval scheme. Meanwhile, the confidentiality of the sensitive cloud data, such as outsourced images, index tree, and query request, is also a key issue. Firstly, a balanced binary clustering algorithm is exploited over the integrated image features composed of basic features, such as HSV histogram and DCT histogram, yielding a balanced binary tree (BBT). In particular, due to the adoption of a balanced index tree, our scheme can achieve logarithmic search time. Secondly, the secure inner product is employed to encrypt the index vector and query feature. Finally, to resist the statistical attack of the frequency distribution of the retrieved results, we copy the database and merge the subtree of encrypted BBT to blind the search results. Security analysis and experimental results show that the proposed scheme is secure and efficient.

1. Introduction

With the arrival of the cloud computing, secure image retrieval on the cloud has attracted more and more attention in recent years [1]. Due to high convenience and economic savings, both enterprises and individuals tend to employ the cloud to store and manage their sensitive data, such as photo albums and personal health documents. To ensure the confidentiality of outsourced data, the common approach is to encrypt the data before it is outsourced to the cloud. However, the traditional encryption may make the basic data operations unfeasible, for example, the information retrieval of encrypted data. In the ciphertext scenario, how to achieve an efficient retrieval while protecting the privacy of customers becomes a challenging problem.

At present, there are three main issues that restrict the development of information retrieval in the encrypted domain. The first issue is to realize searchable functionality on encrypted data and achieve the same precision as plaintext data. Certainly, a naive approach is to download all the ciphertext, decrypt them, and search locally in the plaintext. However, it will incur heavy cost of bandwidth and computation. To address this issue, cryptographic techniques, such as homomorphic encryption [2] and multiparty computation, can be employed to encrypt the plaintext data and support search operation in the ciphertext. However, the above methods are concerned more with data confidentiality than retrieval efficiency, and the cost is expensive in practical applications. In the contrary, some efficient techniques, such as order-preserving encryption (OPE) [3, 4], randomized hash functions [5–7], and asymmetric scalar-product-preserving encryption (ASPE) [8], are widely adopted. The reason is that they take both the data confidentiality and retrieval efficiency into account. The second issue is that although plaintext content of the encrypted data is not leaked in the above schemes, some statistical information, such as the request frequency of encrypted query (*i.e.*, query pattern) or the access frequency of encrypted result (*i.e.*,

access pattern), may leak the privacy of query user. Oblivious RAMs [9] is a solution to protect the access pattern, but not practical enough. The third issue is that a retrieval scheme of linear efficiency is not desirable, because the search time will increase as the dataset becomes larger. In fact, secure information retrieval is commonly used for images or documents stored in a cloud server. We detail them as follows.

On the one hand, abundant works have been put forward to achieve secure retrieval in the encrypted documents. For example, Boolean search based on the single keyword is separately proposed in symmetric key setting [11] and public key setting [12]. Since similarity search is more practical than Boolean search, multi-keyword ranked search [13] is studied to enrich search functionality and improve result accuracy, where each document is associated with an index vector. Each element of the vector indicates whether a keyword exists or represents its "term frequency (TF) × inverse document frequency (IDF)." Then, the k-nearest neighbor is found by comparing the cosine similarity between the query vector and all index vectors, which is linear efficiency. In order to improve retrieval efficiency, a few works based on index tree are proposed. For example, Sun et al. [14] present a tree-based search scheme that constructs index vectors of all documents as an MDB-tree. It achieves sublinear search efficiency via setting the prediction threshold for each level of the index tree. Although a tighter prediction value can obtain logarithmic search efficiency, the result accuracy is sacrificed at the same time. Moreover, Xia et al. [15] build a KBB-tree from a bottom-up approach. In KBB-tree, the element of internal node vector is the maximum value of the corresponding position of its child node vectors. A "Greedy-Depth-First Search" algorithm is executed to find k most relevant leaf nodes, which are stored in an RList. If the correlation score between the query vector and the internal node vector is smaller than the minimum score in the RList, the subtree of the internal node does not need to be searched. Thus, this scheme can also achieve sublinear efficiency.

On the other hand, some works have been proposed for encrypted image retrieval. In [16], a privacy-enhanced face recognition is realized with a help of Paillier homomorphic encryption (HE). The drawback is that the adoption of HE incurs heavy cost of computation and communication. To be practical, Lu et al. propose a secure content-based image retrieval (CBIR) scheme based on feature/index randomization [17] or min-Hash [5]. Meanwhile, the performance comparison between homomorphic encryption and distance-preserving randomization is studied in [18]. Due to the one-way and binary property of hash code, secure CBIR that exploits the hash function to encrypt features is effective and efficient in the large-scale database [6]. However, the access pattern is leaked in the above schemes. To solve this issue, Weng et al. [7] omit certain bits of the hash code of query image. As a result, the cloud returns all possible candidates to the user. Therefore, the query pattern and access pattern are protected. But the user is involved to compare the features of candidates and obtain an accurate result. Also, it is difficult to generate hash codes that uniformly distributed in the feature space. In further, under the vector space model, there is only a handful of works that support efficient index structure. For example, Xia et al. [19] use local-sensitive hash (LSH) to construct a prefilter table, but a refinement of the candidate results is also a linear comparison. Thus, it just achieves sublinear search efficiency. Yuan et al. [20] employ k-means to build an index tree. Since k-means is not a balanced clustering algorithm, it is inevitable to generate an index tree of skewed hierarchies. Subsequently, due to uneven depth in different parts of index tree, the search efficiency tends to be sublinear. In short, under the vector space model, the requirement for secure and efficient image retrieval mechanisms remains open up to date.

Besides the above schemes that based on the encrypted feature, the secure retrieval scheme directly based on encrypted images has been proposed. It is designed for the widely used JPEG image. For example, Zhang and Cheng first use DC histogram preserved in the encrypted image as a feature [21] and then exploit AC histogram [22] and a new block feature [23] for retrieval. However, the efficiency of linear comparison or block-wise comparison is unacceptable. Therefore, the construction of secure index tree is necessary.

In this paper, we propose an efficient image retrieval scheme over encrypted cloud data, which based on a balanced index tree, *i.e.*, balanced binary tree (BBT). We introduce a novel balanced binary clustering (BBC) algorithm and use integrated features to build the BBT. Owing to the fact that this paper realizes similarity search, we tend to utilize global feature (such as color, shape) instead of accurate local feature (such as SIFT, BRIEF). Hence, our integrated feature is composed of widely used HSV histogram and DCT histogram [5, 21, 22, 24]. We focus on the construction of index tree and protection of query pattern and access pattern. As for the image encryption, we adopt partial image encryption technique [23, 25, 26] for JPEG format image and mature cryptographic ciphers (such as AES) for the other format images. Note that the key distribution is an independent issue and out of the scope of this paper. Our main contributions lie in three aspects:

(1) We combine HSV histogram in the color space and DCT histogram in the transform domain as an integrated feature. To improve the search efficiency, we construct a BBT for the image database. The BBT is built from a top-down approach. Since the hierarchical index tree is approximately balanced, the scheme can achieve logarithmic search time.

(2) By incorporating secure inner product (*i.e.*, ASPE), the BBT is encrypted as secure BBT (SBBT) by a secret key, and query feature is encrypted. The database image is encrypted by another secret key. After that, the sensitive data in the cloud is protected; meanwhile, the searchable functionality of the encrypted feature is still valid. Further, the random number extended in the query feature can blind the query pattern.

(3) To blind the search result, we copy the database and merge the subtree to rebuild SBBT. The reduced SBBT and the weighted factor of the integrated feature make the search result of one query image not unique. Therefore, the query privacy about access pattern is protected.

The rest of this paper is organized as follows. In Section 2, we introduce system model, threat model, and design goals. Also, common notations are defined. In Section 3, we propose

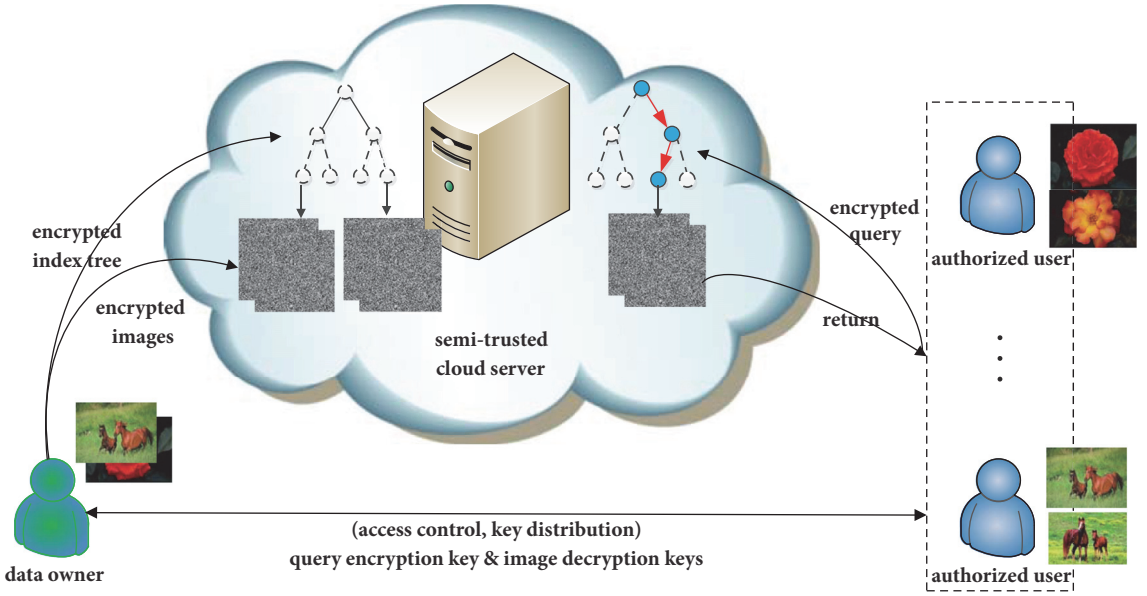


Figure 1: A secure and efficient image retrieval scheme over encrypted data.

BBC algorithm, use BBC to build a BBT, and adopt ASPE to encrypted BBT. Section 4 presents and discusses experimental results. Finally, we conclude the paper in Section 5.

2. Problem Formulation

2.1. System Model. The system contains three parties: the data owner, the cloud server, and authorized users, as illustrated in Figure 1.

The owner first extracts feature vectors, such as HSV histogram and DCT histogram, from each image in his database. They are concatenated into an integrated feature. After the feature extraction, the plaintext image is encrypted as the ciphertext by a secret key. For improving search efficiency, the owner builds an index tree based on the integrated features. Then, he utilizes another secret key to encrypt the index tree. The leaf node of the secure index tree is associated with the encrypted image. Afterward, the owner outsources the encrypted database images and the secure index tree to the cloud. In addition, the owner sends secret keys to each authorized user through a secure channel.

The authorized user extracts feature vectors from the query image. To fetch encrypted images in the secure index tree, he encrypts the integrated feature of a query image and submits it to the cloud.

The cloud stores the encrypted database images and secure index tree from the owner. After receiving an encrypted query feature, the cloud executes the search operation in the secure index tree. After that, he returns encrypted images that associated with retrieved leaf node to the user.

2.2. Thread Model. We assume that three parties follow a semihonest model, in which any two parties do not collude to share information. However, even if the cloud does not know the content of the encrypted query, it may count the query frequency of same encrypted request and the access frequency of the same returned result. Thus, depending on the available information to the cloud, we consider two threat models here.

Known Ciphertext Model. The cloud only knows encrypted database, secure index tree, and the encrypted query.

Known Background Model. In this stronger model, the cloud knows background knowledge about the frequency distribution of the query images. For example, Trump's photos are very popular in the presidential campaign. The cloud can conduct a statistical attack to identify certain image via the access frequency of the returned images.

2.3. Design Goals. To enable secure image retrieval over encrypted cloud data under the above models, our scheme should achieve the following design goals.

Effectiveness. The scheme is designed to support search over encrypted data. The result is as accurate as the search in the plaintext domain.

Efficiency. Under vector space model, we structure the feature vectors of image database as an SBBT. Instead of linear or sublinear search efficiency, our scheme will achieve logarithmic search time.

Privacy. The goal is to protect the privacy of owner and user, which is summarized as follows. (1) The database and its index tree confidentiality: namely, the image and its index tree exist in the ciphertext form in the cloud. (2) The query privacy: it is not possible to determine whether two requests are from the same query image. Note that to ease the overload of computation and communication, our goals are not to fully protect access pattern, *i.e.*, the search path in the SBBT.

2.4. Notations. Assume that the total number of images in the database is N. We define common notations as follows:

(i) $\mathbf{D}$: the database, namely, the set of images

(ii) $\mathbf{C}$: the ciphertext form of $\mathbf{D}$ stored in the cloud

(iii) $\mathbf{p}_i^1$: the normalized feature, such as HSV histogram from the ith image, $1 \le i \le N$, it is a d_1-dimensional row vector

(iv) $\mathbf{p}_i^2$: the normalized feature, such as DCT histogram from the ith image, $1 \le i \le N$, it is a d_2-dimensional row vector

(v) $\overrightarrow{\mathbf{p}}_i$: the integrated feature, $\overrightarrow{\mathbf{p}}_i = [\alpha \mathbf{p}_i^1, (1-\alpha)\mathbf{p}_i^2]$, a weighted factor $\alpha \in [0, 1]$

(vi) $\mathbf{v}$: the index vector in each node of BBT, it is generated from $\overrightarrow{\mathbf{p}}_i$

(vii) $\mathbf{I}$: the $H-$height BBT for the whole image database $\mathbf{D}$, $H \approx \lceil \log_2^N \rceil$

(viii) $\mathbf{V}$: the ciphertext index which corresponds to plaintext $\mathbf{v}$

(ix) $\mathbf{T}$: the ciphertext form of $\mathbf{I}$

(x) $\mathbf{q}^1$: the normalized feature of the query image, such as HSV histogram

(xi) $\mathbf{q}^2$: the normalized feature of the query image, such as DCT histogram

(xii) $\overrightarrow{\mathbf{q}}$: the query vector, $\overrightarrow{\mathbf{q}} = [\beta \mathbf{q}^1, (1-\beta)\mathbf{q}^2]$, a weighted factor $\beta \in [0, 1]$

(xiii) $\mathbf{M}$: the encryption key $\mathbf{M} \in \mathbb{R}^{(d+1)\times(d+1)}$ is a random invertible matrix, $d = d_1 + d_2$

(xiv) $\mathbf{Q}$: the ciphertext query, corresponding to plaintext $\overrightarrow{\mathbf{q}}$

To find the k-nearest neighbor of the query image, we adopt "inner product similarity" to evaluate the distance among the features. Specifically, $\overrightarrow{\mathbf{p}}_i$ is an integrated feature of the database image, including HSV histogram and DCT histogram. $\overrightarrow{\mathbf{q}}$ is the integrated feature of the query image. The final similarity score is obtained by $\overrightarrow{\mathbf{p}}_i \bullet \overrightarrow{\mathbf{q}}$. Based on the score, the most relevant images in the database are returned to the user.

3. Proposed Scheme

In order to achieve logarithmic search efficiency, we first build a balanced index tree for the plaintext image database, in which a novel balanced binary clustering algorithm is proposed. Then, to ensure the confidentiality of index vector $\mathbf{v}$ and query vector $\overrightarrow{\mathbf{q}}$, the secure inner product computation is employed to encrypt the vector $\mathbf{v}$ and $\overrightarrow{\mathbf{q}}$. Also, we adopt a partial image encryption to encrypt database images. Finally, we improve the scheme to enhance privacy protection in the known background model.

3.1. BBC Algorithm: Balanced Binary Clustering. To build the balanced index tree, we need a balanced binary clustering (BBC) algorithm to find a separating vector $\mathbf{v}$ which divides each dataset into two equal (or nearly equal) clusters. However, traditional k-means aims at finding cluster centers so that the sum of the distance between each point and its nearest center is maximized, rather than the equal size of clusters. Thus, frequency sensitive competitive learning (FSCL) [27] is introduced to improve k-means. FSCL is a conscience type algorithm that takes the size of clusters as a weight factor which makes the bigger cluster less likely to win more points in next iteration. It can converge to a local minimum [28]. Although the difference of cluster centers can be used as a separating vector $\mathbf{v}$, FSCL does not ensure that the k clusters are equally sized. In fact, it is experienced as very unstable in high-dimensional vector spaces [29]. On the contrast, balanced k-means [30] can split a dataset into two equal-sized clusters by finding a perfect matching in a bipartite graph. But it does not generate the separating vector $\mathbf{v}$. Thus, we propose a novel BBC to find a separating vector that makes the size of two clusters equal. To our best knowledge, this work is the first endeavor to combine FSCL and balanced k-means to achieve balanced clustering. The detail of BBC algorithm is elaborated in the following.

Given n points $\{\overrightarrow{\mathbf{p}}_i\}_{i=1}^n \in \mathbb{R}^d$, at the tth iteration, the size of cluster is $\tau_k^{(t)} \ge 0$ and the cluster center is $\mathbf{C}_k^{(t)}$, $k = 1, 2$. The BBC algorithm works as follows.

Initialization. We adopt k-means++ to choose two points from $\{\overrightarrow{\mathbf{p}}_i\}_{i=1}^n$ as initial centers $\mathbf{C}_k^{(1)}$, $k = 1, 2$.

First, uniformly choose an initial center $\mathbf{C}_1^{(1)}$ from $\{\overrightarrow{\mathbf{p}}_i\}_{i=1}^n$.

Second, choose the next initial center $\mathbf{C}_2^{(1)}$ from $\{\overrightarrow{\mathbf{p}}_i, \overrightarrow{\mathbf{p}}_i \ne \mathbf{C}_1^{(1)}\}_{i=1}^n$ with probability $\mathrm{D}\langle\overrightarrow{\mathbf{p}}_i, \mathbf{C}_1^{(1)}\rangle / \sum D\langle\overrightarrow{\mathbf{p}}_i, \mathbf{C}_1^{(1)}\rangle$, $D\langle\overrightarrow{\mathbf{p}}_i, \mathbf{C}_1^{(1)}\rangle = 1 - (\overrightarrow{\mathbf{p}}_i \bullet \mathbf{C}_1^{(1)})/\|\overrightarrow{\mathbf{p}}_i\|\|\mathbf{C}_1^{(1)}\|$.

Finally, according to the distance to the initial centers, other points are divided into two categories CL_k, the size of which is $\tau_k^{(1)}$, $k = 1, 2$.

Cluster Assignment. Let new $T_{i,k}^{(t)}$ be the solution of optimization problem as follows.

$$\min \quad J\left(T_{i,k}^{(t)}\right) = \sum_{k=1}^{2}\left(\tau_k^{(t)} \cdot \sum_{i=1}^{n} T_{i,k}^{(t)} D\left\langle\overrightarrow{\mathbf{p}}_i, \mathbf{C}_k^{(t)}\right\rangle\right) \tag{1}$$

$$\text{s.t.} \quad \begin{cases} T_{i,k}^{(t)} \in \{0, 1\}, & i = 1, \ldots, n, \ k = 1, 2 \\ \sum_{k=1}^{2} T_{i,k}^{(t)} = 1, & i = 1, \ldots, n \\ \sum_{i=1}^{n} T_{i,k}^{(t)} \ge \left\lfloor \frac{n}{2} \right\rfloor, & k = 1, 2 \end{cases} \tag{2}$$

where $D\langle\overrightarrow{\mathbf{p}}_i, \mathbf{C}_k^{(t)}\rangle = 1 - (\overrightarrow{\mathbf{p}}_i \bullet \mathbf{C}_k^{(t)})/\|\overrightarrow{\mathbf{p}}_i\|\|\mathbf{C}_k^{(t)}\|$. A smaller value of $D\langle\overrightarrow{\mathbf{p}}_i, \mathbf{C}_k^{(t)}\rangle$ indicates that the two vectors $\overrightarrow{\mathbf{p}}_i, \mathbf{C}_k^{(t)}$ are more similar.

Inspired by balanced k-means, we first turn (1) as a linear assignment problem (LAP) [31]. The cost matrix is defined as

$$CE = \begin{bmatrix} D\left\langle \overrightarrow{\mathbf{p}}_1, \mathbf{C}_k^{(1)} \right\rangle & \cdots & D\left\langle \overrightarrow{\mathbf{p}}_n, \mathbf{C}_k^{(1)} \right\rangle \\ D\left\langle \overrightarrow{\mathbf{p}}_1, \mathbf{C}_k^{(2)} \right\rangle & \cdots & D\left\langle \overrightarrow{\mathbf{p}}_n, \mathbf{C}_k^{(2)} \right\rangle \\ & \cdots & \\ D\left\langle \overrightarrow{\mathbf{p}}_1, \mathbf{C}_k^{(1)} \right\rangle & \cdots & D\left\langle \overrightarrow{\mathbf{p}}_n, \mathbf{C}_k^{(1)} \right\rangle \\ D\left\langle \overrightarrow{\mathbf{p}}_1, \mathbf{C}_k^{(2)} \right\rangle & \cdots & D\left\langle \overrightarrow{\mathbf{p}}_n, \mathbf{C}_k^{(2)} \right\rangle \end{bmatrix}, \tag{3}$$

where $CE \in \mathbb{R}^{2\lceil n/2 \rceil \times n}$. If it is not a square matrix, a column of zeros is added to make it square. Then, we adopt Volgenant-Jonker (VJ) algorithm [32, 33] to find a perfect matching in a bipartite graph. Finally, if the vector $\overrightarrow{\mathbf{p}}_i$ is assigned to center $\mathbf{C}_k^{(t)}$, $T_{i,k}^{(t)} = 1$ and 0 otherwise.

Cluster Update. Update the cluster centers $\mathbf{C}_k^{(t+1)}$, the "selection" variables $T_{i,k}^{(t+1)}$, and the size of clusters $\tau_k^{(t+1)}$ as follows:

$$\begin{aligned} \mathbf{C}_k^{(t+1)} &= \frac{\sum_{i=1}^n T_{i,k}^{(t)} \overrightarrow{\mathbf{p}}_i}{\sum_{i=1}^n T_{i,k}^{(t)}}, \quad k = 1, 2 \\ T_{i,1}^{(t+1)} &= 1, \\ T_{i,2}^{(t+1)} &= 0, \\ &\qquad \text{if } D\left\langle \overrightarrow{\mathbf{p}}_i, \mathbf{C}_1^{(t+1)} - \mathbf{C}_2^{(t+1)} \right\rangle \le 0 \\ T_{i,1}^{(t+1)} &= 0, \\ T_{i,2}^{(t+1)} &= 1, \\ &\qquad \text{if } D\left\langle \overrightarrow{\mathbf{p}}_i, \mathbf{C}_1^{(t+1)} - \mathbf{C}_2^{(t+1)} \right\rangle > 0 \\ \tau_k^{(t+1)} &= \sum_{i=1}^n T_{i,k}^{(t+1)}, \quad k = 1, 2. \end{aligned} \tag{4}$$

Cluster assignment and cluster update are repeated until $\tau_k^{(t+1)} \le 2^{\lceil \log_2^n \rceil - 1}$ or $\mathbf{C}_k^{(t+1)} = \mathbf{C}_k^{(t)}$, $k = 1, 2$.

Output. The above three steps run several times with different initial centers $\mathbf{C}_1^{(1)}, \mathbf{C}_2^{(1)}$.

Owing to the fact that different initial centers may generate different results, the result that the sizes of two clusters are most similar is chosen as a final output. The size similarity of two clusters is defined as

$$s\left(\tau_k^{(t+1)}\right) = \begin{cases} 0, & \prod_{k=1,2} \tau_k^{(t+1)} = 0 \\ \dfrac{\min\left\{\tau_k^{(t+1)}, k = 1, 2\right\}}{\max\left\{\tau_k^{(t+1)}, k = 1, 2\right\}}, & \prod_{k=1,2} \tau_k^{(t+1)} \ne 0. \end{cases} \tag{5}$$

Finally, the output is

$$\begin{aligned} &\overrightarrow{\mathbf{p}}_i \in CL_k \quad \text{if } T_{i,k}^{(t+1)} = 1, \ i = 1, \ldots, n, \ k = 1, 2 \\ &\mathbf{v} = \left[\frac{\mathbf{C}_1^{(t)}}{\left\|\mathbf{C}_1^{(t)}\right\|} - \frac{\mathbf{C}_2^{(t)}}{\left\|\mathbf{C}_2^{(t)}\right\|} \right]. \end{aligned} \tag{6}$$

The dataset of n points can be divided into two equal (nearly equal) clusters by the separating vector $\mathbf{v}$.

3.2. UBBT: Unencrypted Balanced Binary Tree. In this subsection, we build the BBT for the whole database. Firstly, we generate an integrated feature $\overrightarrow{\mathbf{p}}_i$ for each image in the database. Then, an index vector $\mathbf{v}$ of the root node or internal node of the BBT is generated by a top-down approach. BBC algorithm generates $\mathbf{v}$ to separate image features into two clusters of nearly equal size, *i.e.*, CL_1, CL_2. Namely,

$$\begin{aligned} &\overrightarrow{\mathbf{p}}_i \bullet \mathbf{v} \ge 0 \quad \text{if } \overrightarrow{\mathbf{p}}_i \in CL_1 \\ &\overrightarrow{\mathbf{p}}_i \bullet \mathbf{v} < 0 \quad \text{if } \overrightarrow{\mathbf{p}}_i \in CL_2. \end{aligned} \tag{7}$$

Algorithm 1 recursively divides feature vectors of the database images into two clusters until no cluster has more than two feature vectors. The hierarchical index tree is shown in Figure 2. The level of the root node is 0 and the level of the leaf node is H. Thus, the height of entire index tree is H. The subtree that the root node is at hth level of the index tree has the height of $H - h$. The root and internal node u in our BBT are defined as follows:

$$u = \langle ID, \mathbf{v}, uL, uR \rangle \tag{8}$$

where ID is the identity of node, $\mathbf{v}$ is the index vector dividing images on the leaf nodes into two clusters CL_1 and CL_2, and uL and uR are the pointers to the left and right child node. If u is a leaf node, it is defined as

$$u = \langle ID, iL, \ldots, iL \rangle. \tag{9}$$

where iL points to the images.

The search process starts from the root node of UBBT. If $\overrightarrow{\mathbf{q}} \bullet \mathbf{v} \ge 0$, we execute iterative retrieval upon the left subtree, otherwise upon the right subtree until obtaining the leaf node.

3.3. SBBT: Secure Balanced Binary Tree. In UBBT scheme, the leaf node is retrieved by the sign of the inner product $\overrightarrow{\mathbf{q}} \bullet \mathbf{v}$. In order to protect the sensitive data $\overrightarrow{\mathbf{q}}$ and $\mathbf{v}$, we use the secure inner product (*i.e.*, ASPE) [8] to encrypt the vectors and obtain the same result as the plaintext.

First of all, the owner generates a random invertible matrix $\mathbf{M} \in \mathbb{R}^{d \times d}$ to encrypt $\mathbf{v}$. The user also use it to encrypt $\overrightarrow{\mathbf{q}}$, *i.e.*,

$$\begin{aligned} \mathbf{V} &= \mathbf{v}\mathbf{M}^{-1} \\ \mathbf{Q} &= \mathbf{M}\overrightarrow{\mathbf{q}}^{\mathrm{T}}, \end{aligned} \tag{10}$$

where $\mathbf{V}$ is the ciphertext index and $\mathbf{Q}$ is the ciphertext query. In addition, the JPEG images are encrypted via a partial image encryption [22, 25]. Therefore, index tree $\mathbf{I}$ is encrypted as $\mathbf{T}$. The leaf node of $\mathbf{T}$ is associated with the encrypted images. Then, they are outsourced to the cloud for secure image retrieval. When retrieving, the user submits $\mathbf{Q}$ of the query image to the cloud. Since $\mathbf{VQ} = \overrightarrow{\mathbf{q}} \bullet \mathbf{v}$, the cloud can

Input: $\{\vec{\mathbf{p}}_i\}$, $1 \le i \le N$; Balanced Binary Clustering, *i.e.*, BBC.
Output: Balanced Binary Tree **I**.
(1) $h = 0$ //the height of node
(2) *ChildrenGen* $(\{\vec{\mathbf{p}}_i\}_{1 \le i \le N}) \longrightarrow$ root node
(3) **function** *ChildrenGen* (**vectors** $\{\vec{\mathbf{p}}_i\}$)
(4) *BBC* $(\{\vec{\mathbf{p}}_i\}) \longrightarrow$ two clusters CL_1, CL_2, separating vector as the index **v**
(5) **if** the size of $CL_1 \ge 2$ **then**
(6) *ChildrenGen* $(\{\vec{\mathbf{p}}_i\}_{i \in CL_1}) \longrightarrow$ internal node
(7) **else**
(8) $\{i\text{th image}\}_{i \in CL_1} \longrightarrow$ leaf node
(9) **end if**
(10) **if** the size of $CL_2 \ge 2$ **then**
(11) *ChildrenGen* $(\{\vec{\mathbf{p}}_i\}_{i \in CL_2}) \longrightarrow$ internal node
(12) **else**
(13) $\{i\text{th image}\}_{i \in CL_2} \longrightarrow$ leaf node
(14) **end if**
(15) $h = h + 1$
(16) **return v**, CL_1, CL_2
(17) **end function**

Algorithm 1: Building balanced binary tree.

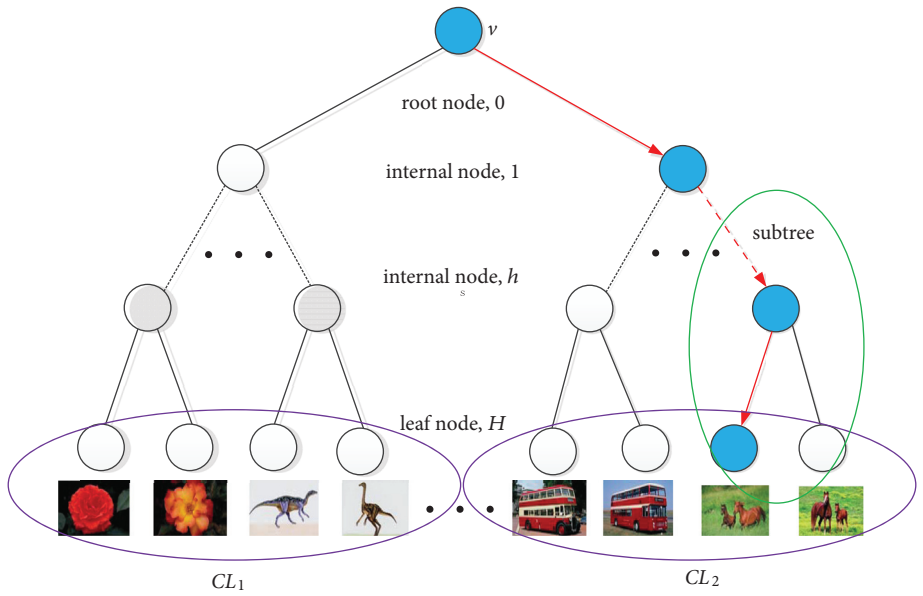


Figure 2: The hierarchical structure of unencrypted BBT.

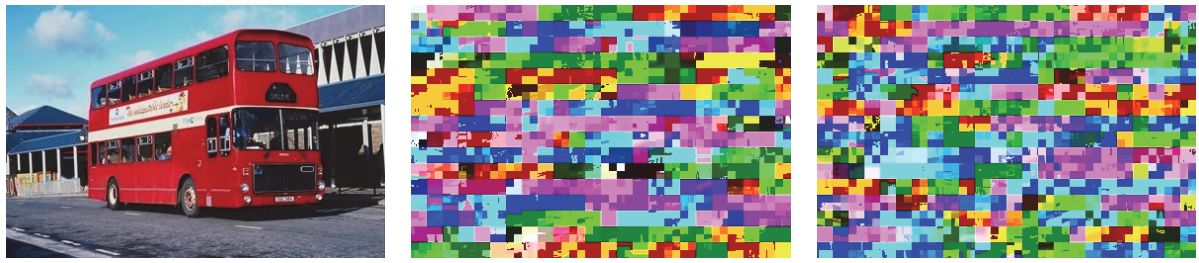
use **Q** to execute an effective search on secure index tree **T**. Therefore, the retrieved result and efficiency in the ciphertext would always be in line with that in the plaintext. Note that the more details of ASPE are described in [8].

Unfortunately, the relationship between $\vec{\mathbf{q}}$ and **Q** is deterministic. According to the same **Q** or **VQ**, the cloud can count the query frequency of $\vec{\mathbf{q}}$. Further, according to the same result, the cloud can also count the access frequency of $\vec{\mathbf{q}}$. Therefore, if the query image and $\vec{\mathbf{q}}$ is a one-to-one relationship, the proposed scheme cannot resist the statistical attack in the known background model.

To break the one-to-one relationship between the query image and ciphertext query **Q**, we extend $\vec{\mathbf{q}}$ and **v** as follows:

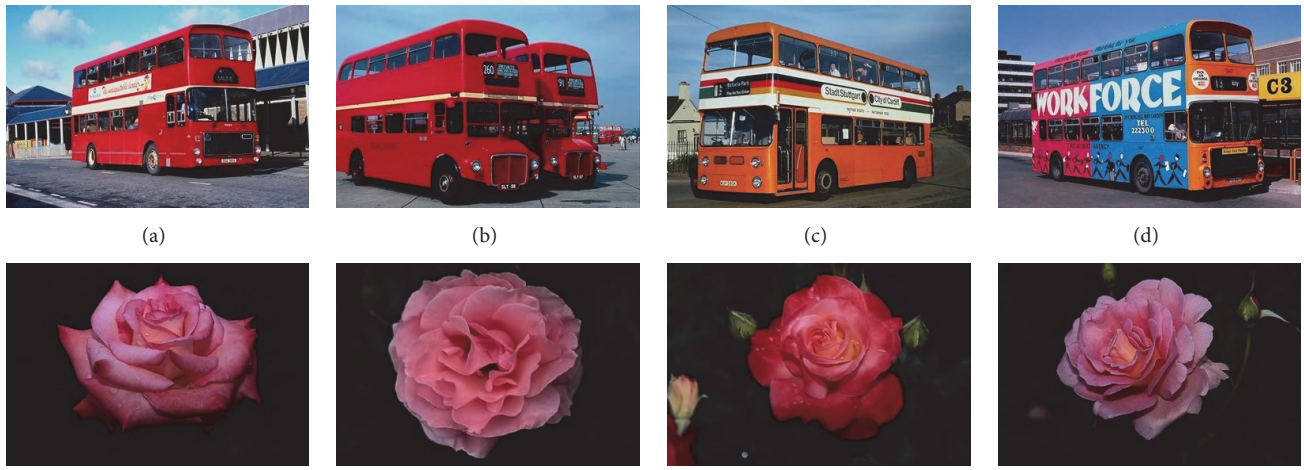
$$\vec{\mathbf{q}} = \theta\left[\gamma, \vec{\mathbf{q}}\right], \quad \mathbf{v} = [0, \mathbf{v}], \tag{11}$$

where the random numbers $\theta > 0$, $\gamma \gg 1$. Thus, the same query image can generate different ciphertext query **Q** by setting θ and γ. Meanwhile, $\mathbf{VQ} = \theta(\vec{\mathbf{q}} \bullet \mathbf{v})$ is blinded by positive θ. The cloud cannot get available statistical



(a) (b) (c)

Figure 3: (a) a query image, (b) one encrypted version, and (c) another encrypted version.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4: In Corel database [10], except for (a),(e) a query image, the most similar image by the retrieval based on (b),(f) HSV histogram, (c),(g) DCT histogram, or (d),(h) an integrated feature.

information from the ciphertext query $\mathbf{Q}$ and the result $\mathbf{VQ}$. However, the sign of $\mathbf{VQ}$ remains the same even if γ and θ are different, so the retrieved result and search path remain the same. The request frequency of a query image can be inferred from the access frequency of the same return image.

To overcome the issue, we first copy the image database. The owner uses partial image encryption [23] to protect plaintext image. According to different encryption keys, same plaintext image can be encrypted as different encrypted images, as shown in Figure 3. Meanwhile, the owner uses the embedding key to hide which encryption key was used [25]. Then, the owner shares encryption keys and the embedding key with authorized users through a secret channel. After obtaining an encrypted image, the user uses the embedding key to extract hidden data and select true encryption key to recover the plaintext image.

Second, different encrypted images derived from the same plaintext image correspond to different integrated features. For the diversity of the integrated feature, we use HSV histogram and DCT histogram as basic features. The reason is that HSV histogram in color space and DCT histogram in transform domain are image features from different perspectives. As shown in Figure 4, different features of the query image can still retrieve a similar image. For ease of description, we define the new ciphertext database as $\mathbf{C}'$. Then, we set $\mathbf{C}$ corresponding to the feature $\overrightarrow{\mathbf{p}}_i = [\alpha_1 \mathbf{p}_i^1, (1-\alpha_1)\mathbf{p}_i^2]$ and $\mathbf{C}'$ corresponds to the feature $\overrightarrow{\mathbf{p}}_i{}' = [\alpha_2 \mathbf{p}_i^1, (1-\alpha_2)\mathbf{p}_i^2]$. Finally, we use proposed BBC algorithm to build an index tree $\mathbf{I}$ from $\{\overrightarrow{\mathbf{p}}_i, \overrightarrow{\mathbf{p}}_i{}', \mathbf{C}, \mathbf{C}'\}$ and employ ASPE to encrypt the plaintext tree as a ciphertext tree $\mathbf{T}$.

To further enhance query privacy, we reduce the height of the index tree $\mathbf{T}$ and return multiple encrypted images to the query user at the one time. Specifically, we set the internal node to be a new leaf node. Then, the new leaf node contains all the encrypted images in the subtree that the internal node of $\mathbf{T}$ is a root node. Due to the fact that the index tree is balanced, leaf nodes of reduced index tree can always point to multiple encrypted images. Please note that some images in a subtree are similar, which is a compromise with effective retrieval.

3.4. Security Analysis. We analyze SBBT concerning the privacy requirements in the design goals:

(1) The database and its index are confidential. In SBBT, index vectors $\mathbf{v}$ of BBT are encrypted as $\mathbf{V}$ by the high-dimensional random matrix which is kept confidential to the

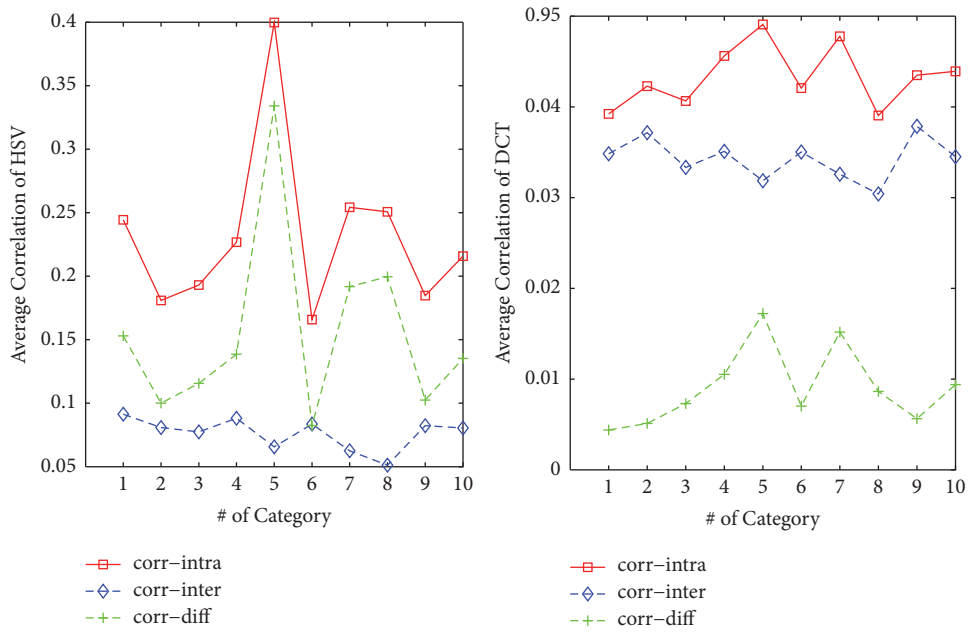


FIGURE 5: The average correlation of HSV histogram (left) and DCT histogram (right) under same category and different categories.

cloud. For images in the database, the partial encryption can protect their confidentiality [23, 25].

(2) Since the feature vector $\overrightarrow{\mathbf{q}}$ of the query image is extended by random numbers γ and θ, it can generate different ciphertext queries $\mathbf{Q}$. The cloud cannot identify the same query image from its different ciphertext queries. The query pattern is protected.

(3) According to different β, different features of same query image can retrieve different images. When β is equal to α_1 or α_2, different results can be recovered to the same query image. Thus, the retrieved result and path of the same query image are not unique but the retrieval is still effective. The cloud cannot link the different returned image to the same query image. The access pattern is protected.

(4) In addition, since the height of SBBT is reduced, one leaf node will point to multiple encrypted images. It further breaks the one-to-one relationship between the query image and the return images. The access pattern is further protected.

In short, the SBBT is secure in the known ciphertext model and can resist the statistical attack in the known background model.

4. Performance Analysis

In this section, we show the experimental results and efficiency analysis of the proposed scheme. As we know that JPEG is the most commonly used image format and accounting for up to 95% of images on the web [34], so we adopt the Corel database [10] for the similarity search. The database contains 1000 JPEG images which belong to 10 categories: African, Beach, Architecture, Buses, Dinosaurs, Elephants, Flowers, Horses, Mountain, and Food. Each of them includes 100 images. As [5], we split an image into 256 blocks and extract hue, saturation, and value of brightness of each block. Then, we perform hierarchical clustering of block features to construct the "bag of word" model. The usage frequency of the visual word is the HSV histogram of the image. As [22], DC/AC coefficients of image blocks are quantized into different bins and the DCT histogram is the usage frequency of quantized DC/AC coefficients. Thus, the color feature $\mathbf{p}_i^1$ and texture feature $\mathbf{p}_i^2$ are set as the 991-dimensional HSV histogram and 780-dimensional DCT histogram, respectively.

4.1. Parameters Setting. First of all, we should set parameters α_1 of $\overrightarrow{\mathbf{p}}_i$ and α_2 of $\overrightarrow{\mathbf{p}}_i{}'$ to construct an index tree. Because the integrated features compose of HSV histograms and DCT histogram, we first evaluate average histogram similarity of the images in the same category and different categories (denoted as corr-intra and corr-inter, respectively), as shown in Figure 5. We can see that the value of corr-intra is larger than corr-inter; thus, images of different categories can be distinguished by each histogram. According to the difference between corr-intra and corr-inter (*i.e.*, corr-diff in Figure 5), we set $\alpha_1 = 0.1$ for the integrated feature $\overrightarrow{\mathbf{p}}_i$ of the database $\mathbf{C}$. The purpose is to make the HSV histogram and DCT histogram have the same contribution to the search results.

For the parameter α_2 of $\overrightarrow{\mathbf{p}}_i{}'$, we hope that the index tree of database $\{\mathbf{C}, \mathbf{C}'\}$ can better protect the access pattern. We use two different metrics to measure the privacy protection achieved by different α_2. The two metrics are the number of different image categories in each subtree and the category entropy of each subtree. Figure 6 shows the average number of image categories contained in the subtree of different levels. Because the index tree of the database based on different α_2 has same height of 11 (*i.e.*, $\lceil \log_2^{2000} \rceil$), the proposed BBC algorithm performs well. However, we can see from

Table 1: Average category entropy of each level.

Parameter	Level of the Index Tree										
α_2	1	2	3	4	5	6	7	8	9	10	11
0.2	2.99	2.49	2.1	1.75	1.38	1.07	0.81	0.62	0.44	0.21	0
0.3	3.2	2.82	2.07	1.81	1.43	1.18	0.91	0.7	0.48	0.24	0
0.4	3.32	2.88	2.28	1.8	1.53	1.28	1.02	0.78	0.54	0.28	0
0.5	3.32	3.03	2.31	1.83	1.56	1.27	1.03	0.79	0.55	0.3	0
0.6	3.32	2.91	**2.37**	**1.96**	**1.65**	**1.41**	**1.11**	**0.89**	**0.61**	**0.32**	0
0.7	3.32	2.6	2.1	1.71	1.46	1.24	1.03	0.81	0.57	0.29	0
0.8	3.32	2.92	2.32	1.85	1.52	1.26	1.04	0.81	0.57	0.3	0
0.9	3.32	2.92	2.25	1.78	1.49	1.27	1.03	0.83	0.58	0.3	0

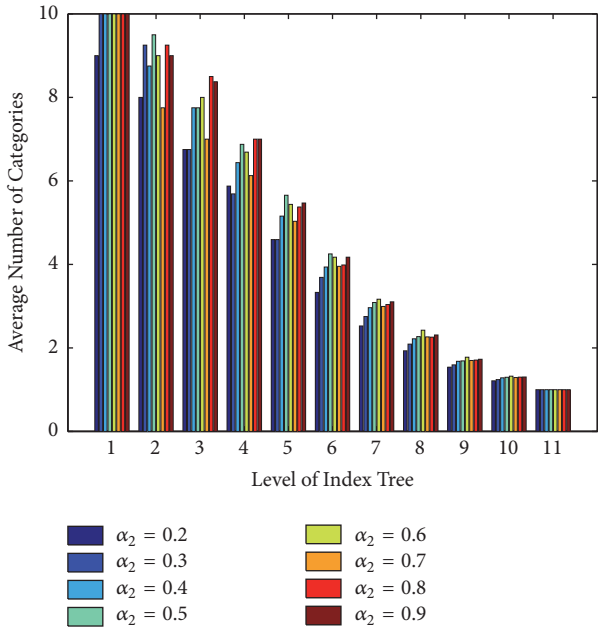


Figure 6: The average number of image categories.

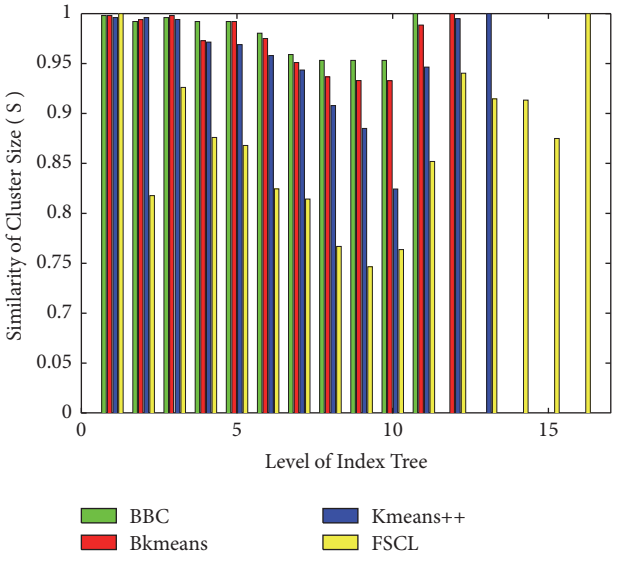


Figure 7: The average size similarity of two clusters.

Figure 6 that the subtree near the leaf node contains a similar number of image categories. We will use category entropy to measure the privacy. The category entropy, which is defined as $-\sum_i (c_i/\sum_i c_i)\log_2(c_i/\sum_i c_i)$ (c_i is the number of existing i category), is used to measure the randomness of image categories in the subtree. From Table 1, we know that the index tree of $\alpha_2 = 0.6$ has the highest average entropy at the 3th-10th level, even its average category number at 3th-6th level is not the biggest in Figure 6. Thus, we set the weighted factor $\alpha_2 = 0.6$ for the integrated feature $\vec{\mathbf{p}}_i'$ of database $\mathbf{C}'$.

4.2. Index Tree Construction. In order to prove that our algorithm can classify feature vectors in a balanced manner, we use average similarity of the cluster size to compare the clustering performance of proposed BBC algorithm, balanced k-means (*i.e.*, Bkmeans), k-means++, and FSCL. The average similarity based on (5) is defined as $S = \overline{s(\tau_k^{(t+1)})}$.

From Figure 7, we can see that the index trees generated by the proposed BBC algorithm, Bkmeans, k-means++, and FSCL have the height of 11, 12, 13, and 16, respectively. The smaller the height of the index tree indicates that the proposed algorithm can classify feature vectors into two clusters more evenly at each level of the index tree. Figure 7 shows that the cluster size of FSCL is the most unbalanced at 1th-11th levels. There are two reasons: the first is that the change of the weight factor affects the stability of the clustering result, and the second is that the value of weight factor impacts the distance between the cluster center and other feature vectors. On contrast, due to the use of conscience weight factor, the proposed algorithm performs better than Bkmeans and k-means++, and the result is much better than FSCL. The reason is that the bipartite graph matching mechanism constrains a balance of the two clusters and restricts the influence of weight factor. In short, the weight factor is still effective in improve the balance of clustering. In fact, above result is also similar in the database of different α_1, α_2.

Table 2: Minimal Dunn index of each level.

Algorithm	Level of the Index Tree															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
BBC	0.47	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.01	**0.02**	∞	-	-	-	-	-
Bkmeans	0.47	0.01	0.02	0.01	0.02	0.02	0.02	0.03	0.04	0.24	**0.73**	∞	-	-	-	-
Kmeans++	0.21	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.03	0.22	**1.05**	∞	-	-	-
FSCL	0.47	0.04	0.05	0.06	0.07	0.1	0.15	0.19	0.23	0.38	0.28	0.58	0.62	0.66	**0.95**	∞

Table 3: Time cost of index tree construction.

Algorithm	BBC	Bkmeans	Kmeans++[20]	FSCL
Time (s)	111	2316	1749	254

In further, we employ Dunn index (DI) [35] to evaluate mentioned clustering algorithms. The Dunn index is defined as

$$DI = \frac{\min_{\forall x \in CL_1, \forall y \in CL_2} \{D\langle x, y\rangle\}}{\max_{k=1,2} \left\{\max_{\forall x,y \in CL_k} \{D\langle x, y\rangle\}\right\}}, \quad (12)$$

where the numerator is the minimal distance of vectors in different clusters and the denominator is the largest within-cluster distance. Thus, we can know that when separating two points into two clusters, the denominator is zero and DI is infinite. A high value of DI means good compactness. Table 2 shows the minimal DI of clustering results in each level. In the same level of the index tree, the most of BBC's DI is smaller than other algorithms (i.e., Bkmeans, K-means++, and FSCL). Meanwhile, BBC's DI of the last level is much smaller than other algorithms. Obviously, for the proposed BBC algorithm, the cluster-size similarity is with higher priority than clustering compactness.

The above algorithms are implemented using MATLAB on a PC with an Intel Core i5 3.2 GHz CPU and 16 GB RAM. Time consumption for index tree construction is shown in Table 3. In particular, we execute 30 times clustering with random initial centers and select the result of highest score of cluster-size similarity. Bkmeans is most time consuming because it is not designed to find a vector that splits a dataset evenly. In contrast, proposed BBC costs the least time.

4.3. Retrieval Performance. To evaluate the retrieval performance, each image in the Corel database is selected as a query image. The retrieval performance is evaluated by average precision-recall (PR) curve, *i.e.*,

$$Precision = \frac{\#\ of\ returned\ positive\ images}{\#\ of\ all\ returned\ images},$$
$$Recall = \frac{\#\ of\ returned\ positive\ images}{\#\ of\ all\ positive\ images}. \quad (13)$$

Figure 8(a) compares the PR curves of the query feature based on different β values. It can be seen that the performance of different β achieves high precision when recall belongs to (0, 0.05). Besides, when only one image is returned, *i.e.*, recall=1%, the precision of $\beta = 0.1$ or 0.6 is 100%. The reason is that we set β as α_1 or α_2. Although two integrated features of a query image are different, the retrieved results are different encrypted versions of the same plaintext image. Since the query features, the corresponding results, and search paths are different, the statistical information of query image is protected.

Figure 8(b) shows the retrieval performance of the reduced index trees with different heights. It can be seen that the precision increases with the height of index tree, even if β is set as 0 or 1. Further, Figure 9 shows the average maximum height of a reduced index tree, in which the returned leaf node contains the exact query image. It can be seen that the height of reduced index tree is 11 when β is set as 0.1 or 0.6. If β is set closer to 0.1 or 0.6, the height of reduced index tree is nearer to 11. When the height is 9, the most of exact query images are retrievable in the reduced index tree and one search returns four (2^2) images. Fortunately, as shown in Figure 8(b), the precision is about 70% at the 9th level of the index tree. Thus, the retrieval is still effective.

4.4. Privacy Protection. We consider the privacy protection from two aspects: query pattern and access pattern. Since the query feature is an integrated histogram, *i.e.*, $\overrightarrow{\mathbf{q}} = [\beta\mathbf{q}^1, (1-\beta)\mathbf{q}^2]$. Different β correspond to different query features. In addition, even if β is determined, the feature will be extended by random numbers θ, γ as $\overrightarrow{\mathbf{q}} = \theta[\gamma, \overrightarrow{\mathbf{q}}]$. Thus, the encrypted feature is $\mathbf{Q} = \mathbf{M}(\theta[\gamma, \beta\mathbf{q}^1, (1-\beta)\mathbf{q}^2])^{\mathrm{T}}$. Obviously, the encrypted feature of a query image is not unique. The query pattern of each query image is protected.

To protect access pattern, the retrieved result of a query image should be not unique. We have built an index tree based on features of original database ($\alpha_1 = 0.1$) and features of the copied database ($\alpha_2 = 0.6$). Then, the integrated features of different β are used to retrieve in the index tree. We know the fact that the returned result is a leaf node of the index tree and different leaf nodes indicate different access paths. So, we define a path privacy to evaluate the protection level of access pattern

$$Privacy = \frac{\#\ of\ different\ search\ paths}{\#\ of\ all\ search\ paths}. \quad (14)$$

Different search path can be obtained by using the query feature of different β. Figures 10(a) and 10(b) show the path

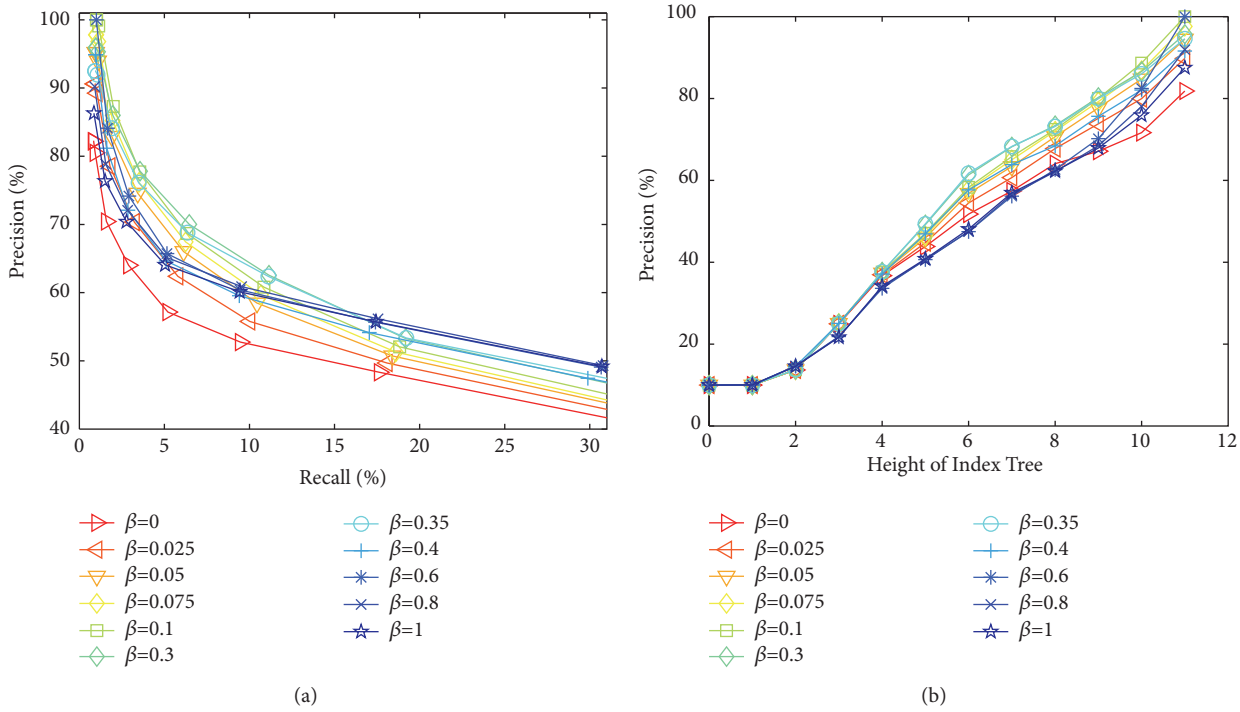


Figure 8: (a) Precision-recall curves of original index tree. (b) Retrieval precision of reduced index trees.

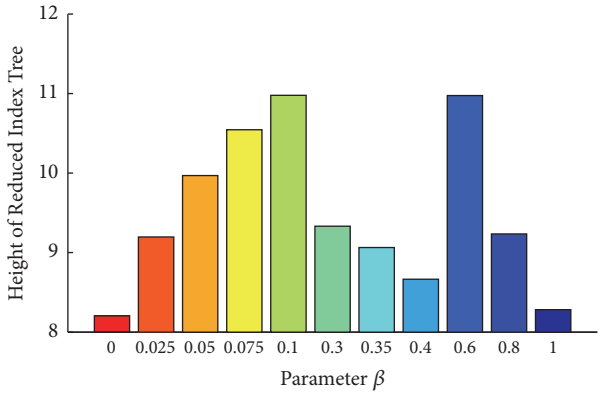


Figure 9: Under different β, the average maximum height of a reduced index tree that the query image can retrieve itself.

privacy of different β, where $\beta = 0.1$ and 0.6 are baselines, respectively. It can be seen that the path privacy is 100% when β is set as 0.6 in Figure 10(a) and 0.1 in Figure 10(b). The reason is that retrieval is separately carried out in the left or right subtree of the root node. Figure 10(a) also shows that search paths are in the same subtree of the root node when β is set to 0.6, 0.8, or 1. But the search path of $\beta = 0.8$ or 1 is still different with $\beta = 0.6$ as shown in Figure 10(b). Similarly, when β is set to 0, 0.0025, 0.05, 0.075, 0.1, 0.3, or 0.35, the search path is in the same subtree of the root node, but still different. Besides, when β is set to 0.4, the search traverses in both subtrees of the root node. In addition, the path privacy increases with the height of the tree. In short, the access pattern can be protected by setting different β. Please note that usage frequency of different β can be optimized based on the frequency distribution of the query images.

In summary, Table 4 compares the proposed scheme with other schemes in the terms of search efficiency, query pattern protection, and access pattern protection. We compare the search efficiency of different schemes by theoretical analysis instead of experiments. The reason is that encrypted image retrieval schemes are based on different image databases according to the application. Also, some schemes utilize encrypted image feature, but other schemes employ the hash code of the image feature.

In the schemes under vector space model, such as Yuan [20], a skew index tree leads to $O(\log N \sim N)$ search time. Xia [19] uses local-sensitive hash (LSH) to construct a prefilter table. Then, with corresponding image feature, the cloud carries out a refinement of candidate results. Thus, it is a sublinear scheme. Cheng [22, 23] does not build an index tree, so the retrieval efficiency is $O(N)$. Weng [6, 7] uses a hash table to improve the search efficiency (*i.e.*, $O(1 \sim N)$). In short, the index tree or hash table is a trade-off between space and efficiency. Although Weng's scheme [7] and proposed scheme both protect query pattern and access pattern, the hash code and feature vector are, respectively, used. Specifically, Weng [7] omits certain bits of the hash code of query image and returns multiple candidates to the user for refinement. The user is involved in completing retrieval and obtaining the accurate result. In our scheme, the user can set β as α_1 or α_2 to retrieve exact query image through different paths. Unlike [7], our scheme does not need the user to participate in the feature comparison. However, to protect

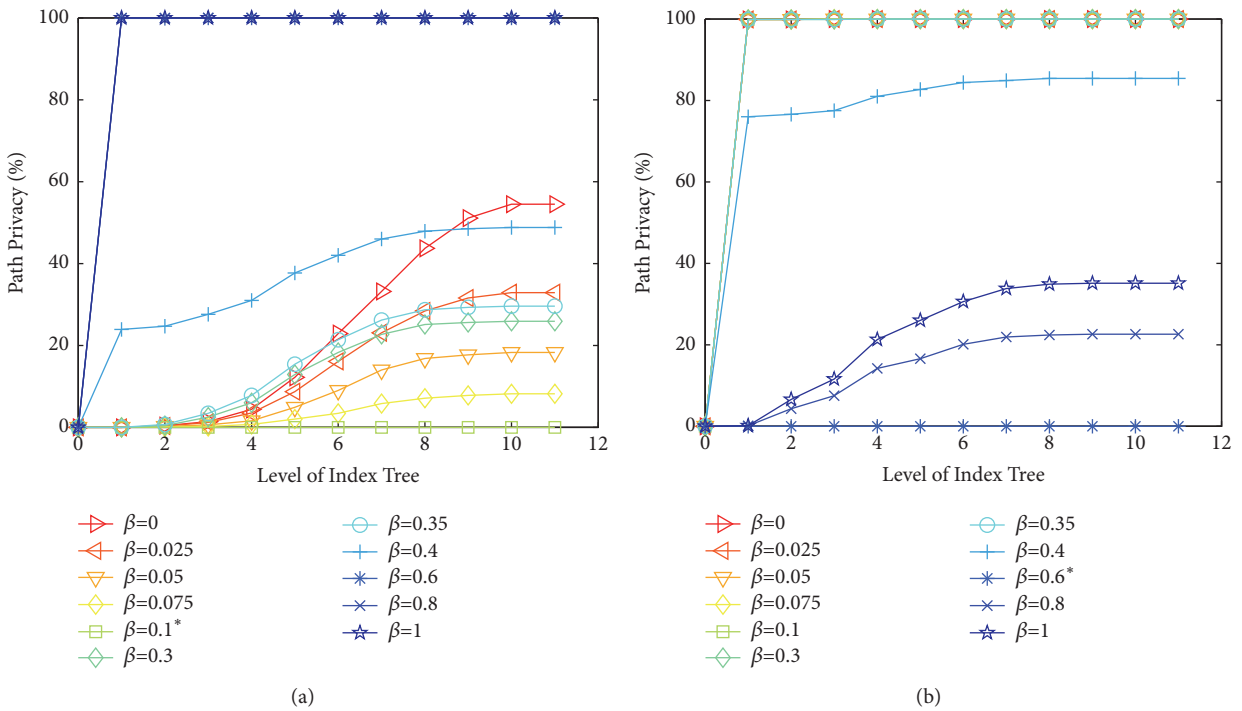


Figure 10: Path privacy of different β: (a) $\beta = 0.1$ is the baseline and (b) $\beta = 0.6$ is the baseline.

Table 4: Comparison of encrypted image retrieval schemes.

Goals	Yuan [20]	Xia [19]	Weng [6]	Weng [7]	Cheng [22, 23]	Proposed
Efficiency	$O(\log N \sim N)$		$O(1 \sim N)$		$O(N)$	$O(\log 2N)$
Query Pattern	√	√	×	√	×	√
Access Pattern	×	×	×	√	×	√

access pattern and alleviate computation burden on the user, we copy the image database and cost more storage space in the cloud. Fortunately, the retrieval efficiency is $O(\log 2N)$.

5. Conclusion

This paper introduces a secure and efficient image retrieval scheme over encrypted cloud data. In this scheme, we propose a novel clustering algorithm, *i.e.*, BBC, to build a balanced index tree, namely, BBT. Thus, our scheme can achieve logarithmic search time. Firstly, to support effective image retrieval in the ciphertext, we employ ASPE to encrypt the index of the image database and feature of the query image. In this case, the proposed SBBT is secure in the known ciphertext model. Secondly, to resist the statistical attack on known background model, we copy the database and build a new SBBT of the original database and the copied database. After that, adjusting the weighted factor of integrated feature, the search results of a query image are different leaf nodes. Finally, through merging the subtree of SBBT, the leaf node of new SBBT contains multiple encrypted images. Therefore, the one-to-one relationship between the query image and the return image is broken. In short, the SBBT can protect query pattern and access pattern. Furthermore, proposed BBC and SBBT can be used as independent tools for encrypted document retrieval.

In future, we will improve the ASPE to support multiuser scenario, where a dishonest user may try to reveal the encrypted query of other users. It is also a meaningful work to generate a generic, high-semantic feature.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (U1636206, 61602295, 61525203, and 61472235), the Shanghai Dawn Scholar Plan (14SG36), the Shanghai Excellent Academic Leader Plan (16XD1401200), the Natural Science Foundation of Fujian Province (2017J01502), and the Scientific Research Foundation of Fuzhou University (510483).

References

[1] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," *IEEE Internet Computing*, vol. 16, no. 1, pp. 69–73, 2012.

[2] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the 41st annual ACM symposium on Theory of Computing (STOC '09)*, pp. 169–178, ACM, Bethesda, Md, USA, 2009.

[3] C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling secure and efficient ranked keyword search over outsourced cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 8, pp. 1467–1479, 2012.

[4] K. Li, W. Zhang, C. Yang, and N. Yu, "Security Analysis on One-to-Many Order Preserving Encryption-Based Cloud Data Search," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1918–1926, 2015.

[5] W. Lu, A. Swaminathan, A. L. Varna, M. Wu et al., "Enabling search over encrypted multimedia databases," *Media Forensics and Security*, vol. 7254, 2009.

[6] L. Weng, L. Amsaleg, A. Morton, and S. Marchand-Maillet, "A privacy-preserving framework for large-scale content-based information retrieval," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, pp. 152–167, 2015.

[7] L. Weng, L. Amsaleg, and T. Furon, "Privacy-preserving outsourced media search," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 10, pp. 2738–2751, 2016.

[8] W. K. Wong, D. W. Cheung, B. Kao, and N. Mamoulis, "Secure kNN computation on encrypted databases," in *Proceedings of the International Conference on Management of Data and 28th Symposium on Principles of Database Systems (SIGMOD-PODS '09)*, pp. 139–152, Providence, RI, USA, July 2009.

[9] P. Williams, R. Sion, and B. Carbunar, "Building castles out of mud: practical access pattern privacy and correctness on untrusted storage," in *Proceedings of the 15th ACM Conference on Computer and Communications Security, CCS'08*, pp. 139–148, October 2008.

[10] http://wang.ist.psu.edu/docs/related/.

[11] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on IEEE*, pp. 44–55, IEEE, 2000.

[12] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Advances in cryptology—EUROCRYPT 2004*, vol. 3027 of *Lecture Notes in Comput. Sci.*, pp. 506–522, Springer, Berlin, 2004.

[13] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 222–233, 2014.

[14] W. Sun, B. Wang, N. Cao et al., "Verifiable privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 11, pp. 3025–3035, 2014.

[15] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 340–352, 2016.

[16] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft, "Privacy-Preserving Face Recognition," in *Privacy Enhancing Technologies*, vol. 5672 of *Lecture Notes in Computer Science*, pp. 235–253, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[17] W. Lu, A. L. Varna, A. Swaminathan, and M. Wu, "Secure image retrieval through feature protection," in *Proceedings of the ICASSP 2009 - 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1533–1536, Taipei, Taiwan, April 2009.

[18] W. Lu, A. L. Varna, and M. Wu, "Confidentiality-preserving image search: A comparative study between homomorphic encryption and distance-preserving randomization," *IEEE Access*, vol. 2, pp. 125–141, 2014.

[19] Z. Xia, N. N. Xiong, A. V. Vasilakos, and X. Sun, "EPCBIR: An efficient and privacy-preserving content-based image retrieval scheme in cloud computing," *Information Sciences*, vol. 387, pp. 195–204, 2017.

[20] J. Yuan, S. Yu, and L. Guo, "SEISA: Secure and efficient encrypted image search with access control," in *Proceedings of the IEEE INFOCOM 2015 - IEEE Conference on Computer Communications*, pp. 2083–2091, Kowloon, Hong Kong, April 2015.

[21] X. Zhang and H. Cheng, "Histogram-based retrieval for encrypted jpeg images," in *Proceedings of the in Signal and Information Processing (ChinaSIP, 2014 IEEE China Summit International Conference on. IEEE*, pp. 446–449, 2014.

[22] H. Cheng, X. Zhang, and J. Yu, "AC-coefficient histogram-based retrieval for encrypted JPEG images," *Multimedia Tools and Applications*, vol. 75, no. 21, pp. 13791–13803, 2016.

[23] H. Cheng, X. Zhang, J. Yu, and Y. Zhang, "Encrypted JPEG image retrieval using block-wise feature comparison," *Journal of Visual Communication and Image Representation*, vol. 40, pp. 111–117, 2016.

[24] Y. Xu, J. Gong, L. Xiong, Z. Xu, J. Wang, and Y. Q. Shi, "A privacy-preserving content-based image retrieval method in cloud environment," *Journal of Visual Communication Image Representation*, vol. 43, no. C, pp. 164–172, 2017.

[25] Z. Qian, H. Zhou, X. Zhang, and W. Zhang, "Separable reversible data hiding in encrypted jpeg bitstreams," *IEEE Transactions on Dependable Secure Computing*, no. 99, pp. 1-1, 2016.

[26] Z. Qian, H. Xu, X. Luo, and X. Zhang, "framework of reversible data hiding in encrypted jpeg bitstreams," *IEEE Transactions on Circuits Systems for Video Technology*, no. 99, pp. 1-1, 2018.

[27] A. Banerjee and J. Ghosh, "Frequency-sensitive competitive learning for scalable balanced clustering on high-dimensional hyperspheres," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 15, no. 3, pp. 702–719, 2004.

[28] A. S. Galanopoulos, R. L. Moses, and S. C. Ahalt, "Diffusion approximation of frequency sensitive competitive learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 8, no. 5, pp. 1026–1030, 1997.

[29] C. T. Althoff, A. Ulges, and A. Dengel, "Balanced clustering for content-based image browsing," in *in GI-Informatiktage, 2011., GI-Informatiktage (Informatik-2011)*, Gesellschaft fr Informatik, Bonn, Germany.

[30] M. I. Malinen and P. Fränti, "Balanced K-means for clustering," in *Structural, Syntactic, and Statistical Pattern Recognition*, P. Fränti, G. Brown, M. Loog, F. Escolano, and M. Pelillo, Eds., vol. 8621 of *Lecture Notes in Computer Science*, pp. 32–41, Springer, Berlin, Germany, 2014.

[31] https://en.wikipedia.org/wiki/Assignment_problem.

[32] R. Jonker and A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," *Computing: Archives for Scientific Computing*, vol. 38, no. 4, pp. 325–340, 1987.

[33] W. Jones, A. Chawdhary, and A. King, "Optimising the Volgenant–Jonker algorithm for approximating graph edit distance," *Pattern Recognition Letters*, vol. 87, pp. 47–54, 2017.

[34] G. Schaefer, "Fast JPEG compressed domain image retrieval," in *Proceedings of the 5th International Conference on Multimedia Computing and Systems, ICMCS 2016*, pp. 148–150, mar, October 2016.

[35] https://en.wikipedia.org/wiki/Dunn_index.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 5841967, 12 pages
https://doi.org/10.1155/2018/5841967

Research Article

Outsourcing Set Intersection Computation Based on Bloom Filter for Privacy Preservation in Multimedia Processing

Hongliang Zhu,[1,2] **Meiqi Chen**,[1,2] **Maohua Sun**,[3] **Xin Liao**,[4] **and Lei Hu**[3]

[1]*Beijing University of Posts and Telecommunications, Beijing, China*
[2]*National Engineering Laboratory for Disaster Backup and Recovery, Beijing, China*
[3]*Capital University of Economics and Business, Beijing, China*
[4]*Hunan University, Hunan, China*

Correspondence should be addressed to Maohua Sun; sunmaohua@cueb.edu.cn

Received 26 September 2017; Accepted 22 February 2018; Published 5 April 2018

Academic Editor: Xinpeng Zhang

With the development of cloud computing, the advantages of low cost and high computation ability meet the demands of complicated computation of multimedia processing. Outsourcing computation of cloud could enable users with limited computing resources to store and process distributed multimedia application data without installing multimedia application software in local computer terminals, but the main problem is how to protect the security of user data in untrusted public cloud services. In recent years, the privacy-preserving outsourcing computation is one of the most common methods to solve the security problems of cloud computing. However, the existing computation cannot meet the needs for the large number of nodes and the dynamic topologies. In this paper, we introduce a novel privacy-preserving outsourcing computation method which combines GM homomorphic encryption scheme and Bloom filter together to solve this problem and propose a new privacy-preserving outsourcing set intersection computation protocol. Results show that the new protocol resolves the privacy-preserving outsourcing set intersection computation problem without increasing the complexity and the false positive probability. Besides, the number of participants, the size of input secret sets, and the online time of participants are not limited.

1. Introduction

Network multimedia comes into fashion in the form of services; there are many methods to protect multimedia data in traditional service mode, such as steganography [1, 2] and data embedding [3]. By providing diversified media services, a new service mode, multimedia computing, has become an attractive technology to generate, edit, process, and search various media contents, like images, videos, audios, graphs, and so on [4]. For purposes of multimedia applications and services based on Internet and mobile Internet, it needs lots of computation resources so as to serve millions of netizens and wireless users, which means a large demand for multimedia cloud computing. Cloud computing is a new computing mode which could provide kinds of data service based on its computational resources. As an important application of cloud computing, outsourcing computation could enable users with narrow computing power to outsource complex function calculations to cloud servers and could guarantee the correctness of outputs and privacy of both inputs and outputs. So in this new multimedia computation mode based on cloud computing, users can store and process distributed multimedia application data without installing multimedia application software in local computer terminals to ease off the load of maintenance and updating. With regard to the large amount of computation of sites, data, and attribute dimensions, we introduce PSI into cloud computing. There is a wide range of applications where Secure Multiparty Computation is introduced into cloud computing considering the privacy-preserving algorithms.

Private Set Intersection (PSI) is an important research branch of Secure Multiparty Computation (SMC), which is a research hotspot in recent years. Privacy-preserving set operation can be described as the situation that multiple

participants wish to complete set intersection computation based on their private secret sets, and they cannot receive additional information other than results after computation. In PSI research model, participants complete secure computation using their private computing resources through mutual communication. Privacy preservation has become a key factor in extending the application of cloud computing, and it is the current research trend. In order to implement PSI in cloud computing successfully to solve the problems mentioned, Privacy-preserving Outsourcing Set Intersection (POSI) is proposed.

1.1. Contributions. The work we have completed in this paper contributes to the study and development of privacy preservation as well as outsourcing computation in several aspects as follows:

(1) We summarize system models of current privacy-preserving technology and propose a system model of privacy-preserving outsourcing computation protocol in cloud computing. It can guarantee the security and correctness of the data.

(2) We study and implement a privacy-preserving set intersection protocol based on GM homomorphic encryption scheme and Bloom filter, and the proposed protocol is proved to be significant.

(3) In detail, the protocol has some characteristics as follows:

(a) The participant encrypts the secret set locally and consigns ciphertexts to the server who completes the outsourcing computation, but the server is unable to know about the participant's secret set because it does not have the private key to decrypt them. So it guarantees security. Participants can check whether one or more items of data are in the intersection.

(b) The protocol does not require sizes of participants' sets being the same as well as public compared to the existing PSI protocols [5–14].

(c) The protocol can implement secure outsourcing computation of more than two participants' secret set intersection without the limitation that participants should be online at the same time, while the existing secure outsourcing computation protocol of set intersection [15] can only solve the situation with two participants online.

(d) The protocol has a lower probability of communication complexity and false positive error verification compared with [15].

(e) The protocol is safe under the semihonest model. We provide a full proof with simulation based security. There are two reasons why we do not design a protocol in the malicious model. (1) The proposed algorithm can be packaged as software. When we use peripheral secure technology to make the software difficult to be tampered with, semihonest model is safe enough. (2) Converting protocol in semihonest model to malicious model is an independent research topic with plenty of achievements currently. If necessary, the algorithm can be converted into one in the malicious model based on the existing research findings.

1.2. Related Work. The following sections describe the research progress of privacy-preserving set intersection and outsourcing computation.

1.2.1. Secure Multiparty Computation. Protocols for Secure Multiparty Computation enable a set of parties to carry out a joint computation on private inputs, without revealing anything but the output. Over the past decade, there has been a major research effort to develop Secure Multiparty Computation. Zhou et al. [16] proposed a secure multiparty subset protocol using the Bloom filter and homomorphic encryption scheme. However, their protocol may yield a false positive. Liu et al. [17] proposed an information-theoretically secure protocol to solve the multiparty millionaires' problem using the vectorization and secret splitting methods; their protocol can resist collusion attacks. Sun et al. [18] proposed a secure outsourcing multiparty computation protocol on lattice-based encrypted data in two-cloud-servers scenario. Their protocol was completely noninteractive between any users, and both of the computation and the communication complexities of each user in our solution were independent of the computing function.

1.2.2. Privacy-Preserving Set Intersection. Privacy-preserving set intersection is a research focus in the field of cryptography. The PSI problem can be described as the situation that multiple participants wish to complete the set intersection computation based on their private secret sets, and they cannot receive additional information other than results after the computation.

According to different implementation principles, we can classify research findings of PSI into the following four types.

(i) The Oblivious Polynomial Evaluation Based Protocols. Oblivious polynomial evaluation is the first method to implement the PSI protocol. Dachman-Soled et al. [5] implemented a PSI protocol in malicious models using Shamir Threshold Secret Sharing technology. The computational complexity of the algorithm is $O(mnk \log n + mk^2 \log^2 n)$, and the communication complexity is $O(nk + mk^2 \log^2 n)$, in which k is the secure parameter, while m and n are the sizes of the participant input sets.

(ii) The Oblivious Pseudorandom Function Based Protocols. At the TCC Conference in 2008, Hazay and Lindell [6] proposed a privacy-preserving set intersection protocol based on the oblivious pseudorandom function. The scheme is safe in the weakly malicious model, which means participants' malicious behavior will be found with a high probability. Later, Hazay and Nissim [7] used zero-knowledge proof and perfectly hiding commitment scheme to implement a privacy-preserving set intersection protocol in malicious

model. The communication complexity of the algorithm is $O(m + n(\log\log m + \sigma))$, and the computational complexity is $O(m + n\sigma)$, in which m and n are the sizes of the two sets. σ is elements' largest binary number of bits in the set. Jarecki and Liu [8] proposed a privacy-preserving intersection protocol under the CRS model based on the Decisional-q-Diffie-Hellman Inversion hypothesis. De Cristofaro and Tsudik [9, 10] proposed a privacy-preserving intersection operation protocol with linear complexity under the semihonest model based on the One-More-Gap-DH hypothesis. Later, De Cristofaro et al. [11] proposed an efficient privacy-preserving intersection operation scheme against malicious attackers based on the DDH hypothesis.

(iii) The Bloom Filter Based Protocols. Bloom filter is a new data structure introduced in recent years, of which the structure is similar to bit-map. Compared to bit-map, Bloom filter saves more space and can quickly judge whether an element is in a set. But there is a certain rate of error recognition in this method. In 2012, Many et al. [12] introduced Bloom filter into the privacy-preserving intersection operations. They used the secure multipart multiplication protocol to get the Bloom filter vector corresponding to the intersection of participants and then get the set intersection. However, the algorithm is insecure because the intersection Bloom filter vector leaked information of each participant's set. In 2013, Dong et al. [13] designed a more efficient privacy-preserving intersection protocol based on Bloom filter, using secret sharing and oblivious transfer. Take the privacy-preserving intersection operation protocol under semihonest model as an example; the scheme Dong et al. [13] proposed requires $2(k + k\log_2 e)n$ times of hash operations and hundreds of public key operations. In 2014, Pinkas and Schneider [14] designed a random confusion Bloom filter to optimize efficiency of the protocol of Dong et al. [13], using oblivious extension protocol.

(iv) The Garbled-Circuit Technology Based Protocols. Using garbled-circuit technology to solve privacy-preserving problems is a common method of Secure Multiparty Computation, but many references in the past suggest that the method is less efficient. In 2012, Huang et al. [19] designed the intersection-specific circuit based on the idea of *"Sort-Compare-Shuffle"* and implemented the privacy-preserving intersection operation protocol using Yao's generic garbled-circuit method. The experimental results of Huang et al. [19] show that the scheme of De Cristofaro and Tsudik [9, 10] is more efficient when the security level is low, and as the security level increases, the scheme of Huang et al. [19] is significantly better than that of De Cristofaro and Tsudik [9, 10] considering efficiency of the program. In 2014, Pinkas and Schneider [14] optimized the GMW scheme using oblivious extension protocol, used the optimized GMW scheme to evaluate the intersecting circuit designed by Huang et al. [19], and implemented a more efficient privacy-preserving intersection operation protocol on Boolean circuits. The computational complexity is $18n\sigma\log n$ times of symmetric encryption operations, while the communication complexity is $O(6nk\sigma\log n)$, in which k is the secure parameter.

1.2.3. Privacy-Preserving Outsourcing Computation. Outsourcing computation in multimedia processing is an emerging technology in recent years. Although the study of privacy-preserving outsourcing computation has just started, it is the current research hot spot.

At the CRYPTO conference in 2010, Gennaro et al. [20] proposed privacy-preserving issues in verifiable computations and designed a privacy-preserving outsourcing computation protocol that can achieve verifiable efficiency based on the homomorphic encryption technology. In 2011, Mohassel [21] designed a noninteractive security outsourcing computation protocol on linear algebraic operations based on homomorphic encryption. In 2013, Parno et al. [22] designed the Pinocchio system which implemented efficient outsourcing computation, but the system did not take into account the privacy-preserving issues of the information input by participants; Schoenmakers et al. [23] designed the Trinocchio system to solve the leakage of Pinocchio system, enabling efficient verifiable secure outsourcing computation. In the same year, Peter et al. [24] designed a secure outsourcing computation protocol for common functional functions, using a dual decryption mechanism scheme with additive homology, and implemented an efficient face recognition system in cloud computing environment based on this protocol. In 2013, Xing et al. [25] constructed a verifiable secure outsourcing computation protocol using the blind product as a matrix product, matrix determinant, and matrix inverse. The security does not depend on any cryptographic assumptions. In 2014, Hu and Tang [26] implemented the secure outsourcing protocol of multiplication on the elliptic curve in the cloud computing environment, which could effectively accelerate the efficiency of signature verification.

Although the PSI protocol has implemented plenty of achievements, they cannot be converted to be used in privacy-preserving set intersection outsourcing computation directly. At present, the research on the privacy-preserving issues in set intersection outsourcing computation has just started, while the findings are still not enough. According to our searching results, Kerschbaum [15] proposed a set intersection secure outsourcing protocol based on SYY homomorphic encryption scheme and Bloom filter. However, the protocol has the following problems: (1) the protocol only solves the secure outsourcing computation of two participants' set intersections, while one of the participants needs to be both common participant and server at the same time; (2) during the process of the protocol, all the participants are required to be online at the same time; (3) there is a high probability of false positive error judgement in the protocol.

1.3. Organizational Structure. In the second session, we introduce secure definition in the scheme and the underlying cryptographic tools. We show the system model in Section 3 and present the privacy-preserving set intersection computation protocol which can be applied into cloud computing in Section 4. In Section 5, we give the correct proof of the protocol, error probability analysis, and security proof as well as efficiency analysis and comparison. Finally, we summarize

prospects of our protocol's application in multimedia processing based on cloud computing in the Conclusion.

2. Background

2.1. Secure Model and Secure Definition. Since the protocol proposed in this paper belongs to one kind of the Secure Multiparty Computation protocols, we use secure models and secure definitions of Secure Multiparty Computation protocols.

Participants of Secure Multiparty Computation are classified into honest participants, semihonest participants, and malicious participants. During the implementation of the protocol, honest participants completely comply with the protocol, with no provision of false data, leakage, eavesdropping, and suspension of the protocol; semihonest participants will finish each step following the requirements of the implementation without behaviors mentioned earlier, but they will keep all the information they collected in order to judge secret messages of other participants; malicious participants completely ignore the requirements of the protocol. They may provide false data, leak all the information they collect, eavesdrop, or even suspend protocols.

The semihonest model is safe and widely used in Secure Multiparty Computation. The model can be intuitively understood as the situation that if a semihonest participant can directly use their input and output of protocols to obtain any information he can reach in the implementation of the protocol by a separate simulation of the entire protocol implementation process, it can be guaranteed in the protocol that the input is private. If a computation protocol can be simulated like this, participants cannot obtain valuable information from the execution of the protocol, and such protocol is safe.

Definition 1 (private computation under semihonest model). In the implementation of protocol Π, the information that participants P_1 and P_2 obtain is recorded as

$$\begin{aligned} \mathrm{VIEW}_1^{\Pi}(x, y) &= \left(x, r^1, m_1^1, m_2^1, \ldots, m_t^1\right) \\ \mathrm{VIEW}_2^{\Pi}(x, y) &= \left(x, r^2, m_1^2, m_2^2, \ldots, m_t^2\right). \end{aligned} \tag{1}$$

In the equations, r^i represents the random number P_i generates and m_j^i represents the jth message P_i receives. After the protocol ends, the output of participant P_i is recorded as $\mathrm{OUTPUT}_i^{\Pi}(x, y)$. We can see that in fact $\mathrm{OUTPUT}_i^{\Pi}(x, y)$ is a part of $\mathrm{VIEW}_i^{\Pi}(x, y)$.

As for the deterministic function f, we can say that protocol Π computes f under the semihonest model privately if and only if probability polynomial time algorithms S_1 and S_2 exist, and it conforms to the equations:

$$\begin{aligned} \left\{S_1(x, f_1(x, y))\right\}_{x,y\in\{0,1\}^*} &\overset{c}{\equiv} \left\{\mathrm{VIEW}_1^{\Pi}(x, y)\right\}_{x,y\in\{0,1\}^*} \\ \left\{S_2(y, f_2(x, y))\right\}_{x,y\in\{0,1\}^*} &\overset{c}{\equiv} \left\{\mathrm{VIEW}_2^{\Pi}(x, y)\right\}_{x,y\in\{0,1\}^*} \end{aligned} \tag{2}$$

for $|x| = |y|$.

2.2. GM Homomorphic Encryption. A high-level description of Gentry's scheme is as follows. The scheme is based on identifying ideals I in polynomial quotient rings $Z[x]/(f(x))$ (with $^\circ(f) = n$) with euclidean lattices $L_I \subseteq R^n$ by mapping eachresidue polynomial $r(x) = a_0 + \cdots + a_{n-1}x^{n-1}$ to its vector of coefficients $(a_0, \ldots, a_{n-1})$. Gentry calls these objects ideal lattices. Ideal lattices provide additive and multiplicative homomorphisms modulo a public key ideal. We obtain an encryption procedure Encrypt such that $\mathrm{Encrypt}(x_1) + \mathrm{Encrypt}(x_2) = \mathrm{Encrypt}(x_1 + x_2)$ and $\mathrm{Encrypt}(x_1) \cdot \mathrm{Encrypt}(x_2) = \mathrm{Encrypt}(x_1 \cdot x_2)$. Therefore, any circuit C with efficient description can be evaluated homomorphically. However, this somewhat fully homomorphic scheme (SWHE) is not perfect. Due to the noisy nature of the scheme, with each homomorphic gate evaluation the noise term in the partial result grows. After the evaluation of only a logarithmic depth circuit, the decryption fails to recover the correct result. To make the scheme work, Gentry uses a number of tricks. He introduces a reencryption procedure called Recrypt that takes a noisy ciphertext and returns a noise-reduced version. In a brilliant move, Gentry manages to obtain Recrypt again from the SWHE scheme by simply homomorphically evaluating the decryption circuit using encrypted secret key bits on the noisy ciphertext. To make this work, the SWHE needs to be able to handle circuits that are deeper than its own decryption circuit before the level of noise becomes too large. SWHE schemes with this property are called bootstrappable.

2.3. XOR Secret Sharing. The secret publisher converts his secret s into n subsecrets and sends them to other participants. The secret sharing scheme is called a (t, n) threshold secret sharing scheme when they can recover the secret s if and only if at least t participants contribute their specific subsecrets.

When the threshold $t = n$, the XOR secret sharing scheme proposed by Ishai et al. [27] is widely used. The details are as follows.

Participants. The participants are secret publisher D and n participants $P_1, P_2, \ldots, P_n$.

Input. The input is secret s that secret publisher D inputs.

Secret Sharing. (1) Secret publisher generates $n - 1$ random numbers $r_1, r_2, \ldots, r_{n-1}$, and the length of each is $|s|$.

(2) Secret publisher calculates the nth secret:

$$r_n = r_1 \oplus r_2 \oplus \cdots \oplus r_{n-1} \oplus s. \tag{3}$$

$\{r_1, r_2, \ldots, r_n\}$ compose subsecrets of s.

(3) As for $i = 1, 2, \ldots, n$, the secret publisher sends subsecret r_i to P_i.

Secret Recovery. When it is necessary to recover the secret s, n participants $P_1, P_2, \ldots, P_n$ contribute their own subsecrets and do the following operation:

$$s = r_1 \oplus r_2 \oplus \cdots \oplus r_n. \tag{4}$$

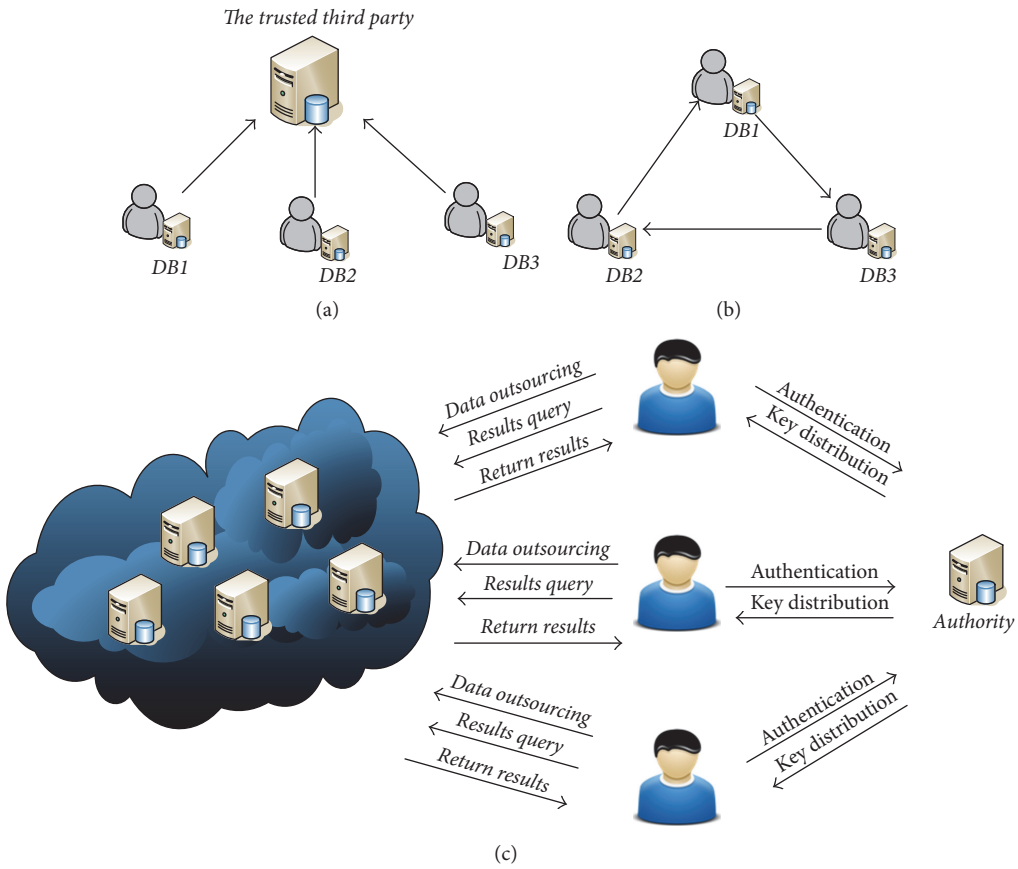


FIGURE 1: Secure Multiparty Computation model.

2.4. Bloom Filter. The Bloom filter [28] set is a data structure used to judge whether an element is in a set. A Bloom filter contains several hash functions hash_i $(i = 1, 2, \ldots, k)$ and a Bloom filter set BF. When building a Bloom filter set, use the hash function first to map the data x which is to be inserted to the kth position of BF, and then set the data on those positions to 1. When all the data is inserted, the Bloom filter set is completed. When verifying whether a data y is in a set, use the hash function first to map y to the kth position of BF. If the values of these k data bits are all 1, there is a great possibility that y is in the set; otherwise it is not in for sure.

3. System Model

A trusted third party is a model that solves the privacy-preserving problem in distributed computation, as shown in Figure 1(a). However, it is difficult to find a completely credible third party in real life, so this system model is rarely used at present. Currently in the field of Secure Multiparty Computation, a widely used system model is shown in Figure 1(b). It needs a number of participants to complete the secure computation of a certain function through information interaction instead of a trusted third party. To achieve the privacy-preserving outsourcing computation, we can not use the model of Figure 1(a) directly because a completely trusted third party does not exist; nor can we use Figure 1(b) model directly, because a lot of computation is consigned to the server.

The system model we use is shown in Figure 1(c). Although a completely trusted third party does not exist, the authority (for example, an authoritative digital certificate authority) does exist. Before the protocol is formally conducted, the participant will be authenticated by the authority first. If the audit passes, the authority sends the system key to participants. In the process of the protocol, participants use the public key to encrypt their own secret sets and consign the ciphertexts to the server. The server computes all the ciphertexts it takes over and saves them. Then every participant may request to verify whether one or some of data is in the intersection of the sets at any time.

Then we describe the behavior pattern of all participants and the server after the authentication in the system model applied in cloud computing shown in Figure 1(c). In this system model, the problem to be solved can be described as follows: m participants $P_1, P_2, \ldots, P_m$ hold secret messages separately, and the participant completes the operation $f(s_1, s_2, \ldots, s_m)$ by leasing a server with powerful computing resources. In terms of security, the participant wishes others not to be informed of other useful information except the results after completing the computation; the server is unable to know the participants' secret messages $s_1, s_2, \ldots, s_m$, and the server can not know the result $f(s_1, s_2, \ldots, s_m)$.

We divide the information interaction between participants and servers into three stages: preprocessing, outsourcing computation, and results query. In the preprocessing stage, behavior of participants and servers is as follows:

$$P_i : S_i \xrightarrow{\pi} \pi(S_i)$$
$$P_i \xrightarrow{\pi(S_i)} \text{Server}. \tag{5}$$

As for each participant P_i, the first step is converting S_i to $\pi(S_i)$ through a certain operation π locally and then sending $\pi(S_i)$ to the server. The operation π should be unidirectional; otherwise the server will be informed of the participant's secret message.

In the outsourcing computation stage, the server converts all the outsourced data of participants to data sources $F(\pi(S_1), \pi(S_2), \ldots)$ of the results query stage through a certain operation F. We can use the following equation to represent the server's behavior pattern:

$$\text{Server} : \{\pi(S_1), \pi(S_2), \ldots\}$$
$$\xrightarrow{F} F(\pi(S_1), \pi(S_2), \ldots). \tag{6}$$

In the results query stage, the behavior pattern of the inquirer P_i and the server is as follows:

$$P_i \xrightarrow{Q_i} \text{Server}$$
$$\text{Server} : \{F(\pi(S_1), \pi(S_2), \ldots), Q_i\} \xrightarrow{\Delta} R_i$$
$$\text{Server} \xrightarrow{R_i} P_i$$
$$P_i : R_i \xrightarrow{\Phi} \Phi(R_i). \tag{7}$$

It means that the participant constructs query data Q_i and sends it to the server first. The server conducts operation Δ using the result $F(\pi(S_1), \pi(S_2), \ldots)$ of the previous stage and Q_i as input and then gets the result R_i and sends it to P_i. Participants P_i conduct a certain decryption Φ to R_i and gets the final result $\Phi(R_i)$. The correctness requirement of this model is $\Phi(R_i) = f(s_1, s_2, \ldots, s_m)$.

4. Privacy-Preserving Set Intersection Outsourcing Computation Protocol

In this section, we design the set intersection secure outsourcing protocol in accordance with three stages of preprocessing, outsourcing computation, and results query. We state in this section that participants and authorities have completed authentication and key distribution in Figure 1(c).

The protocol uses the following symbols: P represents all participants, P_i represents the ith participant, and m represents the number of participants. The secret set of participant P_i is S_i, and its size is represented by $|S_i|$. GF_i represents the Bloom filter set of participant P_i and $\text{GF}_i(j)$ represents the jth element in the Bloom filter set. The number of elements in Bloom filter is t while the number of hash functions used in the process of forming Bloom filters is k. CGF_i represents ciphertexts corresponding to Bloom filter set of P_i. The length of ciphertexts in XOR secret sharing is n, and the length of ciphertexts in GM encryption algorithm is l.

4.1. Preprocessing. In the preprocessing stage, the participant generates Bloom filter set corresponding to his private secret set. In order to reduce the probability of false positives, participants share data of secret sets to the k elements of Bloom filter, using XOR secret sharing. We can get the positions of the k elements by hashing. In order to achieve privacy preservation, participants use the GM algorithm to do encryption operations on their respective Bloom filter sets and send them to server. The preprocessing protocol process is as shown in Algorithm 1.

After the previous computation, participant P_i gets encrypted Bloom filter set CBF_i, and P_i needs to send CBF_i to server to complete the data outsourcing.

4.2. Outsourcing Computation. After the previous stage ends, server receives the encrypted Bloom filter sets CBF_i ($i = 0, 1, \ldots, m$) that all the participants send. Server does the following operations in the outsourcing computation stage:

```
for (j = 0; j < tn; j + +)
{
    CBF(j) = ∏_{i=0}^{m} CBF_i(j)
}
```

4.3. Results Query. In the results query stage as shown in Algorithm 2, participants query whether one of more data is in the intersection.

5. Theoretical Analysis

In this section, we analyze the correctness, error probability, security, and performance of the protocol and compare the results with the existing ones.

5.1. Correctness

Theorem 2. *When the participant is able to construct the Bloom filter successfully, the proposed set intersection secure outsourcing protocol is correct.*

Proof. $\forall q \in I$, then, for $i = 1, 2, \ldots, m$, there is

$$q \in S_i \text{ as well as } \bigoplus_{j=0}^{k-1} \text{BF}_i\left(\text{hash}_j(q)\right) = q. \tag{8}$$

Because GM algorithm has a characteristic of XOR homomorphies, for $j = 0, 1, \ldots, k-1$, there is

$$\text{CBF}\left(\text{hash}_j(q)\right) = \text{Enc}\left(\bigoplus_{i=1}^{m} \text{BF}_i\left(\text{hash}_j(q)\right)\right). \tag{9}$$

```
Participants: P_i
Input: the input sets of P_i: S_i
System parameters: k hash functions hash_i (i = 1, 2, ..., k)
for (j = 0; j < t; j + +)
    BF_i(j) = NULL;
    //In the initial state, the Bloom filter set
    //of participants is empty
for (j = 0; j < |S_i|; j + +)
{
    π = NULL;
    for (δ = 0; δ < k; δ + +)
    {
        if (BF_i(hash_δ(S_i(j))) == NULL)
        {
            π = hash_δ(S_i(j));
BF_i(hash_δ(S_i(j))) = Random(0, 1)^n;
            //generate random numbers
            //with the length of n;
        }
        if (π == NULL)
            Return error1; //error
        else
BF_i(π) = S_i(j) ⊕ BF_i(hash_1(S_i(j))) ⊕
                BF_i(hash_2(S_i(j))) ⊕
                ··· ⊕ BF_i(hash_k(S_i(j)))
    }
}
ε = 0;
for (j = 0; j < t; j + +)
{
    if (BF_i(j) == NULL)
        BF_i(j) = Random(0, 1)^n;
    for (μ = 0; μ < n; μ + +)
    {
        ε = ε + 1;
        CBF_i(ε) = Enc(BF_i(j)_μ);
        //BF_i(j)_μ represents the μth
        //bit in BF_i(j);
    }
}
```

Algorithm 1: Preprocessing protocol.

When participants query whether q is in the set intersection, for $j = 0, 1, \ldots, k-1$, there is

$$\mathrm{QBF}\left(\mathrm{hash}_j\left(q\right)\right) = 1. \tag{10}$$

So participants use extended oblivious transfer protocol to get the set ABF, and there is

$$\mathrm{ABF}\left(\mathrm{hash}_j\left(q\right)\right) = \mathrm{CBF}\left(\mathrm{hash}_j\left(q\right)\right). \tag{11}$$

We can know from (9) and (11) that

$$\mathrm{Dec}\left(\mathrm{ABF}\left(\mathrm{hash}_j\left(q\right)\right)\right) = \bigoplus_{i=1}^{m} \mathrm{BF}_i\left(\mathrm{hash}_j\left(q\right)\right). \tag{12}$$

So

$$\begin{aligned}\pi &= \bigoplus_{\delta=0}^{k-1} \mathrm{Dec}\left(\mathrm{ABF}\left(\mathrm{hash}_\delta\left(q\right)\right)\right)\\ &= \bigoplus_{\delta=0}^{k-1}\left(\bigoplus_{i=1}^{m} \mathrm{BF}_i\left(\mathrm{hash}_\delta\left(q\right)\right)\right)\\ &= \bigoplus_{i=1}^{m}\left(\bigoplus_{\delta=0}^{k-1} \mathrm{BF}_i\left(\mathrm{hash}_\delta\left(q\right)\right)\right) = \bigoplus_{1}^{m} q.\end{aligned} \tag{13}$$

So when m is even, $\pi = \{0\}^n$; when m is odd, $\pi = q$.

Similarly, if $q \notin I$, when m is even, $\pi \neq \{0\}^n$; when m is odd, $\pi \neq q$.

The proof is finished. □

Participants: P_i, Server
Input: P_i inputs query set $Q = \{q_1, q_2, \ldots, q_\tau\}$; Server inputs the Bloom filter set CBF of intersection I.
Output: P_i gets query results $R = \{r_1, r_2, \ldots, r_\tau\}$, if $r_i = 1$, then data $q_i \in I$; otherwise $q_i \notin I$.
Step 1. Server generates random Bloom filter set RBF and Bloom filter set pair PBF following the steps below.

```
for (j = 0; j < tn; j++)
{
    RBF(j) = Random(0, 1)^l;
    PBF(i) = (RBF(j), CBF(j));
}
```

Step 2. P_i generates query Bloom filter set QBF according to query set Q.

```
for (j = 0; j < tn; j++)
    QBF(j) = 0;
for (j = 0; j < Q; j++)
{
    for (δ = 0; δ < k; δ++)
    {
        QBF(hash_δ(q_j)) = 1;
    }
}
```

Step 3. P_i and Server implement oblivious transfer protocol OT_l^t. P_i is set to be a Receiver and Server is set to be a Sender. The input of P_i is QBF while the input of Server is PBF. After the oblivious transfer protocol is completed, P_i gets the set ABF. When $\mathrm{QBF}(j) = 0$, $\mathrm{ABF}(j) = \mathrm{RBF}(j)$; when $\mathrm{QBF}(j) = 1$, $\mathrm{ABF}(j) = \mathrm{CBF}(j)$.
Step 4. P_i checks whether each element of Q is in the intersection following the steps below.

```
for (j = 0; j < τ; j++)
{
    π = {0}^n;
    for (δ = 0; δ < k; δ++)
    {
        π = π ⊕ Dec(ABF(hash_δ(q_j)))
    }
    if (m(mod2) == 0)
    {
        if (π == {0}^n)
            r_j = 1;
        else
            r_j = 0;
    }
    else
    {
        if (π == q_j)
            r_j = 1;
        else
            r_j = 0;
    }
}
```

Algorithm 2: Results query stage.

5.2. Error Probability

Theorem 3. *The probability that participant P_i constructs Bloom filter set based on XOR secret sharing successfully is*

$$P = 1 - p_1^h \times \left(1 + O\left(\frac{h}{p_1}\sqrt{\frac{\ln t - h \ln p_1}{t}}\right)\right) \tag{14}$$

in which $p_1 = 1 - (1 - 1/t)^{h(|S_i|-1)}$.

Proof. The necessary and sufficient condition that participants P_i are unable to map their data x of their secret sets to the Bloom filter set when building a Bloom filter set based on XOR secret sharing is that the k positions in Bloom filter data x gets after being mapped by k hash functions are occupied. And the necessary and sufficient condition of general Bloom filters with false positive authentication is that all the k positions data y gets after being mapped by k hash functions are all set to one. Thus, the probability of participant P_i being unable to construct a Bloom filter set based on XOR secret sharing is the same as the probability of a generic Bloom filter set with false positive. From [29] we can see that the probability is $p' = p_1^h \times (1 + O((h/p_1)\sqrt{(\ln t - h \ln p_1)/t}))$, in which $p_1 = 1 - (1 - 1/t)^{h(|S_i|-1)}$. Thus, the probability of participant P_i successfully constructing a Bloom filter set based on XOR secret sharing is $P = 1 - p_1^h \times (1 + O((h/p_1)\sqrt{(\ln t - h \ln p_1)/t}))$.

The proof is finished. □

Theorem 4. *After the participant constructs Bloom filter successfully, the false positive error probability is* $(1/2)^n$.

Proof. $\forall x \notin I$, when the result is $x \in I$, then there will be false positive verification. Consider the following matrix:

$$Z = \begin{bmatrix} \mathrm{BF}\_x_1^1 & \mathrm{BF}\_x_1^2 & \cdots & \mathrm{BF}\_x_1^k \\ \mathrm{BF}\_x_2^1 & \mathrm{BF}\_x_2^2 & \cdots & \mathrm{BF}\_x_2^k \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{BF}\_x_m^1 & \mathrm{BF}\_x_m^2 & \cdots & \mathrm{BF}\_x_m^k \end{bmatrix} \tag{15}$$

in which $\mathrm{BF}\_x_i^j = \mathrm{BF}_i(\mathrm{hash}_j(x))$.

If the number of participants m is even, $\bigoplus_{i=1}^m (\bigoplus_{j=1}^k \mathrm{BF}\_x_i^j) = \{0\}^n$. We can know from the process of construction that the probability is $(1/2)^n$; if it is odd, $\bigoplus_{i=1}^m (\bigoplus_{j=1}^k \mathrm{BF}\_x_i^j) = x$; the probability is also $(1/2)^n$.

In conclusion, the probability of false positive error is $(1/2)^n$ after the participant constructs the Bloom filter successfully in this scheme.

The proof is finished. □

5.3. Security

Theorem 5. *Assuming that the underlying GM homomorphic encryption scheme and the OT protocol are secure under the semihonest model, the proposed set intersection security outsourcing protocol safely implements the outsourcing computation of the participant's secret set under the semihonest model.*

Proof. The protocol proposed in this paper is asymmetric, which means only the participant is informed of the result. So

$$\begin{aligned} & f(S_1, S_2, \ldots, S_m, Q) \\ & \quad \stackrel{\text{def}}{=} \left(f_P(S_1, S_2, \ldots, S_m, Q), f_S(S_1, S_2, \ldots, S_m)\right) \\ & \quad \stackrel{\text{def}}{=} \left(f_P(S_1, S_2, \ldots, S_m, Q), \Lambda\right) \end{aligned} \tag{16}$$

in which Λ means empty strings and π means the proposed security outsourcing protocol. The security analysis is performed from the server view and the participant view, respectively, as follows.

Server View. First analyze the situation where the serve is attacked. During the execution of the protocol π, the server's view is

$$\begin{aligned} & \text{view}_s^{\pi}(S_1, S_2, \ldots, S_m, Q) \\ & = \left\{\Lambda, r^s, \text{CBF}_1, \text{CBF}_2, \ldots, \text{CBF}_m, \text{CBF}, \text{PBF}, \text{view}_s^{\text{OT}}\right\} \end{aligned} \tag{17}$$

in which Λ means output of the server and $\{\text{CBF}_1, \text{CBF}_2, \ldots, \text{CBF}_m, \text{CBF}, \text{PBF}, \text{view}_s^{\text{OT}}\}$ means the view of the server in the protocol.

Create the simulator Sim_S as follows. Sim_S receives the output Λ of the server and simulates behavior of the server in the protocol. First, Sim_S generates even-distributed random toss r^{Sim} and generates $\text{CBF}_1', \text{CBF}_2', \ldots, \text{CBF}_m'$ in accordance with the following rules:

$$\begin{aligned} & \text{for } (i = 0; j < m; i++) \\ & \quad \text{for } (j = 0; j < t; j++) \\ & \quad\quad \text{CBF}_i'(j) \leftarrow \text{Enc}(\text{Random}(0,1)^n) \end{aligned}$$

Then Sim_S calculates CBF' according to the following rules:

$$\begin{aligned} & \text{for } (j = 0; j < t; j++) \\ & \quad \text{CBF}'(j) = \textstyle\prod_{i=0}^{m} \text{CBF}_i'(j) \end{aligned}$$

Then Sim_S generates intermediate information PBF' of the results query stage:

$$\begin{aligned} & \text{for } (j = 0; j < t; j++) \\ & \quad \text{PBF}'(i) = (\text{Random}(0,1)^l, \text{CBF}'(j)) \end{aligned}$$

Finally, Sim_S simulates the oblivious transfer protocol of results query stage, using PBF' as input and Λ as output, and generates the view $\text{view}_{\text{Sim}}^{\text{OT}}$.

After the whole simulation completes, Sim_S outputs the simulation view:

$$\begin{aligned} & \text{view}_{\text{sim}_S}^{\pi} = \left\{\Lambda, r^{\text{Sim}}, \text{CBF}_1', \text{CBF}_2', \ldots, \text{CBF}_m', \text{CBF}', \text{PBF}', \right. \\ & \quad \left. \text{view}_{\text{Sim}}^{\text{OT}}\right\}. \end{aligned} \tag{18}$$

r^{Sim} and r^s are distributed uniformly, so

$$r^{\text{Sim}} \stackrel{c}{\equiv} r^s. \tag{19}$$

It is assumed that the GM encryption scheme is safe under the semihonest model, and the introduction of random numbers in the GM scheme makes ciphertexts of the GM encryption scheme indistinguishable, so

$$\begin{aligned} & \left\{\text{CBF}_1', \text{CBF}_2', \ldots, \text{CBF}_m', \text{CBF}', \text{PBF}'\right\} \\ & \quad \stackrel{c}{\equiv} \left\{\text{CBF}_1, \text{CBF}_2, \ldots, \text{CBF}_m, \text{CBF}, \text{PBF}\right\}. \end{aligned} \tag{20}$$

In the results query stage, as for the oblivious transfer protocol, the input information PBF' of Sim_S and the server's input information PBF have indistinguishability, and we assume that the underlying OT protocol in the semihonest model is safe, so

$$\text{view}_{\text{Sim}}^{\text{OT}} \stackrel{c}{\equiv} \text{view}_s^{\text{OT}}. \tag{21}$$

In conclusion,

$$\text{view}_{\text{sim}_S}^{\pi}(S_s, S_c) \stackrel{c}{\equiv} \text{view}_s^{\pi}. \tag{22}$$

Participant View. Now we analyze the situation where participant P_1 is attacked. The participant view in protocol π is

$$\begin{aligned} & \text{view}_{P_1}^{\pi}(S_1, S_2, \ldots, S_m, Q) \\ & \quad = \left\{S_1, Q, R, r^P, \text{CBF}_1, \text{QBF}, \text{ABF}, \text{view}_P^{\text{OT}}\right\} \end{aligned} \tag{23}$$

in which S_1 and Q are the input information of P_1, while R is the output information of P_1.

And $\{r^P, \text{CBF}_1, \text{QBF}, \text{ABF}, \text{view}_P^{\text{OT}}\}$ is the information view generated by P_1 in the protocol.

We describe construction of simulator Sim_P as follows. Sim_P receives the input information S_1 and the output R of P_1 and simulates the behavior of the protocol P_1 in the protocol. First, Sim_P generates a uniform distribution of random toss r^{Sim} and generates the encrypted Bloom filter set CBF_1' following steps of the protocol according to inputs. In the results query stage, Sim_P generates the query Bloom filter QBF' following steps of the protocol using Q as input. Sim_P simulates and generates ABF' according to output R (see Algorithm 3).

Finally, Sim_P simulates the oblivious transfer protocol in the results query stage using QBF' as input and ABF' as output and generates the view $\text{view}_{\text{Sim}}^{\text{OT}}$.

After the whole protocol simulation is completed, Sim_P outputs the simulation view

$$\begin{aligned} & \text{view}_{\text{sim}_P}^{\pi} \\ & \quad = \left\{S_1, Q, R, r^{\text{Sim}}, \text{CBF}_1', \text{QBF}', \text{ABF}', \text{view}_{\text{Sim}}^{\text{OT}}\right\}. \end{aligned} \tag{24}$$

r^{Sim} and r^P are distributed uniformly, so

$$r^{\text{Sim}} \stackrel{c}{\equiv} r^P. \tag{25}$$

```
for (j = 0; j < t; j + +)
    PABF(j) = NULL;
for (j = 0; j < |Q|; j + +)
{
    π = NULL;
    if (R(j) == 1)
    {
        for (δ = 0; δ < k; δ + +)
        {
            if (PABF_i(hash_δ(Q(j))) == NULL)
            {
                π = hash_δ(Q(j));
                PABF_i(hash_δ(Q(j))) = Random(0, 1)^n;
            }
        }
PABF_i(π) =
    S_i(j) ⊕ (⊕_{μ=1}^{k} PABF_i(hash_μ(Q(j))))
    }
}
for (j = 0; j < t; j + +)
{
    if (PABF_i(j) == NULL)
        PABF_i(j) = Random(0, 1)^n;
        ABF'(j) = Enc(PABF_i(j));
}
```

Algorithm 3

It is assumed that the GM encryption scheme is safe under the semihonest model, and the introduction of random numbers in the GM scheme makes ciphertexts of the GM encryption scheme indistinguishable, so

$$\left\{\mathrm{CBF}_1', \mathrm{QBF}', \mathrm{ABF}'\right\} \stackrel{c}{\equiv} \left\{\mathrm{CBF}_1, \mathrm{QBF}, \mathrm{ABF}\right\}. \quad (26)$$

In the process of generating the query Bloom filter, according to steps of the protocol, when inputs are the same, there will be identical query Bloom filter sets, so $\mathrm{QBF} = \mathrm{QBF}'$.

In the results query stage, as for the oblivious transfer protocol, the input information QBF' of Sim_S and the server's input information QBF are the same. The output ABF' of Sim_P and the input ABF of participants are indistinguishable. We assume that the underlying OT protocol under the semihonest model is safe, so

$$\mathrm{view}_{\mathrm{Sim}}^{\mathrm{OT}} \stackrel{c}{\equiv} \mathrm{view}_{P}^{\mathrm{OT}}. \quad (27)$$

In conclusion,

$$\mathrm{view}_{\mathrm{sim}_P}^{\pi} \stackrel{c}{\equiv} \mathrm{view}_{P_1}^{\pi}. \quad (28)$$

So we can say that the proposed protocol under semihonest model is safe.

The proof is finished. □

5.4. Performance Analysis. Now we analyze the efficiency of the protocol from two aspects: computational complexity and communication complexity.

5.4.1. Computational Complexity. As for each participant P_i, the hash operation is performed $k|S_i|$ times during the preprocessing stage, and the GM encryption operates kn times; during the results query stage, it is hashed $k|Q_i|$ times and does OT_l^t operation once, while the GM decryption operation is performed at most $kn|Q_i|$ times. As for the server, the ciphertext multiplication operation is performed tmn times in the outsourcing computation stage in all; OT_l^t is performed once in the results query stage.

When implementing OT_l^t using extended OT technology [27], Receiver needs to perform 2λ times of public key operations and $1.44hs$ times of hash operations. Sender needs to perform λ times of public key operations and $1.44hs$ times of hash operations, in which λ represents the security parameter of extended OT protocol. When using the GM algorithm, the encryption operation needs to perform one modular multiplication while the decryption operation needs to perform one modular multiplication, and the multiplication of ciphertexts requires one modular multiplication. Therefore, the participant in this scheme needs to implement the public key algorithm $kn + kn|Q_i| + 2\lambda$ times and the hash algorithm $1.44ks$ times; the server needs to implement the public key algorithm $tmn + \lambda$ times and the hash algorithm $k|S_i| + k|Q_i| + 1.44ks$ times.

5.4.2. Communication Complexity. At the end of preprocessing stage, each participant sends tl bits data to the server, and the server receives tlm bits data in all. In the results query stage, the participant and the server transfer $2\lambda t$ bits of data, respectively.

6. Comparison

There are a number of different parameters due to the fact that existing privacy-preserving set intersection outsourcing protocols are different from privacy-preserving set intersection protocols in principle. Parameters are instantiated in order to compare efficiency of protocols. Common parameters: the sizes of the participant sets are all s. $k = 8$, $t = sk/\ln^2 2$, and $l = 100$. In the proposed protocol, the query set $Q = S$; the Kerschbaum scheme [15] can only achieve security outsourcing computation of two participants, so $m = 2$ in this scheme; the length of ciphertexts in XOR secret sharing is $n = 8$. And, in the Kerschbaum scheme, $\beta = 8$. Construction and query of Bloom filter are based on Dong's open source experimental model [13], which uses SHA-1 to instantiate hash functions; OT protocol uses classical Naor-Pinkas scheme [30].

After summarizing and comparing the existing algorithms in Figure 2 and Table 1, we can see the following. (1) The computational complexity and the communication complexity are lower than that of Huang's scheme and similar to that of Dong's. Also it is slightly lower than Kerschbaum's. (2) The false positive probability is higher than that of Huang's, but the same as Dong's and Kerschbaum's scheme. (3) The proposed algorithm solves the problem of privacy preservation in outsourcing computation considering the cloud computing environment; in the Kerschbaum scheme, a participant is needed to be the server, so it is a traditional

Table 1: Comparison of protocols in applicability.

Comparison of algorithms	Number of participants	System model	Online necessary or not
Dong et al. [13]	2	Traditional Secure Multiparty Computation model	Yes
Huang et al. [19]	2	Traditional Secure Multiparty Computation model	Yes
Kerschbaum [15]	2	Traditional Secure Multiparty Computation model	Yes
The proposed algorithm	≥2	Outsourcing computation model	No

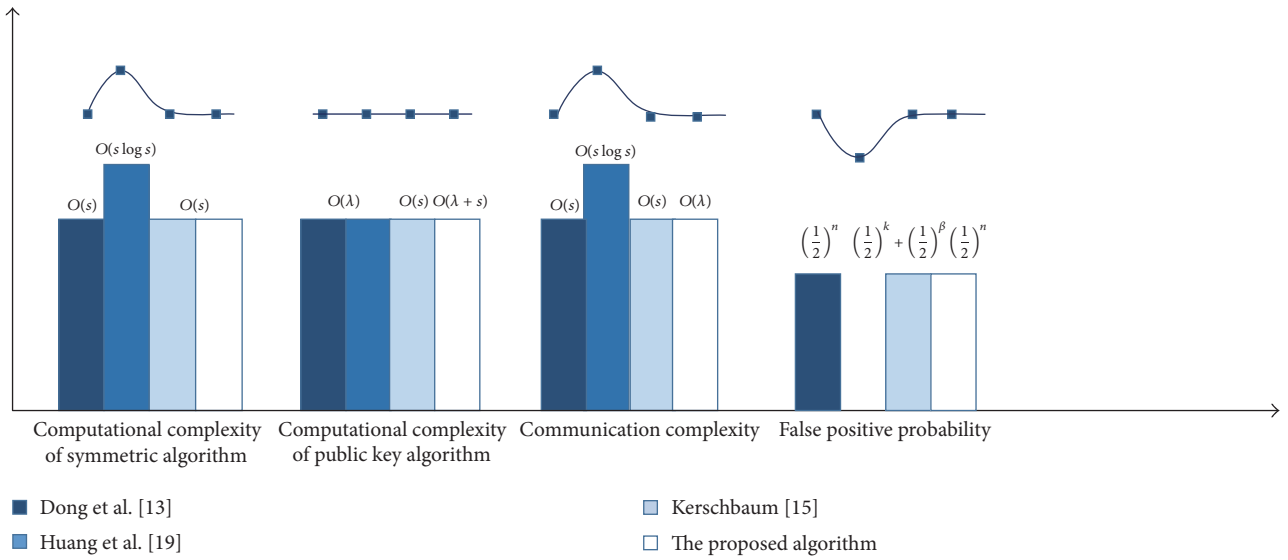


Figure 2: Comparison of protocols in complexity and false positive probability.

secure computation model; in Huang's and Dong's scheme, traditional secure computation model is used to solve PSI problem. (4) The proposed algorithm can solve the secure outsourcing computation with two or more participants, while the others can only deal with the situation of two. (5) It does not need all the participants being online at real time in the proposed algorithm, while the others need them to be online in order to complete the computation at the same time.

In the comparison, we can know from Figure 2 that our algorithm can deal with privacy preservation in outsourcing computation without increasing computational complexity, communication complexity, and false positive probability. In addition, as shown in Table 1, it has great advantages considering the limit of some factors, such as the number of participants, sizes of inputs, and requirement of being online. So, to a large extent, the proposed algorithm improves the solution of privacy preservation in cloud computing.

7. Conclusion

In this paper, we propose a privacy-preserving outsourcing computation system model which can be used in multimedia processing based on cloud computing to solve security and correctness problems. Based on this model, we design a privacy-preserving set intersection outsourcing computation protocol based on GM homomorphic encryption scheme and Bloom filter. The results show that the proposed protocol achieves privacy preservation in the outsourcing computation without increasing computational complexity, the communication complexity, and the false positive probability. And the protocol does not limit the number of participants, the input secret sizes, and whether participant is online in real time. Obviously, not only is the method proposed suitable for multimedia processing, but also it can be used for cloud computing, distributed computing, Internet of things, virtual property transactions, and so on.

In the next few years, we will continue designing the privacy-preserving set intersection outsourcing computation protocol and extending its application in cloud computing. We will focus on the further improvement of efficiency of the algorithm, as well as the design of algorithms against malicious attackers.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant no. 61402162), Hunan Provincial Natural Science Foundation of China (Grant no. 2017JJ3040),

Applied Sci-Tech R&D Special Fund Program of Guangdong Province (no. 2015B010131007), and National High Technology Research and Development Program of China (863 Program) (nos. 2015AA016005, 2015AA017201).

References

[1] X. Liao, G. Chen, and J. Yin, "Content-adaptive steganalysis for color images," *Security and Communication Networks*, vol. 9, no. 18, pp. 5756–5763, 2016.

[2] X. Liao, J. Yin, S. Guo, X. Li, and A. K. Sangaiah, "Medical JPEG image steganography based on preserving inter-block dependencies," *Computers and Electrical Engineering*, 2017.

[3] X. Liao, Z. Qin, and L. Ding, "Data embedding in digital images using critical functions," *Signal Processing: Image Communication*, vol. 58, pp. 146–156, 2017.

[4] G. Han, W. Que, G. Jia, and L. Shu, "An efficient virtual machine consolidation scheme for multimedia cloud computing," *Sensors*, vol. 16, no. 2, article 246, 2016.

[5] D. Dachman-Soled, T. Malkin, M. Raykova, and M. Yung, "Efficient robust private set intersection," in *Proceedings of the International Conference on Applied Cryptography and Network Security. ACNS 2009*, vol. 5536, pp. 125–142, 2009.

[6] C. Hazay and Y. Lindell, "Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries," in *Proceedings of the Theory of Cryptography Conference. TCC 2008*, vol. 4948, pp. 155–175, 2008.

[7] C. Hazay and K. Nissim, "Efficient set operations in the presence of malicious adversaries," in *Proceedings of the International Workshop on Public Key Cryptography – PKC 2010*, vol. 6056, pp. 312–331, 2010.

[8] S. Jarecki and X. Liu, "Efficient oblivious pseudorandom function with applications to adaptive ot and secure computation of set intersection," in *Proceedings of the TCC 2009: Theory of Cryptography*, vol. 5444, pp. 577–594, 2009.

[9] E. De Cristofaro and G. Tsudik, "Practical private set intersection protocols with linear complexity," in *Proceedings of the International Conference on Financial Cryptography and Data Security: FC 2010*, vol. 6052, pp. 143–159, 2010.

[10] E. De Cristofaro and G. Tsudik, "Experimenting with fast private set intersection," in *Proceedings of the International Conference on Trust and Trustworthy Computing: TRUST 2012*, vol. 7344, pp. 55–73, 2012.

[11] E. De Cristofaro, J. Kim, and G. Tsudik, "Linear-complexity private set intersection protocols secure in malicious model," in *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security: ASIACRYPT 2010*, vol. 6477, pp. 213–231, 2010.

[12] D. Many, M. Burkhart, and X. Dimitropoulos, "Fast private set operations with sepia," Tech. Rep. 345, Mar 2012.

[13] C. Dong, L. Chen, and Z. Wen, "When private set intersection meets big data: An efficient and scalable protocol," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS '13)*, pp. 789–800, 2013.

[14] B. Pinkas and T. Schneider, "Faster private set intersection based on OT Extension," in *Proceedings of the 23rd UNSENIX Security Symposium*, 2014.

[15] F. Kerschbaum, "Outsourced private set intersection using homomorphic encryption," in *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, ASIACCS 2012*, pp. 85-86, Republic of Korea, May 2012.

[16] S. Zhou, S. Li, J. Dou, Y. Geng, and X. Liu, "Efficient secure multiparty subset computation," *Security and Communication Networks*, vol. 2017, Article ID 9717580, 2017.

[17] X. Liu, S. Li, X. Chen, G. Xu, X. Zhang, and Y. Zhou, "Efficient Solutions to Two-Party and Multiparty Millionaires' Problem," *Security and Communication Networks*, vol. 2017, pp. 1–11, 2017.

[18] Y. Sun, Q. Wen, Y. Zhang, H. Zhang, Z. Jin, and W. Li, "Two-cloud-servers-assisted secure outsourcing multiparty computation," *The Scientific World Journal*, vol. 2014, Article ID 413265, 2014.

[19] Y. Huang, D. Evans, and J. Katz, "Private set intersection: Are garbled circuits better than custom protocols?" in *Proceedings of the NDSS Symposium 2012*, 2012.

[20] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: outsourcing computation to untrusted workers," in *Advances in cryptology—CRYPTO*, vol. 6223 of *Lecture Notes in Computer Science*, pp. 465–482, Springer, 2010.

[21] P. Mohassel, "Efficient and secure delegation of linear algebra," Cryptology ePrint Archive, Report 2011/605, 2011.

[22] B. Parno, J. Howell, C. Gentry, and M. Raykova, "Pinocchio: Nearly practical verifiable computation," in *Proceedings of the 34th IEEE Symposium on Security and Privacy, SP 2013*, pp. 238–252, usa, May 2013.

[23] B. Schoenmakers, M. Veeningen, and N. de Vreede, "Trinocchio, privacy-preserving outsourcing by distributed verifiable computation," Cryptology ePrint Archive, Report 2015/480, 2015.

[24] A. Peter, E. Tews, and S. Katzenbeisser, "Efficient outsourcing multiparty computation under multiple keys," Cryptology ePrint Archive, Report 2013/013, 2013.

[25] H. Xing, P. DingYi, T. ChunMing, and D. S. Wong, "Verifiable and secure outsourcing of matrix calculation and its application," *SCIENTIA SINICA Informationis*, vol. 43, pp. 842–852, 2013.

[26] X. Hu and C. M. Tang, "Securely outsourcing computation of point multiplication on elliptic curves in cloud computing," *Journal of Hunan University of Science and Technology*, vol. 29, pp. 119–123, 2014.

[27] Y. Ishai, J. Kilian, K. Nissim, and E. Petrank, "Extending oblivious transfers efficiently," in *Proceedings of the Annual International Cryptology Conference: CRYPTO 2003*, vol. 2729 of LNCS, pp. 145–161, Springer.

[28] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[29] P. Bose, H. Guo, E. Kranakis et al., "On the false-positive rate of Bloom filters," *Information Processing Letters*, vol. 108, no. 4, pp. 210–213, 2008.

[30] M. Naor and B. Pinkas, "Efficient oblivious transfer protocols," in *Proceedings of the SODA '01 Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pp. 448–457, SIAM, Washington, DC, USA.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 6256872, 8 pages
https://doi.org/10.1155/2018/6256872

Research Article

A Modification-Free Steganography Method Based on Image Information Entropy

Xia ShuangKui[1] and Jianbin Wu[2]

[1]*School of Electronic Information, Wuhan University, Wuhan, China*
[2]*College of Physical Science and Technology, Huazhong Normal University, Wuhan, China*

Correspondence should be addressed to Jianbin Wu; wujianbin@mail.ccnu.edu.cn

Received 29 September 2017; Revised 4 December 2017; Accepted 6 February 2018; Published 26 March 2018

Academic Editor: Zhenxing Qian

In order to improve the security and robustness of the Information Steganography Algorithm under strictly controlled environment, a new algorithm of modification-free steganography based on image and big data is introduced in this paper. In the proposed algorithm, a mapping relationship between the hot image entropy and the secret information is constructed and the payload information is expressed by the mapping relation. At the same time, turbo code is introduced in order to improve robustness, the hot image comes from Internet image big data, and the library of hot image is established. The performance of the proposed algorithm is analyzed using simulation experiment. Because of its none-modifying on carrier image, the results of experiment show that the proposed algorithm can achieve good performance in robustness analysis, dimension scaling attack, and rotation attack. In particular, in the test of dimension scaling attack and the rotation attack, the rate of data recovering can be over 95%. The proposed algorithm can be very valuable in the covert communication which requires high security and low volume, for example, the key exchange of symmetric encryption system.

1. Introduction

At present, the transmission of important data on the secure communication mainly relies on the cryptography. Cryptography techniques are aimed at encrypting the data in order to ensure its security, thus making it incomprehensible for an adversary. However, the encryption technology has an unavoidable shortage that it clearly indicates the existence of the important data and then easily attracts the attacker's attention.

On the other hand, steganography seeks to provide a covert communication channel between two parties. A common class of steganographic algorithms embeds the secret message in cover works such as images, video, audio, or text. The combination of cover work and secret message is referred to as the stego work and a goal of all steganographic algorithms is to ensure imperceptibility, robustness, capacity, and security. Digital images as important carrier information are widely used in steganography. Presently steganography has achieved lots of research results, mainly in spatial and frequency domain of the image [1–7]. LSB (Least Significant Bit) replacement algorithm is proposed in [1]. The author of [2] proposes an algorithm based on gray level modification and multilevel encryption, which improves the image quality of the carrier and presents a greater challenge to the steganalysis. In literature [3], a steganography algorithm based on run length is proposed. The algorithm has good properties to avoid detection of LSB. Some literature [4–10] discusses the steganography of the frequency domain. In literature [5], the author combines the chaotic sequences and information hiding to propose a DCT (Discrete Cosine Transform) domain method based on Logistic map and achieves good robustness. The algorithm of information hiding based on the discrete wavelet transform is studied in [6]. Although the above steganography research has achieved some good properties, there is a fatal defect that if the adversaries obtain the original image, they can judge whether the carrier image has been modified and then detect the secret information because all algorithms discussed in references must modify the original image according to their principles.

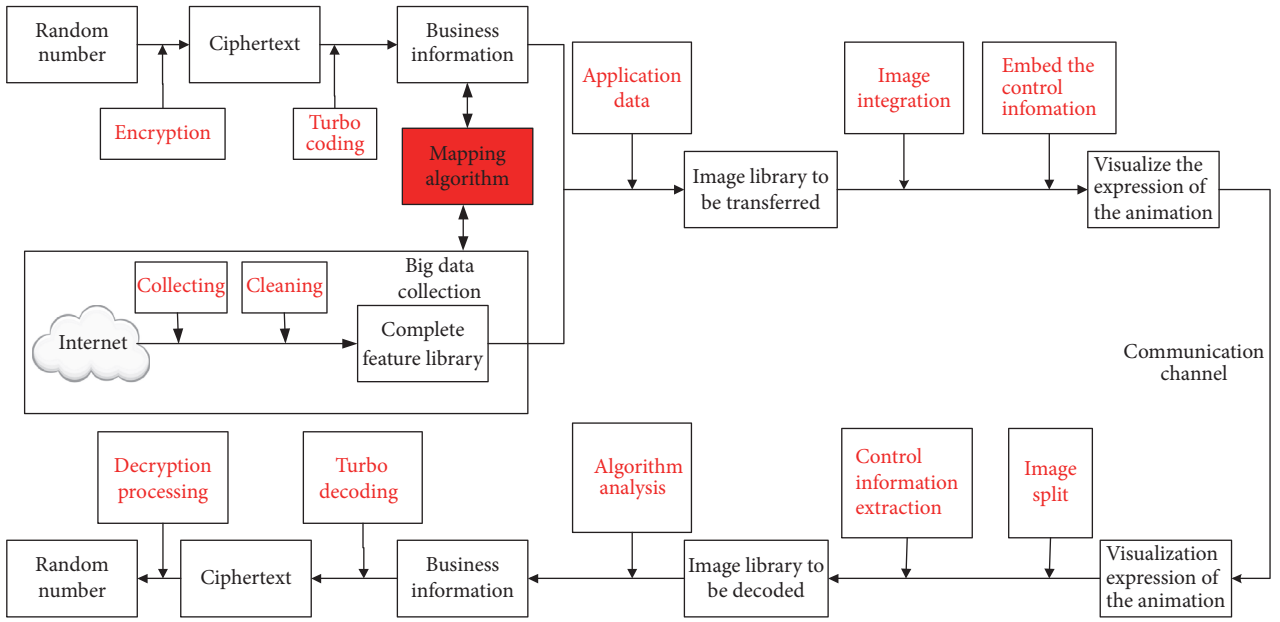


Figure 1: Block diagram of random number visualization expression system.

In this paper, the modification-free steganography algorithm based on image information entropy (MFSA, modification-free steganography algorithm) is proposed, which integrates big data and turbo coding technology. The MFSA establishes a relationship between the image features and the payload information. Through collecting, filtering, and cleaning of big data of network image, these images were divided into different classes according to the relevance of the content of the image, such as landscapes, cars, and small animals. The purpose of classification is to avoid noncooperative concerns. Then, we extract the entropy of the selected image and establish entropy matrix. Finally, the random occult information is mapped onto images according to certain algorithms, so as to construct a complete feature library; thus, we can conduct secret communication without modifying the original images. Because the selected images are hot, highly relative, they will not draw the attention of noncooperative side, and a truly safe and secret communication can be achieved. At the same time, the MFSA also uses error correction coding to further improve the robustness of the secret communication system.

The rest of the paper is organized as follows. Section 2 proposes the MFSA algorithm and elaborates the principle and process of the algorithm from the feature libraries establishment and information extracting, respectively. Section 3 includes the experiments and discusses the performance of the MFSA algorithm, in which the imperceptibility, robustness, and safety are discussed, respectively. Finally, conclusions are drawn in Section 5.

2. The Proposed Algorithm

Given the lack of imperceptible yet robust steganography algorithms, the proposed algorithm adopts a different approach to achieve steganography. The overall block diagram of random number visualization representation system is shown in Figure 1, which includes a complete sender processing flow, a communication channel, and a receiver processing flow.

The random number at the sending end is encrypted to the corresponding ciphertext information. Because the information may be affected by the noise in the process of transmitting, which will result in the error of the binary information, the turbo error correction coding technique is used for correction in the system. Then the turbo encoded information sequence will be the business information of the sending side. Big data acquisition system based on the Java environment is a system that acquires the hot image from network, using network crawler [11] to search and filter the big data. The proposed algorithm is based on a hybrid feature extraction mechanism of the hot image. A complete image feature library is constructed; the follow-up process only needs to update the feature library from time to time. The selected image set is turned into a visual animation in accordance with the control information; then the animation is sent to the receiver through the public communication, and the receiver gets the complete secret information to be delivered by using image resolution, control information extraction, algorithm analysis, turbo decoding, decryption, and other operations. The mapping algorithm module is one of the core parts of the whole system. The modification-free steganography algorithm based on the information entropy of the image not only has the characteristics of high security and robustness, but also establish a mapping relationship between the feature of image and the secure data. Based on the idea of zero-steganography secret communication, the algorithm considers the image information entropy as the starting point and implements the grid description of the image, the extraction of the entropy, the reduction of the entropy matrix, quantization, and a series of mathematical

operations in order to establish a mapping relationship between images and the secure data for secret communication. The use of this kind of hybrid approach for characteristic matrix generation provides a decent robustness against common attacks as well as sufficient capacity. The following section will focus on the principles and implementation of MFSA algorithm.

Suppose that a single image has a capacity of C bits; the number of images required to build a complete image library is 2^c. The length of the payload information is L bits, the number of images sent each time is $N = L/C$ (N upload rounding), the size of each image is p (MB), and the capacity of complete image library is $S = p \cdot 2^c$ (MB). In this paper experiment, the value of parameter c is 32; with consideration of computing power and practical application, the value of parameter c is 16 and the value of parameter p is 0.1.

2.1. The Principle of MFSA Algorithm. There are some features of the image such as color, brightness, histogram, and entropy; by extracting and quantifying, the certain image can express some binary bit sequences. However, some image feature space such as histogram has high dimensionality and poor antinoise ability. The image information entropy is a quantitative description of the image characteristics. It can be seen from the information theory that the information entropy can represent the amount of information contained in the image. From the perspective of image information entropy, the mapping relation between the entropy and the payload information is constructed in this paper, the information entropy of the image is used to represent the payload information, and the zero-steganography covert communication is achieved. Later, the construction method of image information entropy matrix is introduced in detail and the basic principles and implementation of the algorithm will be discussed.

The information entropy is defined as the mathematical expectation of random variables $I(x)$ in the set $\{X, q(x)\}$. Its mathematical expression is shown in

$$H(x) = -\sum_{x \in X} q(x) \log q(x). \tag{1}$$

$H(x)$ is called the information entropy; $q(x)$ is the probability of occurrence of X. In the grayscale image, each pixel can be treated as an argument m (0 to 255). The pixels of the entire image can be viewed as a collection $\{m, p(m)\}$, $p(m)$ is the probability density of the point where the gray value is m, and the expression of the image information entropy F can be obtained, as shown in

$$F = \sum_{m=0}^{k} p(m) \log\left(p(m)\right). \tag{2}$$

k represents the gray value of the pixel, $k = 255$, $p(m)$ represents the probability density of the pixel values appearing in the entire image.

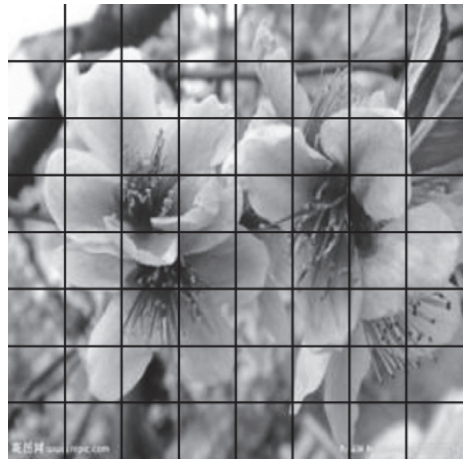


Figure 2: 16 × 16 grid description of image.

$$E_{ij} = \begin{pmatrix} E_{11} & \cdots & E_{1j} \\ \vdots & \ddots & \vdots \\ E_{i1} & \cdots & E_{ij} \end{pmatrix} \quad 1 \le i \le 16,\ 1 \le j \le 16. \tag{3}$$

In (2), the information entropy of the image discussed refers to the global information entropy, which characterizes the statistical distribution of all the pixels of the entire image. The spatial distribution of the image pixels is not taken into account; this will result in different images with the same probability distribution having the same information entropy. In order to solve the problem and use the spatial image information, this paper presents the concept of unit entropy. The grid descriptor is introduced, and Figure 2 shows a grid description of size 16, which maps the original image to the grid, so that any image with different size becomes a $n \times n$ ($n = 16$) phalanx. The information entropy of each grid unit in Figure 2 is obtained by using formula (2); finally we get a 16 × 16 entropy matrix E, which is shown in formula (3).

The $n \times n$ entropy matrix needs dimensionality reduction in order to reduce the redundancy, and the reduced eigenvalue vector $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ is obtained. Since the resulting eigenvalue vector is a floating-point number, we can get the binary bit sequence β after quantization. The index is the binary vector which the carrier image can express actually.

2.2. The Implementation of MFSA Algorithm. Section 2.1 describes the basic principles of the algorithm, which plays an important role in the construction of the complete feature library shown in Figure 1. The implementation flowchart of the algorithm is shown in Figure 3. The size of the selected single image will not exceed 100 kb and the format of the selected images adopts the most popular JPEG in the network. And the measurement of the relevance of images according to the fractal feature of images classification algorithm proposed in [12] is used to select image needed. The algorithm can classify landscape images, artificial drawing images, and computer-generated images, so that the required images achieve a high degree of relevance, giving a very

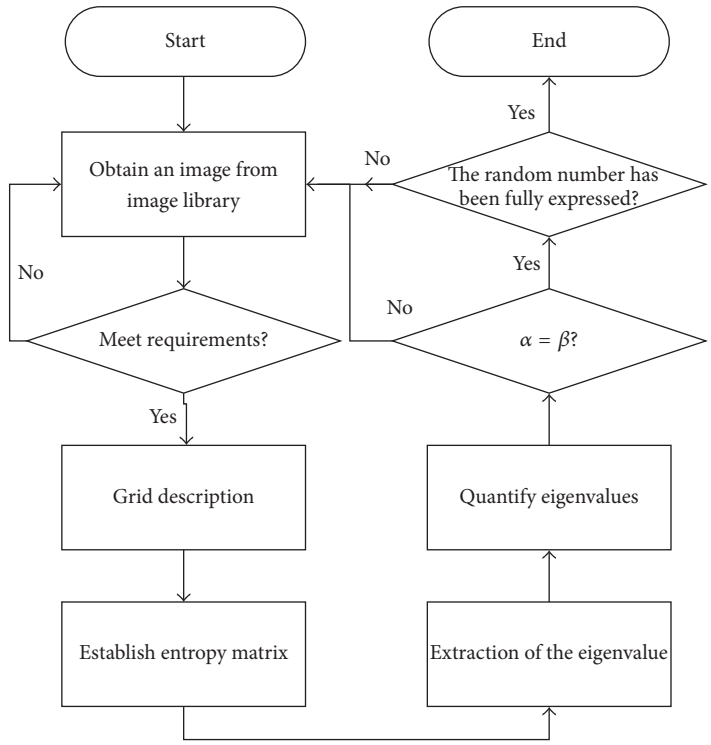


Figure 3: The flowchart of MFSA algorithm.

natural feeling. Referring to the flowchart in Figure 3, the following steps are operated to the filtered picture to obtain the random number that needs to be expressed.

(1) The image is mapped onto a grid with a resolution of 16×16 ($n = 16$), and the entropy matrix is obtained by using formula (2).

(2) Derive the eigenvalue of the entropy matrix, and take the eight largest values to get the eigenvalue vector $(\alpha_1, \alpha_2, \ldots, \alpha_8)$.

(3) The eigenvalue vector is quantized to obtain random number β. The quantitative formula is shown in

$$\beta_i = \text{dec2bin}\left(\text{mod}\left(\text{round}\left(\alpha_i\right), 16\right)\right), \quad (i = 1, 2, \ldots, 8). \tag{4}$$

In the equation, mod( ) represents the congruent operation, round( ) represents rounding operation, dec2bin( ) represents integer conversion to vector operation, α_i represents eigenvalue vector, β_i represents a 4-bit row vector, and finally the line vector $\beta = (\beta_1, \beta_2, \ldots, \beta_8)$.

(4) The extracted random number β is compared with the random number δ to be expressed. If they are matched and there is no image that is able to express the random number in the complete library, the image is added to the complete feature library. If they are not matched, the image is discarded and then the algorithm reselects an image from the image buffer library to repeat the above operations until the completion of the establishment of a complete feature library. Once the complete feature library is established, all we have to do in the calling process is to update it from time to time instead of creating one.

3. Turbo Error Correction Coding

Generally, covert communication implements transmission with the help of the public network link. Although the MFSA algorithm itself has a high fraudulence and concealment because of its characteristics of modification-free steganography, this algorithm incorporates turbo error correction coding technology to further improve the robustness of the system considering the security and complexity of public links.

The turbo code is used to improve the robustness performance of the system and reduce the bit error rate of transmission due to its excellent error correction performance. Turbo code is a high-performance error correction. The principle and performance of turbo code coding are described in detail in [13, 14]. In order to make the turbo code suitable for the system better, this paper improves the QPP (Quadratic Polynomial Permutation) interweaver used in turbo and gives some parameters of turbo coding.

The equation $F(x) = f_0 + f_1 x + \cdots + f_m x^m$ is the permutation polynomial for permuting $\{0, 1, 2, \ldots, N-1\}$. The derivative of $F(x)$ is as $F'(x) = f_1 + 2f_2 x + \cdots + m f_m x^{m-1}$. The selected N'S permutation polynomial structure can be applied to the turbo code interweaver by selecting the appropriate N and polynomial coefficients f. In this system, we select $m = 2$, and the resulting polynomial is $F(x) = f_0 + f_1 x + f_2 x^2$. For the constant term f_0 in the quadratic polynomial only affects the shift in the interweaver and does not work on decoding performance, $F(x) = f_0 + f_1 x + f_2 x^2$ can be further reduced as $F(x) = f_1 x + f_2 x^2$ to simplify the calculation.

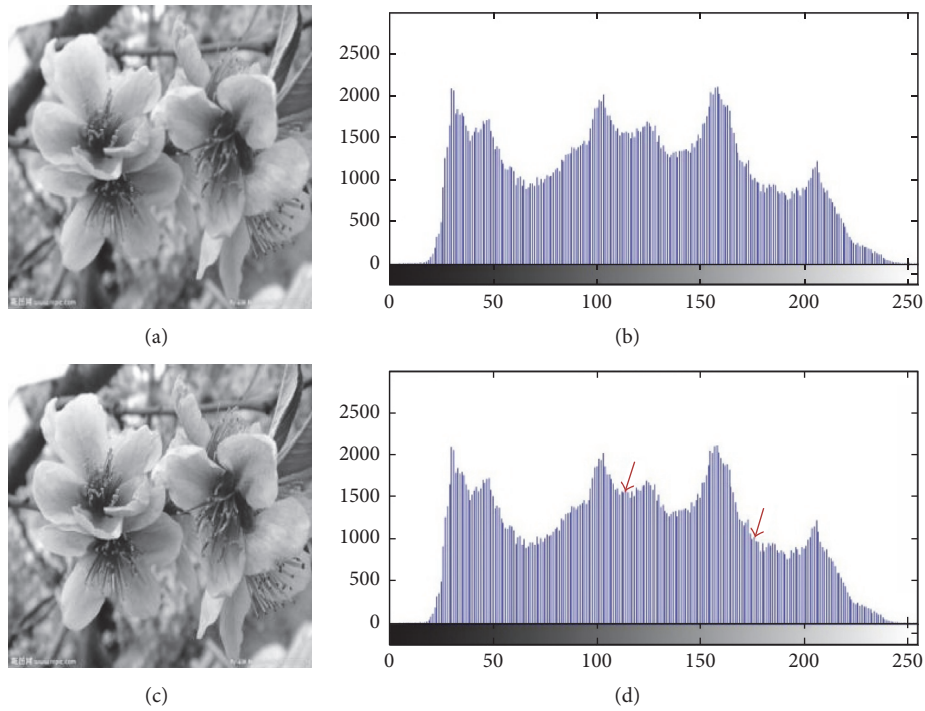


Figure 4: Literature [15] and MFSA algorithm effect contrast. (a) Original carrier image. (b) Histograms of the original carrier image. (c) Secret carrier image. (d) Histograms of the secret carrier image.

Table 1: The coefficient of DPP interweaver permutation multinomial.

k	32	64	128	256	512
f_1	3	7	19	7	7
f_2	10	12	42	16	18

The QPP interweaver uses some particular quadratic polynomial to make them satisfy certain conditions and become QPP structure. So the most critical problem is to solve the polynomial coefficients f_1 and f_2. The polynomial coefficients f_1 and f_2 of the system are satisfied and the length k of the interweaver can be obtained using computer technology, as shown in Table 1. Table 2 shows the turbo encoder and the operating parameters of others used in the paper.

4. Experiment and Analysis

It is known that the statistical analysis of the original carrier is a security risk of the covert communication, and the robustness of the carrier is also an important factor in the secure communication. So this paper designed three experiments and an analysis; the antistatistical analysis ability, the antiscale interference ability, and the antirotation attack capability of the MFSA algorithm are tested, respectively. The safety performance of this MFSA algorithm is analyzed. In the experiment, the MFSA algorithm uses a size of 16×16 grid descriptor. The basic characteristics of the image are mainly concentrated in the larger 5 to 8 eigenvalues, which contains information accounted for more than 90% of the total eigenvalue vector. So the experiment takes the larger 8 eigenvalues; after quantization, the length of the binary vector that can be mapped to a single image is 32 bits.

4.1. Experiment 1. In experiment 1, the immune to statistical analysis was tested between the CSFA algorithm and the proposed algorithm in the literature [15], under the premise of known original "carrier." The result is shown in Figure 4. Figures 4(a) and 4(b) are the original carrier images and histograms, respectively. Since the MFSA algorithm proposed in this paper does not embed the data in the original image, Figures 4(a) and 4(b) are the "carrier" images and histograms of the MFSA algorithm; Figures 4(c) and 4(d) are the secret carrier and its histogram; the embedding algorithm is proposed by the literature [15].

Due to the little payload, from the visual point of view, no difference was found between Figures 4(a) and 4(c). But there is a slight change in Figures 4(b) and 4(d). As indicated by the red arrow in the figure. The arrows on the left of Figure 4(d) indicate that the statistical value is smaller than the corresponding position in Figure 4(b), and the arrow on the right indicates that the statistical value is larger than the corresponding position in Figure 4(b). Other subtle differences with the naked eye cannot be directly found.

The matrix of the original image and confidential image in literature [15] can be expressed as E_{ij} and F_{ij} $(i, j \in (1, 512))$, as shown in

Table 2: The parameters of turbo coding.

Code length (interleaving length)	Bit rate	Interweaving method	Decoding method	Iterations	Production matrix
32 64 128 256 512 variables	0.8	Pseudo-random intertwined	Log-MAP	12	(7, 5), (15, 17)

(a) (b) (c) (d)

Figure 5: Gray scale image of different scale. (a) Image resolution, 1024 × 1024. (b) Image resolution, 512 × 512. (c) Image resolution, 256 × 256. (d) Image resolution, 128 × 128.

$$E_{ij} = \begin{pmatrix} E_{11} & \cdots & E_{1j} \\ \vdots & \ddots & \vdots \\ F_{i1} & \cdots & F_{ij} \end{pmatrix} F_{ij} = \begin{pmatrix} F_{11} & \cdots & F_{1j} \\ \vdots & \ddots & \vdots \\ F_{i1} & \cdots & F_{ij} \end{pmatrix}. \quad (5)$$

Since the essence of the literature [15] is still the alternative method, it is possible to obtain the embedded secret information by performing mathematical operations on E_{ij} and F_{ij}. According to formula (6), the element "1" position and number can be obtained; we can get the length and location of the secret information embedding.

$$Q_{ij} = E_{ij} \odot F_{ij}, \quad (i, j \in (1, 512)). \quad (6)$$

However, the MFSA algorithm is based on the idea of no zero modification through the mapping method to express the secret information. Because it does not make any changes, it is very effective in antistatistical analysis, especially the statistical analysis of known vectors. In addition, because there is no embedded information, the original picture will not show any change, so it will not be due to visual anomalies caused by noncooperation side of the vigilance.

4.2. Experiment 2. In experiment 2, the MFSA algorithm immune to scale attack was tested. As shown in Figure 5, Figure 5(b) is the original "carrier" image, resolution size is 512 × 512, and some different scale images are given as follows: Figure 5(a) is the original image amplification twice the image, and the resolution becomes 1024 × 1024; Figure 5(c) is the image where the original image was shrunk twice and so on. The attack comes from altering the image scale.

The MFSA algorithm is used to demapping the original image and images with varying degrees of scaling attack, as much as possible to restore the random number it represents.

Table 3: Statistical analysis of attack at different scale.

	Figure 5(a)	Figure 5(b)	Figure 5(c)	Figure 5(d)
E (%)	0	0	3.125	6.25
E' (%)	0	0	1.25	3.125

The 32-bit random number vector was recovered from graph (b) $(b_1, b_2, \ldots, b_{32})$; Figures 5(a), 5(c), and 5(d) are similar. The bit error rate is used as its evaluation. Equation (7) gives the solution of the bit error rate E_a (E_a is the bit error rate that is not added to turbo error correction coding; E_a' is the error rate of adding turbo error correction code) for the random number shown in Figure 5(a). Replace a_i with c_i and d_i, respectively; we can find the bit error rates E_c and E_d of graphs (c) and (d).

$$E_a = \frac{p}{n}, \quad \left(p = \sum_{i=1}^{n} a_i \odot b_i\right). \quad (7)$$

n represents the total number of bits, $n = 32$, a_i and b_i represent the random number vectors derived from the images in Figures 5(a) and 5(b). The original "carrier" image is subjected to 100 different scale scaling attacks, and the results of the 100 trials were averaged. The final results are shown in Table 3; E_a is the bit error rate that is not added to turbo error correction coding. E_a' is the error rate of adding turbo error correction code.

As can be seen from Figure 5, although the scale of the original "carrier" image has been attacked, the main content of the image is not changed. The spatial position information of the image and the statistical distribution information of the pixel values are not changed. From the statistical results of Table 3, although the original "carrier" image size has undergone great changes, the correct rate of demapping recovery is also about 95%, and error correction coding can

Figure 6: Different angle rotation of the original secret image. (a) Original secret image. (b) Rotation of 90 degrees anti-counterclockwise of the original secret image. (c) The original secret image horizontally mirrored and vertically mirrored. (d) Rotation of 270 degrees anti-counterclockwise of the original secret image. (e) The original secret image horizontally mirrored. (f) Rotation of 90 degrees anti-counterclockwise and the original secret image vertically mirrored. (g) The original secret image vertically mirrored. (h) Rotation of 270 degrees anti-counterclockwise and the original secret image vertically mirrored.

Table 4: Results of statistical analysis of attack at different rotation angles.

	Figure 6(a)	Figure 6(b)	Figure 6(c)	Figure 6(d)	Figure 6(e)	Figure 6(f)	Figure 6(g)	Figure 6(h)
E (%)	0	7.375	0	5.375	3.125	7.375	3.125	8.375
E' (%)	0	4.125	0	3.125	2.125	5.225	1.125	6.125

further improve the correct rate of demapping. So it can be concluded that the MFSA algorithm has a good effect on the attack defense.

4.3. Experiment 3. In experiment 3, the MFSA algorithm immune to the rotation attack was tested. As shown in Figure 6, the attack comes from image rotation, horizontal flip, horizontal mirror, and vertical mirror of the original image.

Perform 100 independent tests on different "carrier" images; according to the method in experiment 2, the inverse mapping of the original "carrier" image is subjected to different angles of rotation after the attack image is obtained. The result of statistical analysis is shown in Table 4.

From the table, the original "carrier" images subjected to varying degrees of rotation attacks can be seen, the demapping data was recovered, and the data bit error rate remained at about 95%. In particular, as shown in Figure 6(c), the image is horizontally mirrored and vertically mirrored and the random number can be 100% recovered by demapping. Because the "carrier" image suffered a different degree of rotation attack after the basic characteristics did not change, the statistics of the image are not changed. So the feature matrix and its eigenvalue vector remain essentially unchanged. This shows that the MFSA algorithm has a good effect on the antirotation attack.

4.4. Analysis of Safety Performance of MFSA Algorithm. The proposed MFSA algorithm is based on zero modification of the carrier image and it expresses the secure information by using the mapping relationship rather than embedding method, which conceals the existence of the covert communication and has a decisive effect on avoiding the nonpartner's doubt and monitoring. Besides, the MFSA algorithm itself has a certain level of safety.

The safety performance of the MFSA algorithm is dominated by two keys: K_1 is a 6-bit expression algorithm control variable, which controls when to choose what kind of random number expression algorithm; K_2 is the 20-bit quantization coefficient which determines the relative parameters in the quantization process. When extracting the feature from the picture, it is necessary to add the number of the floating-point type to the quantization coefficient and then transform it into the integer number. On the one hand, if the nonpartner obtains the secret key K_1 only, they would not break the MFSA algorithm. The nonpartner cannot obtain the quantified binary vector through iterative computation. On the other hand, even if the nonpartner gets the secret keys K_1 and K_2, they still cannot get the secure information, because the parameters and formulas of the secret key are shared only between the sending and receiving end.

5. Conclusion

Based on the idea of zero modification of the carrier image, the MFSA algorithm is proposed in this paper; the mapping relationship between the image information entropy and information is used to express the secure information. And the network image search engine is used to obtain the "appropriate" image from the big data of image; the big data parallel processing method as the technical support. The mathematical reasoning and the result of experiment demonstrate the proposed MFSA algorithm which has a good performance on the immune statistical analysis and some attacks. Turbo coding technology improves the system robustness and security; the simulation results showed the

noise immunity of our method and this method can resist the scale and rotation attacks. Even if the nonpartner cracks the secret communication channel, and the communication "carrier" is intercepted, the nonpartner cannot judge whether the "carrier" is secret image and cannot get the content of communication. It is very suitable for the covert communication with smaller capacity and high security level, such as the transmission of security system keys, key figures, time, location, and other information transmissions.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Natural Science Foundation of China and China General Technology Research Institute (U1736121).

References

[1] P. Wang and Q. Ma, "Information hiding technology based on least significant bit," *Information Technology Journal*, vol. 12, no. 20, pp. 5681–5684, 2013.

[2] K. Muhammad, J. Ahmad, H. Farman, Z. Jan, M. Sajjad, and S. W. Baik, "A secure method for color image steganography using gray-level modification and multi-level encryption," *KSII Transactions on Internet and Information Systems*, vol. 9, no. 5, pp. 1938–1962, 2015.

[3] J. Q. Xie, Q. Xie, and D. Z. Huang, "Image information hiding algorithm with high security based on run-length," *Computer Science*, 2014.

[4] S. K. Bandyopadhyay, T. U. Paul, and A. Raychoudhury, "A novel steganographic technique based on 3D-DCT approach," *Computer and Information Science*, vol. 3, no. 4, 2010.

[5] J. Song, Y. Zhu, and J. Song, "An information hiding method base on logistic map in DCT domain," *Advances in Information Sciences and Service Sciences*, vol. 4, no. 2, pp. 32–39, 2012.

[6] Q. Sun, P. Guan, Y. Qiu et al., "DWT domain information hiding approach using detail sub-band feature adjustment," *Telkomnika Indonesian Journal of Electrical Engineering*, vol. 11, no. 7, 2013.

[7] M. Bilal, S. Imtiaz, W. Abdul, S. Ghouzali, and S. Asif, "Chaos based Zero-steganography algorithm," *Multimedia Tools and Applications*, vol. 72, no. 2, pp. 1073–1092, 2014.

[8] M. Bilal, S. Imtiaz, W. Abdul et al., "Zero-steganography using DCT and spatial domain," in *Proceedings of the ACS International Conference on Computer Systems and Applications (AICCSA '13)*, pp. 1–7, IEEE, Ifrane, Morocco, May 2013.

[9] H. Ishizuka, I. Echizen, K. Iwamura et al., "A zero-watermarking-like steganography and potential applications," in *Proceedings of the 10th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP '14)*, pp. 459–462, IEEE, Kitakyushu, Japan, August 2014.

[10] H. Ishizuka, I. Echizen, K. Iwamura et al., "Evaluation of a Zero-Watermarking-Type Steganography," in *Digital-Forensics and Watermarking*, pp. 613–624, Springer International Publishing, Berlin, Germany, 2014.

[11] D.-Z. Yang et al., "Application of WebCrawler in information search and data mining," *Computer Engineering and Design*, vol. 30, pp. 5658–5662, 2009.

[12] W.-P. Liu et al., "A fractal feature based method for image classification," *Journal of Image and Graphics*, vol. 10, pp. 754–757, 2005.

[13] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and encoding: turbo-codes," in *Proceedings of the IEEE International Conference on Communications*, pp. 1064–1070, May 1993.

[14] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261–1271, 1996.

[15] H.-J. Zhang and C.-M. Zhu, "Novel LSB steganography algorithm of against statistical analysis," *Computer Engineering*, vol. 34, no. 23, pp. 144–146, 2008.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 1326235, 9 pages
https://doi.org/10.1155/2018/1326235

Research Article

Improved Encrypted-Signals-Based Reversible Data Hiding Using Code Division Multiplexing and Value Expansion

Xianyi Chen, Haidong Zhong, Lizhi Xiong, and Zhihua Xia

Jiangsu Engineering Center of Network Monitoring, Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology, School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China

Correspondence should be addressed to Zhihua Xia; xia_zhihua@163.com

Received 28 September 2017; Revised 1 December 2017; Accepted 21 January 2018; Published 15 February 2018

Academic Editor: Rémi Cogranne

Compared to the encrypted-image-based reversible data hiding (EIRDH) method, the encrypted-signals-based reversible data hiding (ESRDH) technique is a novel way to achieve a greater embedding rate and better quality of the decrypted signals. Motivated by ESRDH using signal energy transfer, we propose an improved ESRDH method using code division multiplexing and value expansion. At the beginning, each pixel of the original image is divided into several parts containing a little signal and multiple equal signals. Next, all signals are encrypted by Paillier encryption. And then a large number of secret bits are embedded into the encrypted signals using code division multiplexing and value expansion. Since the sum of elements in any spreading sequence is equal to 0, lossless quality of directly decrypted signals can be achieved using code division multiplexing on the encrypted equal signals. Although the visual quality is reduced, high-capacity data hiding can be accomplished by conducting value expansion on the encrypted little signal. The experimental results show that our method is better than other methods in terms of the embedding rate and average PSNR.

1. Introduction

Encryption and data hiding are two common approaches for protecting against information leakage [1–3]; the former is used to protect the content itself [4], whereas the latter is used to protect the hidden data [5]. While it is a problem for sensitive applications, such as military images and medical images, reversible data hiding (RDH) is an effective method for these special scenarios, which aims to recover both embedded data and the original image. In the past two decades, many classic RDH algorithms have been proposed, such as lossless image compression-based methods [6], difference expansion- (DE-) based methods [7], histogram shifting- (HS-) based methods [8], integer-to-integer transform-based methods [9], and dual-image-based methods [10].

However, with the popularity of outsourced storage services [11, 12], the traditional RDH is not suitable in these scenarios, especially with regard to the requirement of high security. Therefore, the research of privacy protection in cloud computing has attracted considerable attention in recent years [13–15]. Among these studies, encrypted-image-based reversible data hiding (EIRDH) provides the possibility that the image owner can encrypt the image before uploading it to the service provider, and then the service provider can embed some additional message into the incomprehensible encrypted image for steganography or authentication. The authorized users or receivers can recover both the additional message and the original image. The existing EIRDH methods can be grouped into three categories: vacating room after encryption (VRAE) methods [16–22], reserving room before encryption (RRBE) methods [23–25], and reversible image transform (RIT) methods [26, 27].

The framework of "VRAE" was proposed by Zhang [18], in which secret bits can be embedded after encrypting the original image. Specifically, the data hider can divide the encrypted image block into two sets and embed secret bits by flipping three LSBs of a set. To decrease the extracted-bits error rate, Hong et al. [19] and Liao and Shu [20] evaluated the complexity of image blocks. Recently,

Yi and Zhou [21] proposed a novel EIRDH method using binary-block embedding for joint decryption and extraction, in which a bit-level scrambling process can prevent secret bits from loss. The methods [18–21] can embed an additional message from the decrypted image. To extract secret bits in the encrypted image, Zhang [22] proposed a novel separable EIRDH method, in which three cases are considered according to the encryption key or data hiding key.

However, it is hard to use the traditional RDH method for the data hider in the VRAE since the correlation between neighbor pixels in the encrypted image is destroyed.

The framework of "RRBE" was designed by Ma et al. [23]. The data owner can reverse the room of LSBs using the traditional RDH method and then encrypt the self-embedded image. After that, the data hider embeds secret data into the reversed LSBs of the encrypted image. Cao et al. [24] compressed pixels in the local patch by sparse representation and achieved a higher reversed room than other previous methods.

To transform the original image into an encrypted image which looks like the target image, Zhang et al. [26] proposed the EIRDH framework based on RIT, in which an image block is paired by similar means and standard deviation between the original and target images. Since the correlation of transformed images is not destroyed, the data hider can embed secret bits by the traditional RDH method. However, this method has high image distortion since much auxiliary information must be self-embedded into the transformed image for recovering the original image. Recently, Hou et al. [27] improved the visual quality of camouflage images and reduced the auxiliary information for recording block indexes by adopting k-means clustering.

Different from the EIRDH, Chen et al. [28] designed an encrypted-signals-based reversible data hiding (ESRDH) method, in which the data owner divides each pixel of the cover image into two signals and then encrypts them by a public key. After that, the data hider embeds secret bits into the encrypted signals by the additive homomorphism. To reduce data expansion, Shiu et al. [29] adopted a difference expansion method to embed the message. Zhang et al. [30] proposed a lossless, a reversible, and a combined data hiding scheme for ciphertext images, respectively, which were encrypted by a public key cryptosystem with homomorphic properties. Recently, Wu et al. [31] presented an improved ESRDH method, in which the original image can be recovered completely by decrypting all signals. The method can achieve a higher embedding capacity and better quality of decrypted images than previous ESRDH methods. In addition, compared to the traditional EIRDH method, the cost spent, on a reliable key management system in a multiparty environment, is reduced since the receiver can share the public key with other identities.

In this paper, we propose an improved ESRDH method using code division multiplexing (CDM) and value expansion (VE), in which the traditional CDM technology is utilized to achieve lossless visual quality of decrypted signals, and VE technology is designed according to the RDH method based on DE for further increasing the embedding capacity. Compared to the current ESRDH method, the proposed method achieves a high embedding capacity and good visual quality.

The rest of this paper is arranged as follows. In Section 2, the related works are described. In particular, communication technology based on CDM, RDH method based on DE, and ESRDH method based on signal energy transfer are given. Section 3 presents a lossless visual quality ESRDH method using code division multiplexing and high embedding capacity ESRDH method using value expansion. Section 4 displays the performance of the proposed method by experimental results and Section 5 concludes this paper.

2. Related Works

2.1. Communication Technology Based on CDM. In communication systems, CDM is a kind of spectrum spreading technology for ensuring secure information transmission and channel multiplexing, in which the sender encodes to-be-transmitted bits with a predetermined spreading sequence, and then the receiver can obtain the secret bits by the same spreading sequences. So, the sequences that are derived by Walsh Hadamard matrix play an important role in CDM-based communication technology. Walsh Hadamard matrix consists of 1 and −1. In addition to the first row (column) of the matrix, other row (column) vectors are selected as the spreading sequences, and they have two properties. First, the sum of elements is equal to 0 for any sequence. Second, they are orthogonal to each other of any two different sequences, so the cross-correlation is 0.

Suppose the spreading sequences generated by Walsh Hadamard matrix are S_z, $z = \{1, \ldots, q\}$ and the compound sequence C is the linear combination of S_z; when the secret bit is 0, the coefficient of S_z is set as −1. Otherwise, the coefficient is set as 1. Thus, the receiver can decode the secret bits according to the results of dot product between S_z and C.

For example, three spreading sequences from a 4-level Hadamard matrix are$S_1 = (1, -1, 1, -1)$, $S_2 = (1, 1, -1, -1)$, and $S_3 = (1, -1, -1, 1)$. Obviously, S_1, S_2, and S_3 are zero means and orthogonal to each other. Suppose the secret bits are "010"; then, the sequences are denoted as S_1, $-S_2$, and S_3, respectively, and the three sequences are added to form the compound sequence $C = S_1 - S_2 + S_3 = (1, -3, 1, 1)$.

On the receiver side, for the receiver with the spreading sequence S_1, the decoding result is $(C \cdot S_1)/|S_1|^2 = ((S_1 - S_2 + S_3) \cdot S_1)/|S_1|^2 = (S_1 \cdot S_1)/|S_1|^2 = 1$, which represents the notion that the secret bit is 1; for the receiver with S_2, the result is $(C \cdot S_2)/|S_2|^2 = (-S_2 \cdot S_2)/|S_2|^2 = -1$, which represents bit 0; for the receiver with S_3, the result is $(C \cdot S_3)/|S_3|^2 = (S_3 \cdot S_3)/|S_3|^2 = 1$, which represents bit 1. Therefore, the secret bits can be extracted. Moreover, since a large number of secret bits are represented and transmitted by the compound sequences in different ways and it is impossible for an intruder to guess them, the CDM-based communication technology is secure and can provide high capacity.

2.2. RDH Method Based on DE. The RDH method based on DE was proposed by Tian. Assume the two neighbor pixels of

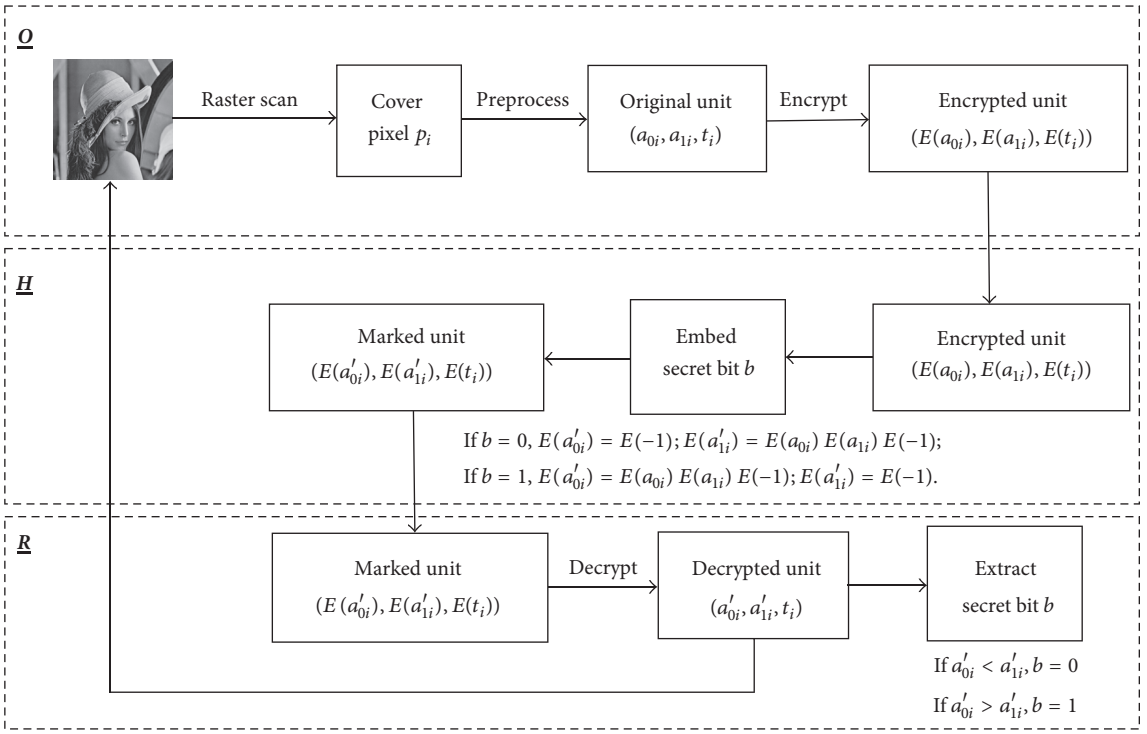


Figure 1: The framework of ESRDH method based on signal energy transfer.

the image are x and y. The sender can compute the average m and the difference h between x and y:

$$m = \left\lfloor \frac{x + y}{2} \right\rfloor, \quad h = x - y. \tag{1}$$

Then, the secret bit b can be embedded as the least significant bit (lsb) of h. And the new difference value h' can be obtained:

$$h' = 2h + b. \tag{2}$$

Finally, the corresponding values x' and y' can be computed and transmitted by the sender:

$$x' = m + \left\lfloor \frac{h' + 1}{2} \right\rfloor, \quad y' = m - \left\lfloor \frac{h'}{2} \right\rfloor. \tag{3}$$

From the pixels pair (x', y'), the receiver can extract secret bits and recover the original image. The initial average value m and new difference value h' can be computed:

$$m = \left\lfloor \frac{x' + y'}{2} \right\rfloor, \quad h' = x' - y'. \tag{4}$$

The secret bit b and difference value h can be obtained by h':

$$b = \mathrm{LSB}\left(h'\right), \quad h = \left\lfloor \frac{h'}{2} \right\rfloor. \tag{5}$$

And the original pair (x, y) can be recovered by m and h:

$$x = m + \left\lfloor \frac{h + 1}{2} \right\rfloor, \quad y = m - \left\lfloor \frac{h}{2} \right\rfloor. \tag{6}$$

DE is a simple and efficient RDH method that utilizes the redundancy between two neighbor pixels to embed secret bits and achieve reversibility. In addition, we can use the data-embedding algorithm for an image more than once for multiple-layer embedding. When the difference is small, a higher embedding capacity and better visual quality of the cover image can be achieved. However, the transformed values x' and y' should be restricted in the range of $[0, 255]$.

2.3. ESRDH Method Based on Signal Energy Transfer. ESRDH method based on signal energy transfer was proposed by Wu et al. It consists of three phases: image encryption phase, data-embedding phase, and data extraction and image recovery phase. There are three active identities in this method. They are image owner O, data hider H, and receiver R. The framework of this method is described in Figure 1.

Suppose the size of the cover image is $N_1 \times N_2$ and the pixel is $p_i \in [0, 255]$, $1 \le i \le N_1 \times N_2$. Inspired by the signal energy transfer, one signal can be represented by the sum of other signals. Therefore, p_i can be represented as $a_{0i} + a_{1i} + t_i$, where a_{0i} is selected randomly from $\{0, 1, 2, \ldots, x_i\}$, x_i is obtained by $x_i = 2\lfloor p_i/2 \rfloor$, a_{1i} is equal to a_{0i}, and t_i is set to be $t_i = p_i - 2a_{0i}$. And then, the image owner can encrypt the three signals according to the public key that was generated by the Paillier encryption algorithm.

Let the encrypted pixels be represented by $(E(a_{0i}), E(a_{1i}), E(t_i))$. The data hider embeds the secret bit b into $E(a_{0i})$ and $E(a_{1i})$ and generates the marked signals $E(a'_{0i})$ and $E(a'_{1i})$. If the secret bit is 0, then $E(a'_{0i}) = E(a_{0i} - a_{1i} - 1)$, $E(a'_{1i}) = E(a_{0i} + a_{1i} + 1)$. If the bit is 1, then $E(a'_{0i}) = E(a_{0i} + a_{1i} + 1)$, $E(a'_{1i}) = E(a_{0i} - a_{1i} - 1)$. Since $a_{0i} = a_{1i}$ and Paillier encryption maintains the additive homomorphic properties, then the embedded formulas can be reduced to

$$E\left(a'_{0i}\right) = \begin{cases} E(-1), & \text{if } b = 0, \\ E\left(a_{0i}\right) E\left(a_{1i}\right) E(1), & \text{if } b = 1, \end{cases}$$
$$E\left(a'_{1i}\right) = \begin{cases} E\left(a_{0i}\right) E\left(a_{1i}\right) E(1), & \text{if } b = 0, \\ E(-1), & \text{if } b = 1. \end{cases} \tag{7}$$

To extract the secret bit and recover the cover image, the receiver can decrypt the marked signals $(E(a'_{0i}), E(a'_{1i}), E(t_i))$ according to the private key generated by Paillier encryption and obtain the decrypted unit (a'_{0i}, a'_{1i}, t_i). Then, the decrypted pixel is denoted by $p_i = a'_{0i} + a'_{1i} + t_i$. Because of $a'_{0i} + a'_{1i} = a_{0i} + a_{1i}$, the decrypted image is a cover image. After that, the bit b can be extracted by comparing a'_{0i} and a'_{1i}:

$$b = \begin{cases} 0, & \text{if } a'_{0i} < a'_{1i}, \\ 1, & \text{if } a'_{0i} > a'_{1i}. \end{cases} \tag{8}$$

This method achieves a lossless visual quality since the decrypted image is the original one. It also can process encoded multimedia since each separated unit of the encoded multimedia can be recovered completely. Therefore, there is no underflow or overflow problem and there is no need to embed any auxiliary information.

3. The Proposed Method

3.1. Lossless Quality ESRDH Method Based on CDM. To maintain the lossless quality of the decrypted image and improve the embedding capacity, inspired by communication technology based on CDM, we propose a lossless visual quality ESRDH method only using CDM.

Suppose that the generated spreading sequences are $S_z, z = \{1, \ldots, q\}$ and the length is denoted as $l = q + 1$, where l must be the power of 2 such as 2, 4, 8. Therefore, the preprocess and encryption can be summarized as follows. Firstly, divide each pixel unit p_i as $p_i = a_{1i} + \cdots + a_{li} + t_i$, where t_i is obtained by $t_i = p_i \bmod l$ and $a_{1i}, \ldots, a_{li}$ are set to be $(p_i - t_i)/l$. Secondly, define the vector $V_i = [a_{1i}, \ldots, a_{li}]$ as an embedded vector, so each unit p_i can be represented as (V_i, t_i). Finally, the image owner encrypts each unit by a public key generated by Paillier encryption, and the encrypted unit $(E(V_i), E(t_i))$ is generated, where $E(V_i) = [E(a_{1i}), \ldots, E(a_{li})]$.

Now, the data hider can embed secret bits b_{iz} ($z = \{1, \ldots, q\}$) into $E(V_i)$ using CDM. This means the number of bits which can be embedded into the ith ($1 \le i \le N_1 \times N_2$) pixel is q. At the beginning, the secret bits b_{iz} can be transformed to w_{iz} which consist of −1 and 1:

$$w_{iz} = \begin{cases} -1, & \text{if } b_{iz} = 0 \\ 1, & \text{if } b_{iz} = 1. \end{cases} \tag{9}$$

Then, we can obtain the compound sequence C_i by $C_i = w_{i1}S_1 + \cdots + w_{iq}S_q$; the elements of C_i are $[C_i(1), \ldots, C_i(l)]$. They can be encrypted by the public key and can generate an encrypted compound sequence $EC_i = [E(C_i(1)), \ldots, E(C_i(l))]$. Since Paillier encryption has additive homomorphic properties, EC_i can be embedded into $E(V_i)$ by the following formula:

$$\begin{aligned} E\left(V'_i\right) &= \left(E\left(a'_{1i}\right), \ldots, E\left(a'_{li}\right)\right) \\ &= \left(E\left(a_{1i} + C_i(1)\right), \ldots, E\left(a_{li} + C_i(l)\right)\right) \\ &= \left(E\left(a_{1i}\right) \times E\left(C_i(1)\right), \ldots, E\left(a_{li}\right) \times E\left(C_i(l)\right)\right), \end{aligned} \tag{10}$$

where the new vector $E(V'_i)$ can be denoted as $E(V'_i) = [E(a'_{1i}), \ldots, E(a'_{li})]$.

Finally, the receiver can decrypt the marked signals $(E(V'_i), E(t_i))$ by a private key and obtain (V'_i, t_i). V'_i can be represented by $V'_i = [a'_{1i}, \ldots, a'_{li}]$ or $V'_i = V_i + C_i$. Because of $a_{1i} = \cdots = a_{li}$ and for any different spreading sequences, they are zero means and orthogonal to each other; the result of dot product between V'_i and S_z ($z = \{1, \ldots, l\}$) can be summarized as

$$V'_i \cdot S_z = C_i \cdot S_z = l^2 \times w_{iz}. \tag{11}$$

Therefore, w_{iz} can be obtained by the following formula:

$$w_{iz} = \frac{V'_i \cdot S_z}{l^2} = \operatorname{sign}\left(V'_i \cdot S_z\right). \tag{12}$$

And b_{iz} can inversely be transformed by formula (9). Since the sum of elements is equal to 0 for any spreading sequences, we have

$$a'_{1i} + \cdots + a'_{li} = a_{1i} + \cdots + a_{li}. \tag{13}$$

The directly decrypted pixel is p'_i, and we have $p'_i = a'_{1i} + \cdots + a'_{li} + t_i = p_i$. As a result, the decrypted pixel is the original one. In other words, based on CDM, ESRDH achieves lossless visual quality of images. And the data-embedding process does not cause an underflow/overflow problem. In addition, the embedding rate can be improved to be q bits per pixel (bpp) such as $q = \{1, 3\}$. So, it can be determined by the number of spreading sequences q.

3.2. High Embedding Capacity ESRDH Method Based on VE. In Wu et al.'s method and ESRDH method based on CDM, the signal t_i is not used to embed secret bits. To further increase the embedding capacity, the ESRDH method based on VE is proposed. In Wu et al.'s method, the range of t_i is $[p_i - 4\lfloor p_i/2 \rfloor, p_i]$, which will decrease the visual quality of the decrypted image because the absolute value of t_i is too large. However, in the ESRDH method based on CDM, the range of t_i is $[1, q]$, and it is enough small, so the ESRDH based on VE is an efficient method.

In Section 3.1, the pixel unit (V_i, t_i) is encrypted and sent to the data hider. Instead of only embedding secret bits into V_i, t_i also can embed bits in this section. Suppose t_i is expanded 2^k $(k = 1, 2)$ times and the decimal number of $(b_{i1} \cdots b_{ik})_2$ is D_i. So, k LSBs of t_i can be emptied to embedded secret bits by $t_i' = 2^k \times t_i + D_i$. Then, p_i' can be constructed by the sum of $a_{1i}', \ldots, a_{li}', t_i'$, and it can be reduced to $p_i + t_i' - t_i$. Since $p_i \in [0, 255]$ and $t_i' - t_i \in [0, R\max]$, thus $p_i' \in [0, 255 + R\max]$, where $R\max = (2^k - 1) \times (q + 1)$. Therefore, data embedding can possibly cause an overflow problem. To solve this problem, t_i should be preprocessed, and the formula can be summarized as

$$ts_i = \begin{cases} t_i, & 0 \le p_i \le 255 - R\max, \\ t_i - R\max, & 255 - R\max < p_i \le 255. \end{cases} \tag{14}$$

The range of t_i is changed from $[0, q]$ to $ts_i \in [-R\max, q - R\max] \cup [0, q]$. So, if $ts_i < 0$, ts_i is the changeable one. Otherwise, it is the original one.

Firstly, the encrypted unit $(E(V_i), E(ts_i))$ can be obtained by a public key.

Secondly, D_i should be encrypted to $E(D_i)$ by the same key. And then it is embedded into $E(ts_i)$, so a new encrypted value $E(ts_i')$ can be generated, which can be summarized as

$$E\left(ts_i'\right) = E\left(2^k \times ts_i + D_i\right) = E\left(ts_i\right)^{2^k} \times E\left(D_i\right). \tag{15}$$

Thirdly, the receiver decrypts $E(ts_i')$ by the corresponding private key and denotes it as ts_i'. After that, the embedded bits $(b_{i1} \cdots b_{ik})_2$ can be obtained by extracting k LSBs of ts_i'. So, ts_i can be calculated by

$$ts_i = \frac{ts_i' - D_i}{2^k}, \tag{16}$$

where D_i is the decimal number of $(b_{i1} \cdots b_{ik})_2$.

Finally, the original t_i should be recovered by

$$t_i = \begin{cases} ts_i, & ts_i \ge 0, \\ ts_i + R\max, & ts_i < 0. \end{cases} \tag{17}$$

And the pixel p_i can be recovered by the sum of other signals.

Based on ESRDH using CDM, the proposed method can further improve the embedding rate using VE. The embedded rate can be increased by k bpp such as $k = \{1, 2\}$. Furthermore, there is no need to embed any auxiliary information to recover the original t_i. However, there is a possibility that an overflow problem arises, which may cause image distortion.

3.3. Algorithm of ESRDH Method Based on CDM and VE. According to Sections 3.1 and 3.2, the framework of ESRDH method based on CDM and VE is shown in Figure 2, and the algorithm can be divided into three phases: preprocess and signal encryption phase, data-embedding phase, and data extraction and image recovery phase. The details are described as in Algorithm 1.

4. Experimental Results

As is shown in Figures 3(a)–3(d), four different grayscale images with different features are selected as the test images, which are Lena, Pepper, Sailboat, and Baboon. The size of these pictures is 512 × 512. The development tool is Myeclipse8.6, which is used on an Intel Core i5 CPU (2.8 GHz) with 8 GB of memory.

The embedding rate (ER) and visual quality of the decrypted image are two important indicators to measure the performance of the EIRDH method, and they can be calculated by

$$\begin{aligned} \mathrm{ER} &= \frac{\text{Embedding Capacity}}{\text{The size of cover image}}, \\ \mathrm{PSNR} &= 10 \lg \frac{255^2 \times N_1 \times N_2}{\sum_{i=1}^{N_1 \times N_2} \left(p_i' - p_i\right)^2}, \end{aligned} \tag{18}$$

where p_i and p_i' are the original and modified pixel values, respectively.

In the proposed method, if the ESRDH method based on CDM is used to embed secret bits, the ER is q bpp, and PSNR is $+\infty$. In Table 1, when $q = 1$ and $q = 3$, the embedding capacity (EC) is $512 \times 512 = 262144$ and $3 \times 512 \times 512 = 786432$, and the corresponding ER is 1 bpp and 3 bpp, respectively. And lossless visual quality of the decrypted image can be achieved. The decrypted images with the hidden data were not further processed, as shown in Figure 4. The plain text images recovered from the four encrypted images were all identical to the original one. In addition, we can utilize the VE method to further improve ER, and up to $q + k$ bpp can be achieved. In addition, PSNR will be changed by setting k. For example, 786432 bits can be embedded using CDM and VE methods when $q = 1$, $k = 2$. However, PSNR of Lena is decreased to 36.85 dB. The other results about EC, ER, and PSNR of different test images are shown in Table 1. Furthermore, although the ER of the four images is 3 bpp when $q = 1$, $k = 2$, or $q = 3$, the PSNR of these decrypted images are different.

To assess the time overheads on encryption, embedding, and decryption, we assessed the statistic efficiency performance of the proposed method for different test images when $q = 3$ and $k = 1$ in Table 2. In this table, the embedding time is small because secret bits are embedded by multiplication, which also reflects the notion that signals encryption and decryption spend a long time. However, since the operations of CDM and VE were all performed in a homomorphic encryption domain, the encrypted signals with the hidden data were protected by the Paillier cryptosystem. Therefore, the Paillier cryptosystem in our method is an important

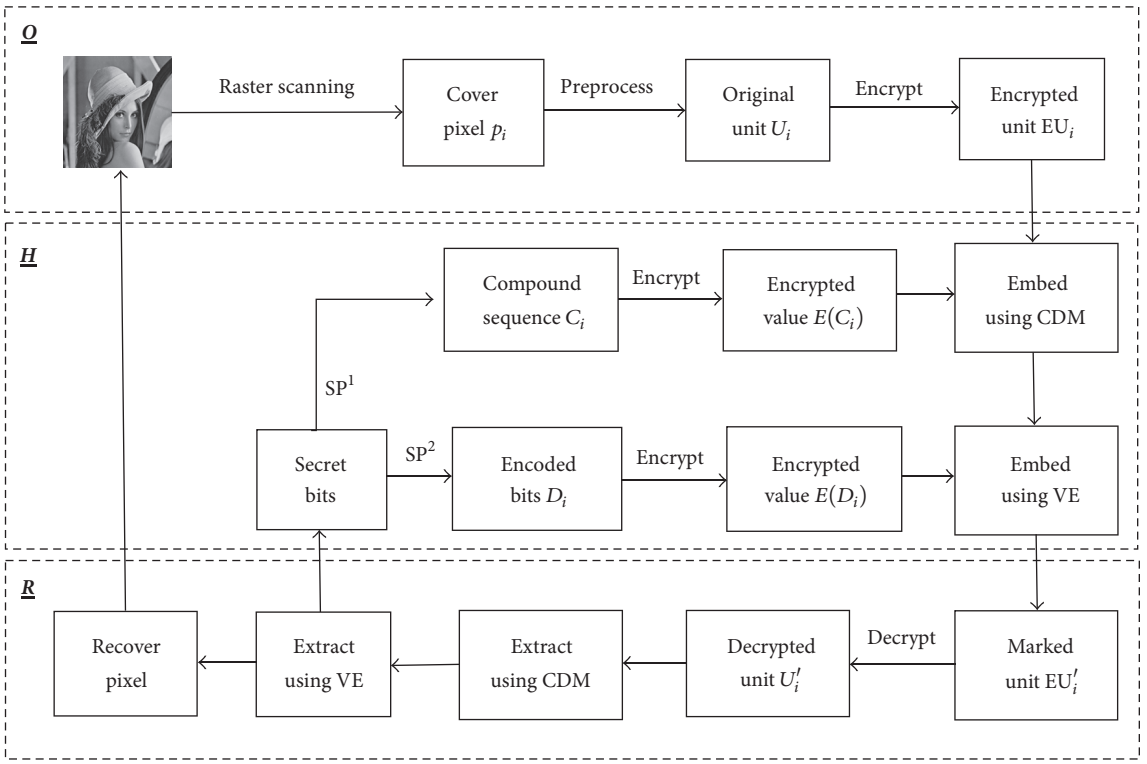
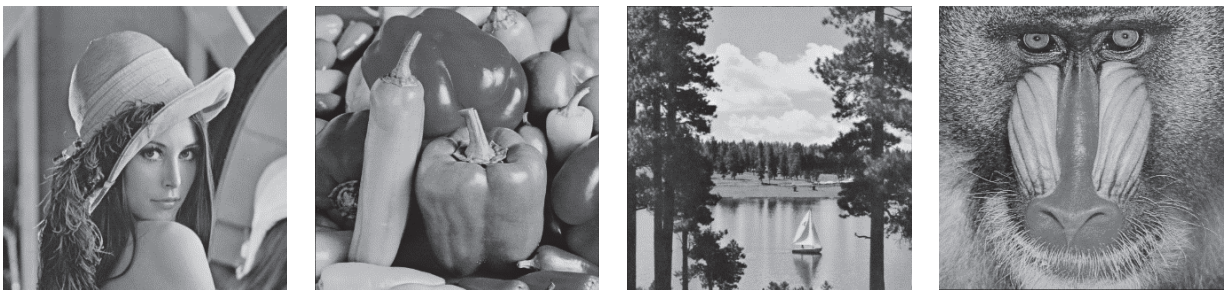
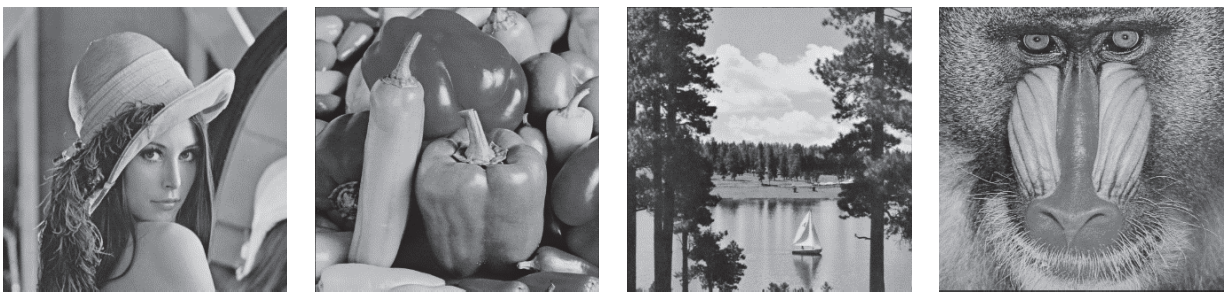


Figure 2: The framework of the ESRDH method based on CDM and VE.



(a) (b) (c) (d)

Figure 3: Four test images. (a) Lena, (b) Pepper, (c) Sailboat, and (d) Baboon.



(a) (b) (c) (d)

Figure 4: Directly decrypted images. (a) Lena, (b) Pepper, (c) Sailboat, and (d) Baboon.

Preprocess and signal encryption phase
Input: An original image with a size of $N_1 \times N_2$, the number of spreading sequences q ($q = \{1, 3\}$), the expanded parameter k ($k = \{1, 2\}$), and public key pk.
Output: The encrypted unit EU_i.
Step 1: Obtain the separated signals from the original pixel $p_i = a_{1i} + \cdots + a_{li} + t_i$, $1 \le i \le N_1 \times N_2$, which can be calculated in Section 3.1. Define the embedded vector $V_i = [a_{1i}, \ldots, a_{li}]$, and each pixel unit is represented as (V_i, t_i).
Step 2: Modify t_i to be ts_i by formula (14) for preventing overflow.
Step 3: Encrypt $U_i = (V_i, ts_i)$ by pk and generate an encrypted unit $\mathrm{EU}_i = (E(V_i), E(ts_i))$.
Data embedding phase
Input: The encrypted unit EU_i, the number of spreading sequences q, the expanded parameter k, public key pk, and secret bits.
Output: Marked unit EU_i'.
Step 1: Divide secret bits into two parts SP^1 and SP^2, where SP^1 contains $q \times N_1 \times N_2$ secret bits and SP^2 contains the remaining bits.
Step 2: Transform secret bits b_{iz}, $z = \{1, \ldots, q\}$ in SP^1 to w_{iz} using formula (9), and then obtain the compound sequence C_i for each unit EU_i.
Step 3: Encrypt C_i by pk and denote it as $E(C_i)$.
Step 4: Embed $E(C_i)$ into $E(V_i)$ using CDM. By formula (10), obtain a new vector $E(V_i')$.
Step 5: Encode secret bits $(b_{i1} \cdots b_{ik})_2$ in SP^2 to D_i. D_i is the decimal of $(b_{i1} \cdots b_{ik})_2$ for each unit EU_i.
Step 6: Encrypt D_i by pk and denote it as $E(D_i)$.
Step 7: Embed $E(D_i)$ into $E(ts_i)$ using VE. By formula (15), obtain a new signal $E(ts_i')$. Then, generate a marked unit $\mathrm{EU}_i' = (E(V_i'), E(ts_i'))$.
Data extraction and image recovery phase
Input: The marked unit EU_i', the number of spreading sequences q, the expanded parameter k, and private key sk.
Output: Secret bits and the original image.
Step 1: Decrypt EU_i' using sk to generate $U_i' = (V_i', ts_i')$.
Step 2: Extract $q \times N_1 \times N_2$ secret bits from V_i' by formulas (9) and (12).
Step 3: Obtain $(b_{i1} \cdots b_{ik})_2$ by extracting k LSBs of ts_i'.
Step 4: Calculate ts_i by formula (16) and recover t_i by formula (17).
Step 5: Recover the original image by calculating $p_i = a_{1i}' + \cdots + a_{li}' + t_i$.

Algorithm 1

Table 1: The performance of EC, ER, and PSNR for different images.

Test image	q and k	EC (bits)	ER (bpp)	PSNR (dB)
Lena	$q = 1$	262144	1	$+\infty$
	$q = 1, k = 1$	524288	2	46.37
	$q = 1, k = 2$	786432	3	36.85
	$q = 3$	786432	3	$+\infty$
	$q = 3, k = 1$	1310720	4	29.52
Pepper	$q = 1$	262144	1	$+\infty$
	$q = 1, k = 1$	524288	2	43.39
	$q = 1, k = 2$	786432	3	36.42
	$q = 3$	786432	3	$+\infty$
	$q = 3, k = 1$	1310720	4	29.53
Sailboat	$q = 1$	262144	1	$+\infty$
	$q = 1, k = 1$	524288	2	46.37
	$q = 1, k = 2$	786432	3	36.87
	$q = 3$	786432	3	$+\infty$
	$q = 3, k = 1$	1310720	4	29.52
Baboon	$q = 1$	262144	1	$+\infty$
	$q = 1, k = 1$	524288	2	46.36
	$q = 1, k = 2$	786432	3	36.87
	$q = 3$	786432	3	$+\infty$
	$q = 3, k = 1$	1310720	4	29.52

Table 2: Efficiency performance of the proposed method.

Test images	Encryption (m)	Embedding (m)	Decryption (m)
Lena	94.76	1.79	181.70
Pepper	93.48	1.46	183.64
Sailboat	95.66	1.42	180.78
Baboon	94.74	1.51	185.29

Table 3: The signal expansion between our method and the selected methods.

Methods	Input signal (bits)	Output signal (bits)
Our method	8	$2\times(q+2)\times 512$
Wu et al. [31]	8	6×512
Shiu et al. [29]	8	2×512
Chen et al. [28]	8	4×512

technology for protecting image content security although it is not efficient enough.

Table 3 denotes signal expansion between our method and the selected methods, including Wu et al. [31], Shiu et al. [29], and Chen et al. [28]. According to Paillier encryption, the size of the cipher space is square of the size of the message space. In the table, Paillier encryption with 512-level security is adopted, and the plain text and the ciphertext are represented by 512 bits and 1024 bits, respectively. The output signal of our method is related to q and can reach $2 \times (q + 2) \times 512$. To solve the problem of data expansion, three alternative methods can be considered. The first is to use a lower security level, such as a 128-bit security level and 64-bit security level. Thus, the length of output signal will be reduced to $2 \times (q + 2) \times 128$ bits and $2 \times (q + 2) \times 64$ bits, respectively. The second is to reduce the value of q. When $q = 1$, it is $6\times512\times512$ and the same as Wu et al.'s method. The third is to use a pixel block instead of a single pixel as the input signal. If a 512-bit security level is used, an 8×8 pixel block can be adopted since such a block consists of $8 \times 8 \times 8 = 512$ bits. Now, the length of the output signal is $2 \times (q + 2)$ times that of the input signal.

The average PSNR of different images when $q = 1$and $k = 2$ are shown in Figure 5. At the beginning, the secret bits can be embedded using CDM, and the embedding rate can reach 1 bpp. Since the decrypted image is the same as the original image, PSNR is $+\infty$. To further improve the embedding rate, the PSNR will be reduced and more secret bits will be embedded using VE. Compared with other related methods including Wu et al. [31], Shiu et al. [29], and Chen et al. [28], the proposed method has a better performance in terms of the embedding rate and PSNR.

Moreover, since the decrypted image is the original image, the proposed method, which only uses CDM, can be applied to deal with encoded media such as H.264 video and JPEG images. Besides, real reversibility can be achieved by the proposed method.

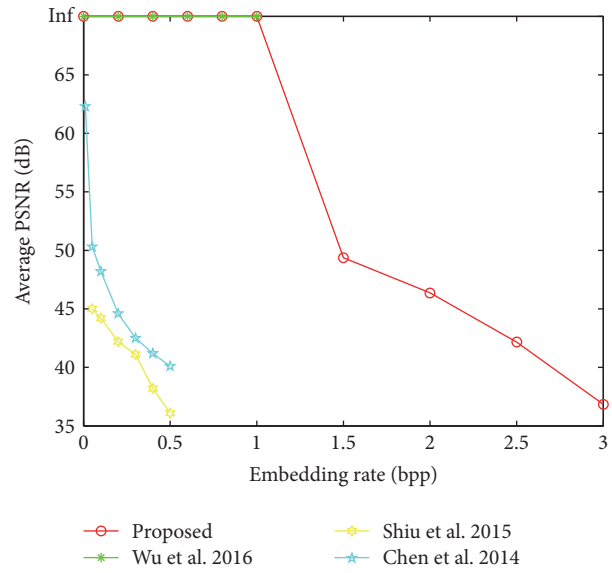


Figure 5: The average PSNR in different test images when $q = 1$ and $k = 2$.

5. Conclusion

This paper proposes improved encrypted signals-based reversible data hiding based on code division multiplexing and value expansion. When only using code division multiplexing to embed bits, lossless visual quality of directly decrypted images can be achieved, and the embedding rate can reach q ($q = \{1, 3\}$) bpp. When using code division multiplexing and value expansion method, the decrypted image will have some distortion, but the embedding rate can be improved to $q + k$ ($k = \{1, 2\}$) bpp. The secret bits and the original image can be recovered from the decrypted image completely. However, the size of the encrypted image will be expanded because of Paillier encryption. In further work, the expansion problem may be solved using other encryption ways, and the property of multiplicative homomorphism may be utilized to further improve the performance on the embedding rate and PSNR.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) (61502242, U1536206, 61672294, 61702276, 61602253, 61373133, 61232016, 71501101, and U1405254), Jiangsu Basic Research Programs-Natural Science Foundation (BK20150925), Startup Foundation for Introducing Talent of Nanjing University of Information Science and Technology (2014r026#), Open Fund of Demonstration Base of Internet Application Innovative Open Platform of Department of Education (KJRP1402), Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) Fund, Collaborative Innovation Center

of Atmospheric Environment and Equipment Technology (CICAEET) Fund, and National Ministry of Science and Technology Special Project Research (GYHY201301030, 2013DFG12860, and BC2013012).

References

[1] J. Shen, D. Liu, Q. Liu, and X. Sun, "A secure cloud-assisted urban data sharing framework for ubiquitous-cities," *Pervasive & Mobile Computing*, 2017.

[2] Z. Fu, F. Huang, X. Sun, A. V. Vasilakos, and C. Yang, "Enabling semantic search based on conceptual graphs over encrypted outsourced data," in *Proceedings of the IEEE Transactions on Services Computing*, vol. 99, 1939.

[3] C. Yuan, X. Sun, and R. Lv, "Fingerprint liveness detection based on multi-scale LPQ and PCA," *China Communications*, vol. 13, no. 7, pp. 60–65, 2016.

[4] J. Shen, S. Chang, J. Shen, Q. Liu, and X. Sun, "A lightweight multi-layer authentication protocol for wireless body area networks," *Future Generation Computer Systems*, 2016.

[5] X. Chen, S. Chen, and Y. Wu, "Coverless information hiding method based on the Chinese character encoding," *Journal of Internet Technology*, vol. 18, no. 2, pp. 313–320, 2017.

[6] J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding-new paradigm in digital watermarking," *EURASIP Journal on Applied Signal Processing*, no. 2, pp. 185–196, 2002.

[7] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.

[8] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Transactions on Circuits Systems for Video Technology*, vol. 16, no. 3, pp. 354–362, 2006.

[9] S. Lee, C. D. Yoo, and T. Kalker, "Reversible image watermarking based on integer-to-integer wavelet transform," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 321–330, 2007.

[10] H. Yao, C. Qin, Z. Tang, and Y. Tian, "Improved dual-image reversible data hiding method using the selection strategy of shiftable pixels' coordinates with minimum distortion," *Signal Processing*, vol. 135, pp. 26–35, 2017.

[11] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 9, pp. 2546–2559, 2016.

[12] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2594–2608, 2016.

[13] Y. Kong, M. Zhang, and D. Ye, "A belief propagation-based method for task allocation in open and dynamic cloud environments," *Knowledge-Based Systems*, vol. 115, pp. 123–132, 2017.

[14] J. Shen, J. Shen, X. Chen, X. Huang, and W. Susilo, "An efficient public auditing protocol with novel dynamic structure for cloud data," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 10, pp. 2402–2415, 2017.

[15] Z. Fu, F. Huang, K. Ren, J. Weng, and C. Wang, "Privacy-preserving Smart Semantic Search based on Conceptual Graphs over Encrypted Outsourced Data," *IEEE Transactions on Information Forensics Security*, no. 99, p. 1, 2017.

[16] X. Zhang, Z. Qian, G. Feng, and Y. Ren, "Efficient reversible data hiding in encrypted images," *Journal of Visual Communication and Image Representation*, vol. 25, no. 2, pp. 322–328, 2014.

[17] L. Xiong, Z. Xu, and Y.-Q. Shi, "An integer wavelet transform based scheme for reversible data hiding in encrypted images," *Multidimensional Systems and Signal Processing*, pp. 1–12, 2017.

[18] X. P. Zhang, "Reversible data hiding in encrypted image," *IEEE Signal Processing Letters*, vol. 18, no. 4, pp. 255–258, 2011.

[19] W. Hong, T.-S. Chen, and H.-Y. Wu, "An improved reversible data hiding in encrypted images using side match," *IEEE Signal Processing Letters*, vol. 19, no. 4, pp. 199–202, 2012.

[20] X. Liao and C. Shu, "Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels," *Journal of Visual Communication and Image Representation*, vol. 28, pp. 21–27, 2015.

[21] S. Yi and Y. Zhou, "Binary-block embedding for reversible data hiding in encrypted images," *Signal Processing*, vol. 133, pp. 40–51, 2017.

[22] X. P. Zhang, "Separable reversible data hiding in encrypted image," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 826–832, 2012.

[23] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, "Reversible data hiding in encrypted images by reserving room before encryption," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 3, pp. 553–562, 2013.

[24] X. Cao, L. Du, X. Wei, D. Meng, and X. Guo, "High Capacity Reversible Data Hiding in Encrypted Images by Patch-Level Sparse Representation," *IEEE Transactions on Cybernetics*, vol. 46, no. 5, pp. 1132–1143, 2016.

[25] W. M. Zhang, K. D. Ma, and N. H. Yu, "Reversibility improved data hiding in encrypted images," *Signal Processing*, vol. 94, no. 1, pp. 118–127, 2014.

[26] W. Zhang, H. Wang, D. Hou, and N. Yu, "Reversible data hiding in encrypted images by reversible image transformation," *IEEE Transactions on Multimedia*, vol. 18, no. 8, pp. 1469–1479, 2016.

[27] D. Hou, W. Zhang, and N. Yu, "Image camouflage by reversible image transformation," *Journal of Visual Communication and Image Representation*, vol. 40, pp. 225–236, 2016.

[28] Y.-C. Chen, C.-W. Shiu, and G. Horng, "Encrypted signal-based reversible data hiding with public key cryptosystem," *Journal of Visual Communication and Image Representation*, vol. 25, no. 5, pp. 1164–1170, 2014.

[29] C.-W. Shiu, Y.-C. Chen, and W. Hong, "Encrypted image-based reversible data hiding with public key cryptography from difference expansion," *Signal Processing: Image Communication*, vol. 39, pp. 226–233, 2015.

[30] X. Zhang, J. Long, Z. Wang, and H. Cheng, "Lossless and Reversible Data Hiding in Encrypted Images with Public-Key Cryptography," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 9, pp. 1622–1631, 2016.

[31] X. Wu, B. Chen, and J. Weng, "Reversible data hiding for encrypted signals by homomorphic encryption and signal energy transfer," *Journal of Visual Communication and Image Representation*, vol. 41, pp. 58–64, 2016.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 7080673, 11 pages
https://doi.org/10.1155/2018/7080673

Research Article

A Novel AMR-WB Speech Steganography Based on Diameter-Neighbor Codebook Partition

Junhui He,[1] **Junxi Chen**,[1] **Shichang Xiao**,[1] **Xiaoyu Huang**,[2] **and Shaohua Tang**[1]

[1]*School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China*
[2]*School of Economics and Commerce, South China University of Technology, Guangzhou 510006, China*

Correspondence should be addressed to Junhui He; hejh@scut.edu.cn

Received 28 September 2017; Accepted 26 December 2017; Published 13 February 2018

Academic Editor: Rémi Cogranne

Steganography is a means of covert communication without revealing the occurrence and the real purpose of communication. The adaptive multirate wideband (AMR-WB) is a widely adapted format in mobile handsets and is also the recommended speech codec for VoLTE. In this paper, a novel AMR-WB speech steganography is proposed based on diameter-neighbor codebook partition algorithm. Different embedding capacity may be achieved by adjusting the iterative parameters during codebook division. The experimental results prove that the presented AMR-WB steganography may provide higher and flexible embedding capacity without inducing perceptible distortion compared with the state-of-the-art methods. With 48 iterations of cluster merging, twice the embedding capacity of complementary-neighbor-vertices-based embedding method may be obtained with a decrease of only around 2% in speech quality and much the same undetectability. Moreover, both the quality of stego speech and the security regarding statistical steganalysis are better than the recent speech steganography based on neighbor-index-division codebook partition.

1. Introduction

With the rapid development of the Internet and the growing popularity of instant messaging application, people are increasingly using audio-based communication. How to avoid interception and secure communication turns into one of the most important research problems. Encryption is a conventional method of protecting communication; however, the transmission of ciphered content may easily arouse attackers' suspicion. In recent years, steganography has been presented as an effective means of covert communication. Audio steganography can transfer important messages secretly by embedding them into cover audio files with the use of information hiding techniques [1]. Data hiding in audio is especially challenging because the human auditory system operates over a wider dynamic range in comparison with human visual system.

Many works on audio steganography have been already reported. Gruhl et al. [2] proposed an audio steganographic method of echo hiding by the introduction of synthetic resonances in the form of closely spaced echoes. Gopalan [3] presented a method of embedding a covert audio message into a cover utterance by altering one bit in each of the cover utterance samples. Gopalan et al. [4] provided two methods of secret message embedding by modifying the phase or amplitude of perceptually masked or significant regions of a host. And a direct-sequence spread-spectrum watermarking method with strong robustness against common audio editing procedures was proposed in [5]. And many audio steganographic applications including Steghide and Hide4PGP can be freely downloaded from the Internet. But most of these methods are not resilient to AMR-WB speech.

Based on segmental SNR analysis of modification to the encoded bits in a frame, Liu et al. [6] selected the perceptually least important bits to embed secret message in G.729 speech. In [7], a simple and effective steganographic approach, which may be applied to 5.3 Kbps G.723.1 speech, was presented based on analyzing the redundancy of code parameters, and augmented identity matrix was utilized to lower the distortion of cover speech. Similarly, by calculating speech quality sensitivity on each encoded bit out of 244 bits using perceptual evaluation of speech quality (PESQ) criterion, a

data hiding approach to embedding data in enhanced full rate (EFR) compressed speech bitstream is proposed in [8]. In addition, Nishimura [9] proposed three methods of hiding data in the pitch delay data of the AMR speech.

Based on complementary neighbor vertices codebook partition algorithm (CNV), Xiao et al. [10] presented an approach to information hiding in compressed speech with the use of quantization index modulation (QIM) [11]. Huang et al. [12] proposed a steganographic algorithm for embedding data in different speech encoding parameters of the inactive frames, the embedding capacity of which is bounded by the number of inactive frames in the cover speech. In [13], Huang et al. also presented a method for steganography in low bit-rate VoIP streams based on pitch period prediction. It can achieve high quality of stegospeech and prevent statistical steganalysis, but the embedding rate is still low (only about 133.3 bps). And an adaptive suboptimal pulse combination constrained (ASOPCC) method was presented in [14] to embed data into compressed speech signal of AMR-WB codec. However, most of the PESQ scores in different coding modes are not high. In [15], a key-based codebook partition strategy, which dynamically determines the adopted division scheme, was designed to improve the security of the QIM steganography in speech bitstream. Although the stegospeech quality is guaranteed to be good, the embedding capacity is very limited and not adjustable. Liu et al. [16] proposed a neighbor-index-division codebook division algorithm (NID) for G.723.1 speech. Differing from the existing CNV method, NID divides neighbor-indexed codewords into separated subcodebooks according to a suitable stegocoding strategy. The embedding capacity is improved by using multiple division and multi-ary coding strategy.

The adaptive multirate wideband (AMR-WB) is a widely adapted format in mobile handsets and is also the recommended speech codec for VoLTE. AMR-WB speech may be a good candidate for cover medium in audio steganography. Therefore, we will focus on AMR-WB speech steganography in this paper. Firstly, a new diameter-neighbor (DN) codebook partition algorithm toward AMR-WB speech is proposed. Based on DN codebook division, we develop a novel AMR-WB speech steganography capable of providing flexible embedding capacity with different iterative parameter N_i. For example, when $N_i = 48$, twice the embedding capacity of CNV-based method may be obtained with a decrease of only about 2% in speech quality and much the same undetectability. Moreover, both the quality of stego speech and the security of defending against statistical steganalysis [17, 18] are better than the recent NID-based speech steganography.

The remainder of this paper is organized as follows. In Section 2, the related work is briefly introduced. In Section 3, the proposed DN codebook partition algorithm and the novel AMR-WB speech steganography are described in detail. The experimental results and analysis are provided in Section 4. Finally, conclusions are presented.

2. Related Work

In this section, a technical overview of AMR-WB codec is firstly presented. Then two related codebook partition algorithms CNV [10] and NID [16] are also briefly reviewed.

2.1. AMR-WB Codec. The AMR-WB speech codec is standardized by 3GPP (3rd Generation Partnership Project) and adopted as the standard G.722.2 by ITU-T in 2002 [19]. It is a multirate wideband speech codec applied in modern mobile communication systems to remarkably improve the speech quality. The AMR-WB codec operates at a multitude of bit rates ranging from 6.6 kbit/s to 23.85 kbit/s.

The input audio signal is separated into 20 ms long frame using 16 kHz sampling rate. Every frame contains a linear prediction analysis (LPA) and the LP coefficients are converted to immittance spectrum pairs (ISP) coefficients. ISP coefficients are then converted to frequency domain (ISF) for quantization. Except for mode 0 (6.6 kbit/s), the ISF coefficients are quantized using two-stage vector quantization with split-by-2 in first stage and split-by-5 in the second stage. Both the second and the third codebooks in the second stage have 128 codewords, and the ISF indices of the codewords in these codebooks may be employed to embed secret message.

In the decoder, the transmitted indices are first parsed from the received bitstream and then decoded to obtain the code parameters for each transmitted frame, such as the ISP vector, the 4 fractional pitch lags, the 4 LTP filtering parameters, the 4 innovative code vectors, and the 4 sets of vector quantized pitch and innovative gains. For a more detailed description, one should refer to [19]. From the received ISF indices, which may have been modified because of secret message embedding, the receiver can recover the embedded secret message.

2.2. Complementary Neighbor Vertices. CNV is a new type of codebook partition algorithm proposed in [10], in which each codeword in a codebook is viewed as a vertex in the multidimensional space. The relationship between two codewords X and Y is described as an edge connecting the two codewords' vertices. And the weight of an edge is defined as the Euclidean distance $D(XY)$ between two codewords X and Y. Small value of $D(XY)$ indicates that X and Y bear a close resemblance to each other. The vertex nearest to X is referred to as X's neighbor vertex, which is denoted by $N(X)$. The vertex set V together with the edge set E form a graph $G(V, E)$ in a multidimensional space.

The codebook partition is realized by the construction of the graph $G(V, E)$ and vertex labelling. First, each vertex X in $G(V, E)$ is connected with its neighbor vertex $N(X)$ using an edge. Thus, the graph $G(V, E)$ would be divided into several isolated subgraphs, each of which may be proved to be acyclic and 2-colorable. Second, every vertex and its neighbor vertex in a subgraph are labelled oppositely using "0" or "1." Third, all of the vertices with same label are collected into a subcodebook; hence, two subcodebooks will be obtained.

Based on the generated subgraphs and the label assigned to each codeword in them, CNV-based steganography applies QIM concept to embed secret message. More specifically, when the label of the codeword X, which is associated with the cover quantization index I_a, agrees with the secret message, I_a remains unchanged, or else it should be replaced

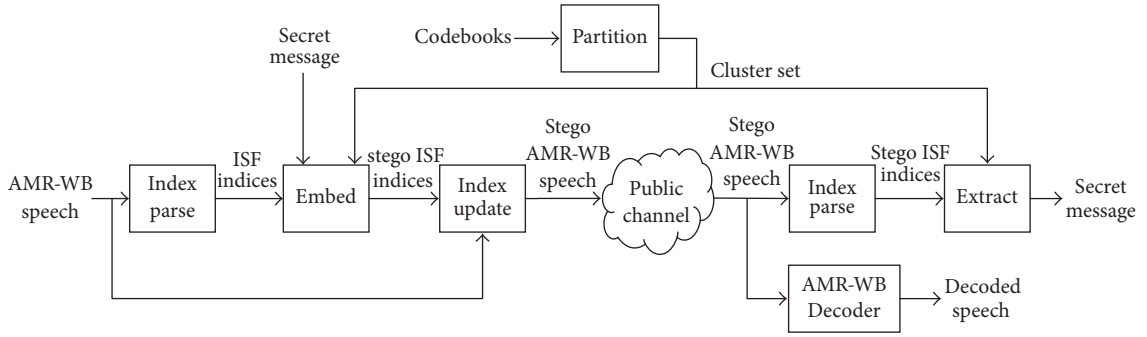


Figure 1: Diagram of the proposed method.

with the quantization index of the neighbor codeword $N(X)$ which belongs to the opposite subcodebook.

The key characteristic of CNV-based steganography is that the distortion is bound even in the worst case. However, the embedding capacity is limited, which is analyzed experimentally in Section 4. Moreover, the number of possible combinations of flipping coefficients which determine whether the labels in a subgraph will be flipped is large. Extra information about the flipping process must be transmitted to the receiver and thus the effective embedding capacity may be decreased further.

2.3. Neighbor Index Division. NID assumes that the codewords of neighbor indices (i.e., neighbor positions) in a codebook would be close together. Hence the codewords in a codebook can be easily separated into subcodebooks according to their indices instead of the Euclidean distance. Specifically, select an appropriate integer k according to the demand for embedding capacity and label the ith codeword with digit $(i-1) \bmod k$, respectively. Then collect all the codewords with same label into a subcodebook and obtain k different subcodebooks.

In order to take full use of the embedding capacity, the binary secret message should be transformed into k-ary digits denoted by m ($m \in \{0, 1, \ldots, k-1\}$). When the codeword related to the cover quantization index belongs to the subcodebook whose label differs from the k-ary digit m to be embedded, this index should be substituted with that of the closest codeword in the corresponding subcodebook$_m$.

NID-based steganography is an information hiding method based on neighbor-index codebook partition, of which the embedding capacity may be controlled by the number of subcodebooks k. However, as illustrated in [16], only about 34% of the pairs of neighbor-index codewords happened to be the pairs of neighbor-vertex codewords. And the mean distance between neighbor-index codewords is apparently larger than that of neighbor-vertex codewords. Therefore, the amount of distortion induced by NID-based steganography may be a little large, which is proved by the experimental results provided in Section 4.

3. Proposed Method

The diagram of the proposed method is shown in Figure 1. Based on DN codebook partition of the codebooks described in Section 2.1, secret message can be embedded into an AMR-WB speech file. After the stego AMR-WB speech file is received, the embedded secret message can be extracted without errors. At the same time, the decoded speech without perceptible distortion will also be obtained. In the following section, the diameter-neighbor codebook partition algorithm (DN) is first introduced. Then the embedding and extraction procedure of our proposed method are described.

3.1. Codebook Partition. A codebook may be viewed as a list of isolated code vectors (i.e., codewords) in the multidimensional space. The codebook partition algorithm used for audio steganography is to divide the codebook into several clusters, in each of which the codewords can be replaced with each other without causing perceptible distortion.

Let B denote the original codebook with N_b codewords, and C denote a cluster with N_c codewords W_t ($t = 1, 2, \ldots, N_c$), and the centroid G of a cluster C is defined as follows:

$$G(i) = \frac{1}{N_c} \sum_{t=1}^{N_c} W_t(i), \tag{1}$$

where $G(i)$ and $W_t(i)$ are the ith components of G and W_t, respectively.

The centroid G (average code vector) is used to represent the corresponding cluster C; hence, the cluster C may also be considered as a vector in the multidimensional codebook space. In order to describe the similarity between two clusters C_1 and C_2, the Euclidean distance between them is defined as follows:

$$D(C_1, C_2) = \sqrt{\sum_{i=1}^{n} \left(G_1(i) - G_2(i)\right)^2}, \tag{2}$$

where G_1 and G_2 are the corresponding geometric center points of the two clusters C_1 and C_2. And n is the dimension of a codeword; $G_1(i)$ and $G_2(i)$ are the ith components of G_1 and G_2, respectively.

Let S denote a cluster set. The diameter of S is defined as the maximal Euclidean distance D_m of all cluster pairs in the cluster set S, that is,

$$D\left(C_p, C_q\right) \le D_m \quad \forall p, q = 1, 2, \ldots, |S|, \tag{3}$$

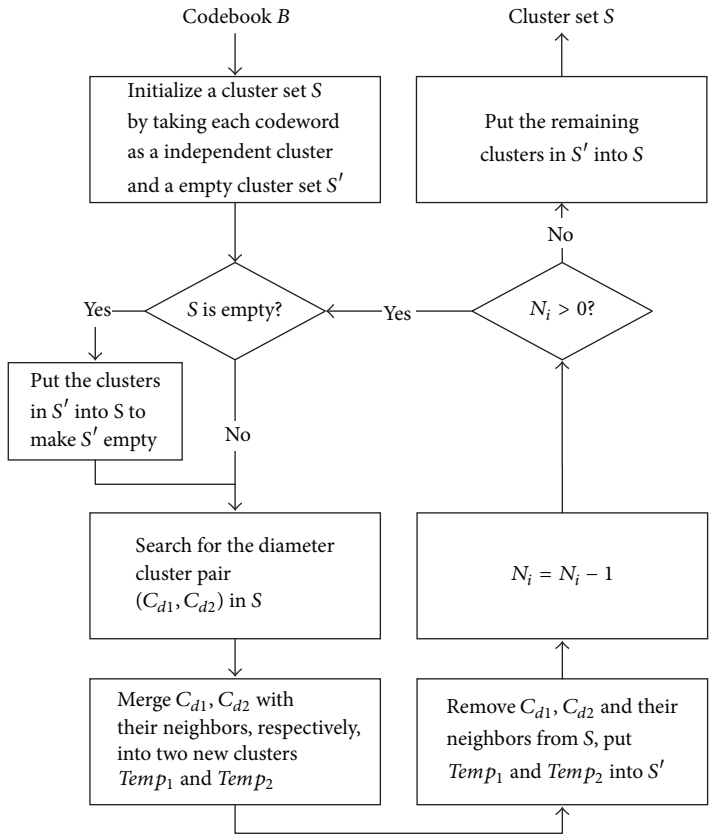


Figure 2: Diagram of our proposed codebook partition.

where $|S|$ is the number of clusters within the cluster set S. The cluster pair with maximal Euclidean distance D_m, called diameter cluster pair, is denoted by (C_{d1}, C_{d2}). And the neighbor of a cluster C in S is represented by $N(C, S)$; then we have

$$D(C, N(C, S)) \leq D\left(C, C_p\right) \quad \forall p = 1, 2, \ldots, |S|. \tag{4}$$

Figure 2 illustrates the diagram of the proposed DN codebook partition algorithm. And its detailed procedure is given in Algorithm 1. The original codebook B will be divided into $|S|$ clusters by iteratively merging the diameter cluster pair with their respective neighbors. An iteration parameter N_i is applied to obtain flexible embedding capacity through controlling the merging procedure. The relationship between N_i and the embedding capacity will be discussed in Section 4.3.

Figure 3 is provided as an example to illustrate the proposed codebook partition algorithm. The white circle "○" denotes a codeword. And the oval "⬭" with shadow denotes a codeword and its neighbor in S being processed, while the oval "⬭" without shadow represents a cluster in S' that has been formed. The "0," "1," "00," "01," "10," or "11" in a circle "○" is the label of a codeword in the cluster. The cross "×" means the centroid of the cluster it belongs to, and a line "–" represents the diameter of a cluster set. The first to third merging iterations are shown in Figures 3(a)–3(c), respectively. The fourth merging iteration is comprised of Figures 3(d) and 3(e), and Figure 3(f) demonstrates the labelling of the codewords.

3.2. Embedding Procedure. In our proposed method, the ISF indices corresponding to the codewords in the codebook are first obtained by parsing the host AMR-WB speech. Then the ISF indices are employed to embed secret message based on codebook partition. Generally, the codewords in the same cluster as the codeword referred by I_a lies in are considered to be replaceable with each other. According to the secret message to be embedded, I_a may be substituted by one of the other codewords' indices within the same cluster. The number of secret message bits that can be embedded depends on the size of the specific cluster. The embedding procedures are given in the following.

Step 1. Search cluster set S for the cluster C which contains the codeword referred by the ISF index I_a.

Step 2. If there are N codewords in C, the number of secret bits that can be embedded into I_a is calculated as $n = \lfloor \log_2 N \rfloor$.

Step 3. Read n not-yet-embedded bits, denoted by m, from the secret message. I_a is replaced with I_b which indexes the codeword with the same label as m.

Step 4. Repeat Steps 1–3 until all the secret bits are embedded.

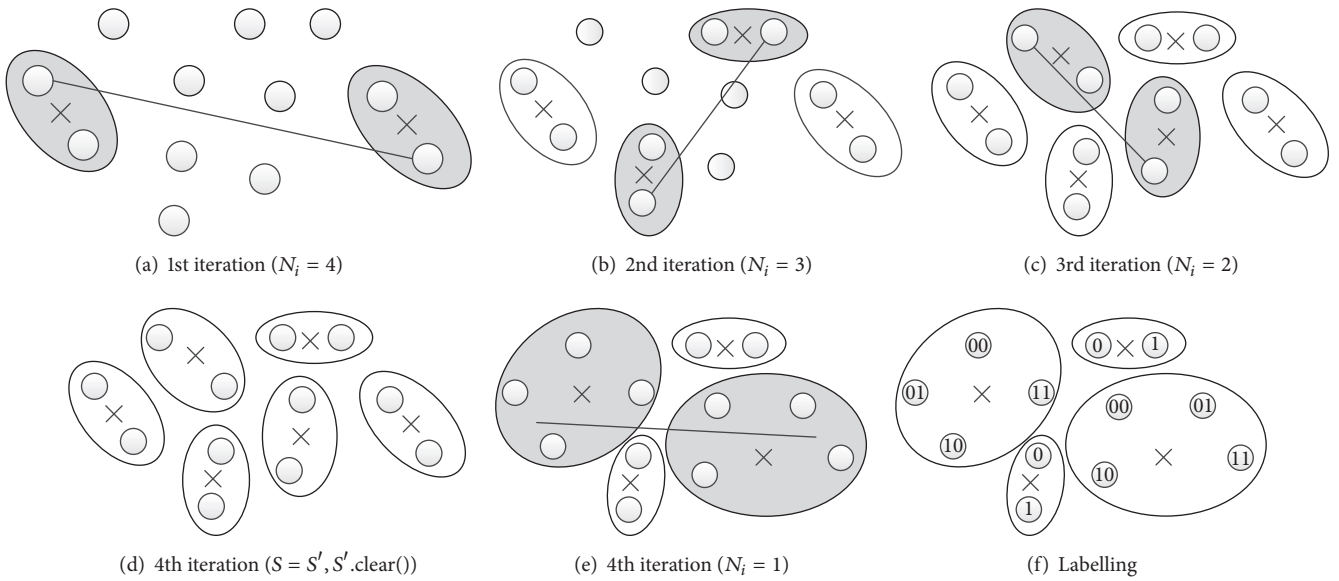


Figure 3: An example of our proposed codebook partition.

Input: Codebook B, iterative parameter N_i
Output: Cluster set S
/* S' is a helper cluster set */
S'.clear();
S.clear();
/* Each codeword is taken as a initial cluster */
for $i = 0$; $i < N_c$; $++i$ **do**
  S.push (C_i);
end
/* Iterative merging */
while $N_i > 0$ **do**
  if S *is empty* **then**
    $S = S'$;
    S'.clear();
  end
  $(C_{d1}, C_{d1}) = \arg\max_{i,j\in\{1,2,\ldots,|S|\}} D(C_i, C_j)$;
  $Temp_1 = C_{d1} \cup N(C_{d1}, S)$;
  $Temp_2 = C_{d2} \cup N(C_{d2}, S)$;
  S'.push ($Temp_1$);
  S'.push ($Temp_2$);
  S.remove (C_{d1});
  S.remove (C_{d2});
  S.remove ($N(C_{d1}, S)$);
  S.remove ($N(C_{d2}, S)$);
  $N_i = N_i - 1$;
end
/* Put the remaining clusters in S' into S */
for iter = S'.begin(); iter < S'.end(); ++iter
 do
  S.push (*iter);
end
return S;

Algorithm 1: DN-based codebook partition algorithm.

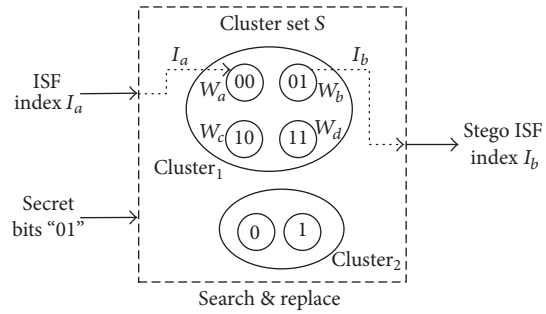


Figure 4: Embedding two bits into one cover ISF index.

Figure 4 is an example of embedding two secret bits into one cover ISF index. Let us assume the cluster set S contains two clusters and the corresponding codeword indexed by I_x is W_x; for example, I_b indexes the codeword W_b. Hence, the ISF index I_a shown in Figure 4 will be replaced with I_b which indexes the codeword W_b with the same label as the secret bits "01."

3.3. Extracting Procedure. When the stego AMR-WB speech is transferred to the intended receiver, the stego indices may be obtained by parsing AMR-WB speech stream and used to extract the embedded secret message. The message extraction procedures from the stegoindex I_b are given below.

Step 1. Search cluster set S, which is the same as that employed in the embedding procedure, for the cluster C which contains the codeword W_b referred by the ISF index I_b.

Step 2. If there are totally N codewords in C, the number of secret bits carried by I_b is computed by $n = \lfloor \log_2 N \rfloor$.

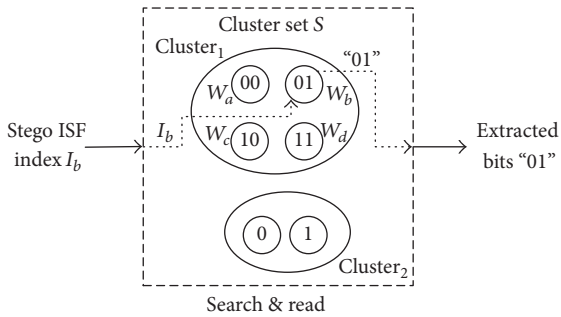


Figure 5: Extracting two bits from one stego-ISF index.

Step 3. Read the label of W_b as the extracted n bits, which are appended to the secret message bit sequence.

Step 4. Repeat Steps 1–3 until all the secret bits are recovered.

Figure 5 is the corresponding example of extracting two secret bits from the stegoindex I_b generated by the previous embedding instance shown in Figure 4. It can be easily seen that the extracted secret bits are identical to the embedded secret bits.

4. Experimental Results and Analysis

In order to demonstrate the performance of the proposed method, the perceptual quality of the stego AMR-WB speech with secret message embedded using our method is computed and compared to that of the stego AMR-WB speech generated with CNV and NID steganography. Moreover, the flexibility of embedding capacity and the security regarding statistical detection are analyzed in detail.

4.1. Audio Database. TIMIT acoustic-phonetic continuous speech corpus (https://catalog.ldc.upenn.edu/ldc93s1) is an audio database which contains broadband recordings of 630 speakers of eight major dialects of American English, each reading ten phonetically rich sentences, and all audio sentences are sampled at 16 kHz. In our experiments, 1000 audio sentences are randomly chosen from TIMIT database. The average, maximum, and minimum length of the chosen audio sentences are 3.47 s, 3.96 s, and 3.12 s. All audio files are converted into AMR-WB format using standard codec.

4.2. Speech Quality Evaluation. The perceptual evaluation of speech quality (PESQ) described in the ITU-T P.862 Recommendation [20] may be employed to evaluate speech quality. Moreover, according to ITU-T P.862.2 [21], the raw PESQ score can be converted to mean opinion score-listening quality objective (MOS-LQO), which is more suitable for evaluating wideband speech. Hence, MOS-LQO is applied in our experiments. The normal range of MOS-LQO score is 1.017 to 4.549. The higher the score, the better the quality.

Figure 6 shows the MOS-LQO scores of the 1000 cover AMR-WB speeches in 23.85 kbit/s mode and the corresponding stego AMR-WB speeches using three different codebook partition algorithms. Three progressive embedding rates, that

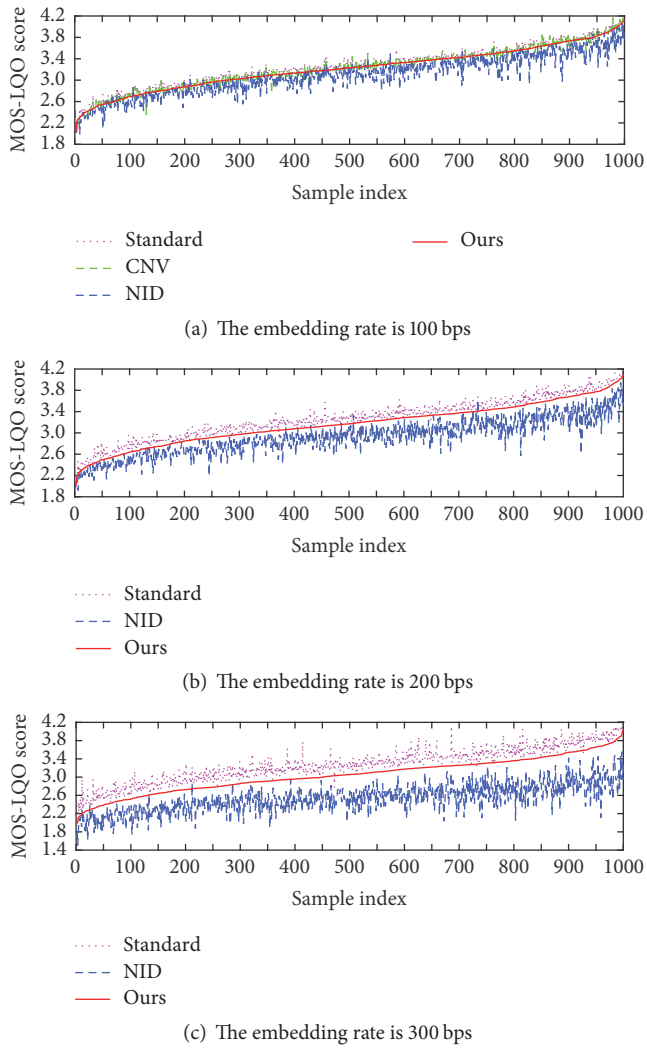


Figure 6: Comparisons of MOS-LQO values for 1000 samples between the standard AMR-WB codec, CNV-based steganography, NID-based steganography, and the proposed DN-based steganography.

is, 100 bps, 200 bps, and 300 bps, are employed in our experiments. The indices of speech samples are sorted according to the MOS-LQO scores of our proposed method. It can be seen from Figure 6 that the overall scores of the stego AMR-WB speeches generated with our method are higher than those of the NID-based stego AMR-WB speeches, especially when the embedding rates are 200 bps and 300 bps. And the MOS-LQO scores of the CNV-based stego AMR-WB speeches are slightly higher than ours when the embedding rate is 100 bps, which means there are no obvious discrepancies in speech quality between them. Besides, when the high embedding rate, that is, 200 bps or 300 bps, is used, the decrease in MOS-LQO scores of our stego AMR-WB speeches is significantly smaller than that of NID-based steganography.

Moreover, the average MOS-LQO scores of the cover AMR-WB speeches and the stego AMR-WB speeches with three different codebook partition algorithms, that is, CNV,

Table 1: MOS-LQO scores of the standard codec, CNV-based, NID-based, and our proposed steganography in four different rate modes and three embedding rates.

Embedding rate	Method	Rate mode (kbit/s)			
		12.65	15.85	19.85	23.85
	Standard	2.929	3.073	3.199	3.269
100 bps	CNV	2.871 (−2.0%)	3.021 (−1.7%)	3.153 (−1.4%)	3.225 (−1.3%)
	NID	2.750 (−6.1%)	2.895 (−5.8%)	3.020 (−5.6%)	3.091 (−5.4%)
	Ours	2.864 (−2.2%)	3.010 (−2.0%)	3.139 (−1.9%)	3.216 (−1.6%)
200 bps	CNV	/	/	/	/
	NID	2.601 (−11.2%)	2.736 (−11.0%)	2.875 (−10.7%)	2.921 (−10.6%)
	Ours	2.807 (−4.2%)	2.955 (−3.8%)	3.084 (−3.6%)	3.164 (−3.2%)
300 bps	CNV	/	/	/	/
	NID	2.284 (−22.0%)	2.386 (−22.3%)	2.475 (−22.6%)	2.533 (−22.5%)
	Ours	2.699 (−7.9%)	2.841 (−7.5%)	2.971 (−7.1%)	3.046 (−6.8%)

NID, and DN, including four rate modes (12.65 kbit/s, 15.85 kbit/s, 19.85 kbit/s, and 23.85 kbit/s) together with three kinds of embedding rate (100 bps, 200 bps, and 300 bps), are given in Table 1. Only the MOS-LQO scores of NID-based and DN-based steganographic methods with embedding rates 200 bps and 300 bps are given in Table 1 because the embedding capacity of CNV-based steganography may not be larger than 100 bps.

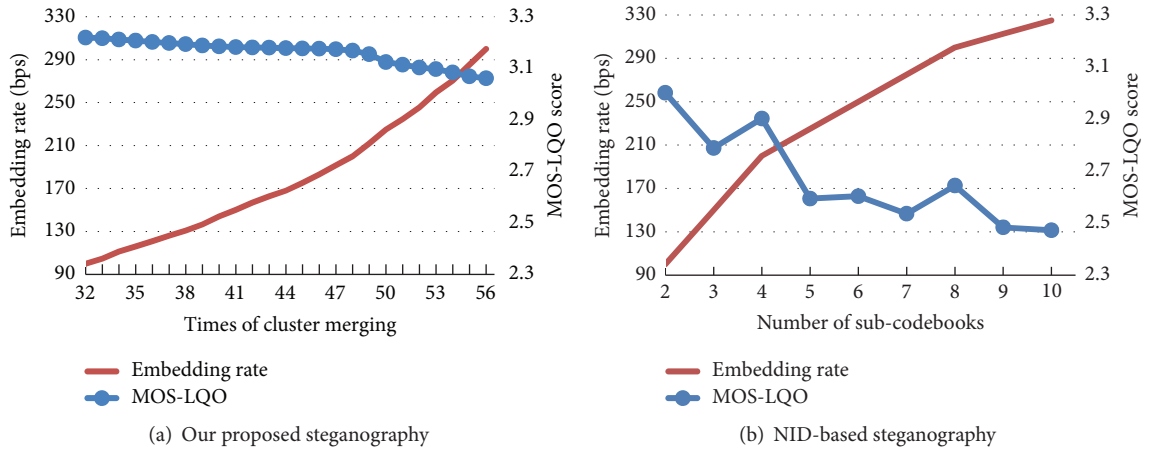
When the embedding rate is 100 bps, which is almost the limit of CNV steganography, we can see from Table 1 that the mean MOS-LQO scores of our proposed method are only about 0.3% worse than CNV-based steganography. The slight decrease may be almost imperceptible by human auditory system (HAS). And there are significant increases of approximately 3.8% in the mean MOS-LQO scores when our presented method is compared to NID-based steganography. And it can be observed that when the embedding rates are 200 bps and 300 bps, the scores of our approach are improved by about 7% and 15% correspondingly in contrast to those of NID-based steganography.

Furthermore, we can also see that the experimental results of four rate modes are analogous. The decrease of speech quality caused by NID-based steganography is more than twice that caused by DN-based steganography. And the proposed method can obtain twice the embedding capacity of CNV-based steganography by sacrificing less than 2% speech quality in average. In addition, only a slight decline in speech quality is observed when 300 bps embedding rate is used in the proposed DN-based method while 200 bps is employed in NID-based method.

4.3. Flexible Embedding Capacity. Compared to CNV-based steganography, flexible embedding capacity may be obtained to satisfy different practical demand with our proposed method. The steganographic capacity can be adjusted by changing the iteration parameter N_i. For different values of N_i, for example, $N_i = 32, 33, \ldots, 54$, the average embedding capacity and the MOS-LQO scores are given in Figure 7(a), and the corresponding results of NID-based steganography are provided in Figure 7(b) for comparison. Without loss of generality, only 23.85 kbit/s mode is used.

From Figure 7, we can observe that the embedding rate significantly increases with N_i while the MOS-LQO score slightly goes down. However, as NID-based steganography is concerned, the MOS-LQO score rapidly declines with the increase of the embedding rate. Therefore, the proposed DN-based steganography can achieve higher embedding capacity with slight decrease in speech quality. For example, when $N_i = 48$, the size of each cluster in S is equal to 4 and we can embed 4 bits per frame; that is, the embedding rate is 200 bps, but, at the same time, the CNV algorithm can embed at most 2 bits per frame (100 bps).

4.4. Resistibility of Statistical Steganalysis. Speech steganography aims to hide secret message into cover speech without arousing suspicion. It is very important for a steganographic method to resist statistical steganalysis, which is the technique of detecting the presence of hidden message. Two state-of-the-art steganalytic methods [17, 18] are used to evaluate the performance of statistical undetectability of our proposed method. In [17], mel-cepstrum coefficients and Markov transition features from the second-order derivative of the audio signal are extracted to capture the statistical distortions caused by audio steganography, while, in [18], the correlation characteristics of split vector quantization codewords of linear predictive coding filter coefficients are



(a) Our proposed steganography (b) NID-based steganography

Figure 7: Relationship between the embedding rates and the MOS-LQO scores for our proposed steganography and NID-based steganography.

Table 2: Steganalysis results of different steganographic methods in 23.85 kbits/s mode.

Training rate	0.4				0.5				0.6			
Method	Markov	MFCC	SS-QCCN	RS-QCCN	Markov	MFCC	SS-QCCN	RS-QCCN	Markov	MFCC	SS-QCCN	RS-QCCN
100 bps												
CNV	49.8%	49.8%	43.7%	49.0%	50.1%	50.2%	44.0%	49.2%	50.0%	50.5%	41.9%	50.0%
NID	51.0%	60.1%	42.2%	50.0%	50.1%	60.9%	42.9%	48.7%	52.1%	59.8%	41.8%	49.4%
Ours	50.0%	50.0%	44.0%	49.4%	50.3%	49.3%	40.3%	49.4%	49.1%	48.6%	41.8%	43.3%
200 bps												
CNV	/	/	/	/	/	/	/	/	/	/	/	/
NID	53.5%	74.5%	46.9%	50.0%	53.3%	76.2%	47.6%	50.0%	53.6%	75.8%	44.4%	50.1%
Ours	51.0%	48.3%	45.2%	50.0%	49.8%	48.7%	42.2%	50.0%	50.5%	48.6%	45.0%	50.0%
300 bps												
CNV	/	/	/	/	/	/	/	/	/	/	/	/
NID	54.8%	74.6%	49.3%	50.0%	56.3%	77.2%	50.0%	50.0%	55.4%	78.3%	50.5%	50.6%
Ours	52.4%	49.7%	47.9%	50.0%	52.8%	60.9%	48.2%	50.0%	53.8%	50.1%	46.6%	50.0%

utilized to steganalyze QIM-based steganography in low-bit-rate speech (such as G.723.1 and G.729). Both steganalytic methods use a support vector machine to predict the existence of hidden message in given audios.

In our experiments, the sentences chosen from "TIMIT" databases as stated in Section 4.1 are first encoded using the standard AMR-WB codec. These AMR-WB recordings constitute the cover speech set. Then secret message is embedded into each cover AMR-WB speech with different embedding rates, that is, 100 bps, 200 bps, and 300 bps, by CNV-based, NID-based, and DN-based steganography. Of course, 200 bps and 300 bps may be omitted for CNV-based steganography because of its limited embedding capacity. And seven stegospeech sets are generated, among which one set is related to CNV-based steganographic method, and each of three sets is associated with NID-based and DN-based steganography, respectively. Moreover, only 23.85 kbit/s mode is used without loss of generality.

In each experiment, a pair of cover and stego speech sets is randomly divided into training and testing sets according to three kinds of training rates, that is, 0.4, 0.5, and 0.6. For example, if the training rate is 0.4, the training set contains 40% speech samples randomly chosen from each of the cover and stegospeech sets, and the remaining 60% samples go into the testing set. As described in [17, 18], LIBSVM [22] is used as a classifier, and radial basis function (RBF) kernel and grid-search technique are employed to obtain better classification performance. For Li et al.'s steganalytic method, the principal component analysis (PCA) is first used, as suggested in [18], to reduce the dimension of feature vectors to 300. Let the samples in cover speech set denote negatives and those in stego speech set stand for positives. Hence, the accuracy may be defined as follows:

$$\text{Accuracy} = \frac{1}{2} \times \left(\frac{\text{TP}}{\text{TP} + \text{FN}} + \frac{\text{TN}}{\text{FP} + \text{TN}} \right), \tag{5}$$

where TP are true positives, TN are true negatives, FN are false negatives, and FP are false positives.

The steganalytic results are given in Table 2, It can be seen that when the embedding rate is 100 bps, the accuracy of detecting both CNV-based and DN-based methods is almost the same, say, 50% or so, while that of detecting

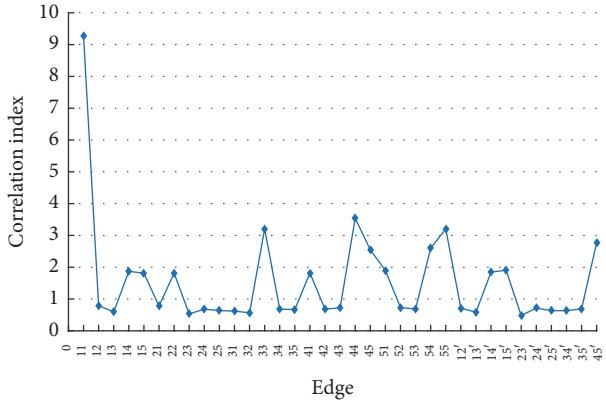


Figure 8: The correlation index of 1000 AMR-WB speeches, where the interframe edge ii connects two vertices $V_i[k]$ and $V_i[k+1]$ in two neighboring frames, and the intraframe edge ij' connects two vertices $V_i[k]$ and $V_j[k]$ in the same frame.

NID-based steganography increases to 60% when MFCC-based steganalytic method is applied. Moreover, there is an apparent increase in the accuracy of detecting NID-based hiding method with the embedding rate increases to 200 bps or 300 bps when Liu et al.'s methods (i.e., Markov and MFCC-based steganalytic methods) are applied. But the accuracy of steganalyzing our proposed method, DN-based steganography, stays at the same level of 50%. Therefore, the proposed method may defend against Liu et al.'s statistical steganalysis [17] even with higher embedding rates.

According to the definition of the correlation index given in [18], the experimental results of the correlation indices of 1000 AMR-WB speeches, which are randomly selected from "TIMIT," are shown in Figure 8. Based on these results, two strong quantization codeword correlation network (QCCN) models, say, SS-QCCN and RS-QCCN, can be constructed as illustrated in Figure 9. These two models are then used to steganalyze our proposed steganography. The steganalytic results are also presented in Table 2. It can be seen from Table 2 that the accuracy of both SS-QCCN and RS-QCCN is less than 50% for all of the AMR-WB stegospeeches. The possible reasons may be that only the second and third codebooks in the second stage are employed in the AMR-WB speech steganography, which means merely the vertices $V_2[k]$ and $V_3[k]$ in the kth frame may be changed during steganography while none of them are utilized in Li et al.'s steganalytic method except for the edge "33" in RS-QCCN model. Besides, we also used an adapted QCCN model (i.e., utilize edges "22," "33," and "$23'$") targeted at AMR-WB speech, but the accuracy is still less than 50%. It may be because the correlation of those edges is not strong enough for steganalysis according to Figure 8. Therefore, it is reasonable to conclude that the AMR-WB speech steganography can defend against the steganalytic method proposed in [18].

In order to visualize the detection performance, we give some receiver operating characteristic (ROC) curves of steganalyzing CNV-based steganography with 100 bps embedding rate and NID-based and DN-based steganography with 100 bps, 200 bps, and 300 bps embedding rates are

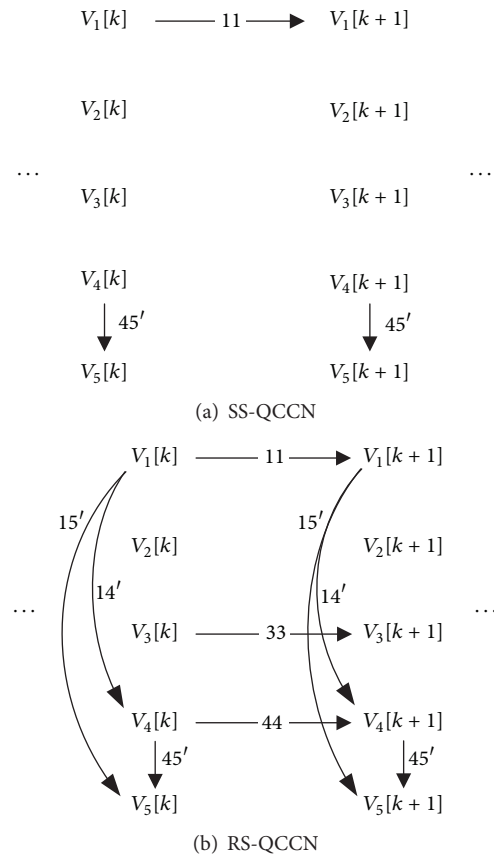


Figure 9: Two AMR-WB strong correlation network models.

provided in Figure 10 (ROC curves for SS-QCCN and RS-QCCN are omitted for these two methods fail to steganalyze AMR-WB steganography in spite of embedding capacity). It shows that all of the three steganographic methods can resist statistical steganalysis when the embedding rate is 100 bps. While the statistical steganalytic methods, especially MFCC-based steganalysis, may detect the existence of hidden message embedded with NID-based steganography when the embedding rate is above 100 bps, the proposed DN-based steganography may still have good security against both Markov-based and MFCC-based steganalysis.

5. Conclusion

The adaptive multirate wideband (AMR-WB) is a widely adapted format in mobile handsets and is also the recommended speech codec for VoLTE. AMR-WB speech may be a good candidate for cover medium in speech steganography. In this paper, a novel AMR-WB speech steganographic method is proposed. The experimental results demonstrated the effectiveness of our proposed method. The main contributions of this paper are as follows:

(1) A novel AMR-WB speech steganography is proposed based on diameter-neighbor codebook partition algorithm. It can provide higher capacity without noticeable decrease in speech quality and better

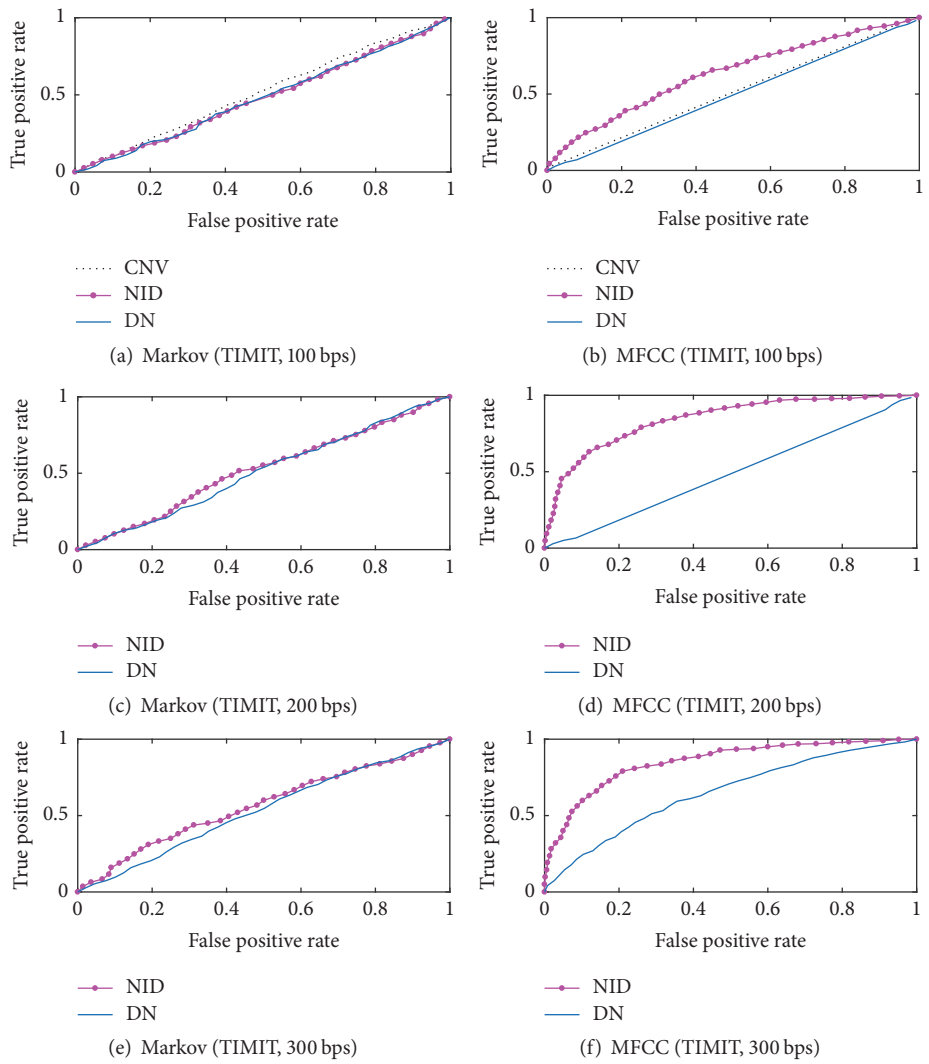


Figure 10: ROC curves for steganalysis of CNV-based, NID-based, and our proposed steganography (50% training rate).

performance against statistical steganalysis than NID-based method.

(2) Flexible embedding capacity may be easily achieved with different iterations of cluster merging. Twice the embedding capacity of CNV-based embedding method may be obtained with $N_i = 48$.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China under Grant no. 61632013.

References

[1] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM Systems Journal*, vol. 35, no. 3-4, pp. 313–335, 1996.

[2] D. Gruhl, A. Lu, and W. Bender, "Echo hiding," in *Information Hiding*, R. Anderson, Ed., vol. 1174 of *Lecture Notes in Computer Science*, pp. 295–315, Springer Berlin Heidelberg, Berlin, Germany, 1996.

[3] K. Gopalan, "Audio steganography using bit modification," in *Proceedings of the 2003 International Conference on Multimedia and Expo, ICME 2003*, pp. I629–I632, USA, July 2003.

[4] K. Gopalan, S. Wenndt, S. Adams, and D. Haddad, "Audio steganography by amplitude or phase modification," in *Proceedings of the Security and Watermarking of Multimedia Contents V*, pp. 67–76, USA, January 2003.

[5] D. Kirovski and H. S. Malvar, "Spread-spectrum watermarking of audio signals," *IEEE Transactions on Signal Processing*, vol. 51, no. 4, pp. 1020–1033, 2003.

[6] L. Liu, M. Li, Q. Li, and Y. Liang, "Perceptually transparent information hiding in G.729 bitstream," in *Proceedings of the 2008 4th International Conference on Intelligent Information Hiding and Multiedia Signal Processing, IIH-MSP 2008*, pp. 406–409, China, August 2008.

[7] T. Xu and Z. Yang, "Simple and effective speech steganography in G.723.1 low-rate codes," in *Proceedings of the 2009*

International Conference on Wireless Communications and Signal Processing, WCSP 2009, China, November 2009.

[8] A. Shahbazi, A. H. Rezaie, and R. Shahbazi, "MELPe coded speech hiding on enhanced full rate compressed domain," in *Proceedings of the Asia Modelling Symposium 2010: 4th International Conference on Mathematical Modelling and Computer Simulation, AMS2010*, pp. 267–270, Malaysia, May 2010.

[9] A. Nishimura, "Data hiding in pitch delay data of the adaptive multi-rate narrow-band speech codec," in *Proceedings of the IIH-MSP 2009-2009 5th International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 483–486, Japan, September 2009.

[10] B. Xiao, Y. Huang, and S. Tang, "An approach to information hiding in low bit-rate speech stream," in *Proceedings of the 2008 IEEE Global Telecommunications Conference, GLOBECOM 2008*, pp. 1940–1944, USA, December 2008.

[11] B. Chen and G. W. Wornell, "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 47, no. 4, pp. 1423–1443, 2001.

[12] Y. F. Huang, S. Tang, and J. Yuan, "Steganography in inactive frames of VoIP streams encoded by source codec," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 2, pp. 296–306, 2011.

[13] Y. Huang, C. Liu, S. Tang, and S. Bai, "Steganography integration into a low-bit rate speech codec," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 6, pp. 1865–1875, 2012.

[14] H. Miao, L. Huang, Z. Chen, W. Yang, and A. Al-Hawbani, "A new scheme for covert communication via 3G encoded speech," *Computers and Electrical Engineering*, vol. 38, no. 6, pp. 1490–1501, 2012.

[15] H. Tian, J. Liu, and S. Li, "Improving security of quantization-index-modulation steganography in low bit-rate speech streams," *Multimedia Systems*, vol. 20, no. 2, pp. 143–154, 2014.

[16] J. Liu, H. Tian, J. Lu, and Y. Chen, "Neighbor-index-division steganography based on QIM method for G.723.1 speech streams," *Journal of Ambient Intelligence and Humanized Computing*, vol. 7, no. 1, pp. 139–147, 2016.

[17] Q. Liu, A. H. Sung, and M. Qiao, "Derivative-based audio steganalysis," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 7, no. 3, article no. 18, 2011.

[18] S. Li, Y. Jia, and C.-C. J. Kuo, "Steganalysis of QIM Steganography in Low-Bit-Rate Speech Signals," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 25, no. 5, pp. 1011–1022, 2017.

[19] ITU-T, Wideband Coding of Speech at around 16 Kbps Using Adaptive Multi-rate Wideband (AMR-WB), International Telecommunication Union Std. G.722.2, 2002.

[20] Perceptual Evaluation of Speech Quality (PESQ): An Objective Method for End-to-end Speech Quality Assessment of Narrow-band Telephone Net-works and Speech Codecs, International Telecommunication Union Std. P.862, 2001.

[21] Wideband Extension to Recommendation P.862 for the Assessment of Wideband Telephone Networks and Speech Codecs, International Telecommunication Union Std. P.862.2, 2007.

[22] C. Chang and C. Lin, "LIBSVM: a Library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, article 27, 2011.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 2516324, 12 pages
https://doi.org/10.1155/2018/2516324

Research Article

Reference Sharing Mechanism-Based Self-Embedding Watermarking Scheme with Deterministic Content Reconstruction

Dongmei Niu,[1,2] **Hongxia Wang**,[1] **Minquan Cheng**,[3] **and Canghong Shi**[1]

[1]*Southwest Jiaotong University, Chengdu 610031, China*
[2]*Southwest University of Science and Technology, Mianyang 621010, China*
[3]*Guangxi Normal University, Guilin 541004, China*

Correspondence should be addressed to Hongxia Wang; hxwang@swjtu.edu.cn

Received 29 September 2017; Revised 19 December 2017; Accepted 9 January 2018; Published 11 February 2018

Academic Editor: Xinpeng Zhang

This paper presents a reference sharing mechanism-based self-embedding watermarking scheme. The host image is embedded with watermark bits including the reference data for content recovery and the authentication data for tampering location. The special encoding matrix derived from the generator matrix of selected systematic Maximum Distance Separable (MDS) code is adopted. The reference data is generated by encoding all the representative data of the original image blocks. On the receiver side, the tampered image blocks can be located by the authentication data. The reference data embedded in one image block can be shared by all the image blocks to restore the tampered content. The tampering coincidence problem can be avoided at the extreme. The maximal tampering rate is deduced theoretically. Experimental results show that, as long as the tampering rate is less than the maximal tampering rate, the content recovery is deterministic. The quality of recovered content does not decrease with the maximal tampering rate.

1. Introduction

With the rapid development of information science and computer network techniques, digital images can be easily copied, altered, and spread over the network. The problems of copyright protection, authentication, and integrity identification of digital images [1, 2] are still a focus of multimedia information security research. Self-embedding watermarking technique is proposed [3] to detect the tampered image areas and recover approximately the tampered content. In most self-embedding watermarking schemes, the original image will be partitioned into nonoverlapping blocks. In addition to the authentication data for detecting the tampered image blocks, the reference data for recovering the tampered image blocks is embedded in the redundant space of the image. The watermark data is composed of the reference data and the authentication data. The performance of self-embedding schemes is commonly evaluated by the peak signal to noise ratio (PSNR) between the original image and the watermarked image, PSNR between the recovered image and the watermarked image, and a bound on the allowed amount of modifications, that is, the maximal tampering rate.

In some self-embedding schemes, the reference data is the representative information of the original image blocks such as the prime DCT (Discrete Cosine Transform) coefficients, the MSB (Most Significant Bit) bits of the block pixels or the vector quantization values. The reference data of an image block is usually embedded into another different image block according to the predetermined rule. In [3], the primary DCT coefficients of an image block were quantized and binary encoded, and the resulting bit string was inserted into the LSB of an offset block. In [4], the reference data is also the quantized and encoded DCT coefficients; the embedding position is determined by a block-mapping sequence. As the analysis in [5], this embedding method will result in the problems of the tampering coincidence and watermark waste. The first problem means that when both the image block and the image block containing its representative information are

tampered or lost, the content recovery will fail. The second problem means that when both of them are reserved, the watermark data embedded in will be useless. To alleviate the problem of the tampering coincidence, in some other schemes the representative information for one image block will be duplicated and embedded in the image for many times. As in [6], for each image block, there are two copies of its representative information hidden in two different blocks that provide the second chance for block recovery in case one copy is destroyed. In [7], the representative information of the image block is the VQ index. Compared with [6], more copies of the representative information are embedded. The same processing method appear in [8–11]. By using this method the probability of tampering coincidence is reduced, but the cost of watermark waste increased.

In some other schemes, coding technique is introduced. The reference information is not the representative data of the original image blocks. The reference data is generated by encoding the representative information [12–23]. In [12, 13], the reference sharing mechanism for watermark self-embedding is proposed. The reference information of one image block will be distributed over many blocks. The 5 MSB bits of the image pixels are pseudo-randomly permuted and divided into groups. Each group is multiplied by a pseudo-random binary matrix to generate the reference data. All the reference bits are then pseudo-randomly permuted and divided into groups, which will be embedded in the image blocks. Scrambling and coding make the reference bits to be embedded in an image block derived from the MSB bits of many different image blocks and shared by these blocks for content restoration. When the number of tampered image blocks is not more than a certain threshold, there is always some part of the reference available. So, the tampering coincidence problem is avoided. This thought is also reflected in the other schemes [14–19]. In [17], the reference sharing mechanism for watermark self-embedding is extended. The numbers of the MSB layers to generate the reference bits are flexible and the numbers of LSB layers to accommodate watermark bits are variable. The relationship for the overall performance of self-embedding scheme, the embedding modes that are used, and the ranges of tampering rates are presented. In [21], the erasure channel is taken as the natural model of the self-embedding problem, and the random linear fountain (RLF) code is used to encode the representative information of all image blocks to generate the reference data. The reference bits to be embedded in an image block will be shared by all the image blocks for content restoration; the tampering coincidence problem can be avoided at the extreme. For that reason, with the same rate of reference information per image block, the proposed approach in [21] allows for working with higher tampering rates than other self-embedding schemes. In these classic reference sharing mechanism-based self-embedding schemes, the binary random matrix is used as the encoding matrix. The tampered image blocks will be restored with probability, but not deterministically. In our scheme, we use the special matrix as the encoding matrix to construct the reference sharing mechanism-based self-embedding scheme. The reference information to be embedded in an image block can be shared by all the image blocks. So the tampering coincidence problem can be avoided at the extreme. As long as the tampering rate is not larger than the maximal tampering rate, the representative data of the tampered image blocks can be recovered deterministically.

The remaining part of the paper is organized as follows. Section 2 reviews the reference data generation method of the prior reference sharing mechanism-based schemes. The detailed procedure of the proposed self-embedding watermarking scheme is presented in Section 3. Section 4 analyzes the bound on the maximal tampering rate of the proposed scheme. The experimental evaluation and comparisons with the existing schemes are presented in Section 5.

2. Related Prior Research

The method of the reference sharing for self-embedding schemes is proposed and described in detail in [12, 13]. This technique is also applied in some other schemes [17–19, 21, 22]. But in these schemes the encoding matrices are the binary random matrices. In [13] the original image is an 8-bit gray-level image. The 5 MSB of each pixel are collected and permuted based on the secret key and then divided into M subsets, each of which containing L bits. For each subset, the reference data generation is performed by

$$\left(r_{m,1}, r_{m,2}, \ldots, r_{m,L/2}\right)^T = \mathbf{R}_m \left(c_{m,1}, c_{m,2}, \ldots, c_{m,L}\right)^T, \quad m = 1, 2, \ldots, M, \tag{1}$$

where $\mathbf{R}_m$ is the encoding matrix, which is a binary random matrix sized $L/2 \times L$, and $(c_{m,1}, c_{m,2}, \ldots, c_{m,L})$ is the mth subset of the 5 MSB. The generated reference data will be stored as part of the watermark in the 3 LSB planes of the image block. The M systems of linear equations in (1) establish a link between the reference data and the 5 MSB of the original image. Scrambling and coding make the reference bits embedded in an image block derived from the MSB bits of many different image blocks and shared by these blocks for content restoration.

After the tampering detection procedure, all the image blocks of the watermarked image will be marked as either "tampered" or "reserved." The ratio between the number of tampered image blocks and the number of all blocks is called the tampering rate, which will be denoted as α. The maximal tampering rate is denoted as $\alpha_{\max}$, which is the upper bound of the tampering rate the scheme can tolerate. Recollect the 5 MSB of each pixel while marking the MSB of the tampered blocks as the unknowns. Separate the reference bits from the watermark while marking the reference bits of the tampered blocks as the unknowns. Reconstruct the M systems of linear equations in (1). For each equation, the invalid equations that the reference bit is unknown are removed:

$$\left(r_{m,e(1)}, r_{m,e(2)}, \ldots, r_{m,e(v)}\right)^T = \mathbf{R}_m^{(E)} \left(c_{m,1}, c_{m,2}, \ldots, c_{m,L}\right)^T, \quad m = 1, 2, \ldots, M, \tag{2}$$

where $\left(r_{m,e(1)}, r_{m,e(2)}, \ldots, r_{m,e(v)}\right)^T$ and $\mathbf{R}_m^{(E)}$ are the constant vector and the coefficient matrix after removing the invalid

equations. Then reformulate the equations for the standard system of equations as follows:

$$\left(r_{m,e(1)}, r_{m,e(2)}, \ldots, r_{m,e(v)}\right)^T - \mathbf{R}_m^{(E,R)}\mathbf{C}_R = \mathbf{R}_m^{(E,T)}\mathbf{C}_T, \quad m = 1, 2, \ldots, M, \tag{3}$$

where $\mathbf{R}_m^{(E,T)}$ is the coefficient matrix of the standard system of equations. If the rank of the coefficient matrix $\mathbf{R}_m^{(E,T)}$ is equal to the number of unknowns, the system of equations in (3) will have the unique solution. In other words, the necessary and sufficient condition for the solution of (3) is that, for any submatrix of the binary random matrix, if the number of rows of it is greater than the number of columns, the submatrix is full column rank. However, this condition can only be satisfied with the probability because both the matrix $\mathbf{R}_m$ and the tampering are random. The probability is dependent upon the tampering rate, image size, and system parameter L, which has been deduced in [13].

In brief, using the binary random matrix as the encoding matrix, the procedure of the encoding and decoding will be simple. However, because of the randomness of the encoding matrix and tampering, the tampered image blocks will be restored with probability, but not deterministically. According to the knowledge of coding theory [24], we knew that if $[\mathbf{I} \mid \mathbf{A}]$ is the generator matrix of the systematic MDS code, any square submatrix of $\mathbf{A}$ will be nonsingular. This is equivalent to saying that, for any submatrix of $\mathbf{A}$, if the number of rows of it is greater than the number of columns, the submatrix is full column rank. Due to the excellent properties of the matrix $\mathbf{A}$, we use it in this paper to construct the reference sharing mechanism-based self-embedding watermarking schemes. We select the appropriate matrix $\mathbf{A}$ to generate the reference data by encoding the representative data of all image blocks based on the matrix $\mathbf{A}$. By this way, the generated reference information embedded in an image block will be shared by all the image blocks. Based on this spreading mechanism, our method can be immune to the tampering coincidence and the reference waste. Moreover, after locating the tampered image blocks by the embedded authentication data, as long as the tampering rate is larger than the maximal tampering rate, the restoration will be deterministic due to the use of the matrix $\mathbf{A}$. The tampered image blocks can be reconstructed by using the recovered representative data.

3. Proposed Self-Embedding Watermarking Scheme

Similar to the common self-embedding schemes, the proposed watermarking scheme includes the following two parts: the first one is the watermark generation and embedding and the second one is the tampering detection and content recovery. The detailed process will be described in the following section.

3.1. Watermark Generation and Embedding. Watermark generation and embedding procedure can be divided into four phases: the first one is the representative data generation, the second is the reference data generation, the third one is the authentication data generation, and the last one is watermarking embedding.

3.1.1. Representative Data Generation

Step 1. Divide the original image $\mathbf{I}$ into K nonoverlapping blocks. They are marked as the first, the second, and so on and the Kth block by the Zig-Zag order.

Step 2. Collect the representative data of each image block. The representative data can be the compressed data of image block, for example, the prime DCT coefficients and the MSB bits of the block pixels. There are K representative data blocks in total, which are denoted as $(\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_K)$.

Step 3. Calculate the ratio R. R is the ratio of the length of the representative data block to the size of the redundant space used to embed the reference data in one image block. In our scheme, we suppose R is an integer or $1/R$ is an integer.

For example, suppose the original image is divided into 9 blocks sized 8×8 pixels. The redundant space of an image block is the 3 LSB of all pixels in the image block. If we use 160 bits to store the reference data block data and we use the prime DCT coefficients to represent the image block and quantified and encoded the DCT coefficients to 80 bits, the calculated ratio R is 1/2. If the 5 MSB of all pixels in an image block are extracted as the representative data of this image block, the length of the representative data block is 320 bits and the calculated ratio R is 2.

3.1.2. Reference Data Generation

Step 1. Encoding the representative data of image blocks to generate the reference data $(\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_K)$, there are two cases needed to be considered.

Case 1. The ration R is less than or equal to 1. Encode the K representative data blocks in the following way:

$$\left(\mathbf{C}_{11}, \mathbf{C}_{12}, \ldots, \mathbf{C}_{1,1/R}, \mathbf{C}_{21}, \mathbf{C}_{22}, \ldots, \mathbf{C}_{2,1/R}, \ldots, \mathbf{C}_{K1}, \mathbf{C}_{K2}, \ldots, \mathbf{C}_{K,1/R}\right) = \left(\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_K\right)\mathbf{A}_{K\times K/R}, \tag{4}$$

where $\mathbf{A}$ is the K rows and K/R columns matrix and $[\mathbf{I} \mid \mathbf{A}]$ is the generator matrix of the systematic $((1/R + 1)K, K)$-MDS code over the finite field. The calculating is finished over the finite field. For this purpose, $\mathbf{D}_i$ $(i = 1, \ldots, K)$ will be transformed to an n-dimensional column vector in the finite field. For example, $\mathbf{D}_{11}$ is transformed to $(d_{11}, d_{21}, \ldots, d_{n1})^T$. So, (4) can be rewrote as

$$\left(\mathbf{C}_{11}, \mathbf{C}_{12}, \ldots, \mathbf{C}_{1,1/R}, \mathbf{C}_{21}, \mathbf{C}_{22}, \ldots, \mathbf{C}_{2,1/R}, \ldots, \mathbf{C}_{K1}, \mathbf{C}_{K2}, \ldots, \mathbf{C}_{K,1/R}\right) = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1,K} \\ d_{21} & d_{22} & \cdots & d_{2,K} \\ \vdots & \vdots & \cdots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{n,K} \end{bmatrix}\mathbf{A}_{K\times K/R}. \tag{5}$$

So, $\mathbf{C}_{ij}$ ($i = 1, \ldots, K$, $j = 1, \ldots, 1/R$) are n-dimensional column vectors in the finite field. Let

$$\mathbf{C}_i = \left(\mathbf{C}_{i1}, \mathbf{C}_{i2}, \ldots, \mathbf{C}_{i,1/R}\right) \quad i = 1, \ldots, K. \tag{6}$$

The reference data $(\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_K)$ in the finite field is generated.

Case 2. The ration R is greater than 1. First, $\mathbf{D}_i$ ($i = 1, 2, \ldots, K$) are divided into R smaller blocks $\mathbf{D}_{i1}, \mathbf{D}_{i2}, \ldots, \mathbf{D}_{iR}$. The length of each smaller block will be equal to the size of the redundant space used to embed the reference data block in an image block. Then generate the K data blocks in the following way:

$$\begin{aligned}\left(\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_K\right) = {} & \left(\mathbf{D}_{11}, \mathbf{D}_{12}, \ldots, \mathbf{D}_{1,R}, \mathbf{D}_{21}, \mathbf{D}_{22}, \ldots, \mathbf{D}_{2,R},\right. \\ & \left. \ldots, \mathbf{D}_{K1}, \mathbf{D}_{K2}, \ldots, \mathbf{D}_{KR}\right) \mathbf{A}_{RK\times K},\end{aligned} \tag{7}$$

where $\mathbf{A}$ is the RK rows and K columns matrix and $[\mathbf{I} \mid \mathbf{A}]$ is the generator matrix of the systematic $((R+1)K, RK)$-MDS code over the finite field. As described above, to finish the calculation over the finite field $\mathbf{D}_{ij}$ ($i = 1, \ldots, K$, $j = 1, \ldots, R$) will be transformed to an n-dimensional column vector in the finite field. So, (7) can be rewrote as

$$\left(\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_K\right) = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1,KR} \\ d_{21} & d_{22} & \cdots & d_{2,KR} \\ \vdots & \vdots & \cdots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{n,KR} \end{bmatrix} \mathbf{A}_{RK\times K}. \tag{8}$$

$(\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_K)$ is the reference data in the finite field.

Step 2. Transform $\mathbf{C}_i$ ($i = 1, \ldots, K$) to bit strings. The bit strings are denoted as $(\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_K)$.

The generated reference data blocks $(\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_K)$ will be embedded as a part of watermark into the redundant space of the corresponding image block. From (4) and (7), it can be seen that the data block $\mathbf{C}_{ij}$ or $\mathbf{C}_i$ is the linear combination of all the representative data blocks. That means each data block carries the information of all the image blocks. The reference data block $\mathbf{R}_i$ embedded in any image block can provide the recovery information equally for any tampered image, or the reference data block $\mathbf{R}_i$ embedded in any image block will be shared by all the image blocks. By this way, a global reference share mechanism has been realized.

3.1.3. Authentication Data Generation. For the ith ($i = 1, \ldots, K$) image block, the representative data $\mathbf{D}_i$ and the reference data $\mathbf{R}_i$ are connected and then fed into a hash function to generate the hash bits $\mathbf{H}_i$. The hash values $\{\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_K\}$ are the authentication data blocks which will be embedded into the redundant space of the image block as a part of the watermark. The redundant space of the image block is divided into two parts, one for the reference data and the rest for the hash data. So, the length of the hash data is equal to the length of the rest redundant space. In our experiment, we use the MD5 function; the output is shortened by exclusive disjunction on neighboring bit pairs to generate the required length hash data.

3.1.4. Watermark Embedding

Step 1. The reference data $\mathbf{R}_i$ and the authentication data $\mathbf{H}_i$ are connected and permuted based upon the secret key to generate the watermark $\mathbf{W}_i$ ($i = 1, \ldots, K$).

Step 2. $\mathbf{W}_i$ is embedded into the redundant space of the ith image block. After all the image blocks have been processed, the watermarked image is produced. In our experiment, the 3 LSB of the ith image block is replaced by the watermark $\mathbf{W}_i$ to generate the watermarked image.

3.2. Tampering Detection and Content Recovery. Suppose the watermarked image has been altered without changing the size. For the receiver, the tampered image blocks will be identified and located firstly; then the tampered image blocks will be recovered. So, the tampering detection and content recovery procedure can be divided into two phases: tampered blocks detection and tampered blocks recovery.

3.2.1. Tampered Blocks Detection

Step 1. The received image is divided and the representative data of all the image blocks is collected as in the preprocessing.

Step 2. For the ith image block, the watermark is extracted from the redundant space, scrambled inversely using the same secret key and decomposed into two parts: the reference data block and the hash data.

Step 3. For each image block, input the representative data and the extracted reference to the HASH function to recalculate the hash value.

Step 4. Compare the recalculated hash value and the extracted hash data. If they are different, the image block is judged as a "tampered" image block; otherwise, it is judged as a "reserved" image block.

3.2.2. Tampered Blocks Recovery. As long as the tampering rate is not larger than the maximal tampering rate, we can perfectly recover the failed representative data of the tampered image blocks. The maximal tampering rate will be derived theoretically in Section 4. The procedure of tampered blocks recovery can be illustrated as follows. Take the case that the ration R is less than or equal to 1 as an example; the same result can be derived when R is greater than 1.

Step 1. Reconstruct the linear equations (4). Suppose there are t tampered image blocks. In order to explain the problem simply, it may be assumed that the front t blocks are tampered. The remaining $K - t$ blocks are the reserved image blocks. The reference data extracted from the tampered image blocks and the representative data of them are denoted as $(\mathbf{C}_1^*, \mathbf{C}_2^*, \ldots, \mathbf{C}_t^*)$ and $(\mathbf{D}_1^*, \mathbf{D}_2^*, \ldots, \mathbf{D}_t^*)$. The reference data

extracted from the reserved image blocks and the representative data of them are denoted as $(\mathbf{C}_{t+1}, \mathbf{C}_{t+2}, \ldots, \mathbf{C}_{K-t})$ and $(\mathbf{D}_{t+1}, \mathbf{D}_{t+2}, \ldots, \mathbf{D}_{K-t})$. Divide $\mathbf{C}_i^*$ and C_i into $1/R$ parts:

$$\begin{aligned} \mathbf{C}_i^* &= \left(\mathbf{C}_{i1}^*, \mathbf{C}_{i2}^*, \ldots, \mathbf{C}_{i1/R}^*\right) \quad i = 1, 2, \ldots, t, \\ \mathbf{C}_i &= \left(\mathbf{C}_{i1}, \mathbf{C}_{i2}, \ldots, \mathbf{C}_{i,1/R}\right) \quad i = t+1, t+2, \ldots, K. \end{aligned} \tag{9}$$

Then we can reconstruct the linear equations (4) as

$$\begin{aligned} &\left(\mathbf{C}_{11}^*, \mathbf{C}_{12}^*, \ldots, \mathbf{C}_{11/R}^*, \ldots, \mathbf{C}_{t1}^*, \mathbf{C}_{t2}^*, \ldots, \mathbf{C}_{t1/R}^*, \mathbf{C}_{(t+1)1}, \mathbf{C}_{(t+1)2},\right. \\ &\left.\ldots, \mathbf{C}_{(t+2)1/R}, \ldots, \mathbf{C}_{K1}, \mathbf{C}_{K2}, \ldots, \mathbf{C}_{K1/R}\right) = \left(\mathbf{D}_1^*, \mathbf{D}_2^*, \ldots,\right. \\ &\left.\mathbf{D}_t^*, \mathbf{D}_{(t+1)}, \ldots, \mathbf{D}_K\right) \mathbf{A}_{K\times K/R} \end{aligned} \tag{10}$$

The representative data of the tampered image blocks $(\mathbf{D}_1^*, \mathbf{D}_2^*, \ldots, \mathbf{D}_t^*)$ are the unknowns of the linear equations. We need to solve them by (10).

Step 2. Eliminate the equations that are invalid. $(\mathbf{C}_{11}^*, \mathbf{C}_{12}^*, \ldots, \mathbf{C}_{11/R}^*, \ldots, \mathbf{C}_{t1}^*, \mathbf{C}_{t2}^*, \ldots, \mathbf{C}_{t1/R}^*)$ are from the tampered image blocks. The data may have been tampered. So, their equations are invalid. Cross out the invalid equations and reformulate the system of equations as a standard form of equations.

$$\begin{aligned} &\left(\mathbf{C}_{(t+1)1}, \mathbf{C}_{(t+1)2}, \ldots, \mathbf{C}_{(t+2)1/R}, \ldots, \mathbf{C}_{K1}, \mathbf{C}_{K2}, \ldots, \mathbf{C}_{K1/R}\right) \\ &\quad = \left(\mathbf{D}_1^*, \mathbf{D}_2^*, \ldots, \mathbf{D}_t^*, \mathbf{D}_{(t+1)}, \ldots, \mathbf{D}_K\right) \mathbf{A}_{K\times(K-t)/R}^E, \end{aligned} \tag{11}$$

where $\mathbf{A}_{K\times(K-t)/R}^E$ is the $(K-t)/R$ columns taken from $\mathbf{A}_{K\times K/R}$ corresponding to the extracted correct reference data blocks $(\mathbf{C}_{(t+1)1}, \mathbf{C}_{(t+1)2}, \ldots, \mathbf{C}_{(t+2)1/R}, \ldots, \mathbf{C}_{K1}, \mathbf{C}_{K2}, \ldots, \mathbf{C}_{K1/R})$.

Step 3. Rearranging (11) as the standard form and moving the portion with the unknowns to the right of the equations, we can reformulate (11) as follows:

$$\begin{aligned} &\left(\mathbf{C}_{(t+1)1}, \mathbf{C}_{(t+1)2}, \ldots, \mathbf{C}_{(t+2)1/R}, \ldots, \mathbf{C}_{K1}, \mathbf{C}_{K2}, \ldots, \mathbf{C}_{K1/R}\right) \\ &\quad - \left(\mathbf{D}_{(t+1)}, \ldots, \mathbf{D}_K\right) \mathbf{A}_{(K-t)\times(K-t)/R}^{(E,R)} \\ &\quad = \left(\mathbf{D}_1^*, \mathbf{D}_2^*, \ldots, \mathbf{D}_t^*\right) \mathbf{A}_{t\times(K-t)/R}^{(E,T)}, \end{aligned} \tag{12}$$

where $\mathbf{A}_{(K-t)\times(K-t)/R}^{(E,R)}$ and $\mathbf{A}_{t\times(K-t)/R}^{(E,T)}$ are the rows of $\mathbf{A}_{K\times r/R}^{(E)}$ corresponding to the representative data blocks $(\mathbf{D}_{(t+1)}, \ldots, \mathbf{D}_K)$ and $(\mathbf{D}_1^*, \mathbf{D}_2^*, \ldots, \mathbf{D}_t^*)$, respectively.

Step 4. Solve the t unknowns $(\mathbf{D}_1^*, \mathbf{D}_2^*, \ldots, \mathbf{D}_t^*)$ according to the equations. The calculation will be finished over the finite field. It can be demonstrated that if the tampering rate is not larger than the maximal tampering rate, the number of equations is more than the number of the unknowns. So, we can rewrite (12) as

$$\begin{aligned} &S - \left(\mathbf{D}_{(t+1)}, \ldots, \mathbf{D}_K\right) \mathbf{A}_{(K-t)\times t}^{(E,R,t)} \\ &\quad = \left(\mathbf{D}_1^*, \mathbf{D}_2^*, \ldots, \mathbf{D}_t^*\right) \mathbf{A}_{t\times t}^{(E,T,t)}, \end{aligned} \tag{13}$$

where S is the front t data blocks of $(\mathbf{C}_{(t+1)1}, \mathbf{C}_{(t+1)2}, \ldots, \mathbf{C}_{(t+2)1/R}, \ldots, \mathbf{C}_{K1}, \mathbf{C}_{K2}, \ldots, \mathbf{C}_{K1/R})$. $\mathbf{A}_{t\times t}^{(E,T,t)}$ is the $t \times t$ matrix whose columns are the first t columns of the matrix $\mathbf{A}_{t\times(K-t)/R}^{(E,T)}$. We can see that the matrix $\mathbf{A}_{t\times t}^{(E,T,t)}$ is the square submatrix of $\mathbf{A}$. From [24], we have known that $[\mathbf{I} \mid \mathbf{A}]$ is the generator matrix of the systematic MDS code if and only if any square submatrix of $\mathbf{A}$ is nonsingular. So, $\mathbf{A}_{t\times t}^{(E,T,t)}$ will be nonsingular because $[\mathbf{I} \mid \mathbf{A}]$ is the generator matrix of systematic MDS. Therefore, (13) has a unique solution. We can solve (13) over the finite field to retrieve the original values of $(\mathbf{D}_1^*, \mathbf{D}_2^*, \ldots, \mathbf{D}_t^*)$. So, we can recover the representative data of tampered image blocks definitely. Similarly, the same result can be derived when $R > 1$.

The recovered representative data can be used to reconstruct the tampered image blocks. The quality of recovered content depends on the method of generating the representative data. Provided that the tampering rate is not larger than the maximal tampering rate, the quality of the reconstructed content does not degrade with the tampering area increasing.

4. The Upper Bound on the Tampering Rate

Suppose t image blocks are tampered. Because any square submatrix of the coding matrix $\mathbf{A}$ is nonsingular, as long as the number of equations is more than the number of the unknowns in (12), (13) will have the unique solution. From this the maximal tampering rate $\alpha_{\max}$ can be easily derived theoretically.

Case 1. The ration R is less than or equal to 1. If an image block is identified as a tampered block, there will be one data block and $1/R$ reference data blocks stored in the image block identified as the failed data blocks. After crossing out the invalid equations, there will be $(K-t)/R$ valid equations. In order to make (13) have unique solution, we should have $(K-t)/R \geq t$. From this inequality, we can work out $t/K \leq 1/(R+1)$. t/K is the ratio of the tampered image blocks to all the image blocks. So, in this case the maximal tampering rate $\alpha_{\max}$ is $1/(R+1)$.

Case 2. The ration R is greater than 1. In this case, the same conclusion can be drawn according to the discussion method in Case 1. So, In all cases, the maximal tampering rate

$$\alpha_{\max} = \frac{1}{R+1}. \tag{14}$$

Figure 1 shows the curve about the maximal tampering rate with respect to the ratio R. The curve indicate that the maximal tampering rate will decrease as the ratio R decreases. If we want to improve the restoration capability, we should try to reduce the ratio R by reducing the length of the representative data block or improve the size of the redundant space.

5. Experimental Evaluation and Comparisons

Experiments and comparison were conducted to demonstrate the effectiveness and evaluate the performance of the proposed scheme.

Figure 2(a) is the standard test gray scale image lake sized 512×512 which is used as the host image. The host image is

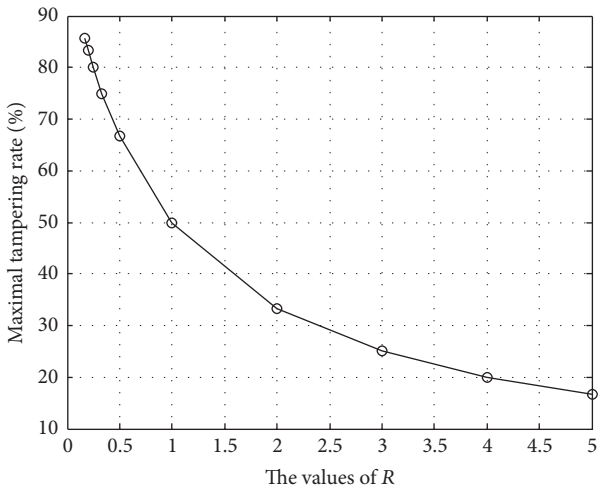


Figure 1: The maximal tampering rate with different R.

divided into blocks sized 8×8 pixels. So, the number of image blocks $K = 2^{12}$. The representative data of an image block is all the 5 MSB of the pixels in the image block. So, the length of the representative data of an image block is 320 bits. The redundant space of an image block is all the 3 LSB of the pixels in the image block. The size of it is 192 bits. We will use 160 bits to store the reference data and the remaining 32 bits to store the hash data. So, we can calculate the ratio R is 2. The coding matrix $\mathbf{A}$ should be a $2^{13} \times 2^{12}$ matrix and $[\mathbf{I} \mid \mathbf{A}]$ should be the systematic MDS code generator matrix. Here we generate the matrix $\mathbf{A}$ by constructing the $2^{13} \times 2^{12}$ Cauchy matrix over $G(2^{16})$ [24].

The 320-bit protected data of each image block will be divided into two smaller blocks with size of 160 bits. So, there are 2^{13} smaller data blocks in total. Each data block will be represented as a column vector of 10 elements in the finite field $G(2^{16})$. Then we calculate the 2^{12} reference data blocks according to (7). Each reference data block will be a column vector of 10 elements in the finite field $G(2^{16})$ and can be transformed into a bits string of length 160 bits. For each image block, the representative data will be linked with the corresponding reference data and then is fed into the MD5 function to produce the hash value. The output hash bits are shortened by exclusive disjunction on neighboring bit pairs to generate the 32 bits hash data. Then, the 160 bits reference data and the 32 bits hash data are linked and permuted. We permuted the 192 bits based on a pseudo-random sequence from the logistic chaotic system and use the initial condition as the secret key. The permuted 192 bits are embedded into the image block by replacing the three LSB planes of the block. This way, the watermarked image is produced. Figure 2(b) is the watermarked Lake. The values of PSNR due to watermark embedding are 37.9 dB. According to (14), it can be derived that the maximal tampering rate of the tested self-embedding scheme is 1/3.

Figure 3 shows the results of the meaningful tampering experiments. The watermarked lake is maliciously tampered with tampering rate $\alpha = 9.8\%$, 21.83%, and 32.69%. The tampered watermarked images are shown in Figure 3((a1)–(a3)) and their corresponding identification and restoration results are shown in Figure 3((b1)–(b3)) and Figure 3((c1)–(c3)). We can see all tampered blocks are located correctly. The tampered blocks are represented by the extreme white. The original MSB of tampered blocks were recovered without any error. In the three cases, PSNR values in the restored area are all 40.7 dB when regarding original image as reference. The quality of the recovered content does not degrade with the growth of tampering rate. We applied the method in [13] and forced the first and second LSB of the restored area as 0 and the third LSB as 1. The experiment demonstrates that if the ratio $R = 1/2$, the proposed scheme can perfectly recover the representative data of the tampered image blocks as long as the tampering rate $\alpha \leq 1/3$.

To evaluate the performance of the proposed scheme, we also conduct the random tampering experiments, seen in Figure 4. The gray scale image Lena, Baboon, Gold Hill, and Airplane in Figure 4((a1)–(d1)) are used as the host images. The watermarked images, shown in Figure 4((a2)–(d2)), are generated as the lake. The values of PSNR due to watermark embedding are 37.9. The watermarked images are tampered randomly with tampering rate $\alpha = 10\%$, 18%, 24%, 33%. The tampered images are shown in Figure 4((a3)–(d3)). The corresponding identification and restoration results are shown in Figure 4((a4)–(d4)) and Figure 4((a5)–(d5)). It can be seen that the perfect recovery has been realized in all the experiments. PSNR values in the restored area are all 40.7 dB when regarding original image as reference.

Another experiment was conducted to test the performance of the proposed scheme when the ratio R is less than 1. In the experiment, the standard test gray scale image lake sized 512×512 (shown in Figure 5(a)) is still used as the host image. The host image is divided into blocks sized 8×8 pixels. The representative data of an image block is the quantified and encoded DCT coefficients. The quantization procedure is the same as that employed in [21]. The quantified DCT coefficients in each block are converted to binary sequences by the following allocation vector:

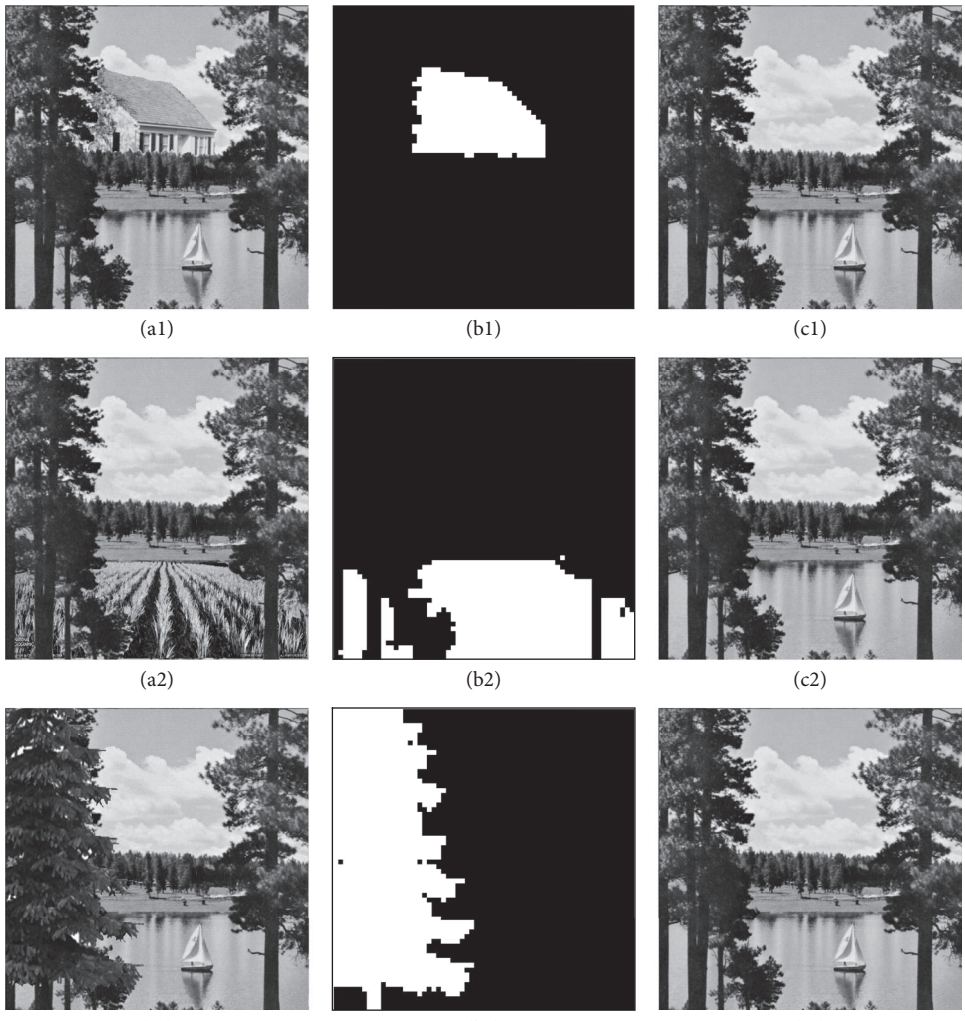
$$\{8, 7, 4, 3, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0\} \tag{15}$$

which results in a total of 56-bit representative data of an image block.

The redundant space of an image block is still all the 3 LSB of each pixel in the image block. So, the size of redundant space is still 192 bits. We will use 168 bits to store the reference data block and the remaining 24 bits to store the hash data. So the ratio R is 1/3. The coding matrix $\mathbf{A}$ will be 2^{12} rows and 3×2^{12} columns matrix and $[\mathbf{I} \mid \mathbf{A}]$ should be the generator matrix of a systematic MDS code. Just as in the first experiment, we generate the matrix A by constructing the 2^{12} rows and 3×2^{12} columns Cauchy matrix over $G(2^{14})$. The 56-bit representative data of each image block will be transformed into 4 elements in the finite field $G(2^{14})$. According to (4), the 3×2^{12} column vectors in the finite field $G(2^{14})$ are generated. For i ($i = 1, \ldots, 2^{12}$), we transform C_{ij} ($j = 1, 2, 3$) to bit strings and then connect the three bit

(a) (b)

Figure 2: (a) Original image lake. (b) Watermarked lake.



(a1) (b1) (c1)

(a2) (b2) (c2)

(a3) (b3) (c3)

Figure 3: Results of the meaningful tampering experiments. ((a1)–(a3)) Tampered lake with α = 9.8%, 21.83%, 32.69%. ((b1)–(b3)) Tampered blocks identification result of ((a1)–(a3)). ((c1)–(c3)) Restored version of ((a1)–(a3)).

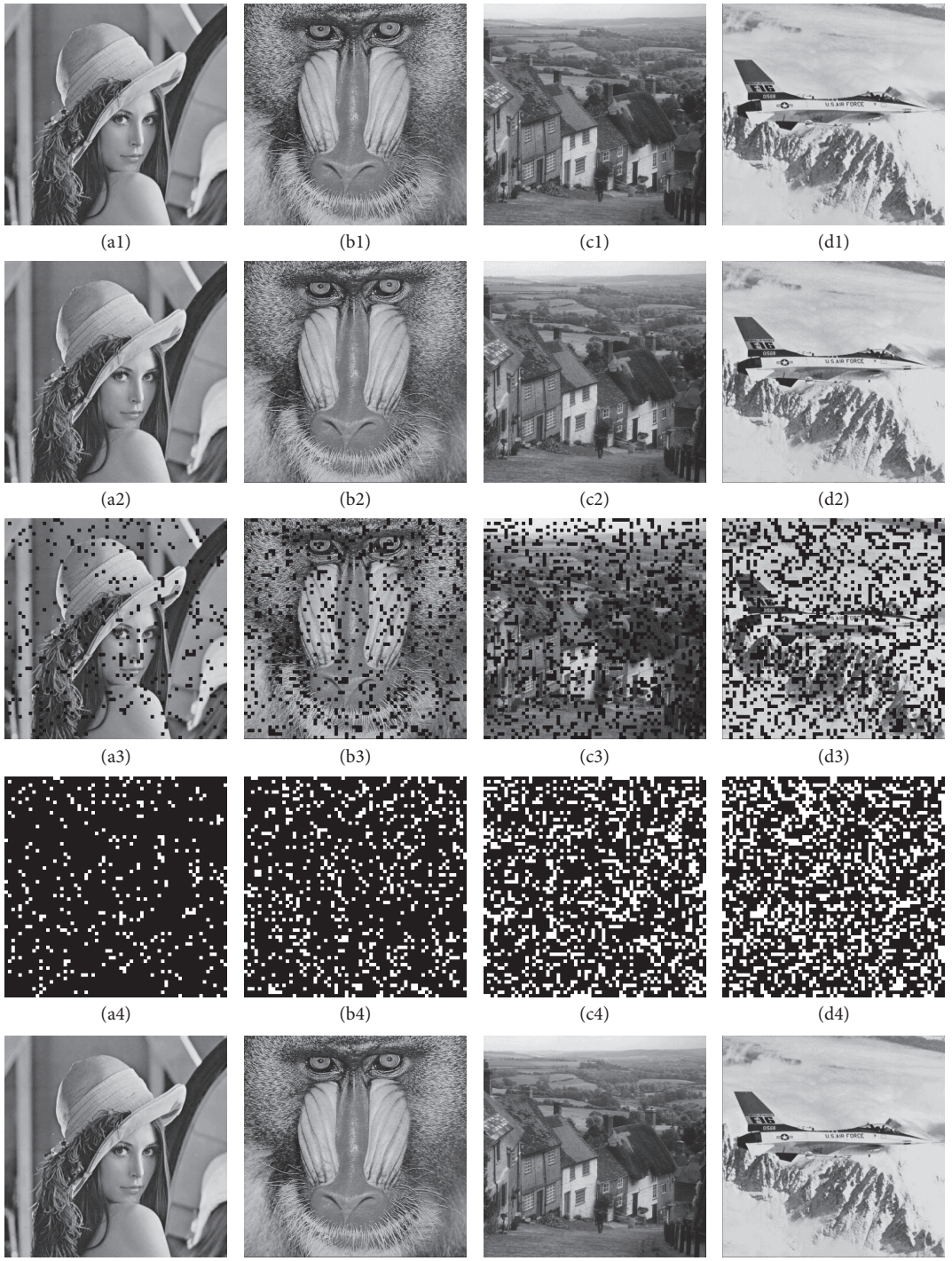


Figure 4: Results of the random tampering experiments. (a1)–(d1) are the original gray images; (a2)–(d2) are the watermarked images; (a3)–(d3) are the random tampered watermarked images with α = 10%, 18%, 24%, 33%; (a4)–(d4) are the tampered blocks identification results; (a5)–(d5) are the tampering restoration results.

strings to generate the 168 bits reference data R_i. The 24-bit hash data and the watermarked image (shown in Figure 5(b)) are generated as in the first experiment. The values of PSNR due to watermark embedding are 37.9 dB. According to (14), it can be calculated that the maximal tampering rate of the tested self-embedding scheme is 3/4.

Figure 6 shows three meaningful tampering experiments. The watermarked lake is maliciously tampered with tampering rate α = 32.69%, 54.35%, and 74.76%. The tampered watermarked images are shown in Figure 6((a1)–(a3)), and their corresponding identification and restoration results are shown in Figure 6((b1)–(b3)) and Figure 6((c1)–(c3)). Just

(a) (b)

Figure 5: (a) Original image. (b) Watermarked.

(a1) (b1) (c1)

(a2) (b2) (c2)

(a3) (b3) (c3)

Figure 6: Results of the meaningful tampering experiments. ((a1)–(a3)) Tampered lake with α = 32.69%, 54.37%, 74.76%. ((b1)–(b3)) Tampered blocks identification result of (a1)–(a3). ((c1)–(c3)) Restored version of (a1)–(a3).

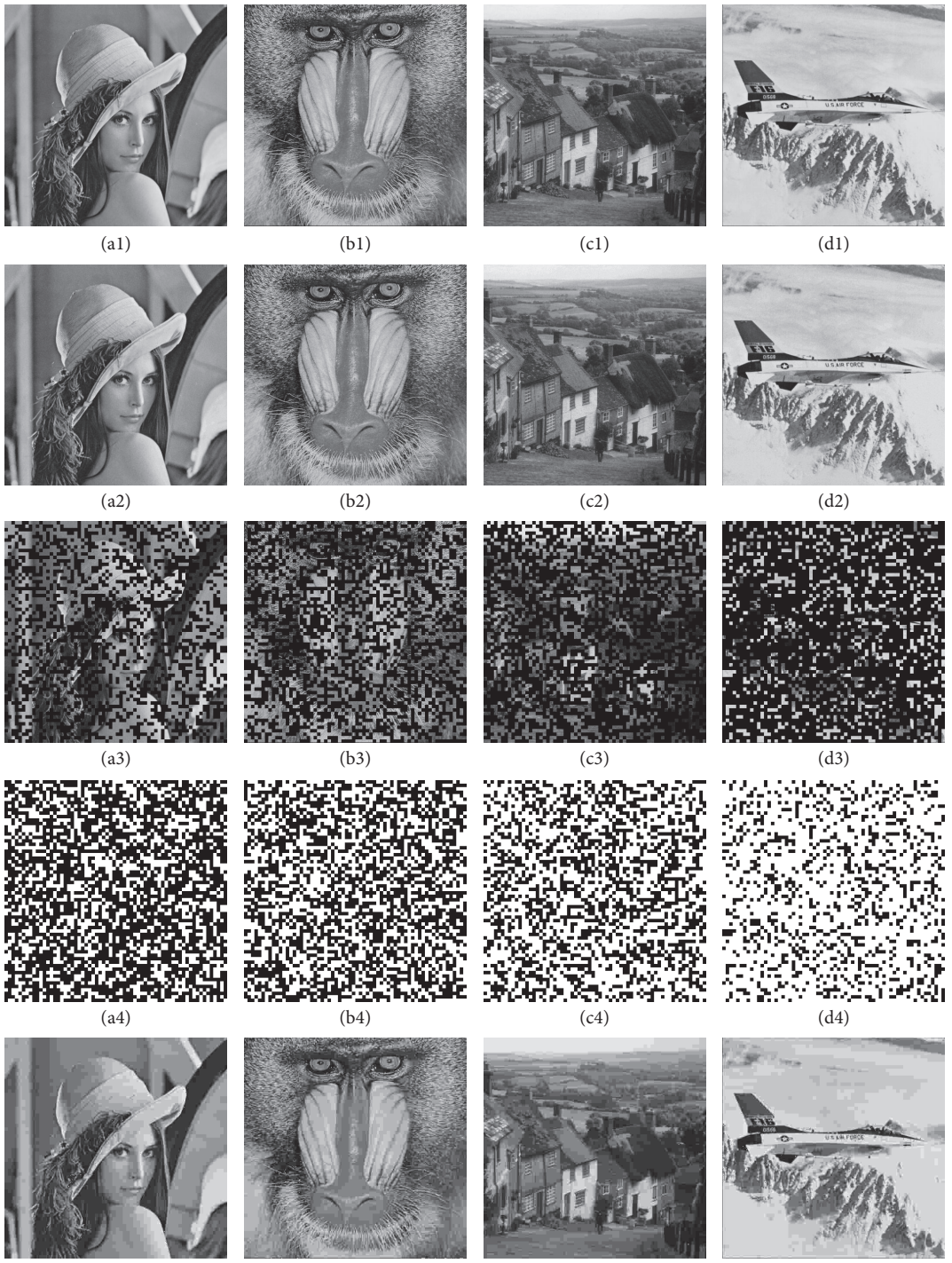


Figure 7: Results of the random tampering experiments. (a1)–(d1) are the original gray images; (a2)–(d2) are the watermarked images; (a3)–(d3) are the random tampered watermarked images with α = 40%, 49%, 58%, 75%; (a4)–(d4) are the tampered blocks identification results; (a5)–(d5) are the tampering restoration results.

as in the first experiment, the tampered blocks are located correctly. The quantified and encoded DCT coefficients of tampered blocks were recovered without any error. In the three cases, PSNR values in the restored area are all 25.2 dB when regarding original image as reference. The experiment demonstrates that if the ratio $R = 3$, the proposed scheme can perfectly recover the tampered image as long as the tampering rate $\alpha \leq 3/4$.

The random tampering experiments were also conducted (as seen in Figure 7). The gray scale images Lena, Baboon, Gold Hill, and Airplane in Figure 7((a1)–(d1)) are used as the host images. The watermarked images, shown in

Figure 7((a2)–(d2)), are generated as the lake. The values of PSNR due to watermark embedding are 37.9. The watermarked images are tampered randomly with tampering rate $\alpha = 40\%, 49\%, 58\%$, and 75%. The tampered images are shown in Figure 7((a3)–(d3)). The corresponding identification and restoration results are shown in Figure 7((a4)–(d4)) and Figure 7((a5)–(d5)). It can be seen that the recovery has been realized in all the experiments. PSNR values in the restored area are 27.8, 21.7, 26.9, and 26.9, respectively, when regarding original image as reference.

We compared the restoration capability of the proposed scheme with that of several other self-embedding watermark schemes. For the proposed scheme, the reference data embedded in one image block will be shared by all the image blocks. By using this global reference sharing mechanism and the special coding matrix, the problem of tampering coincidence is avoided absolutely. In the same experimental condition, the most extensive tampering area could be recovered and the recovery process is deterministic. Moreover, the quality of the restored content does not decrease as the percentage of tampering increases. However, the reference sharing mechanism had not been employed in the schemes [11, 20]. The reference data is embedded in another image block according to the block mapping. By this way the tampering coincidence cannot be avoided absolutely but only with the high probability. In the two schemes, if the tampering coincidence happened, the tampered blocks will be recovered with the neighborhood average. The maximal tampering rates are about 50% and 80%, respectively. But the data is obtained only by experiments, not by rigorous theoretical proof. Moreover, the PSNR of restored content in the two schemes decreases as the proportion of tampered area increases.

We also compare the restoration capability among different schemes based on the reference sharing mechanism. The PSNR between the original image and the watermarked image, the PSNR between the recovered image and the watermarked image or the original image, and the maximal tampering rate are considered. The experimental parameters of method 1 in [13] are the same as the proposed scheme when the ratio R is 2, which has been tested in the above experiment. All the two methods exploit 3 LSB watermark embedding. Therefore, the PSNR due to watermarking embedding is identical and equals 37.9 dB. The representative data blocks are all the 5 MSB of pixels in an image block and the length of the generated reference data blocks are all 160 bits. When the tampering rate is not larger than the maximal tampering rate, all the schemes can recover the representative data. PSNR values in restored area are identical and equal 40.7 dB when regarding original image as reference. But the maximal tampering rate of our proposed method is 33%, which is better than 24%, the maximal tampering rate of method 1 in [13]. The reason is the reference data embedded in one image block is shared by some image blocks but not all the image blocks. The local reference sharing method cannot get the maximum tamper ratio. The experimental parameters of the method in [21] are the same as the proposed scheme when the ratio R is 1/3, which has been tested in the above experiment. Both the proposed method and the method in [21] are based on the global reference sharing mechanism. They have the same restoration performance, while the encoding matrix applied to methods in [13, 21] is the random matrix. The random matrix can only promise the restoration can be successful with a great probability. In contrast the proposed method offers a deterministic self-embedding scheme by using the different encoding matrix.

6. Conclusion

In this paper, we proposed a self-embedding watermarking scheme based on the reference sharing mechanism. In the proposed scheme the special coding matrix is adopted and the global reference sharing mechanism is realized. The tampering coincidence and the reference waste are avoided. Based on our model, the maximal tampering rate can be derived in theory and considering the trade-off between the quality of recovered content and the restoration condition become more feasible. As long as the tampering rate is not larger than the maximal tampering rate, the representative data of the tampered image blocks can be recovered deterministically. The quality of the recovered content does not decrease as the proportion of tampered area increases. Moreover, the proposed scheme is reconfigurable. We experimentally evaluated the scheme in two sorts of configurations. Our experimental results demonstrate that the proposed method is effective.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper. The mentioned received funding in the "Acknowledgment" section did not lead to any conflict of interests regarding the publication of this manuscript.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) under Grant no. U1536110.

References

[1] Z. Zhou, Y. Wang, Q. M. J. Wu et al., "Effective and Efficient Global Context Verification for Image Copy Detection," *IEEE Transactions on Information Forensics Security*, vol. 12, no. 1, pp. 48–63, 2017.

[2] J. Wang, T. Li, Y. Shi, S. Lian, and J. Ye, "Forensics feature analysis in quaternion wavelet domain for distinguishing photographic images and computer graphics," *Multimedia Tools and Applications*, no. 76, pp. 23721–23737, 2017.

[3] J. Fridrich and M. Goljan, "Images with self-correcting capabilities," in *Proceedings of the International Conference on Image Processing (ICIP'99)*, pp. 792–796, October 1999.

[4] H. J. He, J. S. Zhang, and F. Chen, "Adjacent-block based statistical detection method for self-embedding watermarking techniques," *Signal Processing*, vol. 89, no. 8, pp. 1557–1566, 2009.

[5] X. Zhang, Z. Qian, Y. Ren, and G. Feng, "Watermarking with flexible self-recovery quality based on compressive sensing and

compositive reconstruction," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 4, pp. 1223–1232, 2011.

[6] T. Lee and S. D. Lin, "Dual watermark for image tamper detection and recovery," *Pattern Recognition*, vol. 41, no. 11, pp. 3497–3506, 2008.

[7] C.-W. Yang and J.-J. Shen, "Recover the tampered image based on VQ indexing," *Signal Processing*, vol. 90, no. 1, pp. 331–343, 2010.

[8] C. Li, Y. Wang, B. Ma, and Z. Zhang, "A novel self-recovery fragile watermarking scheme based on dual-redundant-ring structure," *Computers Electrical Engineering*, vol. 37, no. 6, pp. 927–940, 2011.

[9] C. Qin, C.-C. Chang, and T.-J. Hsu, "Effective fragile watermarking for image authentication with high-quality recovery capability," *KSII Transactions on Internet and Information Systems*, vol. 7, no. 11, pp. 2941–2956, 2013.

[10] C. Qin, C.-C. Chang, and K.-N. Chen, "Adaptive self-recovery for tampered images based on VQ indexing and inpainting," *Signal Processing*, vol. 93, no. 4, pp. 933–946, 2013.

[11] D. Singh and S. K. Singh, "DCT based efficient fragile watermarking scheme for image authentication and restoration," *Multimedia Tools and Applications*, vol. 76, pp. 1–25, 2015.

[12] X. Zhang, S. Wang, and G. Feng, "Fragile watermarking scheme with extensive content restoration capability," in *Proceedings of the International Workshop on Digital Watermarking*, pp. 268–278, 2009.

[13] X. Zhang, S. Wang, Z. Qian, and G. Feng, "Reference sharing mechanism for watermark self-embedding," *IEEE Transactions on Image Processing*, vol. 20, no. 2, pp. 485–495, 2011.

[14] X. Zhang and S. Wang, "Fragile watermarking scheme using a hierarchical mechanism," *Signal Processing*, vol. 89, no. 4, pp. 675–679, 2009.

[15] Z. Qian, G. Feng, X. Zhang, and S. Wang, "Image self-embedding with high-quality restoration capability," *Digital Signal Processing*, vol. 21, no. 2, pp. 278–286, 2011.

[16] X. Zhang and S. Wang, "Fragile watermarking with error-free restoration capability," *IEEE Transactions on Multimedia*, vol. 10, no. 8, pp. 1490–1499, 2008.

[17] C. Qin, H. Wang, X. Zhang, and X. Sun, "Self-embedding fragile watermarking based on reference-data interleaving and adaptive selection of embedding mode," *Information Sciences*, vol. 373, pp. 233–250, 2016.

[18] X. Zhang, Y. Xiao, and Z. Zhao, "Self-embedding fragile watermarking based on DCT and fast fractal coding," *Multimedia Tools and Applications*, vol. 74, no. 15, pp. 5767–5786, 2015.

[19] C. Qin, P. Ji, X. Zhang, J. Dong, and J. Wang, "Fragile image watermarking with pixel-wise recovery based on overlapping embedding strategy," *Signal Processing*, vol. 138, pp. 280–293, 2017.

[20] C. Qin, P. Ji, J. Wang, and C.-C. Chang, "Fragile image watermarking scheme based on VQ index sharing and self-embedding," *Multimedia Tools and Applications*, vol. 76, no. 2, pp. 2267–2287, 2017.

[21] P. Korus and A. Dziech, "Efficient method for content reconstruction with self-embedding," *IEEE Transactions on Image Processing*, vol. 22, no. 3, pp. 1134–1147, 2012.

[22] P. Korus and A. Dziech, "Adaptive self-embedding scheme with controlled reconstruction performance," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 2, pp. 1134–1147, 2014.

[23] W.-C. Wu and Z.-W. Lin, "SVD-based self-embedding image authentication scheme using quick response code features," *Journal of Visual Communication and Image Representation*, vol. 38, pp. 18–28, 2016.

[24] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error Correcting Codes*, North Holland Publishing Co., Amsterdam, Netherlands, 1977.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 1734961, 14 pages
https://doi.org/10.1155/2018/1734961

Research Article

Separable Reversible Data Hiding in Encrypted Images Based on Two-Dimensional Histogram Modification

Dawen Xu,[1] **Kai Chen**,[1] **Rangding Wang**,[2] **and Shubing Su**[1]

[1]*School of Electronics and Information Engineering, Ningbo University of Technology, Ningbo 315211, China*
[2]*CKC Software Lab, Ningbo University, Ningbo 315211, China*

Correspondence should be addressed to Dawen Xu; dawenxu@126.com

Received 29 September 2017; Revised 30 November 2017; Accepted 21 December 2017; Published 7 February 2018

Academic Editor: Xinpeng Zhang

An efficient method of completely separable reversible data hiding in encrypted images is proposed. The cover image is first partitioned into nonoverlapping blocks and specific encryption is applied to obtain the encrypted image. Then, image difference in the encrypted domain can be calculated based on the homomorphic property of the cryptosystem. The data hider, who does not know the original image content, may reversibly embed secret data into image difference based on two-dimensional difference histogram modification. Data extraction is completely separable from image decryption; that is, data extraction can be done either in the encrypted domain or in the decrypted domain, so that it can be applied to different application scenarios. In addition, data extraction and image recovery are free of any error. Experimental results demonstrate the feasibility and efficiency of the proposed scheme.

1. Introduction

With the rapid developments occurring in mobile internet and cloud storage, privacy and security of personal data have gained significant attention nowadays. There are no guarantees that stored data will not be accessed by unauthorized entities, such as the cloud provider itself or malicious attackers. Under these specific circumstances, sensitive images, such as medical and personal images, need to be encrypted before outsourcing for privacy-preserving purposes [1, 2]. In other words, the consumers would like to give the untrusted cloud server only an encrypted version of the data instead of the original content. The cloud service provider (who stores the data) is not authorized to access the original content (i.e., plaintext). However, in some application scenarios, the cloud servers or database managers need to embed some additional messages, such as authentication or notation data, directly into an encrypted data for tamper detection or ownership declaration purposes. For example, patient's information can be embedded into his/her encrypted medical image to avoid unwanted exposure of confidential information.

To address this problem, researchers have been studying the possibility of hiding data directly in the encrypted domain. Over the past few years, a considerable amount of schemes about data hiding in encrypted images or videos has been reported in the literature [3–10]. However, within these schemes, the host image/video is permanently distorted caused by data embedding. In general, the cloud service provider has no right to introduce permanent distortion. This implies that, for a legal receiver, the original plaintext content should be recovered without any error after image decryption and data extraction. To solve this problem, reversible data hiding (RDH) in the encrypted domain is preferred.

RDH is a technique that slightly alters digital media (e.g., images or videos) to embed secret data while the original digital media can be recovered without any error after the hidden messages have been extracted [11]. This specific data hiding technique has been found to be useful in some important and sensitive areas, that is, military communication, medical science, law-enforcement, and error concealment [12, 13], where the original media is required to be reconstructed without any distortion. So far, three major approaches, that is, lossless compression [14], histogram modification [11, 15], and difference expansion [16], have already been developed for RDH. For more details of these methods and other RDH methods, refer to the latest review of recent research [17].

Although RDH techniques have been studied extensively, these techniques are suitable for plaintext instead of ciphertext.

RDH in the encrypted domain has emerged as a new and challenging research field. In recent years, some RDH methods for encrypted images have been proposed. In general, these methods can be divided into three categories, that is, methods by vacating room after encryption (VRAE) [18–24], methods by reserving room before encryption (RRBE) [25–28], and methods based on homomorphic encryption [29–34]. In VRAE framework, the original signal is encrypted directly by the content owner, and the data hider embeds the additional bits by modifying some bits of the encrypted data. The advantage of this framework is that the operation of the end user is simple and efficient. However, as the entropy of an encrypted image has been maximized, the embedding capacity is limited. Moreover, the accuracy of data extraction and the quality of restored image are not satisfactory. In RRBE framework, the embedding room is created in the plaintext domain, that is, vacating room before encryption. The advantages of this framework are mainly reflected in two aspects; namely, embedding capacity is relatively large and pure reversibility is achieved. But this framework might be impractical because it requires the content owner to perform an extra preprocessing before content encryption [17]. In general, the content owner expects to send only an encrypted image to the manager without extra information. In addition to VRAE and RRBE, another type of method has recently been proposed by using homomorphic encryption. With the additive homomorphic property of Paillier cryptosystem, Chen et al. [29] firstly proposed a homomorphic encryption based RDH approach. Shiu et al. [31] improved Chen et al.'s method [29] by adopting the concept of difference expansion into homomorphic encryption. Moreover, RDH in the homomorphic encrypted domain has also been investigated in [32, 33]. However, the used public-key cryptosystems lead to data expansion after image encryption. In [30, 34], the additive homomorphic property of modulo operation is utilized to realize the RDH in the encrypted domain. The advantage is that encryption does not cause data expansion.

In this paper, we develop an effective and reliable framework for RDH in the encrypted domain. In fact, the proposed method belongs to the third category. Its main contribution is the combination of the modular addition and two-dimensional (2D) histogram modification. Its advantages are mainly manifested in four aspects. First of all, room for data hiding does not need to be vacated before encryption, which is more reasonable compared with the methods in [25–28]. Secondly, completely separable and completely reversible can be achieved, which is more reliable than the methods in [18–21]. Thirdly, the modular arithmetic addition operation, which has additive homomorphism, is utilized for image encryption. It does not cause data expansion, unlike the public-key cryptosystems in [29, 31–33]. Finally, since data embedding in encrypted domain is accomplished by using pairwise coefficient modification, embedded capacity has been greatly improved compared with the methods in [30, 34]. The rest of the paper is organized as follows. In Section 2, we describe the proposed scheme, which includes image encryption, data embedding in encrypted image, data extraction, and original image recovery. Experimental results and analysis are presented in Section 3. Finally, in Section 4, conclusions and future work are drawn.

2. Proposed Scheme

In this section, a RDH method in encrypted images is illustrated. It is composed of three parts, that is, generation of the encrypted image, generation of the marked encrypted image, data extraction, and image recovery. First, the content owner encrypts the original image with encryption key to produce an encrypted image. Then, the data hider without knowing the actual content of the original image can embed some additional data into the encrypted image. Here, the data hider can be a third party, for example, a database manager or a cloud provider, who is not authorized to access the original content of the signal (i.e., plaintext). At the receiving end, maybe the content owner himself or an authorized third party can extract the hidden data either in encrypted or decrypted image. For illustrative purposes, the framework of the proposed scheme is given in Figure 1.

2.1. Image Encryption. Assume the original image X is an 8-bit gray-scale image with size $M \times N$ and pixels $x(i, j) \in [0, 255]$, $0 \le i \le M-1$, $0 \le j \le N-1$. As we know, in the plaintext image, the correlation will gradually decrease with the increase of the distance between two pixels. In order to make good use of the correlation among pixels for RDH, the cover image is divided into a number of nonoverlapping blocks of size 3×3 as shown in Figure 2. If both M and N can be divisible by 3, the number of nonoverlapping blocks is $(M/3)\times(N/3)$. If M or N cannot be divisible by 3, the image is divided into $\lceil M/3 \rceil \times \lceil N/3 \rceil$ blocks, including $\lfloor M/3 \rfloor \times \lfloor N/3 \rfloor$ blocks of size 3×3. Here, $\lceil M/3 \rceil$ denotes the smallest integer greater than or equal to $M/3$, and $\lfloor M/3 \rfloor$ denotes the greatest integer less than or equal to $M/3$.

To ensure that pixels in the same block are encrypted with the same random value, the encryption matrix $R_a = \{r_a(i, j) \mid r_a(i, j) \in [0, 255]\}$ is obtained using the following equation:

$$r_a(i, j) = c\left(\left\lfloor \frac{i}{3} \right\rfloor, \left\lfloor \frac{j}{3} \right\rfloor\right), \tag{1}$$

where $C = \{c(p, q) | c(p, q) \in [0, 255], 0 \le p \le \lfloor M/3 \rfloor, 0 \le q \le \lfloor N/3 \rfloor\}$ is a pseudo-random matrix generated with the encryption key $\mathrm{En}_{\mathrm{key}}$. After getting the encryption matrix R, image encryption is done as follows.

$$S = E(X, R) = \left(x(i, j) + r_a(i, j)\right) \bmod 256 = s(i, j) \quad \forall i = 0, 1, \ldots, M-1, \; j = 0, 1, \ldots, N-1, \tag{2}$$

where S represents an encrypted image. The corresponding decryption can be done in the following manner:

$$X = D(S, R) = \left(s(i, j) - r_a(i, j)\right) \bmod 256 = x(i, j) \quad \forall i = 0, 1, \ldots, M-1, \; j = 0, 1, \ldots, N-1. \tag{3}$$

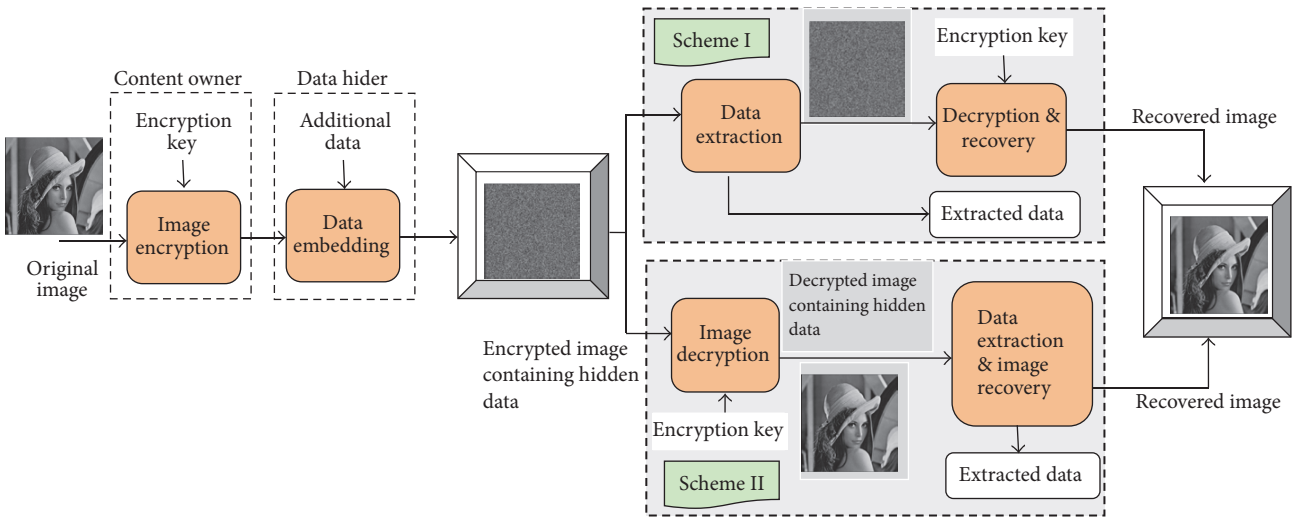


Figure 1: The framework of proposed scheme.

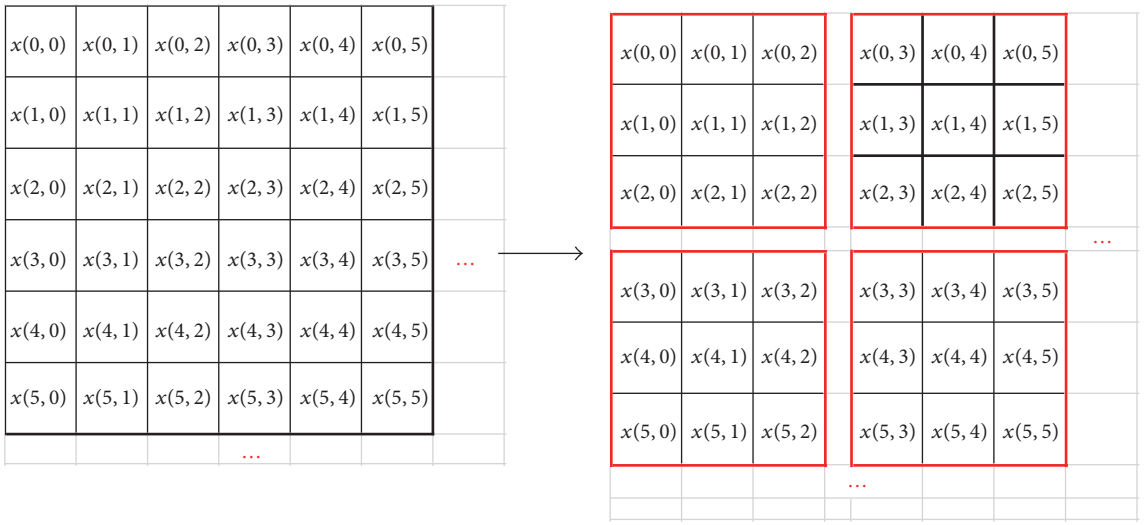


Figure 2: Example of image partition.

2.2. Data Embedding in Encrypted Image. After receiving the encrypted image, the data hider can embed some additional information into it for the purpose of media notation or integrity authentication. In order to achieve reversibility, the idea of histogram shifting is introduced in ciphertext based on homomorphic encryption. The whole process consists of two parts, namely, difference histogram generation and difference histogram modification.

(1) Difference Histogram Generation. Before performing the data embedding operation, a two-dimensional difference histogram of the encrypted image needs to be generated. The detailed procedure can be described as follows.

Step 1. Divide the encrypted image into nonoverlapping 3×3 blocks, which is the same as Figure 2. If the width or height of the image is not a multiple of 3, then the edge block will be ignored during the data embedding process.

Step 2. Calculate the difference between the basic pixel and the remaining pixels in each 3×3 block. Here, the pixel located in the center coordinate (m, n) is taken as the basic pixel for prediction. Then the difference can be calculated by using the following equation:

$$\begin{aligned} &f(m+u, n+v) \\ &\quad = (s(m+u, n+v) - s(m, n)) \bmod 256, \end{aligned} \tag{4}$$

where $u, v \in \{-1, 0, 1\}$. Note that the values of u and v cannot be zero at the same time. Obviously, eight differences can be obtained in each 3×3 block.

Although $s(m+u, n+v)$ is the encrypted value, it is easy to prove the following equation:

$$\begin{aligned} &(s(m+u, n+v) - s(m, n)) \bmod 256 \\ &\quad = (x(m+u, n+v) - x(m, n)) \bmod 256. \end{aligned} \tag{5}$$

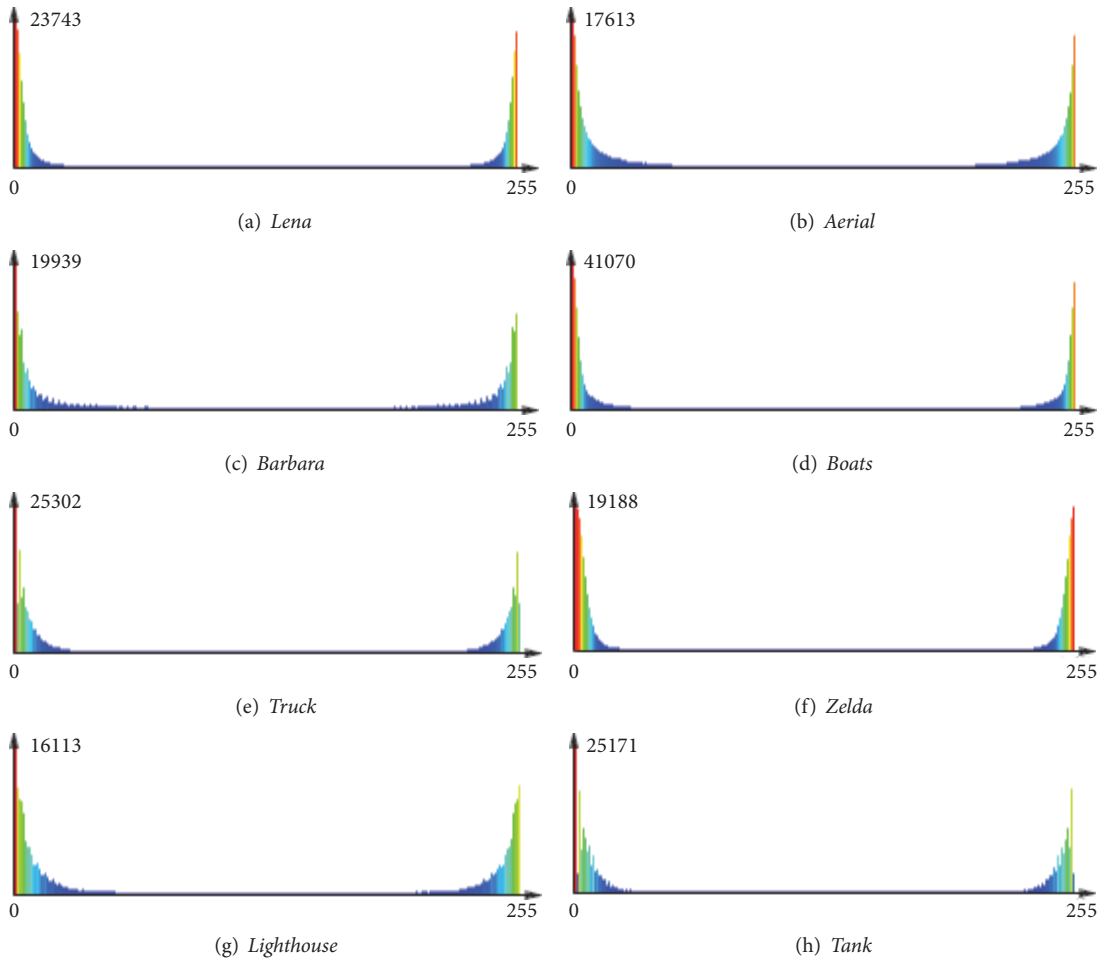


Figure 3: *1D* histogram of image difference.

Proof. One has

$$\begin{aligned}
&(s(m+u,n+v)-s(m,m)) \bmod 256\\
&\quad=\left(\left(x(m+u,n+v)+r_a(m+u,n+v)\right) \bmod 256\right.\\
&\quad\left.-\left(x(m,n)+r_a(m,n)\right) \bmod 256\right) \bmod 256\\
&\quad=\left(\left(x(m+u,n+v)+r_a(m+u,n+v)\right)\right.\\
&\quad\left.-\left(x(m,n)+r_a(m,n)\right)\right) \bmod 256,\\
&r_a(m+u,n+v)=r_a(m,n)\quad(\text{Refer to Eq. (1)}),\\
&\therefore (s(m+u,n+v)-s(m,n)) \bmod 256\\
&\quad=(x(m+u,n+v)-x(m,n)) \bmod 256.
\end{aligned} \tag{6}$$

□

According to the above proof, the correlation between the neighboring pixels in the local area of the plaintext image is preserved; that is, the difference remains unchanged even after encryption. All other 3 × 3 blocks can be processed in the same manner.

Step 3. Generate the difference histogram using differences in each 3 × 3 block. There is a high degree of correlation between adjacent pixels in a local region of an image. That is, they have similar gray values, or even the same gray value. Thus, the resulting difference histogram has a higher peak than the histogram of the original image. To demonstrate the distribution of the image difference, the histograms of some residual images are shown in Figure 3. It is clearly seen that the distribution is approximately symmetrical. The methods in [30, 34] mainly focus on exploiting one-dimensional (1D) coefficient histogram for RDH. The 1D coefficient histogram is usually defined as

$$h(r)=\#\left\{f_k(m+u,n+v)\mid f_k(m+u,n+v)=r\right\}, \tag{7}$$

where # denotes the cardinal number of a set, r is an integer, and k represents the block number. By considering every two differences together, the associated two-dimensional (2D) histogram can be defined as

$$\begin{aligned}
h(r_1,r_2)=\#\left\{\left(f_k\left(M_j\right),f_k\left(M_{j+1}\right)\right)\mid f_k\left(M_j\right)\right.\\
\left.=r_1,f_k\left(M_{j+1}\right)=r_2\right\},
\end{aligned} \tag{8}$$

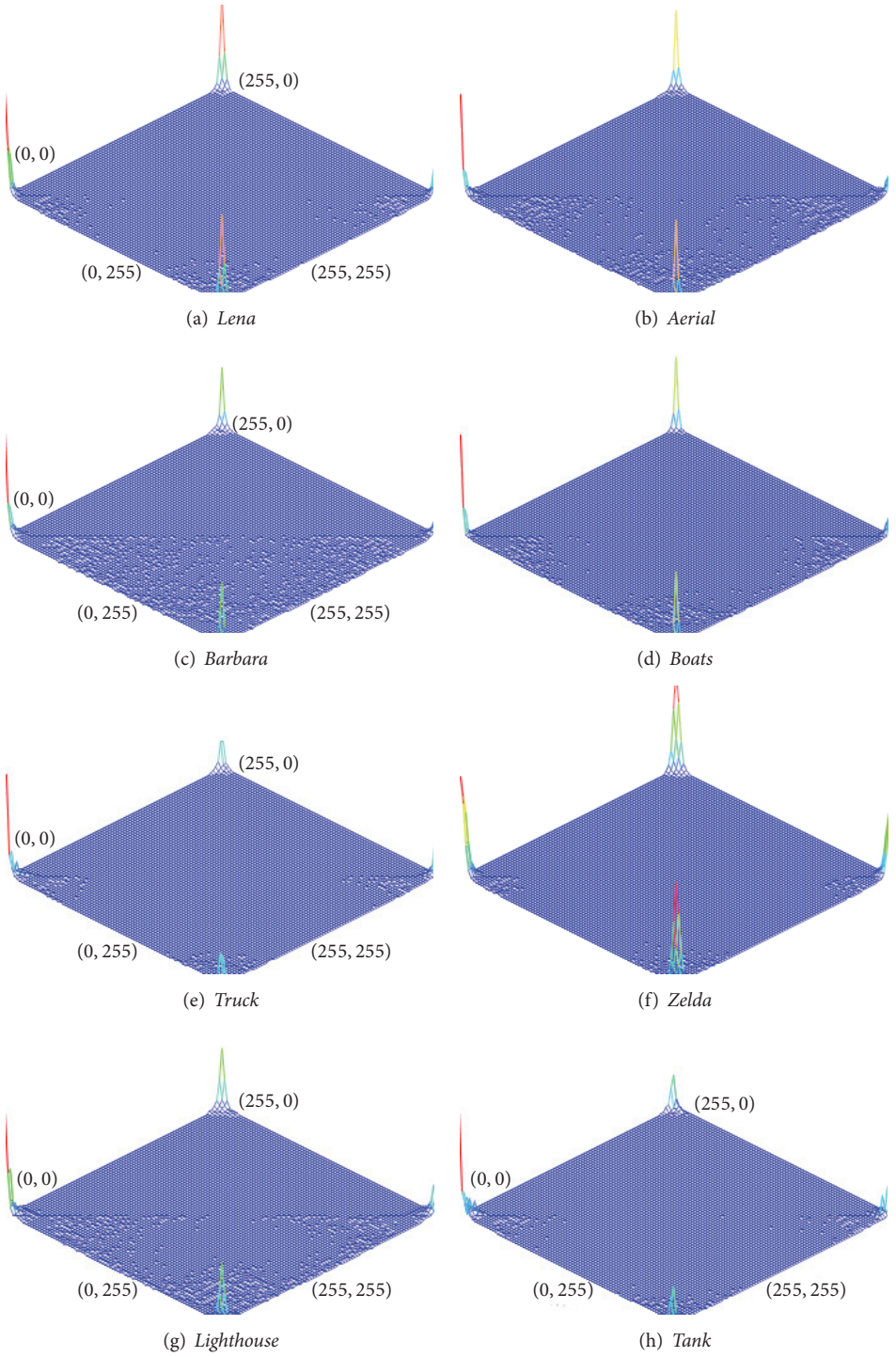


Figure 4: *2D* histogram of image difference.

where $f_k(M_j)$ denotes the jth difference in the kth 3×3 block. More specifically, $(f_k(M_1), f_k(M_2)) = (f_k(m-1, n-1), f_k(m, n-1))$, $(f_k(M_3), f_k(M_4)) = (f_k(m+1, n-1), f_k(m+1, n))$, $(f_k(M_5), f_k(M_6)) = (f_k(m-1, n), f_k(m-1, n+1))$, and $(f_k(M_7), f_k(M_8)) = (f_k(m, n+1), f_k(m+1, n+1))$. The distribution of the two-dimensional histogram is presented in Figure 4.

(2) Difference Histogram Modification. When the difference histogram is generated, reversible data hiding can be accomplished by using histogram shifting method. In [30], the conventional 1D histogram shifting technique is adopted. If the highest bin T_p is located in the left side of the difference histogram, for example, $T_p = 0$, the graphical representation of data embedding is shown in Figure 5(a).

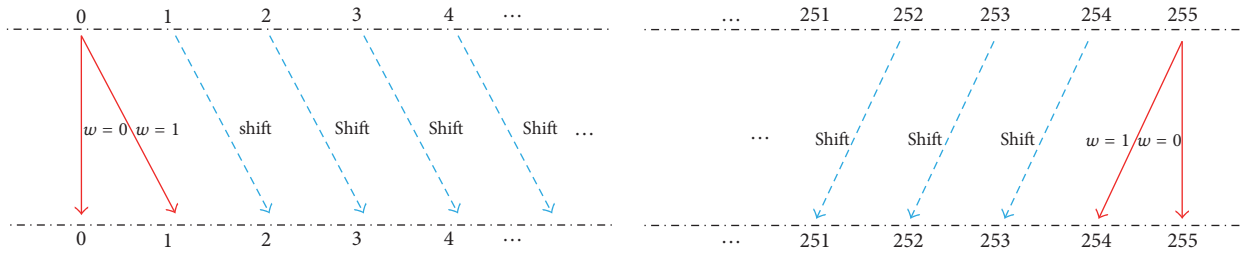


Figure 5: Illustration of the 1D histogram modification.

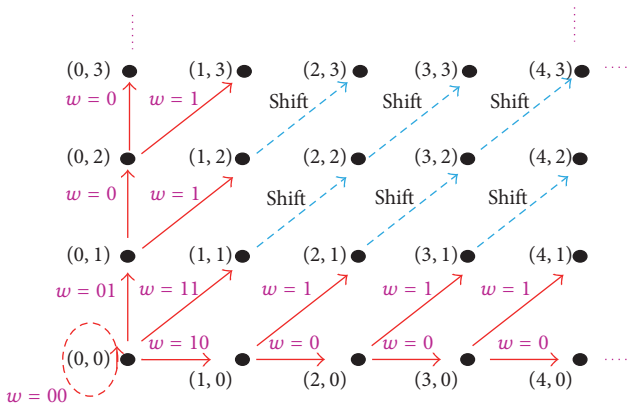


Figure 6: Illustration of the 2D histogram modification.

Otherwise, if the highest bin is located in the right side of the difference histogram, for example, $T_p = 255$, then its graphical representation is shown in Figure 5(b). Specifically, the conventional 1D RDH [30] can also be implemented in an equivalent way by modifying the 2D histogram [35].

For example, histogram modification in Figure 5(a) is in fact equivalent to the one shown in Figure 6. To further illustrate this case, some examples are provided below.

(i) For the coefficient pair $(f_k(M_j), f_k(M_{j+1})) = (0,0)$, in the method of 1D RDH shown in Figure 5(a), $f_k(M_j)$ is expanded to 0 or 1 for embedding a data bit $w \in \{0,1\}$, and $f_k(M_{j+1})$ is expanded similarly. Consequently, in the method of 2D RDH shown in Figure 6, the coefficient pair $(0,0)$ will be expanded to $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$ when the to-be-embedded bits are $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$, respectively.

(ii) For $(f_k(M_j), f_k(M_{j+1})) = (0,1)$, in the method of 1D RDH, $f_k(M_j)$ is expanded to 0 or 1 for embedding a data bit $w \in \{0,1\}$, and $f_k(M_{j+1})$ is shifted to 2. Correspondingly, in the method of 2D RDH, the pair $(0,1)$ is expanded to $(0,2)$ if $w = 0$, and $(1,2)$ if $w = 1$.

(iii) For $(f_k(M_j), f_k(M_{j+1})) = (2,1)$, in the method of 1D RDH, $f_k(M_j)$ and $f_k(M_{j+1})$ are shifted to 3 and 2, respectively. Accordingly, in the method of 2D RDH, the pair $(2,1)$ is shifted to $(3,2)$.

In particular, various histogram modification strategies can be designed based on 2D histogram. A reasonable histogram modification strategy directly contributes to the superior performance. The purpose of our design is to provide high embedding capacity while maintaining good visual quality. According to the statistical distribution of the difference histogram in Figure 4, we find that the probability of occurrence is larger when the difference is closer to 0 or 255. Based on this, a novel RDH technology is presented as shown in Figure 7.

Suppose the message to be embedded is a binary sequence denoted as $B = \{b(l) \mid l = 1,2,\ldots,K,\ b(l) \in \{0,1\}\}$. In order to enhance the security, a stream cipher is used to encrypt the message according to the data-hiding key $\mathrm{Dh}_{\mathrm{key}}$. Thus, the to-be-embedded binary information, that is, $W = \{w(l) \mid l = 1,2,\ldots,K, w(l) \in \{0,1\}\}$, is an encrypted version of B. It is difficult for anyone who does not retain the data hiding key to recover the message. The 2D histogram modification in the encrypted domain can be described as follows. According to the symmetry in Figure 4, only the modification in the lower-left quadrant is described for simplicity.

(1) If $(f_k(M_j), f_k(M_{j+1})) = (0,0)$, it has eight candidate directions for modification. In this case, three bits can be embedded. Specifically, the marked coefficient pair $(f'_k(M_j), f'_k(M_{j+1}))$ is determined as follows:

$$\left(f'_k\left(M_j\right), f'_k\left(M_{j+1}\right)\right) = \begin{cases} (0,0) & \text{if } w = '000' \\ (0,1) & \text{if } w = '001' \\ (1,0) & \text{if } w = '010' \\ (1,1) & \text{if } w = '011' \\ (0,2) & \text{if } w = '100' \\ (2,0) & \text{if } w = '101' \\ (1,2) & \text{if } w = '110' \\ (2,1) & \text{if } w = '111' \end{cases} \tag{9}$$

(2) If $(f_k(M_j), f_k(M_{j+1})) = (y,0)$, 1 bit can be embedded in each coefficient pair. Then, the marked coefficient pair $(f'_k(M_j), f'_k(M_{j+1}))$ is determined as follows

$$\left(f'_k\left(M_j\right), f'_k\left(M_{j+1}\right)\right) = \begin{cases} \left(f_k\left(M_j\right) + 2, f_k\left(M_{j+1}\right)\right) & \text{if } w = '0' \\ \left(f_k\left(M_j\right) + 2, f_k\left(M_{j+1}\right) + 1\right) & \text{if } w = '1', \end{cases} \tag{10}$$

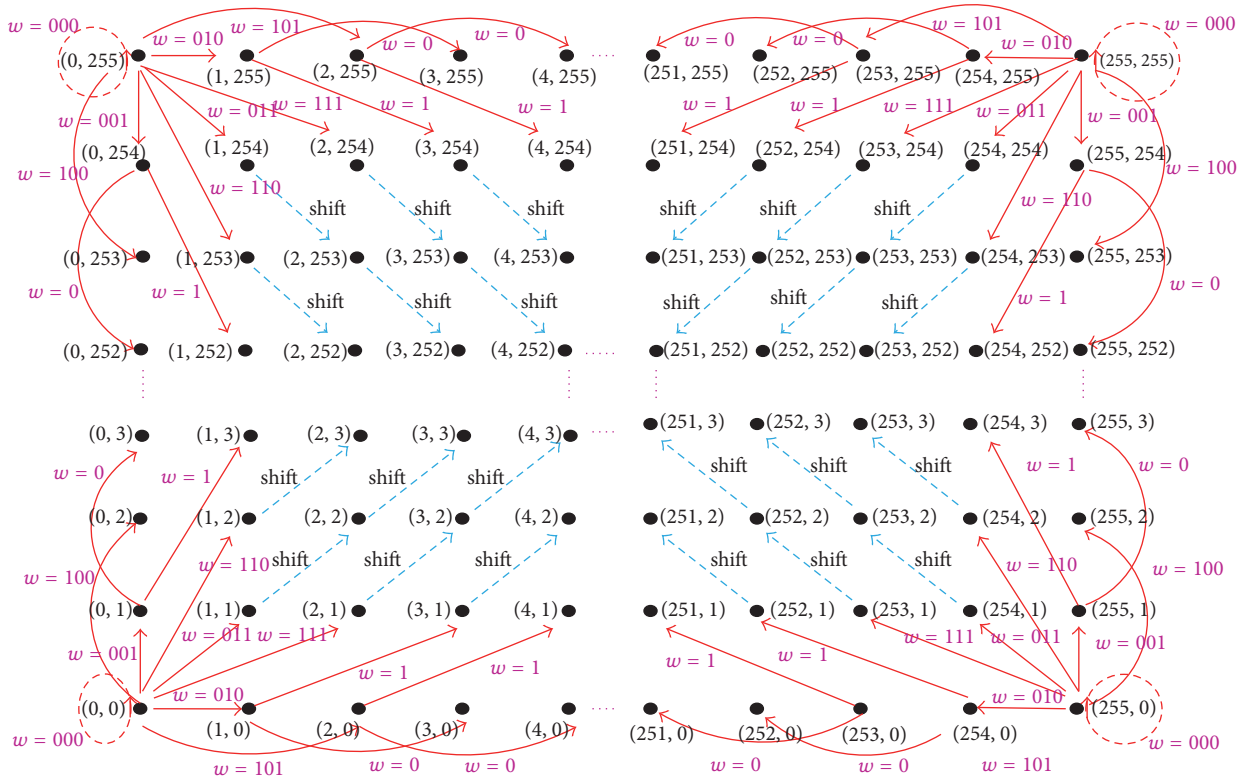


Figure 7: Illustration of the proposed 2D histogram modification.

where $1 \le y \le 125$. Although the middle part of the difference histogram is usually empty, ambiguities arise when the bins from two sides overlapped in the middle after expansion. To avoid it, the differences of 126 and 127 will not be expanded. However, ambiguities still arise when difference is changed from 125 to 127 or from 124 to 126 during the embedding process. The overlapping problem can be resolved by using a location map. It is a binary array with its every element corresponding to 126 and 127, 0 for genuine, and 1 for pseudo. The location map and the secret information will be embedded together in the encrypted domain.

(3) If $(f_k(M_j), f_k(M_{j+1})) = (0, z)$, the marked coefficient pair $(f_k'(M_j), f_k'(M_{j+1}))$ is determined as follows:

$$\left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = \begin{cases} \left(f_k\left(M_j\right), f_k\left(M_{j+1}\right) + 2\right) & \text{if } w = 0 \\ \left(f_k\left(M_j\right) + 1, f_k\left(M_{j+1}\right) + 2\right) & \text{if } w = 1, \end{cases} \tag{11}$$

where $1 \le z \le 125$.

(4) If $1 \le f_k(M_j) \le 126$ and $1 \le f_k(M_{j+1}) \le 126$, the coefficient pair $(f_k(M_j), f_k(M_{j+1}))$ is shifted to $(f_k'(M_j), f_k'(M_{j+1}))$ as follows:

$$\left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = \left(f_k\left(M_j\right) + 1, f_k\left(M_{j+1}\right) + 1\right). \tag{12}$$

According to the characteristic of modulus function, the following equation can be established.

$$\begin{aligned} & f\left(m+u, n+v\right) \pm 1 \\ & \quad = \left(s\left(m+u, n+v\right) - s\left(m, n\right) \bmod 256 \pm 1\right) \\ & \quad \cdot \bmod 256 \\ & \quad = \left(\left(s\left(m+u, n+v\right) \pm 1\right) \bmod 256 - s\left(m, n\right)\right) \\ & \quad \cdot \bmod 256. \end{aligned} \tag{13}$$

According to (13), the operation of $f(m+u, n+v) \pm 1$ can be accomplished by replacing $s(m+u, n+v)$ with $(s(m+u, n+v) \pm 1) \bmod 256$. Thus, in (10)~(12), the modification of the difference is equivalent to the modification of the pixel value. Then the marked and encrypted image $S' = \{s'(i, j) \mid s'(i, j) \in [0, 255]\}$ of the proposed scheme is obtained. The embedding capacity in the lower-left quadrant denoted as EC_2 can be computed by

$$EC_2 = 3 \cdot h\left(0, 0\right) + \sum_{1 \le y \le 125} h\left(y, 0\right) + \sum_{1 \le z \le 125} h\left(0, z\right). \tag{14}$$

2.3. Data Extraction and Original Image Recovery. In this scheme, data extraction and image decryption are completely separable. In other words, the hidden data can be extracted either in encrypted or in decrypted domain. Furthermore, our method is also reversible, where the hidden data could be removed to obtain the original image. We will first discuss the extraction in the encrypted domain followed by the decrypted domain.

(1) Scheme I: Data Extraction in the Encrypted Domain. In order to protect the users' privacy, the database manager (e.g., a cloud server) does not have sufficient permissions to access original video content due to the absence of encryption key. But the manager sometimes need to note and mark the personal information in corresponding encrypted images as well as verify their integrity. In this case, both data embedding and extraction should be manipulated in the encrypted domain. In the encrypted domain, the hidden data extraction can be accomplished by the following steps. According to the symmetry in Figure 4, only the extraction in the lower-left quadrant is described for simplicity.

Step 1. Divide the encrypted image into nonoverlapping 3×3 blocks, which is the same as Figure 2. The center pixel in each block is selected as the basic pixel for prediction.

Step 2. Calculate the difference between the basic pixel and the remaining pixels in each 3×3 block by using the following equation:

$$\begin{aligned} &f'(m+u, n+v) \\ &\quad = \left(s'(m+u, n+v) - s'(m,n)\right) \bmod 256. \end{aligned} \tag{15}$$

Step 3. The associated 2D histogram can be generated in the same way as in Section 2.2

$$\begin{aligned} h(r_1, r_2) = \#\left\{\left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) \mid f_k'\left(M_j\right)\right. \\ \left. = r_1, f_k'\left(M_{j+1}\right) = r_2\right\}. \end{aligned} \tag{16}$$

Step 4. According to the previous embedding rules, the hidden data can be extracted as

$$\widetilde{w} = \begin{cases} '000' & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (0,0) \\ '001' & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (0,1) \\ '010' & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (1,0) \\ '011' & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (1,1) \\ '100' & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (0,2) \\ '101' & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (2,0) \\ '110' & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (1,2) \\ '111' & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (2,1) \\ '0' & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (y,0),\ 2 < y \le 127 \\ '1' & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (y,1),\ 2 < y \le 127 \\ '0' & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (0,z),\ 2 < z \le 127 \\ '1' & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (1,z),\ 2 < z \le 127, \end{cases} \tag{17}$$

where $\widetilde{w}$ denotes the extracted message bits. Since the whole process is entirely operated in encrypted domain, it effectively avoids the leakage of original content.

Step 5. With the data-hiding key, $\mathrm{Dh}_{\mathrm{key}}$, the extracted hidden bits could be further decrypted to obtain the original message. In addition, the image difference value can be further restored as follows:

$$f_k\left(M_j\right), f_k\left(M_{j+1}\right) = \begin{cases} (0,0) & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) \in \{(1,0),(0,1),(1,1)\} \\ \left(f_k'\left(M_j\right) - 2, f_k'\left(M_{j+1}\right)\right) & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (y,0),\ 2 \le y \le 127 \\ \left(f_k'\left(M_j\right) - 2, f_k'\left(M_{j+1}\right) - 1\right) & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (y,1),\ 2 \le y \le 127 \\ \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right) - 2\right) & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (0,z),\ 2 \le z \le 127 \\ \left(f_k'\left(M_j\right) - 1, f_k'\left(M_{j+1}\right) - 2\right) & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (1,z),\ 2 \le z \le 127 \\ \left(f_k'\left(M_j\right) - 1, f_k'\left(M_{j+1}\right) - 1\right) & \text{if } 2 \le f_k'\left(M_j\right) \le 127,\ 2 \le f_k'\left(M_{j+1}\right) \le 127. \end{cases} \tag{18}$$

It should be noted that the boundary difference can be restored according to the location map. Similarly, according to (13), the operation of $f'(m+u, n+v) \pm 1$ can be accomplished by replacing $s'(m+u, n+v)$ with $(s'(m+u, n+v) \pm 1) \bmod 256$. Thus the encrypted image without the hidden data, that is, $S = \{s(i,j) \mid s(i,j) \in [0,255]\}$, is obtained.

Step 6. With the encryption key, $\mathrm{En}_{\mathrm{key}}$, the original cover image can be accurately restored by performing the decryption operation as in (3).

(2) Scheme II: Data Extraction in the Decrypted Domain. In scheme I, both data embedding and extraction are performed in the encrypted domain. However, in some cases, users want to decrypt the image first and then extract the hidden data from the decrypted image when it is needed. For example, with the encryption key, an authorized user wants to achieve the decrypted image containing the hidden data, which can be used to trace the source of the data. In this case, data extraction after image decryption is suitable. The whole process of decryption and data extraction comprised the following steps.

Step 1. Image decryption can be accomplished according to the following equation:

$$\begin{aligned} \widetilde{X} = \left(s'(i,j) - r_a(i,j)\right) \bmod 256 = \widetilde{x}(i,j) \\ \forall i = 0, 1, \ldots, M-1,\ \ j = 0, 1, \ldots, N-1. \end{aligned} \tag{19}$$

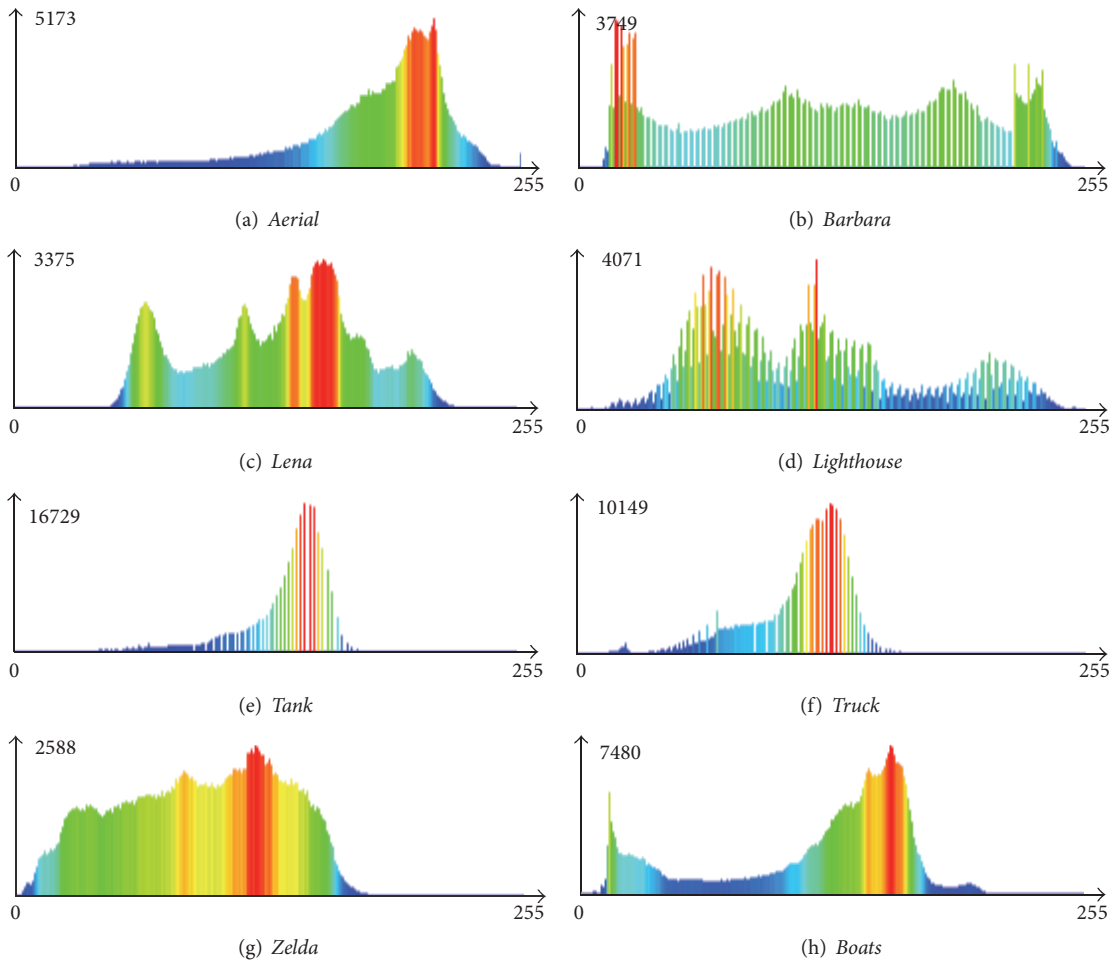


Figure 8: Histogram of the original image.

No visible distortions can be observed in the marked and decrypted images, as will be shown in later experimental results.

Step 2. Divide the marked and decrypted image $\widetilde{X} = \{\tilde{x}(i, j) \mid \tilde{x}(i, j) \in [0, 255]\}$ into nonoverlapping 3×3 blocks, which is the same as Figure 2. The center pixel in each block is selected as the basic pixel for prediction.

Step 3. Calculate the difference between the basic pixel and the remaining pixels to form the prediction error

$$f''(m+u, n+v) = (\tilde{x}(m+u, n+v) - \tilde{x}(m, n)) \bmod 256. \quad (20)$$

According to (5), the following equation is established:

$$f''(m+u, n+v) = f'(m+u, n+v). \quad (21)$$

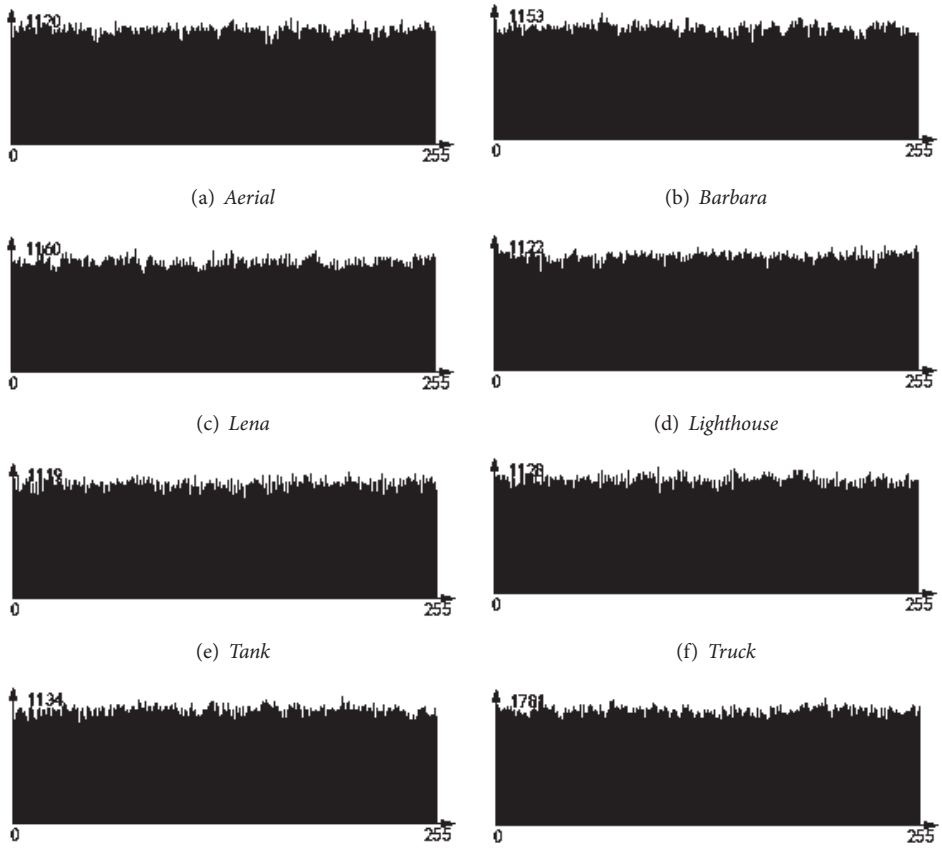
Step 4. The hidden data $\widetilde{w}$ can be extracted in a manner similar to (17). That is, it is only necessary to replace $(f'_k(M_j), f'_k(M_{j+1}))$ in (17) with $(f''_k(M_j), f''_k(M_{j+1}))$.

Step 5. The image difference can also be restored in the same manner as in (18). The only thing that needs to be adjusted is to replace $f'_k(M_j)$ and $f'_k(M_{j+1})$ with $f''_k(M_j)$ and $f''_k(M_{j+1})$, respectively. Similarly, the operation of $f''(m+u, n+v) \pm 1$ can be accomplished by replacing $\tilde{x}(m+u, n+v)$ with $(\tilde{x}(m+u, n+v) \pm 1) \bmod 256$. Therefore, the original image, that is, $X = \{x(i, j)\}$, is successfully restored.

3. Experimental Results and Analysis

Eight well-known standard gray images, that is, *Aerial*, *Barbara*, *Lena*, *Lighthouse*, *Tank*, *Truck*, *Zelda*, and *boats* [36], are considered for experimental purposes. The size of first 7 images is $512 \times 512 \times 8$, and the size of "Boats" is $720 \times 576 \times 8$. The secret data is a binary sequence created by pseudo-random number generator.

3.1. Scrambling Effect and Security Analysis. For an image encryption scheme, the security depends on cryptographic security and perceptual security. Cryptographic security denotes the security against cryptographic attacks, which relies on the underlying cipher. In the proposed scheme, pseudo-random sequence $r_a(i, j)$ is used to encrypt image. Figure 8 illustrates the histogram of the original image. After encryption, the corresponding histogram is shown in Figure 9. By comparing Figures 8 and 9, it can be observed



(a) *Aerial* (b) *Barbara*

(c) *Lena* (d) *Lighthouse*

(e) *Tank* (f) *Truck*

(g) *Zelda* (h) *Boats*

Figure 9: Histogram of the corresponding encrypted image.

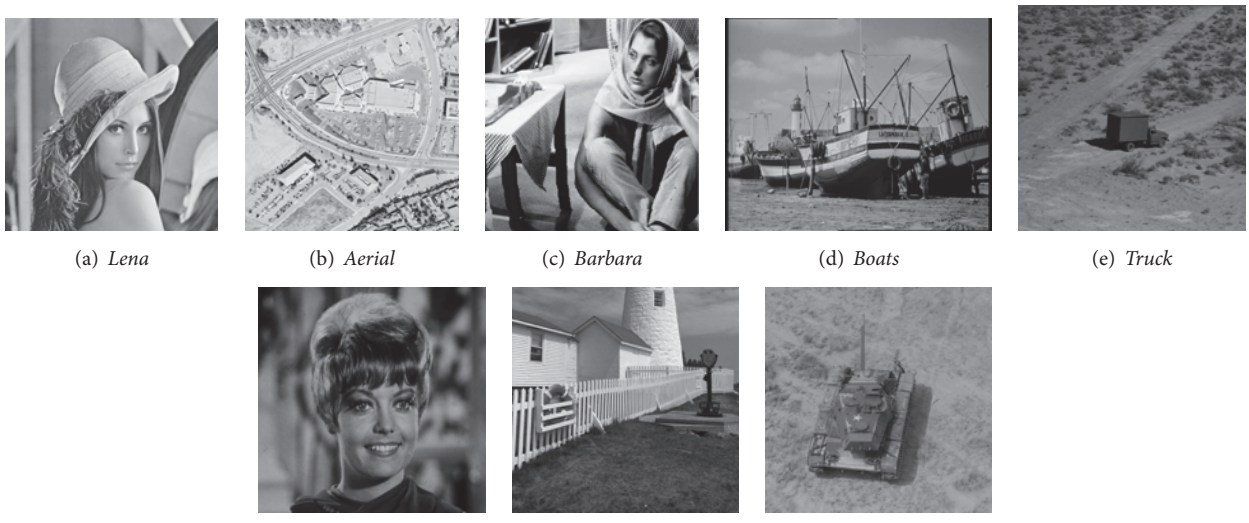
that the modified distribution appears to be uniform, which suggests that a statistical analysis would not be effective for evaluating the original content.

Perceptual security refers to the encrypted image being unintelligible. The original images are given in Figure 10, and their corresponding encrypted results are shown in Figure 11. As can be observed, the marked and encrypted image is a noise-like image. The visual information of the original image is damaged, which means that the data hider has extreme difficulty to obtain any useful information from it. In addition, for standard gray images, that is, *Aerial*, *Barbara*, *Lena*, *Lighthouse*, *Tank*, *Truck*, *Zelda*, and *boats*, PSNR (Peak Signal to Noise Ratio) values are 8.17 dB, 7.87 dB, 9.53 dB, 8.82 dB, 10.17 dB, 9.95 dB, 8.90 dB, and 9.11 dB, respectively. Obviously, scrambling performance of the described encryption system is more than adequate.

3.2. Visual Quality of Marked and Decrypted Image. Since the embedding scheme is reversible, the original cover content can be perfectly recovered after extracting the hidden data. In some scenarios, the encrypted image containing the hidden data provided by the server needs to be decrypted by the authorized user. Therefore, the visual quality of the decrypted image containing the hidden data is also expected to be equivalent or very close to that of the original image. In other words, the degradation of the image quality should be maintained at an acceptable range, even if the hidden data has not been removed. In the proposed method, since the maximum change in pixel value is 2, the artifacts introduced will not be perceptible. To verify this, a series of tests have been conducted. The original images and their corresponding decrypted versions containing the hidden data are shown in Figures 10 and 12, respectively. From our subjective examination, it is concluded that the marked content cannot be visually distinguished from nonmarked content. In addition to subjective observation, PSNR values are also given in Figure 8. In addition to *Zelda*, PSNR values of the remaining images are all above 47 dB. Generally, it is almost impossible to detect the degradation in image quality caused by data hiding.

3.3. Embedding Capacity. According to the embedding process described in Section 2.2, the embedded capacity can be calculated as follows:

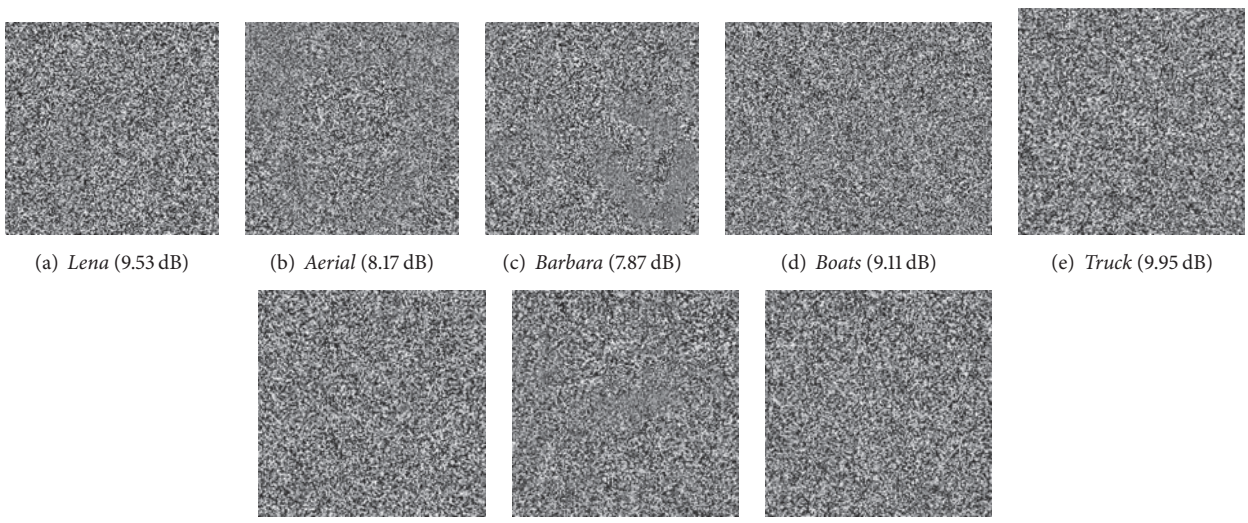
$$\begin{aligned} \mathrm{EC} &= 3 \\ &\quad \cdot \left(h\left(0,0\right) + h\left(0,255\right) + h\left(255,0\right) + h\left(255,255\right)\right) \end{aligned}$$



(a) *Lena* (b) *Aerial* (c) *Barbara* (d) *Boats* (e) *Truck*

(f) *Zelda* (g) *Lighthouse* (h) *Tank*

Figure 10: Original images.



(a) *Lena* (9.53 dB) (b) *Aerial* (8.17 dB) (c) *Barbara* (7.87 dB) (d) *Boats* (9.11 dB) (e) *Truck* (9.95 dB)

(f) *Zelda* (8.90 dB) (g) *Lighthouse* (8.82 dB) (h) *Tank* (10.17 dB)

Figure 11: The corresponding encrypted images.

$$+ \sum_{(1\le y\le 125)\cup(130\le y\le 254)} \left(h\left(y,0\right)+h\left(y,255\right)\right) + \sum_{(1\le z\le 125)\cup(130\le z\le 254)} \left(h\left(0,z\right)+h\left(255,z\right)\right). \tag{22}$$

For standard gray images, that is, *Aerial*, *Barbara*, *Lena*, *Lighthouse*, *Tank*, *Truck*, *Zelda*, and *boats*, the maximal embedding capacities of one-layer embedding strategy are 0.1432 bpp (bit per pixel), 0.1417 bpp, 0.1942 bpp, 0.1164 bpp, 0.1160 bpp, 0.1428 bpp, 0.1570 bpp, and 0.2136 bpp, respectively. It can be observed that the embedding capacity of the proposed scheme depends strongly on the characteristics of the original cover image. As expected, for images with high spatial activity (e.g., *Lighthouse*, *Tank*), low embedding rate is achieved. On the other hand, images with lower spatial activity (e.g., *Lena*, *Boats*) achieve higher embedding rate. The main reason is that most adjacent pixels have similar values in a smooth region. Therefore, they can contribute higher number of differences associated with the peak point compared with those in a complex region.

In our experiments, the size of the encrypted block is set to 3×3. In general, with the increase of the block size, the embedding capacity will increase whereas the security performance of the encryption algorithm will decrease. According to our analysis in Section 2.2, in any block, the difference of those pixel pairs remains unchanged even after encryption. With the increase of the block size, more correlation between the neighboring pixels may be preserved, and thus the

(a) *Lena* (47.29 dB) (b) *Aerial* (47.67 dB) (c) *Barbara* (47.69 dB) (d) *Boats* (47.27 dB) (e) *Truck* (47.60 dB)

(f) *Zelda* (43.98 dB) (g) *Lighthouse* (47.76 dB) (h) *Tank* (47.70 dB)

FIGURE 12: Decrypted images containing the hidden data.

embedding capacity will increase. On the other hand, the difference value between any pixel pair in each block of the plain image can be recovered in encrypted domain. In addition, higher capacities can be also achieved by applying multiple-layer embedding strategy. However, its cost is the decrease in perceptual quality.

3.4. Comparison and Discussion. As mentioned in Section 1, the methods in [18–21] may introduce some errors on data extraction and/or image recovery, while the complete reversibility can be achieved in the proposed method. More importantly, these methods are designed to carry only small payloads. Taking Zhang's method [18], for instance, the embedding rate is 0.0156 bpp associated with block size 8×8. If error correction mechanism is introduced, the actual embedding rate will be further decreased. It can be observed that our method achieves significantly higher embedding rate. For methods in [25–28], completely error-free data extraction and image recovery can be obtained. But it requires the content owner to perform an extra preprocessing before content encryption, which might be impractical. Instead, the proposed method overcomes these two problems.

Furthermore, Figure 13 shows the comparison of the embedding capacity between the proposed method and the methods in [30, 34]. Here, the maximum embedding capacity in one-layer embedding strategy is provided. As can be seen, in one-layer embedding strategy, the embedding capacity has been greatly improved. In fact, it can also be seen from Figures 6 and 7 that the embedding capacity of the proposed method can certainly be larger in one-layer embedding strategy. For example, when the coefficient pair is (0, 0), two bits can be embedded in the method of [30]. However, three bits can be embedded in the proposed method. The direct benefit is that a larger capacity can be achieved by one-layer histogram shifting. If two-layer embedding is used, the visual quality reduction is relatively large. Taking *Lena*

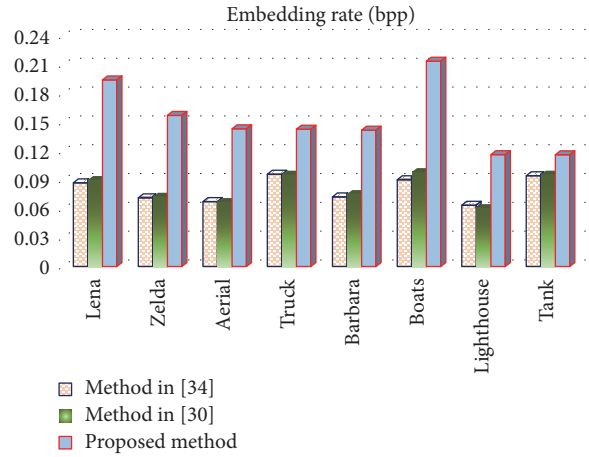
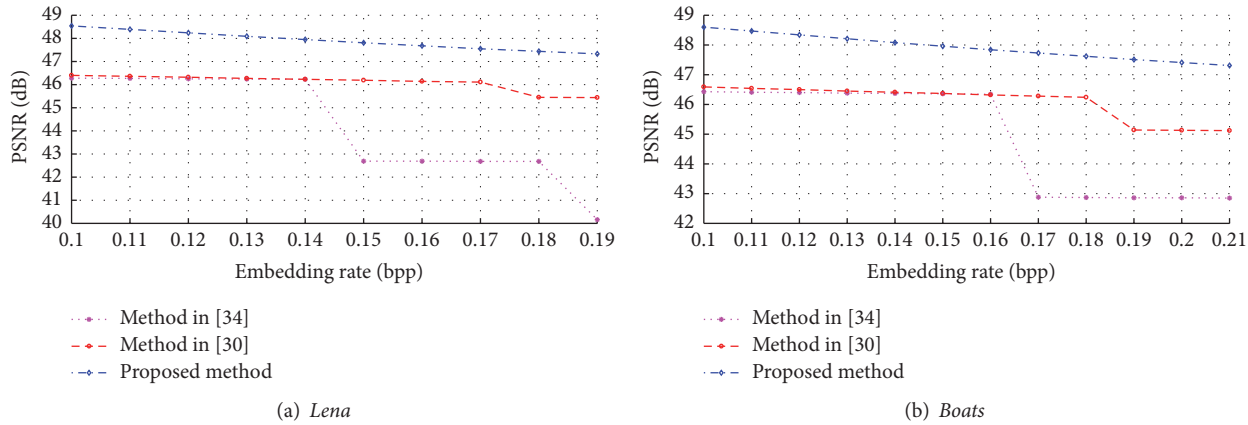


FIGURE 13: The comparison results of embedding capacity.

and *Boats* as an example, the performance comparison of different embedding rates is given in Figure 14. Obviously, the proposed method can provide better performance when the embedding capacity exceeds the maximum capacity of one-layer embedding strategy.

4. Conclusions and Future Work

In this paper, an algorithm to reversibly embed secret data in encrypted images is presented. A specific modulo operation is utilized to encrypt the image, which can preserve some correlation between the neighboring pixels. With the preserved correlation, the data hider can embed the secret data into the encrypted image by using 2D histogram modification, even though he does not know the original image content. Since the embedding process is done on encrypted data, our scheme preserves the confidentiality of content. Data



(a) *Lena* (b) *Boats*

Figure 14: The performance comparison of different embedding rates.

extraction is separable from image decryption; that is, the additional data can be extracted either in the encrypted domain or in the decrypted domain. Furthermore, this algorithm can achieve real reversibility and high quality of marked and decrypted images. One of the possible applications of this method is image annotation in cloud computing where high image quality and reversibility are greatly desired.

Although RDH technology and cryptography have been studied extensively, RDH in the encrypted domain is a highly interdisciplinary area of research. Technical research in this field has only just begun, and there is still an open space for research in this interdisciplinary research area. In future, more considerable effort is needed to determine the optimal modification on the histogram for achieving the best rate-distortion performance. Moreover, future work also aims at designing more efficient scheme for RDH in encrypted videos [37].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61771270, 61672302), Zhejiang Provincial Natural Science Foundation of China (LY17F020013, LZ15F020002), and Public Welfare Technology Application Research project of Zhejiang Province (2015C33237).

References

[1] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 340–352, 2016.

[2] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2594–2608, 2016.

[3] T. Bianchi and A. Piva, "Secure watermarking for multimedia content protection: a review of its benefits and open issues," *IEEE Signal Processing Magazine*, vol. 30, no. 2, pp. 87–96, 2013.

[4] B. Zhao, W. Kou, H. Li, L. Dang, and J. Zhang, "Effective watermarking scheme in the encrypted domain for buyer-seller watermarking protocol," *Information Sciences*, vol. 180, no. 23, pp. 4672–4684, 2010.

[5] J. Guo, P. Zheng, and J. Huang, "Secure watermarking scheme against watermark attacks in the encrypted domain," *Journal of Visual Communication and Image Representation*, vol. 30, pp. 125–135, 2015.

[6] A. V. Subramanyam, S. Emmanuel, and M. S. Kankanhalli, "Robust watermarking of compressed and encrypted JPEG2000 images," *IEEE Transactions on Multimedia*, vol. 14, no. 3, pp. 703–716, 2012.

[7] H. Liu, D. Xiao, R. Zhang, Y. Zhang, and S. Bai, "Robust and hierarchical watermarking of encrypted images based on Compressive Sensing," *Signal Processing: Image Communication*, vol. 45, pp. 41–51, 2016.

[8] D. Xu, R. Wang, and Y. Q. Shi, "Data hiding in encrypted H.264/AVC video streams by codeword substitution," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 4, pp. 596–606, 2014.

[9] D. Xu and R. Wang, "Context adaptive binary arithmetic coding-based data hiding in partially encrypted H.264/AVC videos," *Journal of Electronic Imaging*, vol. 24, no. 3, Article ID 033028, 2015.

[10] D. Xu, R. Wang, and Y. Q. Shi, "An improved scheme for data hiding in encrypted H.264/AVC videos," *Journal of Visual Communication and Image Representation*, 2015.

[11] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354–361, 2006.

[12] D. Xu, R. Wang, and Y. Q. Shi, "An improved reversible data hiding-based approach for intra-frame error concealment in H.264/AVC," *Journal of Visual Communication and Image Representation*, vol. 25, no. 2, pp. 410–422, 2014.

[13] D. Xu and R. Wang, "Two-dimensional reversible data hiding-based approach for intra-frame error concealment in H.264/AVC," *Signal Processing: Image Communication*, vol. 47, pp. 369–379, 2016.

[14] J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding-new paradigm in digital watermarking," *EURASIP Journal on Applied Signal Processing*, vol. 2, pp. 185–196, 2002.

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

[16] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.

[17] Y.-Q. Shi, X. Li, X. Zhang, H.-T. Wu, and B. Ma, "Reversible data hiding: Advances in the past two decades," *IEEE Access*, vol. 4, pp. 3210–3237, 2016.

[18] X. P. Zhang, "Reversible data hiding in encrypted image," *IEEE Signal Processing Letters*, vol. 18, no. 4, pp. 255–258, 2011.

[19] W. Hong, T.-S. Chen, and H.-Y. Wu, "An improved reversible data hiding in encrypted images using side match," *IEEE Signal Processing Letters*, vol. 19, no. 4, pp. 199–202, 2012.

[20] C. Qin and X. Zhang, "Effective reversible data hiding in encrypted image with privacy protection for image content," *Journal of Visual Communication and Image Representation*, vol. 31, pp. 154–164, 2015.

[21] X. P. Zhang, "Separable reversible data hiding in encrypted image," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 826–832, 2012.

[22] Z. Qian and X. Zhang, "Reversible Data Hiding in Encrypted Images with Distributed Source Encoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 4, pp. 636–646, 2016.

[23] J. Zhou, W. Sun, L. Dong, X. Liu, O. C. Au, and Y. Y. Tang, "Secure reversible image data hiding over encrypted domain via key modulation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 3, pp. 441–452, 2016.

[24] F. Huang, J. Huang, and Y.-Q. Shi, "New framework for reversible data hiding in encrypted domain," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2777–2789, 2016.

[25] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, "Reversible data hiding in encrypted images by reserving room before encryption," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 3, pp. 553–562, 2013.

[26] W. M. Zhang, K. D. Ma, and N. H. Yu, "Reversibility improved data hiding in encrypted images," *Signal Processing*, vol. 94, no. 1, pp. 118–127, 2014.

[27] X. Cao, L. Du, X. Wei, D. Meng, and X. Guo, "High Capacity Reversible Data Hiding in Encrypted Images by Patch-Level Sparse Representation," *IEEE Transactions on Cybernetics*, vol. 46, no. 5, pp. 1132–1143, 2016.

[28] D. Xu and R. Wang, "Separable and error-free reversible data hiding in encrypted images," *Signal Processing*, vol. 123, pp. 9–21, 2016.

[29] Y.-C. Chen, C.-W. Shiu, and G. Horng, "Encrypted signal-based reversible data hiding with public key cryptosystem," *Journal of Visual Communication and Image Representation*, vol. 25, no. 5, pp. 1164–1170, 2014.

[30] D. W. Xu, K. Chen, R. D. Wang, and S. B. Su, "Completely separable reversible data hiding in encrypted images," in *International Workshop on Digital-forensics and Watermarking (IWDW 2015), Tokyo, Japan*, vol. 9569 of *LNCS*, pp. 365–377, 2016.

[31] C.-W. Shiu, Y.-C. Chen, and W. Hong, "Encrypted image-based reversible data hiding with public key cryptography from difference expansion," *Signal Processing: Image Communication*, vol. 39, pp. 226–233, 2015.

[32] X. Zhang, J. Long, Z. Wang, and H. Cheng, "Lossless and Reversible Data Hiding in Encrypted Images with Public-Key Cryptography," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 9, pp. 1622–1631, 2016.

[33] H.-T. Wu, Y.-M. Cheung, and J. Huang, "Reversible data hiding in Paillier cryptosystem," *Journal of Visual Communication and Image Representation*, vol. 40, pp. 765–771, 2016.

[34] M. Li, D. Xiao, Y. Zhang, and H. Nan, "Reversible data hiding in encrypted images using cross division and additive homomorphism," *Signal Processing: Image Communication*, vol. 39, pp. 234–248, 2015.

[35] B. Ou, X. Li, Y. Zhao, R. Ni, and Y.-Q. Shi, "Pairwise prediction-error expansion for efficient reversible data hiding," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 5010–5021, 2013.

[36] Test Images (Online), Available: http://www.hlevkin.com/TestImages/.

[37] D. Xu and R. Wang, "Efficient reversible data hiding in encrypted H.264/AVC videos," *Journal of Electronic Imaging*, vol. 23, no. 5, Article ID 053022, 2014.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 1301290, 11 pages
https://doi.org/10.1155/2018/1301290

Research Article

Nonoverlapping Blocks Based Copy-Move Forgery Detection

Yu Sun, Rongrong Ni, and Yao Zhao

Institute of Information Science and Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing Jiaotong University, Beijing 100044, China

Correspondence should be addressed to Rongrong Ni; rrni@bjtu.edu.cn

Received 29 September 2017; Revised 12 December 2017; Accepted 18 December 2017; Published 22 January 2018

Academic Editor: Zhenxing Qian

In order to solve the problem of high computational complexity in block-based methods for copy-move forgery detection, we divide image into texture part and smooth part to deal with them separately. Keypoints are extracted and matched in texture regions. Instead of using all the overlapping blocks, we use nonoverlapping blocks as candidates in smooth regions. Clustering blocks with similar color into a group can be regarded as a preprocessing operation. To avoid mismatching due to misalignment, we update candidate blocks by registration before projecting them into hash space. In this way, we can reduce computational complexity and improve the accuracy of matching at the same time. Experimental results show that the proposed method achieves better performance via comparing with the state-of-the-art copy-move forgery detection algorithms and exhibits robustness against JPEG compression, rotation, and scaling.

1. Introduction

With the development of computer technique, there are more and more image editing tools, such as Photoshop and Fireworks. As a result, it becomes much easier for people even nonprofessionals to do some operations on digital images. However, it also brings convenience for people to tamper maliciously. Once these tampered images are applied to court forensics, newspaper, or academic research, the social credibility crisis will be aroused. Therefore, the image forgery detection becomes necessary. There are many types of tampering operations, and copy-move is one of the most common operations among them. The copy-move forgery is to copy a region and paste it into another place in the same image.

In the last few years, a large number of methods have been proposed to detect copy-move forgery that are mainly concentrated on two categories. One is based on block feature, and the other is based on keypoint feature. The block-based methods usually first divide the image into overlapping blocks, then extract each block's feature, and finally find the duplicate regions after matching them. Fridrich et al. [1] first proposed the copy-move forgery detection (CMFD) method by using the coefficients of Discrete Cosine Transform (DCT), and they applied dictionary sorting to the matching process. However, the computing complexity of this algorithm is very high. As a result, many dimensionality reduction-based algorithms have been proposed. Popescu and Farid [2] reduced the feature dimensions via principal component analysis (PCA). Bashar et al. [3] improved the performance further by Kernel-PCA (KPCA). Li et al. [4] combined discrete wavelet transform (DWT) and Singular Values Decomposition (SVD) to extract block feature. Kang and Wei [5] extracted the singular values of a reduced-rank approximation (SVD) as the block feature. Bayram et al. [6] computed the Fourier-Mellin Transform (FMT) for each block, and this method had high robustness in rotation and scaling. Mahdian and Saic [7] proposed a method based on blur moment invariants (Blur). Wang et al. [8] applied Hu moments (Hu) to each block to extract features. Ryu et al. [9, 10] used Zernike moments (Zernike) as block feature, which was only robust in rotation. Luo et al. [11] used the average of red, green, and blue components, respectively, and computed directional information for each block. Wang et al. [12] used the mean intensities of circles with different radii around the block center to obtain the robust feature. Lin et al. [13] got a 9-dimensional feature vector via calculating the gray average intensities of each block and its subblocks. Bravo-Solorio and Nandi [14] extracted the entropy as block feature. Though the block-based method can locate the tampered

region in pixel level accurately, the performance may decrease a lot when the suspicious image was attacked by some operations, such as noise, JPEG compression, and scaling. In addition, it is hard for these algorithms to estimate an accurate geometrical transform. Moreover, the computing complexity is very high in the matching process due to a large number of overlapping blocks, and the computation time would increase rapidly as the image size becomes larger.

The keypoint-based methods rely on the extraction of keypoints in the images. Huang et al. [15] proposed an algorithm based on Scale Invariant Feature Transform (SIFT), which is robust and sensitive to postimage processing. Amerini et al. [16] matched the SIFT descriptors via g2NN, resulting in managing multiple regions matching successfully. Speeded-Up Robust Feature (SURF) [17, 18] is used to improve matching efficiency, and the length of this feature is half of SIFT. SIFT and SURF are the most widely used keypoints for CMFD. Some other local features are also proposed to detect duplicate regions, like Local Binary Pattern (LBP) [19], Binary Robust Invariant Scalable Keypoints (BRISK) [20], and DAISY [21]. Though the computing complexity of these algorithms in matching process is much lower, they also have a main drawback: the performance will be very poor when the copy-move regions are smooth.

To overcome the shortcomings in block-based and keypoint-based approaches, a novel matching strategy is proposed to detect the copy-move forgery in digital images. We divide image into nonoverlapping blocks; then the image is segmented into smooth part and texture part according to the distribution of keypoints. In texture part, we use the extracted keypoints to find duplicate regions, while in smooth part, in order to reduce the computational complexity, blocks with similar color are clustered into a group, and searching duplicated blocks will also be carried out in the same group. Because we use nonoverlapping blocks, we cannot find the two same blocks in the content, which results in mismatching. So we need to update the candidate block by registration if the query block and the candidate block are partly consistent in content before projecting them to hash space. Finally, LSH algorithm is used to group the updated candidate blocks into several hash buckets. The strategy based on nonoverlapping can significantly reduce computational complexity, while maintaining the high accuracy levels for CMFD. The rest of the paper is organized as follows. In Section 2, we show the framework in our proposed method. The experimental results are presented in Section 3, and Section 4 draws conclusions.

2. Nonoverlapping Blocks Based CMFD Approach

The framework of our method is shown in Figure 1, which consists of three key steps: image division, feature matching, and postprocessing. In the first step, an image is divided into two parts: the yellow part is the texture region, while the blue part is the smooth region. In the second step, two types of features for the two parts are extracted to obtain the matched parts. In the last step, falsely matched pairs are removed; then we exploit morphology operations to generate a forgery detection map.

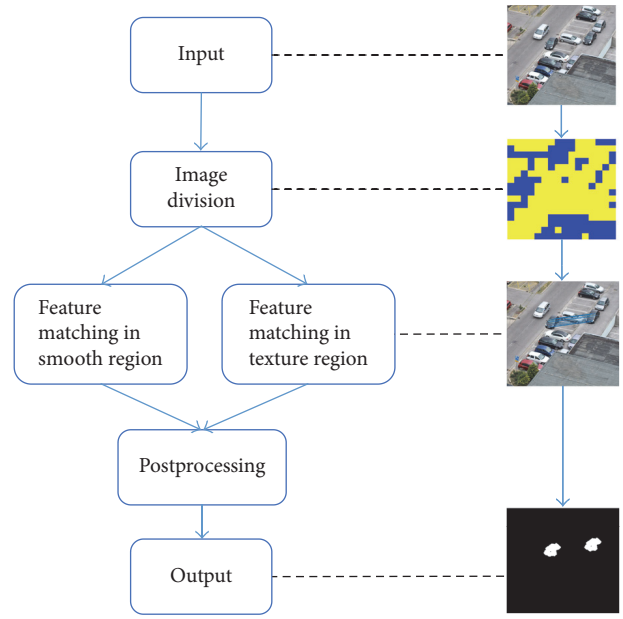


Figure 1: The framework of the proposed method.

2.1. Image Division. In this section, we will introduce the image division. When we get a suspicious image, we first divide the image into nonoverlapping blocks; then we extract the keypoints in each of them. There are many kinds of keypoints, and we use SIFT in this paper. The reason why we choose SIFT keypoint will be explained in Section 2.2. Finally, we count the number of keypoints N_i in each block. As we all know, the identification and selection of the keypoints rely on the high-entropy regions. As a result, most keypoints can only exist in texture regions. Consequently, we select a threshold τ_1 to decide whether the current block is a smooth block or a texture block.

$$\begin{aligned} N_i \geqslant \tau_1 \quad & \text{block}^i \in \text{texture block} \\ N_i < \tau_1 \quad & \text{block}^i \in \text{smooth block}. \end{aligned} \tag{1}$$

2.2. Feature Matching. According to the paper [23], we know that SIFT feature and Zernike moments are recommended for their excellent performance among the keypoint-based and block-based features, respectively. In our implementation, we employ vlFeat software to help us to detect and describe the keypoints. Once the parameters are set, the distribution and quantity of the keypoints are fixed. SIFT feature can detect forged regions even after some attacks but fail to detect flat area. Zernike moments are invariant to rotation and show effective for flat area. Therefore, in our implementation we choose SIFT and Zernike moments as features for detection in texture blocks and smooth blocks, respectively.

2.2.1. Feature Matching in Texture Region. After image division, we obtain a set of keypoints $X = \{x_1, x_2, \ldots, x_n\}$

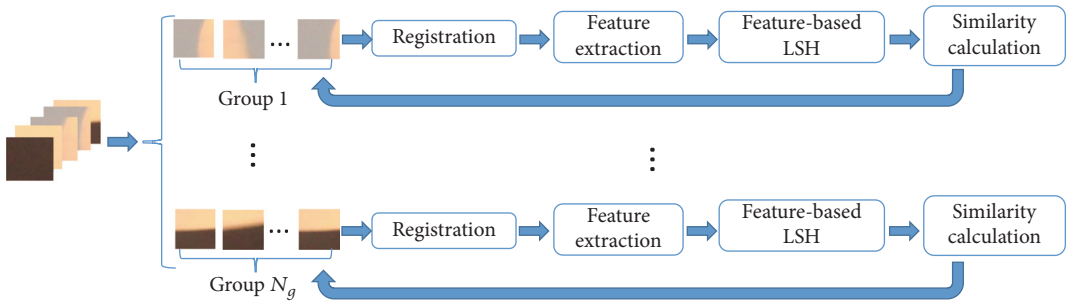


Figure 2: The framework of feature matching in smooth region.

with their corresponding descriptors $S = \{s_1, s_2, \ldots, s_n\}$. We use g2NN matching in this step, because this method can well solve multiple regions matching problem. The candidate matches for each keypoint x_i are found by calculating the Euclidean distance between all the other $(n-1)$ keypoints; then we obtain the vector $D = \{d_1, d_2, \ldots, d_{n-1}\}$ which represents these distances sorted from nearest to farthest. Ratios d_i/d_{i+1} are calculated in turn $T_i = d_i/d_{i+1}\,(i = 1, 2, 3, \ldots, n-2)$. The iteration stops in the value k when $T_k < \tau_2$ and $T_{k+1} > \tau_2$. Each keypoint in correspondence to a distance in $\{d_1, d_2, \ldots, d_k\}$ is considered as a matched keypoint.

2.2.2. Feature Matching in Smooth Region. Due to the fact that only a small number of keypoints can be extracted in the smooth region, we propose a new matching strategy. Firstly, blocks with similar color are clustered into a group. Secondly, blocks are updated by registration within a group. Feature-based LSH is used to accelerate finding correct matches in the last. Noting that after the first round of matching we need to do registration according to the next block and execute the same operation until all the blocks in this group are traversed. The framework of feature matching in smooth region is shown in Figure 2.

We assume that the source region and its corresponding target region in the image share the same color distribution. Hence, when we detect a pair of suspicious regions, we just need to search in blocks which have similar color. For example, if the content of the coping source region is ocean, its matching region must exist in the blue region in the image. In addition, in this way, we can reduce computational complexity. In order to remove the influence of brightness, we convert these selected smooth blocks from RGB color space to HSV color space to get new ones B_{hsv}. The H and S components are uniformly quantized to the k levels, where $H = 0, 1, \ldots, k-1$; $S = 0, 1, \ldots, k-1$. Then we compute the mean value of H and S in each block $B_{\text{hsv}}^{(i,j)}$ to obtain $h_{\text{mean}}^{(i,j)}$ and $s_{\text{mean}}^{(i,j)}$, where (i, j) represents the coordinates of the upper left corner of the block. We use $(h_{\text{mean}}^{(i,j)}, s_{\text{mean}}^{(i,j)})$ as block's color feature; then some of them will be divided into a group if their color features are very similar, since H and S are quantized to the k levels, respectively, and the total number of color groups is k^2. Finally, we count all the nonempty color groups; then the color group can be represented as $\{\text{group}_1, \text{group}_2, \ldots, \text{group}_{Ng}\}$, where N_g is the number of the nonempty color groups.

The traditional block-based methods usually use overlapping blocks to extract feature, which are very time consuming, so we use nonoverlapping blocks in this paper. As shown in Figure 3, the copying source region is marked by a green line and the pasting target region is marked by a blue line. The two red solid lines a and b are candidate blocks to be matched in the next step. If we use the LSH algorithm directly to find similar blocks, the mismatching is more likely to occur due to disalignment. To solve this problem, before hashing them, we need to calibrate the remaining blocks according to each block in turn within a color group.

Phase correlation is a common method of image registration [24]. Given an block $b_1(x, y)$, shift it by $(\Delta x, \Delta y)$, and we will get $b_2(x, y)$, where

$$b_2(x, y) = b_1(x - \Delta x, y - \Delta y). \tag{2}$$

Their Fourier transforms satisfy

$$B_2(u, v) = B_1(u, v) \cdot e^{j(u\Delta x + v\Delta y)}. \tag{3}$$

The cross power spectrum of them is

$$P(u, v) = \frac{B_1(u, v)\, B_2^*(u, v)}{\left\| B_1(u, v)\, B_2^*(u, v) \right\|}. \tag{4}$$

As shown in Figure 4, the inverse Fourier transformation of (4) is a two-dimensional pulse function, with a peak at $(-\Delta x, -\Delta y)$. We make block$^{(i,j)}$ and block$^{(u,v)}$ stand for two candidate matching blocks a and b in Figure 3. We first compute the inverse Fourier transform of their cross power spectrum. If the peak value is larger than τ_3 and the values of other positions are lower than τ_4, we consider that the two blocks satisfy the registration condition. Then we use the position of the peak $(\Delta x, \Delta y)$ as a displacement vector to change block$^{(u,v)}$ to block$^{(u-\Delta x, v-\Delta y)}$. In Figure 2, the updated block is marked by a red dashed line. On the contrary, if these two blocks cannot satisfy the registration condition, we will not update the current candidate block.

We use Zernike moments as block feature here. Zernike moments of order n and repetition m of a digital image $I(\rho, \theta)$ are defined as

$$Z_{n,m} = \frac{n+1}{\pi} \sum_{(\rho,\theta) \in \text{uint disk}} \sum I(\rho, \theta)\, V_{n,m}^*(\rho, \theta), \tag{5}$$

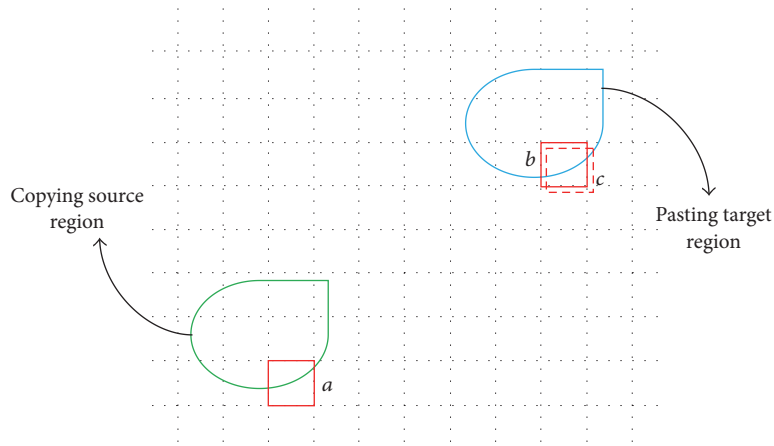


Figure 3: Alignment of candidate blocks.

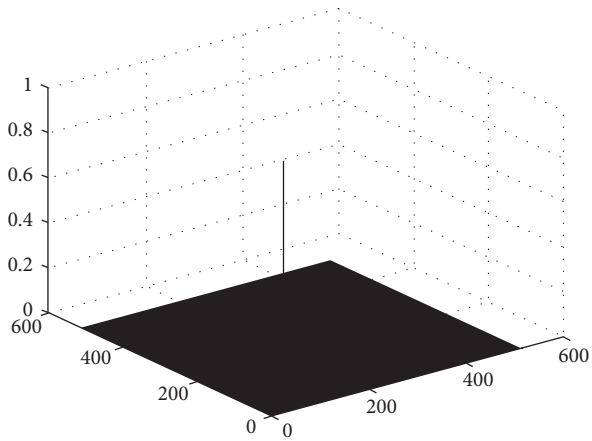


Figure 4: The inverse Fourier transformation of cross power spectrum.

where $V^*_{n,m}(\rho,\theta)$ is a Zernike polynomial of order n and repetition m.

$$V_{n,m}(\rho,\theta) = R_{n,m}(\rho)\, e^{jm\theta}, \tag{6}$$

where $n = 0, 1, \ldots, 0 \leqslant |m| \leqslant n$, $n - |m|$ is even, and $R_{n,m}(\rho)$ are real-valued radial polynomials. We choose the order of the Zernike moment, $n = 5$. Consequently, the Zernike moments extracted from all blocks can be grouped as follows:

$$Z_l = \left\{ z^{(i,j)}, z^{(p,q)}, \ldots, z^{(u,v)} \right\}, \quad l = 1, 2, \ldots, N_g. \tag{7}$$

We make the blocks corresponding to feature vectors in the set Z_l satisfying

$$\left(\text{block}^{(i,j)}, \text{block}^{(p,q)}, \ldots, \text{block}^{(u,v)} \right) \in \text{group}_l. \tag{8}$$

Feature vectors can be regarded as data points distributed in high dimensional character space. The closer the data points, the higher the similarity they have. LSH algorithm can project data points from original data space to hash space through hash functions, and the hash functions satisfy the intuitive notion that the probability of a hash collision for two points is related to the similarity between the points.

Each feature vector $z^{(i,j)}$ can be projected to hash space and get hash value. We choose a hash function based on p-stable distribution:

$$\text{hash}^{(a,b)}\left(z^{(i,j)}\right) = \left\lfloor \frac{a \cdot z^{(i,j)} + b}{w} \right\rfloor, \tag{9}$$

where $\lfloor \cdot \rfloor$ represents an operation of rounding down, w is quantization step, b is a random real number located in $[0, w]$, p is set to 2, and a is a random vector, $a = (a_0, a_1, \ldots, a_{N_{\text{moments}}-1})$. In order to increase the accuracy of collision between similar vectors, we use Q groups of hash functions $g_0, g_1, \ldots, g_{Q-1}$ to project the feature vectors, respectively, in which each group has P hash functions $g(z) = (\text{hash}^{(a_0,b_0)}(z), \text{hash}^{(a_1,b_1)}(z), \ldots, \text{hash}^{(a_{P-1},b_{P-1})}(z))$. Projecting each feature vector with a group of hash functions can obtain a set of hash values that will be used as their bucket numbers. Therefore, we can get Q different hash index tables for similarity vectors searching. The establishment of a hash tables is shown in Figure 5. After projecting feature vector $z^{(i,j)}$ by Q groups of hash functions, we get Q different bucket numbers. All the feature vectors of these buckets are taken out as a set to be screened. Then the similarity between $z^{(i,j)}$ and all of these vectors is calculated. It is worth noting that we do not use the first order $Z_{0,0}$ as it represents the average intensity and its value is much higher than that of others. In our implementation, L2-norm is used to represent similarity between feature vectors. The similarity is therefore

$$\text{similar}\left(z^{(i,j)}, z^{(k,l)}\right) = \left\| z^{(i,j)} - z^{(k,l)} \right\|. \tag{10}$$

If similar $(z^{(i,j)}, z^{(k,l)}) < D$, the two blocks are detected to be similar. Then, we calibrate the rest of the blocks according to another block in this color group until all the blocks are traversed.

A lot of measures are taken to try to reduce the computational complexity of our algorithm. We divide image into

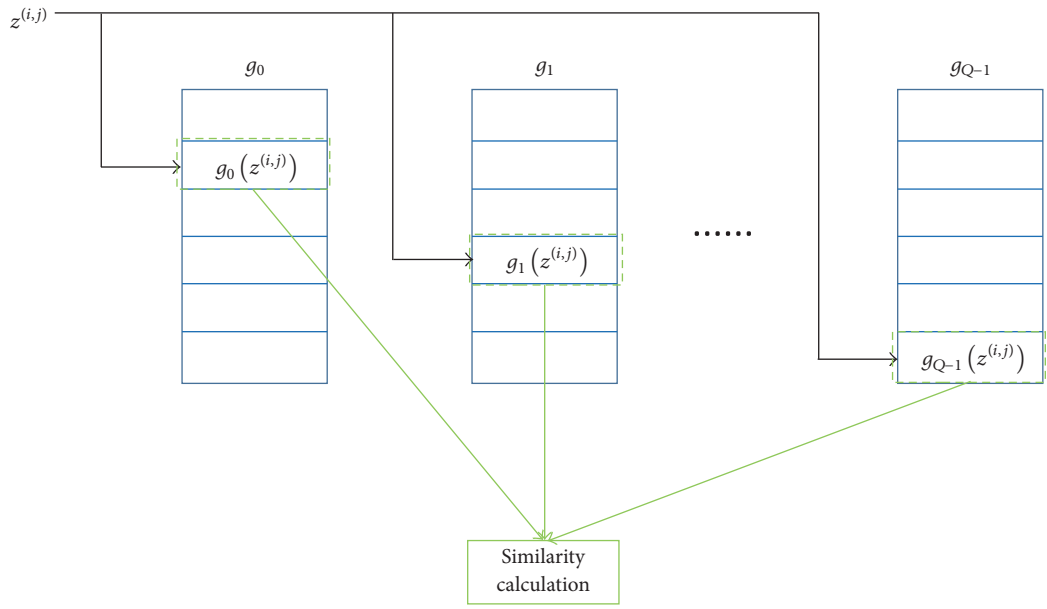


Figure 5: Establishment of hash tables. We use Q groups of hash functions to project the feature vectors, respectively. Projecting each feature vector with a group of hash functions can get a set of hash values as their bucket numbers. Finally, we take out all the feature vectors of these Q hash buckets and calculate the similarity.

texture part and smooth part; therefore we need to calculate their computational complexity separately. In texture part, we extract SIFT feature and use g2NN matching strategy, so the computational complexity of this part is $O(n^2)$, where n is the number of keypoints, while in smooth part blocks with similar color are clustered into a group. Since H and S are quantified to 10 levels, the number of color groups is 100. And in registration step, we update all the other blocks with reference to a query block each time. So the computational complexity of this part is $O(100*(m/100)^3)$ on average, where m is the number of smooth blocks. Special to note is that traditional block-based approaches use overlapping blocks to find duplicated regions, and the total number of blocks is $(M - B + 1) * (N - B + 1)$, where M and N are the width and height of the image, respectively, and B is the length of square block. It is obvious that its computational complexity is $O((M * N)^2)$. Because we use nonoverlapping blocks, $m = (M * N)/B^2$ at most, and $B = 16$ in most cases. Therefore, in the worst case the computational complexity of our method is $O(n^2 + 6 * 10^{-12} * (M * N)^3)$. The number of keypoints is usually not much, and for the most frequently used datasets, our method can be much faster than the traditional block-based methods.

2.3. Postprocessing. The goal of this last step is to present to us more accurate matches. In this paper, we consider three steps to remove falsely matched pairs, including distance, relative position, and affine transformation.

The adjacent keypoints or blocks might be misregarded as the matching pairs, due to their similar characteristics. So the matching pairs will be eliminated, if the distance between them is smaller than the threshold D_2.

The true matching pairs are often distributed densely, so a merging method named Broad First Search Neighbors (BFSN) clustering [25] is performed on spatial locations of the matching pairs to help delete the isolated keypoints or blocks. In this algorithm, if the distance between point a and point b is smaller than a threshold, the two points are defined as neighbors. For a cluster C with m elements, point p can be merged into C, while p is the neighbor of at least λm elements in C. The radio factor λ, which ranges from 0 to 1, controls size and shape of the cluster. First of all, we create an empty class A_1 and put the first row (x_1', y_1') of coordinate matrix C' into A_1. Secondly, we search for all neighbors of (x_1', y_1') in a breadth first order and determine whether the current vector can be incorporated into the class A_1 in terms of clustering conditions. Finally, we delete the vectors in C' which have been incorporated into A_1 and put the first row of the current matrix C' into A_2. Repeat the above three steps until $C' = \varnothing$. The cluster whose inner elements are less than 3 will be regarded as the isolated cluster and the elements in them will be deleted.

Truly matched pairs satisfy the same affine transformation, which means they should exhibit similar amounts of translation, scaling, and rotation. The relationship between these two matched blocks $\text{block}^{(i,j)}$ and $\text{block}^{(k,l)}$ can be represented as follows, where H is a 3×3 matrix:

$$\begin{bmatrix} i \\ j \\ 1 \end{bmatrix} = H \cdot \begin{bmatrix} k \\ l \\ 1 \end{bmatrix}. \tag{11}$$

We assume that if the two matching pairs are from the same duplicated regions, the slopes of their matching lines are

Table 1: Detection result on Benchmark database.

Methods	Precision (%)	Recall (%)	$F1$ (%)
SIFT [16]	79.59	81.25	80.41
Segmentation [22]	70.18	83.33	76.19
Proposed method	90.91	83.33	86.96

consistent. BFSN clustering is performed on the slopes of matching lines to divide the matching pairs into different clusters. RANSAC algorithm [26] is then performed on each group, respectively, to eliminate false matches. This method can avoid misdeleting in the situation of managing the multiduplicated matching. In addition, the test image will be identified as the tampered one, if the number of remaining matches is more than a threshold.

3. Experimental Results

In this section, error measures are first introduced, and experiments are conducted on three databases that are available online, Benchmark presented by Christlein et al. [23], CoMoFoD [27], and GRIP [28]. The first one is composed of 48 images, which range from 800×533 to 3888×2592 in size, having about 10% of the whole image as tampered regions. The second one comprises 200 original images and their corresponding tampered images with size of 512×512. There are 7 groups of tampered images, one is without any attacks, and the rest are manufactured by adding various attacks, including JPEG compression, blurring, noise adding, and color reduction. The last one consists of 80 images, and these images have fixed size 1024×768 pixels; each of them has only one pair of duplicated regions. The hardware environment of the experiments is a 3.60 GHz Intel Core i7-4790 processor; the software used is Windows 7 MATLAB R2014b.

3.1. Error Measures. We often evaluate the performance of a CMFD method at two levels, namely, image level and pixel level. At image level, we are concerned about whether or not this image has been tampered, while in pixel level we pay more attention to the accuracy of location. Precision and recall are used here. The detection error at the image level is defined as

$$\begin{aligned} \text{precision} &= \frac{T_P}{T_P + F_P}, \\ \text{recall} &= \frac{T_P}{T_P + F_N}, \end{aligned} \tag{12}$$

where T_P is the number of images that have been correctly identified as forged, F_P represents the number of images that have been erroneously detected as forged, and F_N indicates the falsely missed forged images. The detection error at the pixel level is defined as

$$\begin{aligned} \text{precision} &= \frac{\left|\Phi \cap \Phi'\right|}{|\Phi|}, \\ \text{recall} &= \frac{\left|\Phi \cap \Phi'\right|}{\left|\Phi'\right|}, \end{aligned} \tag{13}$$

where Φ is the number of pixels detected by the CMFD method; Φ' is the number of all forgery pixels marked by ground truth.

We also use F_1 score as a comprehensive measure.

$$F_1 = \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}. \tag{14}$$

3.2. Detection Results on the Benchmark Database. We first test the proposed method on Benchmark database. Several example test cases are shown in Figure 6. The first row of Figure 6 shows the original images, the second row shows the tampered images, the third row shows the ground truth maps, and the last row shows the detection results of the proposed method. In Table 1 we compare the proposed algorithm with two other methods, and the result indicates that our method can achieve 90.91% precision, 83.33% recall, and 86.96% F_1 score, which preforms better than SIFT [16] and Segmentation [22]. In the third column of Figure 6, though the tampered area is smooth, we can still detect it.

A practical CMFD algorithm should have a relatively low computational complexity in addition to maintaining a certain degree of accuracy. In order to measure the performance of our proposed method, we compare the time complexity of different algorithms on this dataset. The average execution times are illustrated in Figure 7. It should be noted that the implementation platforms of evaluated methods are different. For instance, Zernike is implemented in C++ for speed, and segmentation and the proposed method are implemented in MATLAB. Because of the high resolution, all methods demand more time. The proposed method is the fastest, except for SIFT [16].

3.3. Detection Results on the CoMoFoD Database. Next the experiments are carried out on CoMoFoD database. Illustration of four cases is shown in Figure 8. Each row from top to bottom represents original images, tampered images, ground truth, and detection results of the proposed method, respectively. Table 2 shows the detection results comparing with other methods. From this table, it can be clearly seen that the results of the proposed method outperform [16, 22] under ideal conditions.

In addition, we conduct experiments to evaluate the robustness of our method against various attacks. That will make the CMFD more difficult. The forgery images are

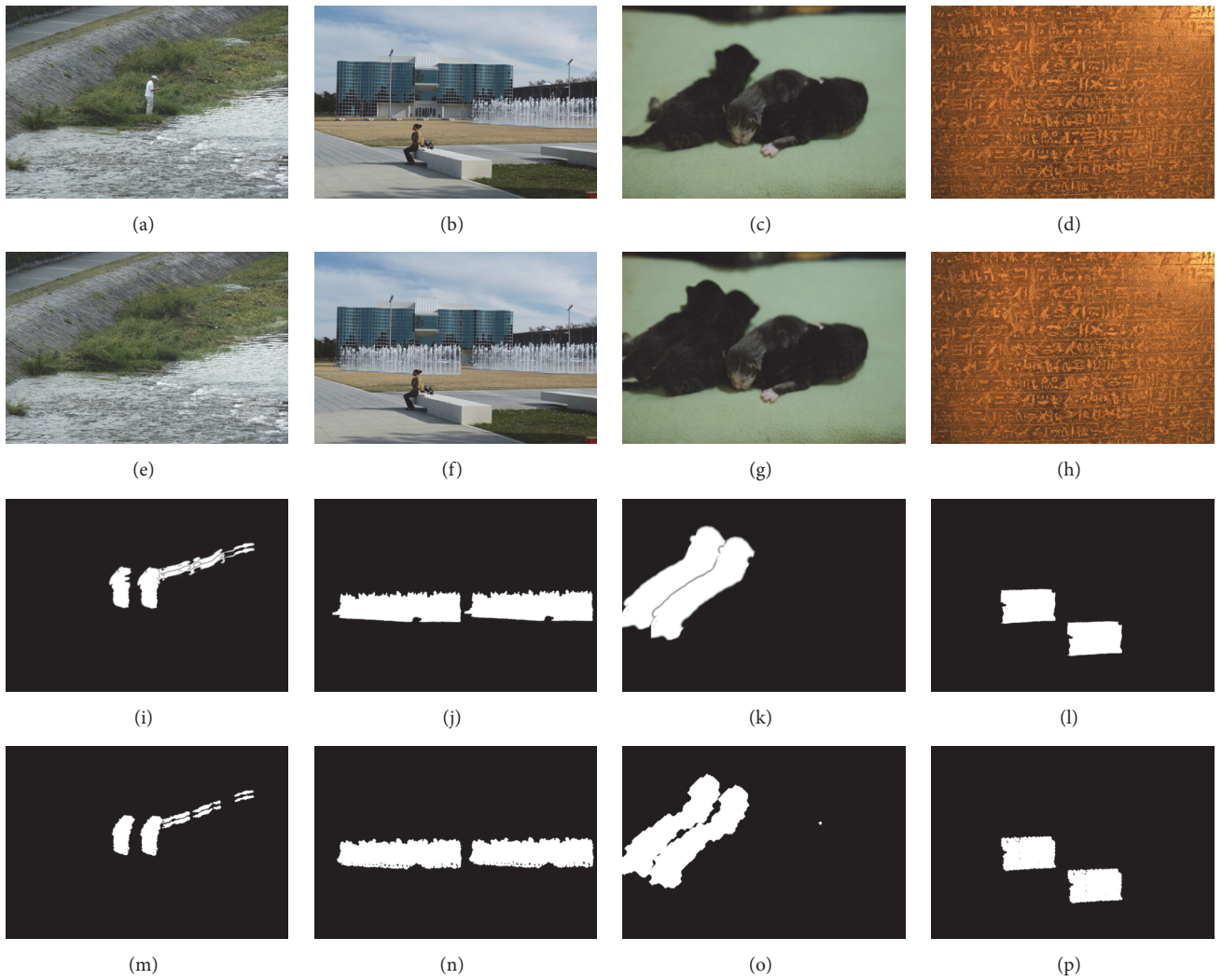


FIGURE 6: Example test cases. ((a), (b), (c), and (d)) Original. ((e), (f), (g), and (h)) Tampered. ((i), (j), (k), and (l)) Ground truth. ((m), (n), (o), and (p)) Proposed.

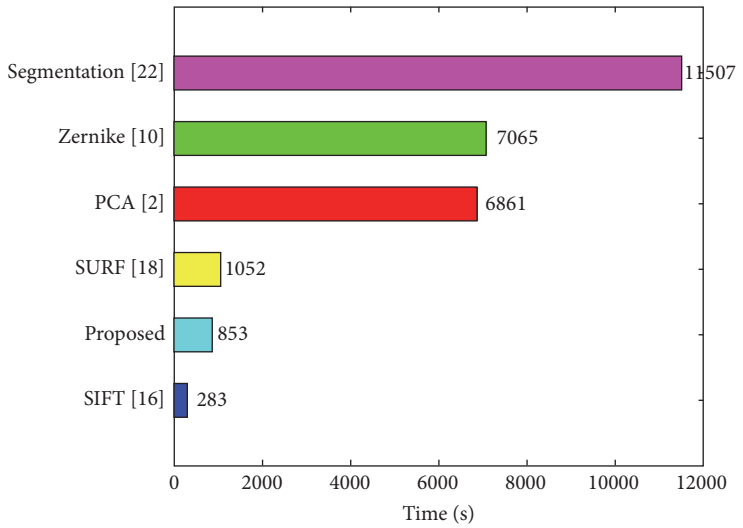


FIGURE 7: The average execution times [2, 10, 16, 18, 22].

Table 2: Detection result on CoMoFoD database.

Methods	Precision (%)	Recall (%)	$F1$ (%)
SIFT [16]	77.83	82.50	80.10
Segmentation [22]	77.19	66.00	71.16
Proposed method	89.30	83.50	87.55

Table 3: Detection result on GRIP database.

Methods	Precision (%)	Recall (%)	$F1$ (%)
SIFT [16]	78.87	70.00	74.17
Segmentation [22]	69.47	82.50	75.43
Proposed method	90.54	83.75	87.01

Table 4: Detection results on pixel level.

Serial number	Precision (%)	Recall (%)	$F1$ (%)
8	99.18	60.05	74.80
10	86.04	48.54	62.06
12	99.56	70.13	82.29
30	91.26	89.57	90.40
31	98.51	72.30	83.39
54	86.88	66.62	75.42
61	98.80	85.76	91.82

generated by making the copied snippets undergo 3 kinds of attacks, namely, JPEG compression, rotation, and scaling.

(1) JPEG compression: the image quality-factor varies from 20 to 100 with the step of 10. The results are shown in first row of Figure 9. For most of the quality factors, the precision, recall, and F_1 score of our method are higher than those of [16, 22]. The curve of red line has smaller slope than others; that means our method has better robustness against JPEG compression.

(2) Rotation: we rotate the copied snippets with the rotation angles varying from 2° to 10° with the step of 2°. The results are shown in second row of Figure 9. It can be clearly seen that our method is much better than [16, 22].

(3) Scaling: the copied snippets are scaled with the scale factors varying from 91% to 109% with the step of 2%. As shown in third row of Figure 9, our method outperforms [16, 22] in terms of precision and F_1 score, and recall is almost as same as that of [16].

3.4. Detection Results on the GRIP Database. We also test the performance of the proposed method on the GRIP database. The detection results are given in Table 3. It can be observed that all evaluation indicators of our method are well over those of [16, 22]. In particular, each evaluation indicator is more than 11% higher than that of [16]. This is because the database contains a lot of smooth images, and our method divides the smooth area of the image into nonoverlapping blocks to find correct duplicated regions.

We first classify the blocks into several groups according to their color distribution. In order to prove that this step can improve the efficiency of the algorithm, a baseline reference technique with registration and LSH is considered. From Figure 10, we are able to see that the advantage of grouping is obvious, and the computational complexity is decreased by 94.67%.

Moreover, we present some experiments to assess the impact of registration. Some smooth images from this dataset are picked out for this purpose. Results reported in Table 4 confirm the effectiveness of the registration step in matching.

4. Conclusion

In this paper, we propose a new CMFD method, which is able to solve the problem of high computational complexity in traditional block-based methods. First, image is divided into two different types of regions. Next, features are extracted and matched in texture region and smooth region using SIFT and Zernike moments, respectively. In particular, we use nonoverlapping blocks as candidates in smooth area and do the registration via the phase correlation algorithm before matching. Finally, the exact forgery regions will be generated after removing falsely matched pairs and exploiting morphology operations. There are three main contributions in our proposed method as follows. (1) Dividing image into texture region and smooth region according to the number of keypoints can not only avoid the poor performance using SIFT in smooth area but also reduce computational complexity. (2) We use nonoverlapping blocks as candidates in smooth area instead of using all overlapping blocks

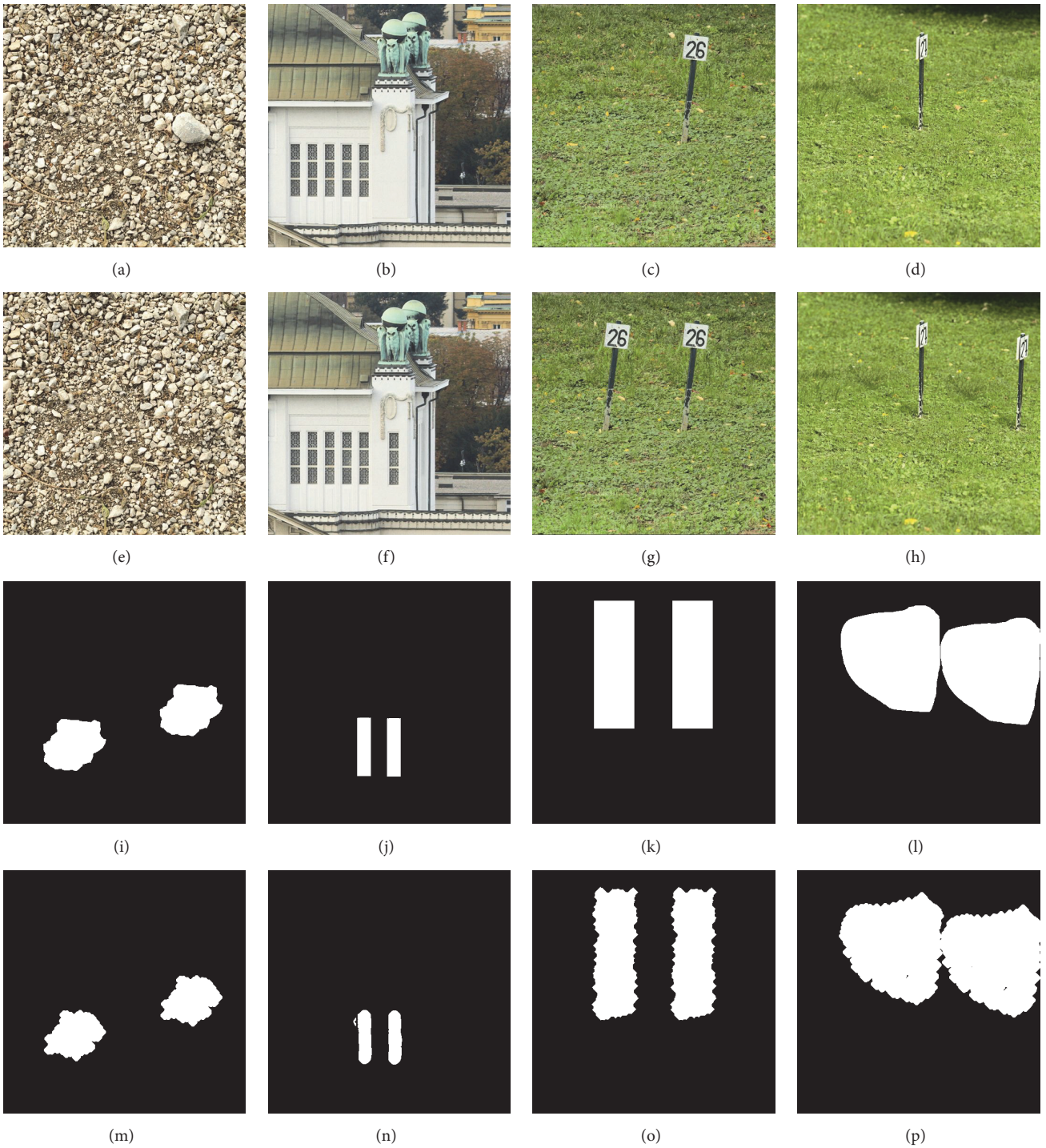


FIGURE 8: Illustration of four cases. ((a), (b), (c), and (d)) Original. ((e), (f), (g), and (h)) Tampered. ((i), (j), (k), and (l)) Ground truth. ((m), (n), (o), and (p)) Proposed.

and apply color based feature to decrease the number of blocks needed to be matched. (3) In order to prevent mismatching due to disalignment, we do registration before using LSH algorithm to obtain more accurate candidate blocks.

Experimental results show that the performance of the proposed algorithm is better than the state-of-the-art CMFD methods. In addition, our method can manage multiple copied regions. Meanwhile, it exhibits robustness to JPEG compression, rotation, and scaling. However, when dealing with very smooth blocks, such as sky and wall, the effect of registration is not particularly good. In our future work, we will try to improve the accuracy of location and the detection speed.

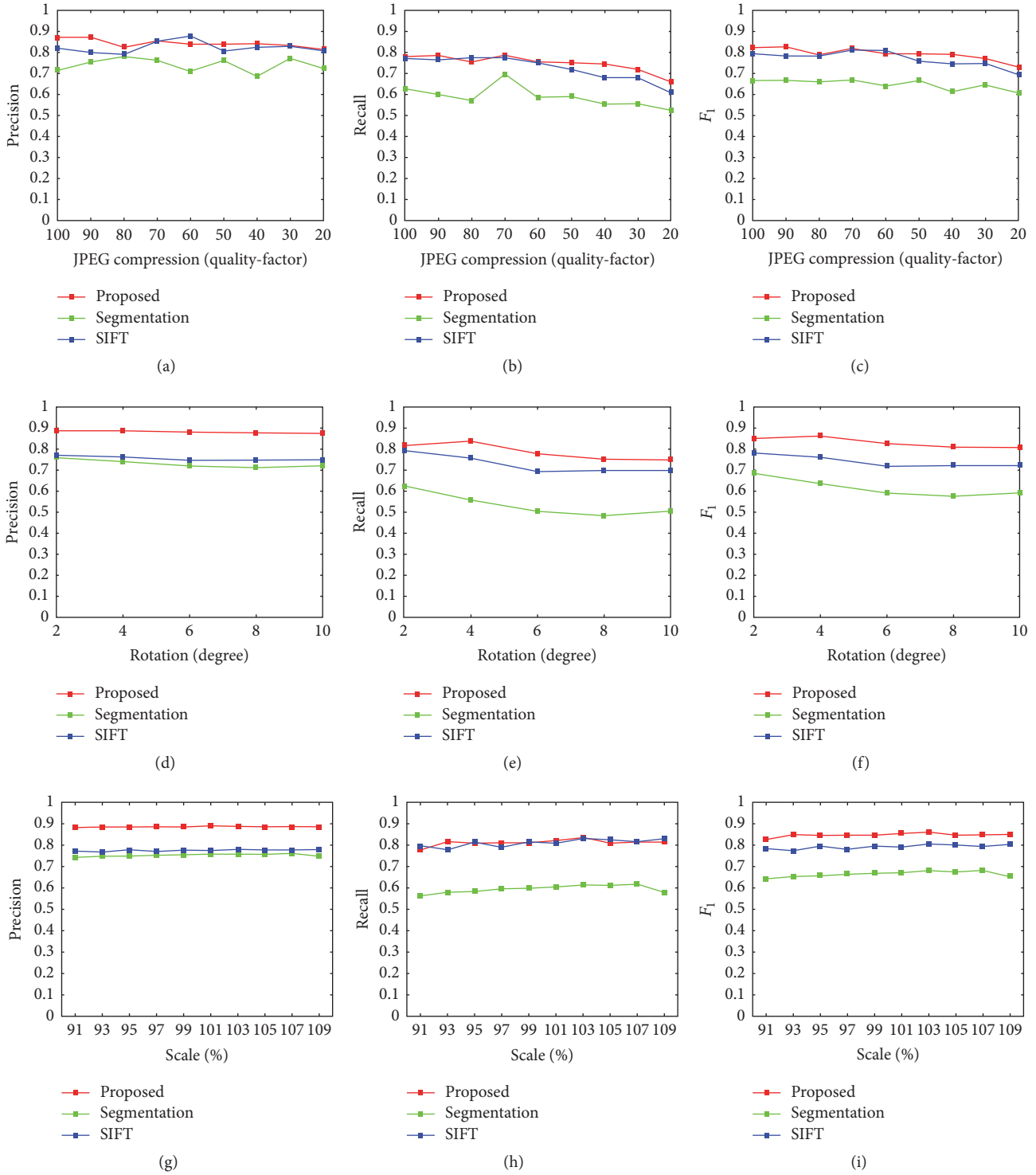


Figure 9: Detection results under 3 kinds of attacks. The three rows from (a–c), (d–f), and (g–i) are JPEG compression, rotation, and scaling, respectively. The three columns from (a–g), (b–h), and (c–i) give the precision, recall, and F_1 score, respectively.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by National Key Research and Development of China (2016YFB0800404), National

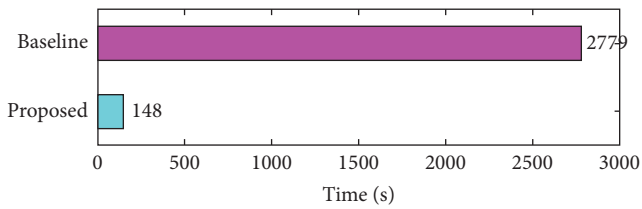


Figure 10: Average execution time per image for the proposed method and baseline reference.

NSF of China (61672090, 61332012), and Fundamental Research Funds for the Central Universities (2015JBZ002).

References

[1] J. Fridrich, D. Soukal, and J. Lukāiåi, "Detection of copy-move forgery in digital images," in *Proceedings of Digital Forensic Research Workshop*, Citeseer, 2003.

[2] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 758–767, 2005.

[3] M. Bashar, K. Noda, N. Ohnishi, and K. Mori, "Exploring duplicated regions in natural images," *IEEE Transactions on Image Processing*, no. 99, pp. 1–40, 2010.

[4] G. Li, Q. Wu, D. Tu, and S. Sun, "A sorted neighborhood approach for detecting duplicated regions in image forgeries based on DWT and SVD," in *Proceedings of the IEEE International Conference onMultimedia and Expo (ICME '07)*, pp. 1750–1753, Beijing, China, July 2007.

[5] X. B. Kang and S. M. Wei, "Identifying tampered regionsusing singular value decomposition in digital image forensics," in *Proceedings of the International Conference on Computer Science and Softare Engineering (CSSE '08)*, vol. 3, pp. 926–930, IEEE, December 2008.

[6] S. Bayram, H. T. Sencar, and N. D. Memon, "An efficient and robust method for detecting copy-move forgery," in *Proceedings of the Proceeding of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '09)*, pp. 1053–1056, Taipei, Taiwan, April 2009.

[7] B. Mahdian and S. Saic, "Detection of copy-move forgery using a method based on blur moment invariants," *Forensic Science International*, vol. 171, no. 2-3, pp. 180–189, 2007.

[8] J.-W. Wang, G.-J. Liu, Z. Zhang, Y.-W. Dai, and Z.-Q. Wang, "Fast and robust forensics for image region-duplication forgery," *Acta Automatica Sinica*, vol. 35, no. 12, pp. 1488–1495, 2009.

[9] S. Ryu, M. Kirchner, M. Lee, and H. Lee, "Rotation invariant localization of duplicated image regions based on zernike moments," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 8, pp. 1355–1370, 2013.

[10] S.-J. Ryu, M.-J. Lee, and H.-K. Lee, "Detection of copy-rotate-move forgery using zernike moments," in *Information Hiding*, vol. 6387 of *Lecture Notes in Computer Science*, pp. 51–65, Springer, 2010.

[11] W. Luo, J. Huang, and G. Qiu, "Robust detection of region-duplication forgery in digital image," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR '06)*, vol. 4, pp. 746–749, Hong Kong, August 2006.

[12] J. Wang, G. Liu, H. Li, Y. Dai, and Z. Wang, "Detection of image region duplication forgery using model with circle block," in *Proceedings of the 1st International Conference on Multimedia Information Networking and Security (MINES '09)*, vol. 1, pp. 25–29, IEEE, November 2009.

[13] H. J. Lin, C. W. Wang, and Y. T. Kao, "Fast copy-move forgery detection," *WSEAS Transactions on Signal Processing*, vol. 5, pp. 188–197, 2009.

[14] S. Bravo-Solorio and A. K. Nandi, "Exposing duplicated regions affected by reflection, rotation and scaling," in *Proceedings of the 36th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '11)*, pp. 1880–1883, Prague, Czech Republic, May 2011.

[15] H. Huang, W. Guo, and Y. Zhang, "Detection of copy-move forgery in digital images using Sift algorithm," in *Proceedings of the Pacific-Asia Workshop on Computational Intelligence and Industrial Application (PACIIA '08)*, pp. 272–276, Computer Society, December 2008.

[16] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra, "A SIFT-based forensic method for copy-move attack detection and transformation recovery," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1099–1110, 2011.

[17] B. Xu, J. Wang, G. Liu, and Y. Dai, "Image copy-move forgery detection based on SURF," in *Proceedings of the 2nd International Conference on Multimedia Information Networking and Security (MINES '10)*, pp. 889–892, IEEE, Nanjing, China, November 2010.

[18] B. L. Shivakumar and S. Baboo, "Detection of region duplication forgery in digital images using SURF," *International Journal of Computer Science Issues*, vol. 8, no. 4, pp. 199–205, 2011.

[19] R. Davarzani, K. Yaghmaie, S. Mozaffari, and M. Tapak, "Copy-move forgery detection using multiresolution local binary patterns," *Forensic Science International*, vol. 231, no. 1–3, pp. 61–72, 2013.

[20] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: binary robust invariant scalable keypoints," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 2548–2555, Barcelona, Spain, November 2011.

[21] R. Sekhar and R. S. Shaji, "A study on segmentation-based copy-move forgery detection using DAISY descriptor," *Advances in Intelligent Systems and Computing*, vol. 398, pp. 223–233, 2016.

[22] J. Li, X. Li, B. Yang, and X. Sun, "Segmentation-based image copy-move forgery detection scheme," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 507–518, 2015.

[23] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou, "An evaluation of popular copy-move forgery detection approaches," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 6, pp. 1841–1854, 2012.

[24] A. Yousef, J. Li, and M. Karim, "High-speed image registration algorithm with subpixel accuracy," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1796–1800, 2015.

[25] J. Qian and Y. Dong, "Clustering algorithm based on broad first searching neighbors," *Journal of Southeast University (Natural Science Edition)*, vol. 34, no. 1, pp. 109–112, 2004.

[26] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[27] D. Tralic, I. Zupancic, S. Grgic, and M. Grgic, "CoMoFoD - New database for copy-move forgery detection," in *Proceedings of the 55th International Symposium ELMAR 2013*, pp. 49–54, hrv, September 2013.

[28] D. Cozzolino, G. Poggi, and L. Verdoliva, "Copy-move forgery detection based on PatchMatch," in *IEEE International Conference on Image*, pp. 5312–5316, 2013.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 6853696, 11 pages
https://doi.org/10.1155/2018/6853696

Research Article

Perceptual Hashing-Based Image Copy-Move Forgery Detection

Huan Wang and Hongxia Wang

School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China

Correspondence should be addressed to Hongxia Wang; hxwang@swjtu.edu.cn

Received 29 September 2017; Revised 10 December 2017; Accepted 20 December 2017; Published 22 January 2018

Academic Editor: Zhenxing Qian

This paper proposes a blind authentication scheme to identify duplicated regions for copy-move forgery based on perceptual hashing and package clustering algorithms. For all fixed-size image blocks in suspicious images, discrete cosine transform (DCT) is used to obtain their DCT coefficient matrixes. Their perceptual hash matrixes and perceptual hash feature vectors are orderly addressed. Moreover, a package clustering algorithm is proposed to replace traditional lexicographic order algorithms for improving the detection precision. Similar blocks can be identified by matching the perceptual hash feature vectors in each package and its adjacent package. The experimental results show that the proposed scheme can locate irregular tampered regions and multiple duplicated regions in suspicious images although they are distorted by some hybrid trace hiding operations, such as adding white Gaussian noise and Gaussian blurring, adjusting contrast ratio, luminance, and hue, and their hybrid operations.

1. Introduction

Copy-move forgery as a popular digital image tampering technology is extensively used by forgers. In a digital image, some regions are copied and then pasted into other regions in this same image to achieve the purpose of hiding some targets or emphasizing some important objects [1]. Its authenticity is broken. Since the original regions and the duplicated regions come from the same image, their most important characteristics, such as the color palettes, noises, and dynamic ranges, are compatible with the remainder of the image [2]. One may neglect this malicious operation if forgers deliberately hide the tampering traces. A typical copy-move forgery example is shown in Figure 1, where the traffic flow is exaggerated by tampering cars. It is urgent to propose effective copy-move forgery detection methods to detect and locate the tampered regions for digital images. Blind authentication for copy-move forgery mainly focuses on the identifying of tampered regions in digital images without any additional information except for themselves. Based on this advantage, it becomes to a valuable research in image authentication fields [3].

Block-based methods and keypoint-based methods are the common techniques for copy-move forgery detection. Block-based methods indicate that a suspicious image is divided into overlapped and fix-sized blocks. The tampered regions can be identified by matching the similar feature vectors that are extracted from the blocks. Fridrich et al. in [1] first proposed a block-based detection scheme using quantized discrete cosine transform (DCT) coefficients, which is one of the landmark methods for copy-move forgery detection. Popscu and Farid in [4] presented a novel method that use principal component analysis (PCA) to derive an alternative representation for each image block. However, it cannot resist other robustness attacks and identify some little tampered regions. Babak and Stanislav in [5] presented a copy-move forgery detection scheme to extract image features for overlapped blocks based on blur moment invariants. Cao et al. in [6] exploited the mean of DCT coefficients to propose their algorithm that not only can resist the attacks of blurring and additive noise operations, but also considers the detection accuracy rate (DAR) and false positive rate (FPR). However, it is weak to resist the attack of hue or contrast ratio adjustments. Thajeel and Sulong in [7] presented an approach to improve the detection precision based on completed robust local binary pattern. Wang et al. in [8] proposed a copy-move forgery detection scheme to improve the detection precision based on DCT and package clustering algorithms. However,

(a) (b)

Figure 1: An example of copy-move forgery: (a) an original image and (b) a copy-move forgery image.

the robustness of feature vectors has been concentrated less. Zhong et al. in [9] presented a scheme that divides the tampered image into overlapped circular blocks. The features of circular blocks are extracted by the discrete radial harmonic Fourier moments. This method obtains outstanding performance under image geometrical distortions. Dixit et al. in [10] proposed a method for detecting copy-move forgery using stationary wavelet transform. The detection accuracy of the proposed method is also considered. Bi and Pun in [11] presented a fast reflective offset guided searching method for image copy-move forgery detection. It aims to reduce the computational complexity. The block-based methods mentioned above need to divide the image into overlapped and fix-sized blocks and then handle each of them. These algorithms can resist some plain postprocessing operations, such as JPEG compression, blurring, and noise interference. However, they did not achieve satisfactory results to resolve a common problem of reducing similar region matching times.

Keypoint-based methods rely on the identification of high-entropy image regions [12]. A feature vector can be extracted for each keypoint. Fewer feature vectors are estimated since the number of keypoints is reduced. Therefore, keypoint-based methods theoretically have lower computational costs for feature vectors matching and postprocessing. Amerini et al. in [12, 13] presented scale invariant feature transform (SIFT) to filter, sort, and classify the keypoint pairs for copy-move forgery detection. Li et al. in [14] try to reduce the similar region matching times and improve the DAR and FPR by segmenting a suspicious image into nonoverlapped patches. Wang et al. in [15] introduced a keypoints-based image passive detecting method based on Harris detector and region growth technology. It is robust for JPEG compression, gamma adjustment, and luminance enhancement. Li et al. in [16] proposed a hierarchical cluster algorithm based on maximally stable color region detector and Zernike moments to extract all keypoint features. Wang et al. in [17] presented a method to segment a suspicious image into irregular superpixels that are classified into smooth, texture, and strong texture. The stable image keypoints can be extracted from each superpixel. The above-mentioned algorithms have moved the copy-move forgery detection field ahead rapidly. However, they did not achieve satisfactory results on the improving of DAR and FPR in order to reduce the matching times. The resistance for other postprocessing operations is less considered, such as adjusting contrast ratio, luminance, hue, and their hybrid operations.

Perceptual hashing [18] is a class of one-way mappings from multimedia presentations to perceptual hash values in terms of the perceptual content. It is widely applied to perform multimedia content identification, retrieval, and authentication. In similar image searching and target tracking, perceptual hash algorithms are applied to generate fingerprints for digital images and then are used to compare them with each other. In addition, perceptual hash values are robust to take into account transformations or "attacks" on a given input and, yet, flexible enough to distinguish between dissimilar files. Such attacks include skew, contrast adjustment and different compression. Perceptual hash values are analogous if features are similar [19]. In a copy-move forgery image, the copy regions are similar with their paste regions. Therefore, perceptual hash algorithms can also be used to generate robust features for detecting the tampered regions.

In this study, a passive authentication scheme is proposed to perform authenticating for copy-move forgery based on perceptual hashing. The novelty of the proposed scheme includes the following: (1) Using perceptual hashing algorithms, the feature vectors of image blocks are robust for improving the DAR and FPR. (2) A package clustering algorithm is used to replace traditional lexicographic order methods to reduce the block matching times, where each package is used to represent a cluster. (3) Using perceptual hash algorithms, the proposed method can effectively identify and locate multiple duplicated regions in digital images that may be distorted by adding white Gaussian noise and Gaussian blurring, adjusting contrast ratio, luminance, hue, and their hybrid operations.

The rest of this paper is organized as follows. Section 2 introduces the proposed method. Section 3 shows the performance of the proposed scheme with a series of experiments. Finally, this paper is concluded in Section 4.

2. The Proposed Scheme

It is impossible in general that there are two identical regions in a naturally formed picture unless it contains large area

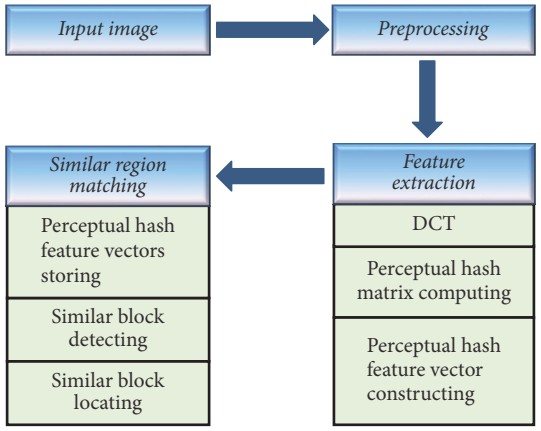


Figure 2: The framework of the proposed copy-move forgery scheme.

smooth regions, such as a blackboard or a piece of blue sky [20]. In this study, we suppose that all images do not contain large area smooth regions.

It is an incontestable fact that each suspicious image contains at least two similar regions, that is, an original region and a copy-move forgery region, if the suspicious image is tampered with copy-move forgery. By concluding many existing schemes, the task of passive authentication for copy-move forgery is to detect and locate tampered regions for suspicious images. In our proposed method, two main steps, that is, feature extraction and feature matching, are separately introduced. In the feature extraction step, perceptual hashing algorithms are extended to generate perceptual hash feature vectors that can be used to represent the image blocks in a suspicious image. In feature matching step, the idea of a package clustering algorithm is used to replace general lexicographically sorting algorithms to improve the detection precision and reduce the feature vector comparing times. Figure 2 shows the framework of the proposed scheme.

2.1. Preprocessing Operation. Let A be a suspicious image. It should be converted into a gray-scale image by $I = 0.299\mathrm{R} + 0.587\mathrm{G} + 0.114\mathrm{B}$ if it is a color image, where R, G, and B represent the red, green, and blue components of A, respectively, and I represents the pixel value of gray-scale image.

2.2. Feature Extraction Using Perceptual Hashing. In this step, suspicious image A is divided into different image blocks. DCT is applied to generate the DCT coefficient matrix for each image block. Finally, perceptual hashing is used to extract a perceptual hash feature vector for each image block according to its generated DCT coefficient matrix. The details of the feature extracting algorithm are shown in Algorithm 1.

In Step 1, suspicious image A with the size of $H \times W$ pixels is divided into $(H-b+1) \times (W-b+1)$ overlapping blocks by sliding a square window with the size of $b \times b$ pixels along with image A from the upper-left corner right down to the lower-right corner; that is, the adjacent overlapping blocks only have

Input: A suspicious gray-scale image A.
Output: All perceptual hash feature vectors for image blocks in A.

Step 1. Suspicious image A is divided into $(W-b+1) \times (H-b+1)$ overlapping blocks, denoted as B_{ij}, where $0 < b \ll W$, $0 < b \ll H$, $1 \le i \le (W-b+1)$, and $1 \le j \le (H-b+1)$.
Step 2. For each block B_{ij}
Step 3. The pixel mean of B_{ij}, denoted as P_{ij}, is computed.
Step 4. DCT is applied to generate the coefficient matrix for block B_{ij}, denoted as C_{ij}.
Step 5. The coefficient matrix C_{ij} is divided into four sub-blocks, denoted as C_{ij}^1, C_{ij}^2, C_{ij}^3, and C_{ij}^4, respectively.
Step 6. The mean of the first sub-block C_{ij}^1 is calculated, denoted as m_{ij}.
Step 7. The perceptual hashing matrix for each sub-block C_{ij}^k is computed, denoted as H_{ij}^k, where $k \in \{1, 2, 3, 4\}$.
Step 8. Each perceptual hashing matrix H_{ij}^k is converted into a decimal number, denoted as d_{ij}^k, to represent feature value for block B_{ij}, where $k \in \{1, 2, 3, 4\}$.
Step 9. The feature vector of block B_{ij} is created, denoted as $F_{ij} = (P_{ij}, d_{ij}^1, d_{ij}^2, d_{ij}^3, d_{ij}^4)$, according to its pixel mean and four feature values.
Step 10. End For

Algorithm 1: Feature extraction.

one different row or column. Each block is denoted as B_{ij}, where $1 \le i \le (H-b+1)$, $1 \le j \le (W-b+1)$, i and j indicate the starting point of the block's row and column, respectively. Therefore, the original regions and their copy-move forgery regions are also divided into different blocks in which there is at least a pair of identical or similar blocks. The main task of the proposed scheme is to detect and locate these identical or similar blocks in pairs.

In Steps 2–10, the feature vector of each block is computed using DCT and perceptual hashing algorithms. It is unideal to directly use pixel values to match similar blocks in suspicious images since the forgers may distort the content of the tampered images. An ideal method is extracting robust features to represent blocks and then the similar blocks can be diagnosed by matching these robust features. The purpose is to strengthen the robustness and improve the detection accuracy of the proposed scheme. In this algorithm, perceptual hash features play this role, which are used to represent image blocks.

In Step 3, the pixel mean of image block B_{ij}, denoted as P_{ij}, is calculated as follows:

$$P_{ij} = \frac{\sum_{x=1}^{b} \sum_{y=1}^{b} f_{ij}(x, y)}{b^2}, \quad (1)$$

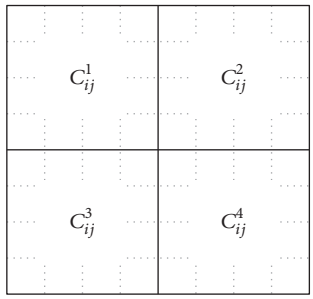


Figure 3: The method of dividing a block into four subblocks.

where $f_{ij}(x, y)$ represents the pixel value of xth row and yth column in B_{ij}. For the pair of two identical or similar image blocks, their pixel means are also identical or similar.

In Step 4, DCT is applied to exploit the DCT coefficient matrix for each block B_{ij}, denoted as C_{ij}, where the DCT coefficient matrix has the same size with block B_{ij}, $1 \le i \le (H - b + 1)$, and $1 \le j \le (W - b + 1)$.

In Step 5, coefficient matrix C_{ij} is divided into four subblocks. A typical characteristic of DCT is that the energy of an image focuses on the low frequency part and the high frequency coefficients play insignificant roles. This means that not all elements are equally important in C_{ij} and the top-left part of C_{ij} represents most features of block B_{ij}. In the proposed method, each DCT coefficient matrix C_{ij} will be equally divided into four subblocks, denoted as C_{ij}^1, C_{ij}^2, C_{ij}^3, and C_{ij}^4, as shown in Figure 3.

In Steps 6–9, the feature vector of block B_{ij} is created using perceptual hashing algorithm. According to the typical characteristic of DCT, the energy of the first subblock can be used to approximately represent the energy of whole block B_{ij}. This means that the average energy of block B_{ij} can be approximately represented by the energy of first subblock C_{ij}^1. In Step 6, the average energy of subblock C_{ij}^1, denoted as m_{ij}^1, is calculated as follows:

$$m_{ij}^1 = \frac{\sum_{x=1}^{b/2} \sum_{y=1}^{b/2} c_{ij}^1(x, y)}{(b/2)^2}, \tag{2}$$

where $c_{ij}^1(x, y)$ indicates the element of xth row and yth column in subblock C_{ij}^1.

In Step 7, the perceptual hash matrixes of the four subblocks C_{ij}^1, C_{ij}^2, C_{ij}^3, and C_{ij}^4, denoted as H_{ij}^1, H_{ij}^2, H_{ij}^3, and H_{ij}^4, respectively, are generated according to the average energy of subblock C_{ij}^1,that is, m_{ij}^1. Therefore, $\forall h_{ij}^k(x, y) \in H_{ij}^k$ is calculated as follows:

$$h_{ij}^k(x, y) = \begin{cases} 1, & c_{ij}^k(x, y) \in C_{ij}^k,\ c_{ij}^k(x, y) \ge m_{ij}^1 \\ 0, & c_{ij}^k(x, y) \in C_{ij}^k,\ c_{ij}^k(x, y) < m_{ij}^1, \end{cases} \tag{3}$$

where $1 \le x \le b/2$, $1 \le y \le b/2$, and $k \in \{1, 2, 3, 4\}$. Obviously, perceptual hash matrix H_{ij}^k can be considered as a perceptual digest from an image block to a binary matrix. It is used to represent the image block.

In Step 8, perceptual hash matrixes are converted into decimal numbers. In practical application, it is easier to calculate and store decimal numbers than binary matrixes. The four perceptual hash matrixes H_{ij}^1, H_{ij}^2, H_{ij}^3, and H_{ij}^4 are converted into four decimal numbers, denoted as d_{ij}^1, d_{ij}^2, d_{ij}^3, and d_{ij}^4, respectively, along with their rows.

In Step 9, the perceptual hash feature vector of block B_{ij} is created. For block B_{ij}, it has five special values that are considered as above, that is, the pixel mean P_{ij} and the four decimal numbers d_{ij}^1, d_{ij}^2, d_{ij}^3, and d_{ij}^4. In order to more accurately represent block B_{ij}, its perceptual hash feature vector, denoted as F_{ij}, is constructed as follows:

$$F_{ij} = \left(P_{ij}, d_{ij}^1, d_{ij}^2, d_{ij}^3, d_{ij}^4\right). \tag{4}$$

Obviously, perceptual hash feature vector F_{ij} has the properties of simpleness and robustness. Therefore, it can be considered as the feature vector for image block B_{ij}.

2.3. Similar Region Matching. In the matching stage of the existing methods, their feature vectors are sorted first by some sorting algorithms, such as traditional lexicographic order algorithms, and then used to detect and locate the similar blocks using block matching methods. However, two kinds of issues in these existing methods should be improved to achieve better matching results. One is the block matching times, it will cause that the proposed matching algorithm has higher time complexity. Another is the precision of locating duplicated regions, which is dissatisfactory. In our proposed scheme, a package clustering algorithm is proposed to detect and locate the tampered regions with the purpose of improving the detection precision. The details of the proposed similar region matching algorithm are described in Algorithm 2.

In Algorithm 2, Steps 1 and 2 construct a package clustering algorithm that stores all perceptual hash feature vectors into the prepared packages according to the pixel means of blocks. Steps 4–7 compare all perceptual hash feature vectors to detect and locate the similar blocks according to the proposed package matching rule.

In Step 1, a set of packages is created. Let n be a preset threshold that represents the maximum capacity of all packages. Therefore, $\xi = (\lfloor 256/n \rfloor + 1)$ packages are created, denoted as $PA_1, PA_2, \ldots$, and PA_ξ, since the suspicious image is a gray-scale image (its pixel range is zero to 255), where $\lfloor\ \rfloor$ is a floor function.

In Step 2, all perceptual hash feature vectors are stored into the ξ packages. Let B_{ij} be a block and $F_{ij} = (P_{ij}, d_{ij}^1, d_{ij}^2, d_{ij}^3, d_{ij}^4)$ be the perceptual hash feature vector of block B_{ij}. Then, block B_{ij} will be put into package PA_θ, where $\theta = (\lfloor P_{ij}/n \rfloor + 1)$. For example, assume $i = 6$, $j = 5$, $n = 4$, and $P_{65} = 129$. Block B_{65} will be put into package PA_{33}, where $(\lfloor P_{65}/n \rfloor + 1) = (\lfloor 129/4 \rfloor + 1) = 33$. This indicates that the pixel mean range of package PA_{33} is $\{128, 129, 130, 131\}$.

For any two image blocks B_{ij} and B_{mn}, their pixel values are similar if the two image blocks are duplicated. Naturally, their average pixel values P_{ij} and P_{mn} are also similar, where

Input: All perceptual hash feature vectors $F_{ij} = (P_{ij}, d_{ij}^1, d_{ij}^2, d_{ij}^3, d_{ij}^4)$, where $1 \le i \le (H - b + 1)$ and $1 \le j \le (W - b + 1)$, which are the output of Algorithm 1.

Output: A map that includes the detection results.

Step 1. Creating ξ packages, denoted as PA_1, $PA_2, \ldots$, and PA_ξ, where $\xi = \lfloor 256/n \rfloor + 1$ and n is a preset threshold.

Step 2. All perceptual hash feature vectors $F_{ij} = (P_{ij}, d_{ij}^1, d_{ij}^2, d_{ij}^3, d_{ij}^4)$ are stored into the ξ packages, respectively, according to the value of P_{ij}.

Step 3. A map is created with the same size of suspicious image and all its initial pixel values are set to zero.

Step 4. For each package PA_θ $(1 \le \theta \le \xi)$

Step 5. The block pairs contained in PA_θ will be matched according to their perceptual hash feature vectors and coordinate positions. The values of the corresponding coordinate positions in the map will be set to a same pixel value "255" according to the coordinates of the suspicious image if the block pairs are diagnosed as similar.

Step 6. For each block contained in package PA_θ, it will be matched with all blocks contained in package $PA_{\theta+1}$ if $1 \le \theta \le \xi - 1$ with the same method of Step 5.

Step 7. End For

Step 8. Outputting the map.

Algorithm 2: Similar region matching.

$P_{ij} = P_{mn}$ or $P_{ij} \approx P_{mn}$. Therefore, the perceptual hash feature vectors of blocks B_{ij} and B_{mn} will be stored into the same package PA_θ or two adjacent packages PA_θ and $PA_{\theta+1}$, where $\theta = (\lfloor P_{ij}/n \rfloor + 1)$ and $1 \le \theta \le \xi - 1$. Let the average pixel values of blocks B_{ij} and B_{mn} be $P_{ij} = 131$ and $P_{mn} = 132$, respectively. We have that the pixel mean range of package PA_{33} is $\{128, 129, 130, 131\}$ and the pixel mean range of package PA_{34} is $\{132, 133, 134, 135\}$. The two perceptual hash feature vectors of blocks B_{ij} and B_{mn} will be stored into the two adjacent packages. We need to match the perceptual hash feature vectors to diagnose the similar blocks in the same package and the adjacent package in the proposed similar region matching algorithm.

In Step 3, a map is created to mark the coordinate positions of all duplicated regions. It is the output in Algorithm 2. At the initial state, all of its values are set to zero. This means that there is no duplicated region at the initial state.

In Step 5, the similar image blocks that belong to the same package will be located according to their perceptual hash feature vectors and their actual coordinate distance. $\forall \theta \in \{1, 2, \ldots, \xi\}$, all perceptual hash feature vectors contained in package PA_θ will be compared with each other. Let B_{ij} and B_{mn} be two image blocks such that $B_{ij} \ne B_{mn}$, $F_{ij} = (P_{ij}, d_{ij}^1, d_{ij}^2, d_{ij}^3, d_{ij}^4)$, and $F_{mn} = (P_{mn}, d_{mn}^1, d_{mn}^2, d_{mn}^3, d_{mn}^4)$ be their perceptual hash feature vectors, respectively, and $N \ge 0$ be a preset threshold. Blocks B_{ij} and B_{mn} can be considered as similar blocks if $\forall k \in \{1, 2, 3, 4\}$ such that $S(d_{ij}^k \oplus d_{mn}^k) \le N$, where $\oplus$ is an exclusive-OR operation for binary strings d_{ij}^k and d_{mn}^k and S is a function that is used to count the number of "1" in $d_{ij}^k \oplus d_{mn}^k$.

Specially, blocks B_{ij} and B_{mn} that are diagnosed as similar blocks should be excluded if their coordinate positions are adjacent in a suspicious image since the adjacent pixels of the suspicious image are generally smooth. Therefore, the coordinate distance of the two similar blocks B_{ij} and B_{mn} should be considered. Let (x_{ij}, y_{ij}) and (x_{mn}, y_{mn}) be the coordinates of blocks B_{ij} and B_{mn}, respectively. Their actual coordinate distance, denoted as α, can be calculated as follows:

$$\alpha = \sqrt{\left(x_{ij} - x_{mn}\right)^2 + \left(y_{ij} - y_{mn}\right)^2}. \tag{5}$$

If $\alpha > M$, similar blocks B_{ij} and B_{mn} are considered as actual similar blocks, where M is a preset threshold. If blocks B_{ij} and B_{mn} are diagnosed as actual similar blocks, the values of the coordinate positions (x_{ij}, y_{ij}) and (x_{mn}, y_{mn}), which are also the coordinate positions of similar blocks B_{ij} and B_{mn} in the suspicious image, should be marked with the same value, such as 255.

In Step 6, each block that belongs to package PA_θ will be matched with all blocks that belong to package $PA_{\theta+1}$ with the same method of Step 5 if $1 \le \theta \le \xi - 1$.

Let B_{ij} and B_{mn} be two image blocks and $F_{ij} = (P_{ij}, d_{ij}^1, d_{ij}^2, d_{ij}^3, d_{ij}^4)$ and $F_{mn} = (P_{mn}, d_{mn}^1, d_{mn}^2, d_{mn}^3, d_{mn}^4)$ be the perceptual hash feature vectors of B_{ij} and B_{mn}, respectively. We have $F_{ij} \approx F_{mn}$ if $B_{ij} \approx B_{mn}$, where $F_{ij} = F_{mn}$ is a particular case of $F_{ij} \approx F_{mn}$. It can be explained as follows. For blocks B_{ij} and B_{mn}, $\forall f_{ij}(x, y) \in B_{ij}$ and $\forall f_{mn}(x, y) \in B_{mn}$ such that $f_{ij}(x, y) \approx f_{mn}(x, y)$ if $B_{ij} \approx B_{mn}$, where $f_{ij}(x, y)$ (resp., $f_{mn}(x, y)$) represents the pixel of xth row and yth column in B_{ij} (resp., B_{mn}), we have $P_{ij} \approx P_{mn}$ since they are the pixel means of B_{ij} and B_{mn}, respectively.

Let C_{ij} and C_{mn} be the DCT coefficient matrixes of B_{ij} and B_{mn}, respectively. According to the independence and stability characteristics of DCT [21], we have $C_{ij} \approx C_{mn}$ if $B_{ij} \approx B_{mn}$. Naturally, we also have $C_{ij}^k \approx C_{mn}^k$, where C_{ij}^k and C_{mn}^k are the four subblocks of C_{ij} and C_{mn}, respectively, and $k \in \{1, 2, 3, 4\}$. Let m_{ij} and m_{mn} be the mean values of C_{ij}^1 and C_{mn}^1, respectively. We have $m_{ij} \approx m_{mn}$ since $C_{ij}^1 \approx C_{mn}^1$. Therefore, we have $H_{ij}^k \approx H_{mn}^k$ according to (3), where H_{ij}^k and H_{mn}^k are the perceptual hash matrixes of subblocks C_{ij}^k and C_{mn}^k, respectively. Note that H_{ij}^k and H_{mn}^k are two binary matrixes, which contain only "1" or "0." Automatically, we have $d_{ij}^k \approx d_{mn}^k$ since the decimal numbers d_{ij}^k and d_{mn}^k are uniquely computed from the two binary matrixes H_{ij}^k and H_{mn}^k, respectively. Therefore, we have $F_{ij} \approx F_{mn}$ since $P_{ij} \approx P_{mn}$ and $d_{ij}^k \approx d_{mn}^k$, where $k \in \{1, 2, 3, 4\}$. It indicates that

image blocks B_{ij} and B_{mn} may be a similar block pair if their perceptual hash feature vectors F_{ij} and F_{mn} are similar. In order to authenticate a suspicious image, we should detect all blocks of the suspicious image by comparing the perceptual hash feature vectors of these blocks.

3. Experiment and Analysis

In this section, the performance of the proposed scheme is tested and analyzed with many suspicious images that are involved in three image databases. The first one is the Columbia photographic images and photorealistic computer graphics database [22], which is made open for passive-blind image authentication research communities. In this database, about 1200 images are involved. We used Photoshop 8.0 to tamper images. All tampered suspicious images form the first experiment database. The second database contains two datasets MICC-F2000 and MICC-F220 that are introduced by Serra in [23]. The two datasets provide 1110 original images and 1110 tampered suspicious images with copy-move forgery. The original images contain animals, plants, men, artifacts, and natural environment. Moreover, to further evaluate the performance of the proposed scheme, 200 supplemented images are downloaded from the Internet and tampered with copy-move forgery to form the third database.

3.1. Evaluation Criteria Introduction. To evaluate the performance of copy-move forgery detection methods, researchers usually consider their test results at two different levels, that is, image level and pixel level [3]. At image level, it mainly focuses on the detection of whether an image is tampered or not. Let T_c be the number of tampered images that are correctly detected, F_c be the number of images that are erroneously detected to be the tampered images, and F_m be the number of falsely missed forgery images. The precision ratio p and recall ratio r [3] can be calculated by the following formulas:

$$p = \frac{T_c}{T_c + F_c} \times 100\%, \qquad r = \frac{T_c}{T_c + F_m} \times 100\%, \tag{6}$$

where the precision ratio p denotes the probability of a detected forgery being truly a forgery and the recall ratio r denotes the probability of a forgery being not missed.

At pixel level, it is used to evaluate the accuracy of duplicated regions. Let ω_s and ω_t be the pixels of an original region and a copy-move region in a suspicious image, respectively, and $\widetilde{\omega}_s$ and $\widetilde{\omega}_t$ be the pixels of an original region and a copy-move region in a detected result image, respectively. The detection accuracy rate (DAR) and false positive rate (FPR) are calculated as follows:

$$\mathrm{DAR} = \frac{|\omega_s \cap \widetilde{\omega}_s| + |\omega_t \cap \widetilde{\omega}_t|}{|\omega_s| + |\omega_t|} \times 100\%, \qquad \mathrm{FPR} = \frac{|\widetilde{\omega}_s - \omega_s| + |\widetilde{\omega}_t - \omega_t|}{|\widetilde{\omega}_s| + |\widetilde{\omega}_t|} \times 100\%, \tag{7}$$

where "||" means the area of region, "∩" means the intersection of two regions, and "−" means the difference of two regions. In this sense, the DAR shows the proportion of identified pixels that simultaneously belong to the really duplicated regions and all really duplicated pixels in all suspicious images. The FPR shows the ratio of some identified pixels that actually do not belong to the really duplicated regions and all identified pixels in all suspicious images. The four criteria indicate how precisely the proposed schemes can locate copy-move regions. Then, we can analyze the performance of the proposed scheme at the image level and the pixel level with the four criteria.

Table 1: Comparison of precision ratio p and recall ratio r.

Methods	[6]	[7]	[9]	[P]
Precision ratio (p)	0.865	0.896	0.822	0.902
Recall ratio (r)	0.900	0.850	0.864	0.910

Table 2: Parameters for the five kinds of attacks.

Attacks	Parameters
AWGN	SNR (10, 20, 30, 40, 50)
GB	ω (0.5, 1, 1.5, 2, 2.5)
ACR	gain (G) (0.3, 0.6, 0.9, 1.2, 1,5)
AL	bias (b) (3, 6, 9, 12, 15)
AH	H (5, 10, 15, 20, 25)

3.2. Effectiveness and Accuracy. In this experiment, 400 color images are selected to test the effectiveness and accuracy of the proposed scheme, including 100 original images, 100 forgery images, and other 200 images that are tampered with Photoshop 8.0. All tampered images do not suffer any postprocessing operation. Owing to space constrains, just a part of experimental results is shown in Figure 4. The DAR and FPR are calculated to illustrate the performance of the proposed scheme. In Figure 4, the DAR is generally greater than 0.85 and the FPR is also smaller. It indicates that the duplicated regions can be detected using the proposed scheme even though the duplicated regions are nonregular. Table 1 shows the comparison result of the proposed scheme and other existing schemes that are presented in [6, 7, 9]. It indicates that the p and r in the proposed scheme are better.

3.3. Robustness Test. In addition to the plain copy-move forgery, the detection for tampered images that are attacked by some postprocessing operations is also considered in the proposed scheme. Therefore, a series of experiments have been done to overall analyze the performance of the proposed scheme. It involves 1000 different suspicious images that come from the three databases. In this experiment, five kinds of attacks are considered, that is, adding white Gaussian noises (AWGN), adjusting contrast ratio (ACR), luminance (AL), hue (AH), and Gaussian blurring (GB), and their hybrid operations. Table 2 presents the parameters for the five kinds of attacks and Figure 5 shows a part of experimental results for the proposed scheme.

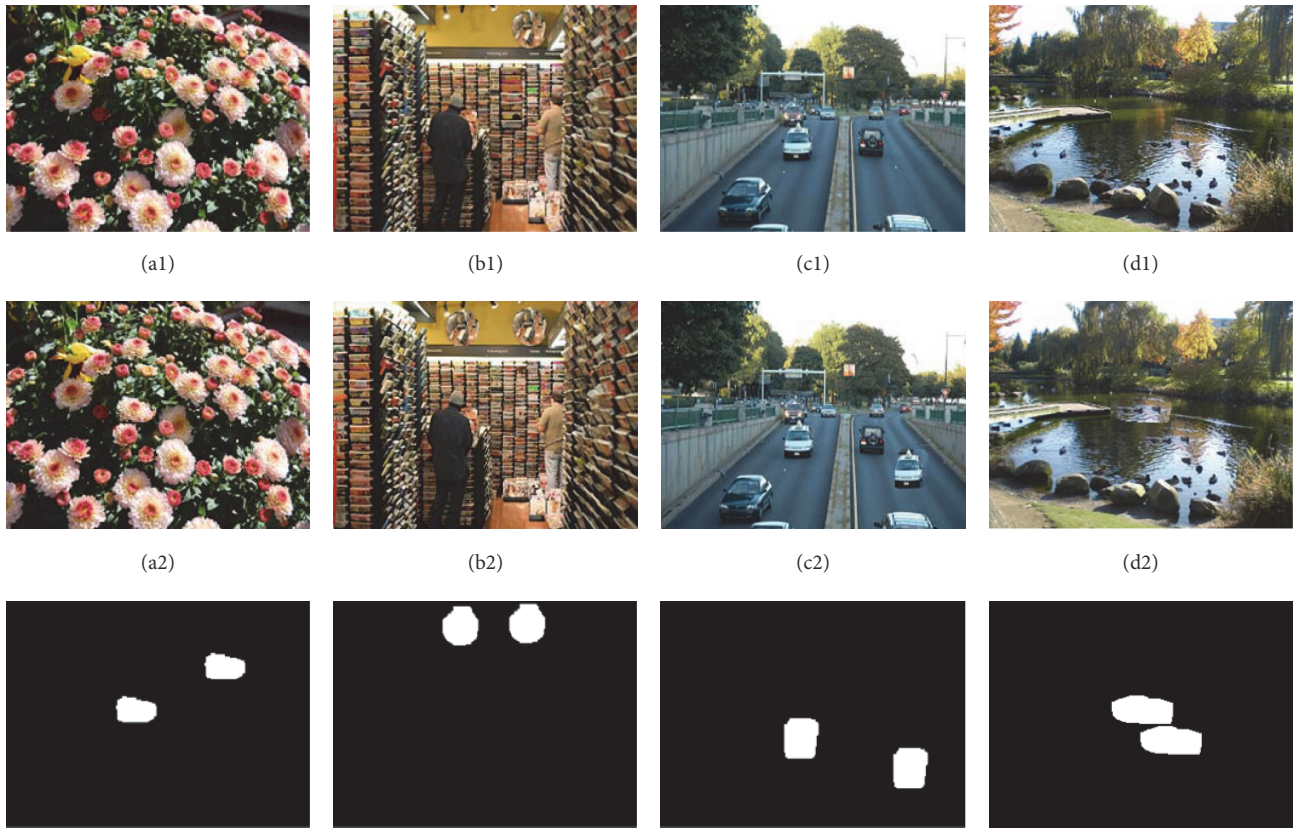


Figure 4: A part of experimental results for the proposed scheme. From top to bottom, the first row (a1)–(d1) shows the original images, the second row (a2)–(d2) shows the tampered images, and the last row (a3)–(d3) shows the detection results.

Table 3: Detection results for adding white Gaussian noise.

	SNR = 25	SNR = 35	SNR = 45
P	0.980	0.970	0.970
r	0.980	0.970	0.990
DAR	0.910	0.846	0.823
FPR	0.123	0.137	0.125

Table 4: Detection results for Gaussian blurring.

	$\omega = 1$	$\omega = 3$	$\omega = 5$
P	0.990	0.980	0.990
r	0.960	0.950	0.990
DAR	0.847	0.894	0.906
FPR	0.112	0.132	0.116

Table 5: Detection results for adjusting contrast ratio.

	$G = 0.9$	$G = 1.1$	$G = 1.4$
P	0.970	0.980	0.990
r	0.940	0.990	0.960
DAR	0.749	0.802	0.896
FPR	0.163	0.174	0.184

In this experiment, the proposed scheme is evaluated by DAR and FPR at the pixel level. The results indicate that the proposed scheme can locate multiple duplication regions although the suspicious images are attacked with different postprocessing operations.

In order to quantitatively evaluate the robustness of the proposed algorithm and analyze its ability to resist different image distortions, 100 tampered images are selected from the three databases. These tampered images are distorted by five kinds of attacks that are shown in Table 2. Then, there are 500 tampered images that will be detected in this experiment. For each kind of attacks, 100 tampered images are selected to be detected. Tables 3–7 show the detection results with the overall averages of p, r, DAR, and FPR. The robustness of the proposed scheme is evaluated at image level and pixel level.

Tables 3 and 4 show that the detection results of the proposed scheme are satisfactory for suspicious images that are attacked by adding white Gaussian noises and Gaussian blurring although the suspicious images have poor quality (SNR = 45 or $\omega = 5$). Only 14 images in all 600 tampered images are failed to be detected ($r = 0.9767$). The detection results of tampered images that are distorted by adjusting

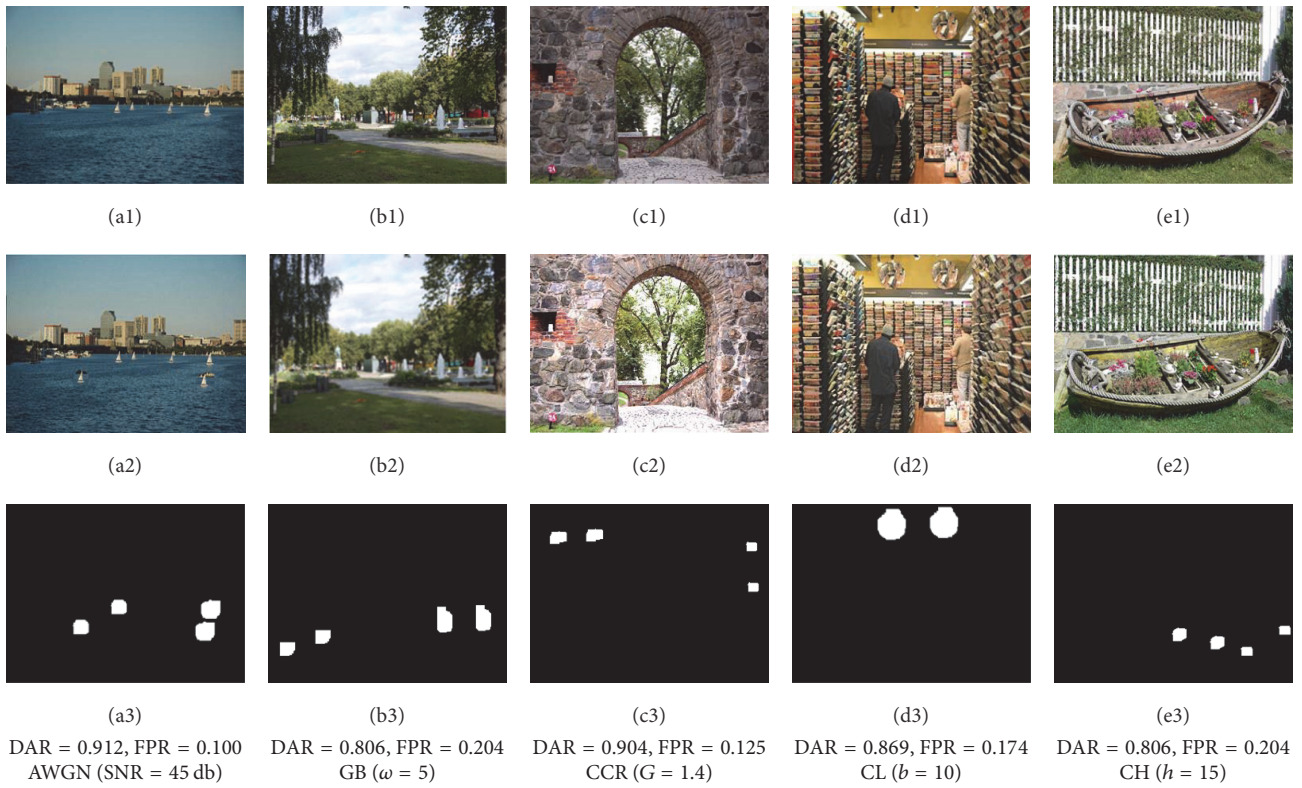


Figure 5: A part of the experimental results for the proposed scheme. From top to bottom, the first row (a1)–(e1) shows the five original images; the second row (a2)–(e2) shows the tampered images that are attacked with adding white Gaussian noises and Gaussian blurring, adjusting contrast ratio, luminance, and hue; the last row (a3)–(e3) shows the detection results of the proposed method.

Table 6: Detection results for adjusting luminance.

	$b = -3$	$b = 3$	$b = 10$
P	0.960	0.980	0.990
r	0.980	0.960	0.990
DAR	0.902	0.876	0.868
FPR	0.168	0.182	0.191

Table 7: Detection results for adjusting hue.

	$H = 10$	$H = 20$	$H = 30$
P	0.980	0.990	0.990
r	0.990	0.990	0.970
DAR	0.759	0.782	0.826
FPR	0.145	0.139	0.136

contrast ratio, luminance, and hue with different parameters are shown in Tables 5, 6, and 7, respectively. We can draw a conclusion from the three tables that the proposed scheme performs well also for attacks of adjusting contrast ratio, luminance, and hue.

3.4. Performances Comparison. In the last experiment, the performance of the proposed scheme is compared with other schemes presented in [6, 7, 9]. In this experiment, 400 tampered images are randomly selected from the three databases. They are tested by the proposed scheme and other schemes provided in [6, 7, 9], respectively. Figure 6 shows the performance comparison of these schemes with the overall averages of DAR and FPR for the 400 tampered images. We can see that the scheme proposed in [9] has the best detection results for the two kinds of attacks by adding white Gaussian noises and Gaussian blurring. However, its performance clearly drops down if the intensity of these attacks is gradually increased. Conversely, the proposed scheme is more robust for resisting various attacks. In most cases, the proposed scheme can also achieve better results for other three kinds of attacks with adjusting contrast ratio, luminance, and hue. Moreover, the proposed scheme has the lowest FPR results, which means that the proposed scheme can detect most duplicated regions in the selected suspicious images. The precision of the proposed scheme is higher than that obtained in [6, 7, 9].

The experimental results show that the proposed method can locate the tampered regions in a tampered image although it is distorted by some hybrid trace hiding operations, such as adding white Gaussian noise, Gaussian blurring, adjusting contrast ratio, luminance, and hue, and their hybrid operations. The proposed forensic technique can be used in politics, military, jurisprudence, and academic research. For example, a journalist takes a photo for a traffic accident. However, the journalist finds that the influence will be better if some crowds appear in this photo. Therefore, he

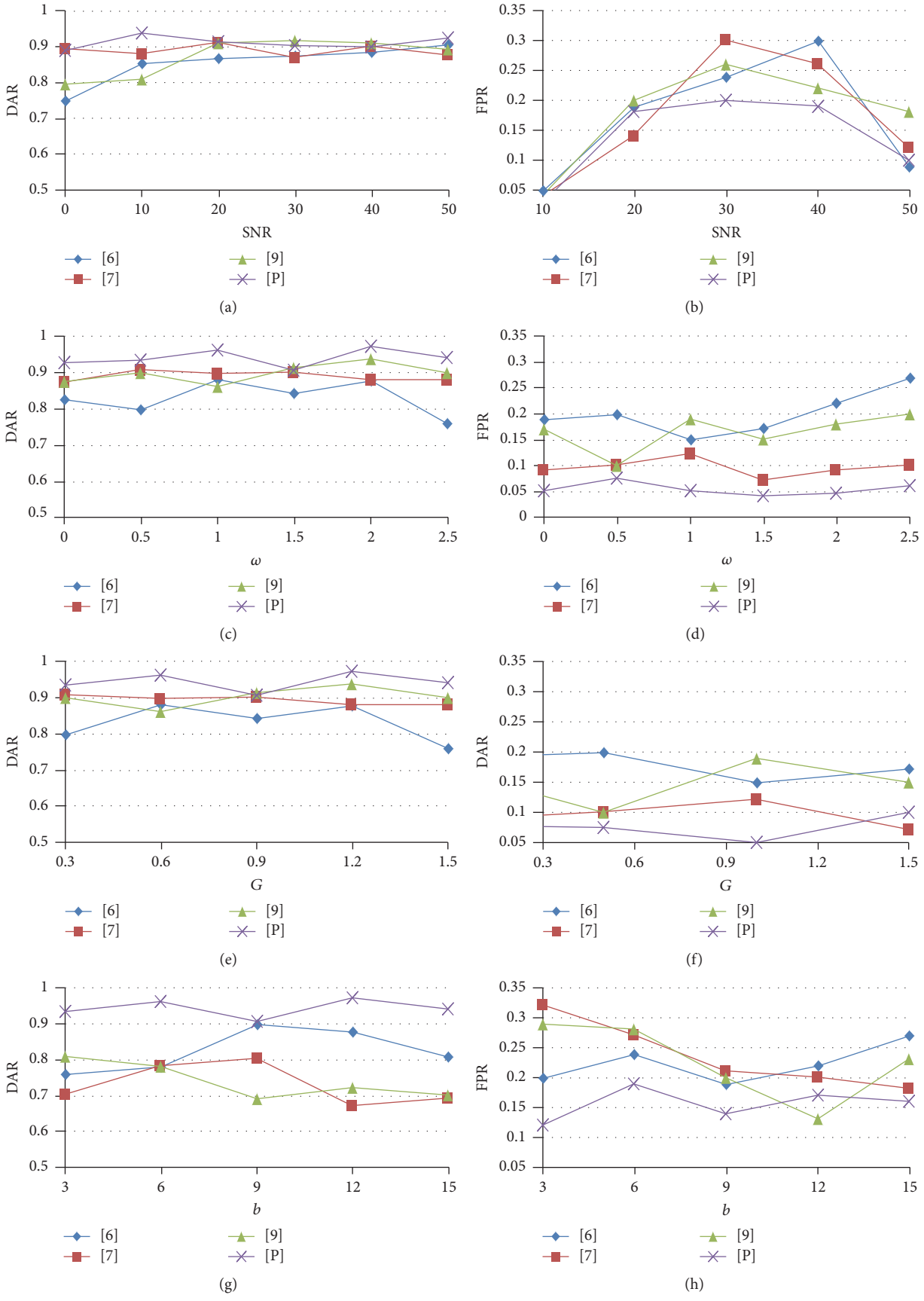


Figure 6: Continued.

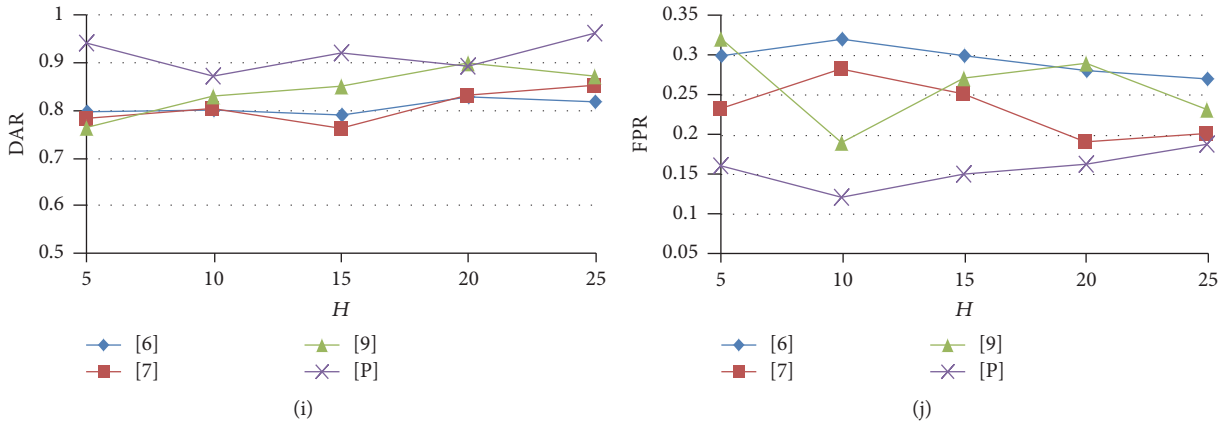


Figure 6: Comparison of different copy-move forgery schemes with 5 kinds of attacks. The two columns represent the result of DAR and FPR, respectively. (a), (b) Adding white Gaussian noise. (c), (d) Gaussian blurring. (e), (f) Adjusting contrast ratio. (g), (h) Adjusting luminance. (i), (j) Adjusting hue.

can use image processing tools to copy some people from the other side of this photo and paste them into the scene and use white Gaussian noise to conceal all tampering traces. Therefore, the authenticity of this traffic accident is broken. The news organization can detect this photo by using the proposed scheme to ensure its authenticity before this news is reported.

4. Conclusion

In this study, a passive authentication scheme is proposed based on perceptual hashing and package clustering algorithms to detect and locate the duplicated regions for copy-move forgery. The experiment results show that the proposed scheme based on perceptual hashing algorithms is robust for some special attacks, such as adjusting contrast ratio, luminance, and hue. A technology application of using perceptual hash strings to construct a feature vector to represent an image block can resist some conventional attacks, such as adding white Gaussian noises and Gaussian blurring. The proposed package clustering algorithm that is used to replace traditional lexicographic order algorithms can improve the performance of the proposed scheme. The evaluation criteria p, r, DAR, and FPR from the experiments show that the proposed scheme is better but the proposed scheme also has some weaknesses. For example, the time complexity is still unsatisfactory because of the previous image block dividing. Furthermore, the proposed scheme cannot resist some complex attacks, such as block rotation and scaling. In future work, we will focus on the studies of improving the time complexity and extending the robustness for more kinds of complex attacks, such as the rotation and scaling.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) under Grant no. U1536110.

References

[1] J. Fridrich, D. Soukalm, and J. Lukas, "Detection of copy-move forgery in digital images," *Digital Forensic Research Workshop*, pp. 19–23, 2003.

[2] L. Kang and X. Cheng, "Copy-move forgery detection in digital image," in *Proceedings of the 3rd International Congress on Image and Signal Processing (CISP '10)*, pp. 2419–2421, Yantai, China, October 2010.

[3] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou, "An evaluation of popular copy-move forgery detection approaches," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 6, pp. 1841–1854, 2012.

[4] A. C. Popscu and H. Farid, "Exposing digital forgeries by detecting duplicated image regions," Tech. Rep. TR2003-515, Dartmouth College, 2004.

[5] M. Babak and S. Stanislav, "Detection of copyCmove forgery using a method based on blur moment invariants," *IEEE Transactions on Information Forensics*, vol. 10, no. 3, pp. 507–518, 2007.

[6] Y. Cao, T. Gao, L. Fan, and Q. Yang, "A robust detection algorithm fosr region duplication in digital images," *International Journal of Digital Content Technology and its Applications*, vol. 5, no. 6, pp. 95–103, 2011.

[7] S. A.-N. Thajeel and G. Sulong, "A novel approach for detection of copy move forgery using completed robust local binary pattern," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 6, no. 2, pp. 351–364, 2015.

[8] H. Wang, H.-X. Wang, X.-M. Sun, and Q. Qian, "A passive authentication scheme for copy-move forgery based on package clustering algorithm," *Multimedia Tools and Applications*, vol. 76, no. 10, pp. 12627–12644, 2017.

[9] J. Zhong, Y. Gan, J. Young, L. Huang, and P. Lin, "A new block-based method for copy move forgery detection under image geometric transforms," *Multimedia Tools and Applications*, vol. 76, no. 13, pp. 14887–14903, 2017.

[10] R. Dixit, R. Naskar, and S. Mishra, "Blur-invariant copy-move forgery detection technique with improved detection accuracy utilising SWT-SVD," *IET Image Processing*, vol. 11, no. 5, pp. 301–309, 2017.

[11] X. L. Bi and C. M. Pun, "Fast reflective offset-guided searching method for copy-move forgery detection," *Information Sciences*, pp. 531–545, 2017.

[12] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra, "A SIFT-based forensic method for copy-move attack detection and transformation recovery," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1099–1110, 2011.

[13] I. Amerini, M. Barni, R. Caldelli, and A. Costanzo, "Counter-forensics of SIFT-based copy-move detection by means of keypoint classification," *EURASIP Journal on Image and Video Processing*, pp. 1–17, 2013.

[14] J. Li, X. Li, B. Yang, and X. Sun, "Segmentation-based image copy-move forgery detection scheme," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 507–518, 2015.

[15] X. Wang, G. He, C. Tang, Y. Han, and S. Wang, "Keypoints-Based Image Passive Forensics Method for Copy-Move Attacks," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 30, no. 3, Article ID 1655008, 2016.

[16] J. Li, F. Yang, W. Lu, and W. Sun, "Keypoint-based copy-move detection scheme by adopting MSCRs and improved feature matching," *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 20483–20497, 2017.

[17] X.-Y. Wang, S. Li, Y.-N. Liu, Y. Niu, H.-Y. Yang, and Z.-L. Zhou, "A new keypoint-based copy-move forgery detection for small smooth regions," *Multimedia Tools and Applications*, vol. 76, no. 22, pp. 23353–23382, 2017.

[18] X. M. Niu and Y. H. Jiao, "An overview of perceptual hashing," *ACTA Electronica Siniica*, vol. 36, no. 7, pp. 1405–1411, 2008.

[19] W. Johannes, "Detecting visual plagiarism with perception hashing," *Degree Project in Computer science*, 2015.

[20] Z. Yin, X. Niu, Z. Zhou, J. Tang, and B. Luo, "Improved Reversible Image Authentication Scheme," *Cognitive Computation*, vol. 8, no. 5, pp. 890–899, 2016.

[21] E. Y. Lam and J. W. Goodman, "A mathematical analysis of the DCT coefficient distributions for images," *IEEE Transactions on Image Processing*, vol. 9, no. 10, pp. 1661–1666, 2000.

[22] T. T. Ng, S. F. Chang, J. Hsu, and M. Pepeljugoski, Columbia photographic images and photorealistic computer graphics dataset ADVENT, Columbia University, New York, NY, USA, 2004.

[23] G. Serra, "A SIFT-based forensic method for copy-move detection. Giuseppe Serra 2014," http://giuseppeserra.com/content/sift-based-forensic-method-copy-move-detection.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 1214681, 15 pages
https://doi.org/10.1155/2018/1214681

Research Article

A Wireless Covert Channel Based on Constellation Shaping Modulation

Pengcheng Cao,[1] Weiwei Liu,[1] Guangjie Liu,[1] Xiaopeng Ji,[1] Jiangtao Zhai,[2] and Yuewei Dai[2]

[1] *The School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China*
[2] *The School of Electrical and Computer Engineering, Jiangsu University of Science and Technology, Zhenjiang 212003, China*

Correspondence should be addressed to Guangjie Liu; gjieliu@njust.edu.cn

Received 29 September 2017; Accepted 4 December 2017; Published 8 January 2018

Academic Editor: Rémi Cogranne

Wireless covert channel is an emerging covert communication technique which conceals the very existence of secret information in wireless signal including GSM, CDMA, and LTE. The secret message bits are always modulated into artificial noise superposed with cover signal, which is then demodulated with the shared codebook at the receiver. In this paper, we first extend the traditional KS test and regularity test in covert timing channel detection into wireless covert channel, which can be used to reveal the very existence of secret data in wireless covert channel from the aspect of multiorder statistics. In order to improve the undetectability, a wireless covert channel for OFDM-based communication system based on constellation shaping modulation is proposed, which generates additional constellation points around the standard points in normal constellations. The carrier signal is then modulated with the dirty constellation and the secret message bits are represented by the selection mode of the additional constellation points; shaping modulation is employed to keep the distribution of constellation errors unchanged. Experimental results show that the proposed wireless covert channel scheme can resist various statistical detections. The communication reliability under typical interference is also proved.

1. Introduction

Covert channel is a specific application of data hiding with the requirement that the hidden secret data is undetectable. It is always classified by the employed carrier; the most popular type of covert channels is network covert channels, which is based on network traffic; information is embedded by manipulating the packet timing information [1–4] or pudding some bits into the packet headers [5]. As the youngest branch of covert channels, wireless covert channel conceals the very existence of secret information by modulating it into the delivered wireless signal [6] or modifying some redundant fields of wireless communication protocols [7]. Wireless covert channels have received increasing attentions because the ubiquitous nature of wireless devices and their localized transmission make it difficult to detect their presence.

Several kinds of wireless covert channels have been proposed [6–16]. In [7, 13], the secret data is embedded in the redundant fields of wireless communication protocols such as padding of frames, headers of the MAC, RLC, and PDCP. In [8], the subcarriers in OFDM-based system which are reserved for channel spacing or synchronization of sender and receiver and to mitigate poor channel response are used to transmit the secret data. The covert transmission in the unused subcarriers has little effect on the normal information transmission. In [12], several wireless covert channels are introduced with the secret data embedded in the phase of short training field, the frequency of long training field, and cyclic prefix in WiFi system. In [14, 15], the secret data are transmitted by covert relay on top of the cover data in wireless relay networks. In [16], the wireless covert channels are presented based on the coordinated operations in the control channel and data channel of MIMO system. These kinds of wireless covert channels are effective but applicable to the specific wireless communication system.

In OFDM-based wireless communication, the deviations of the received signal from the ideal signal which can be called constellation errors are found to widely exist due to channel

impairments and hardware impairments. So the secret data can be modulated into signal similar to constellation errors to resist the detection. In [9], the artificial noise signal generated by secret data is added to the cover signal directly. In this scheme, the spread spectrum technique is applied in the artificial noise signal so that the artificial noise signal has little influence on the transmission of cover signal. The informed receiver can extract the secret data by removing the cover signal. This covert channel is generalized to the MIMO system later [11]. However, those covert channels are easily interfered by wireless channel noise.

Recently, a wireless covert channel based on dirty constellation is proposed which modulates the secret information bits into constellation errors around the normal constellation points [6]. The additional constellation points are added into the original constellation which will be used for modulating instead of the original points. The secret information bits are represented by the selection of these additional constellation points; for example, four additional constellation points can be used to represent two secret message bits. The generation mode of these additional constellation points can be controlled by a shared secret key. However, when the detector is near the sender, the regularity of dirty constellation may result in the poor resistance to some statistical detection [17].

In this paper, the wireless covert channel with constellation shaping modulation is proposed. The model of constellation error is used for the design of dirty constellation. For each subcarrier of OFDM, the distribution of the in-phase and quadrature (I/Q) vectors of the constellation error is calculated with the constellation at the normal receiver; the secret information is modulated into artificial noise that distributes as the real channel noise. Compared with the existing dirty constellation scheme, the undetectability and reliability can both be improved. In addition, the mapping sequence is unnecessary to be synchronized in the proposed scheme.

This paper is organized as follows. In the next section, some background and related works including wireless covert channel with dirty constellation are introduced. In Section 3, some typical detection schemes in the field of covert timing channels are developed into wireless covert channels. In Section 4, we describe the proposed wireless covert channel scheme based on constellation shaping modulation. Section 5 gives the experimental results on undetectability and reliability. Finally, Section 6 concludes the whole paper.

2. Wireless Covert Channel with Dirty Constellation

In the wireless covert channels with dirty constellation (WCC-DC), the secret message bits can be transmitted as the constellation error of the normal cover signal in order to reduce the suspicion by all uninformed observers. The wireless covert channel relies on that the cover message bits are transmitted at a low rate (BPSK or QPSK) with supplemental redundancy that can be utilized as an additional QPSK signal by an informed receiver.

With the example of QPSK, the process at the sender in the wireless covert channel with dirty constellation is demonstrated in Figure 1. First, the mapping sequence bits

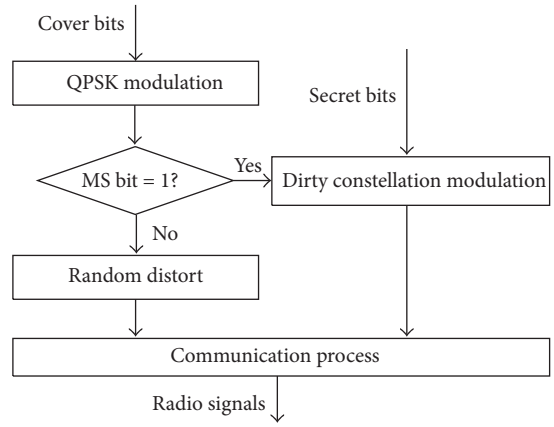


Figure 1: Process at the sender in WCC-DC.

are checked after the cover message bits are modulated by QPSK constellation. The mapping sequence bits are used to select the appropriate mapping for covert and noncovert subcarriers. For example, the mapping sequence bit can be set to "1" if corresponding cover signal of the subcarrier is chosen to embed the secret message bits by dirty constellation, and the mapping sequence bit is set to "0" if corresponding cover signal of the subcarrier is just random distorted to mimic the influence of various interferences. The embedding rate of the wireless covert channels depends on the proportion of the mapping sequence length for the number of "1." The mapping sequence bits must be shared between the sender and informed receiver. Then the covert signal and noncovert signal are blended for transmission. With some communication processes such as IFFT, adding cyclic prefix and so on, the covert signal is transmitted as radio signal. At the informed receiver, the covert signal is picked out by the mapping sequence bits and the secret message bits are extracted by the corresponding demodulation of dirty constellation. Even when an adversary has access to the I/Q vectors of the covert signal, they will interpret the point cloud as a noisy version of a valid QPSK constellation and would not suspect the presence of a covert channel.

The dirty constellation based on QPSK is shown in Figure 2(a). In this paper, normal constellation point denotes the ideal constellation point of cover message bits, covert constellation point denotes the corresponding constellation point of the secret message bits around normal constellation points, and the cover signal denotes the corresponding signal of the ideal constellation point of cover message bits. The covert signal denotes the signal of secret constellation point which contains both cover and secret message bits. I vector and Q vector denote the components in I-plane and Q-plane of the constellation.

To generate the covert signal, the covert constellation points are located around the ideal QPSK constellation points of the cover signal. To modulate a covert subcarrier carrying the cover and covert bits together, the cover constellation point is first chosen, specifying the quadrant, followed by remapping that point to one of the four covert QPSK points around the chosen cover QPSK point. Then the dirty

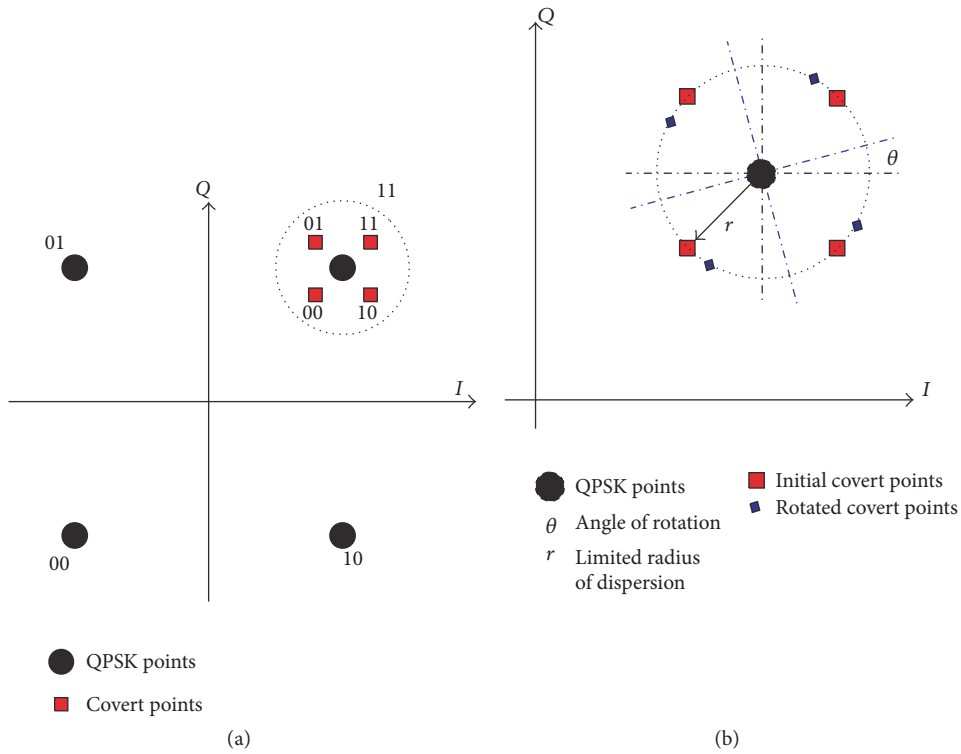


Figure 2: Dirty constellation: (a) covert QPSK constellation and (b) covert constellation.

constellation is further improved to reduce the probability of detection by adversaries. The covert points are put closer to the ideal QPSK point and remapped symmetrically around the QPSK points with a mutual separation of $2/\sqrt{42}$, which is a distance equal to that of a 64 QAM constellation. The I/Q vectors of the covert constellation points are randomized with a Gaussian distribution but limit their dispersion to a radius of $\sqrt{2/42}$. The limitation of dispersion ensures that the covert constellation points are hidden in the cloud of a dispersed (noisy) QPSK point cloud. To have the covert symbols blend with the cloud of the random distorted QPSK points, the covert constellation points are rotated round the corresponding ideal QPSK point for every subcarrier that is mapped to a covert constellation as shown in Figure 2(b). The rotation is performed using a monotonically increasing angle θ; the sender and receiver both start with $\theta = 0^\circ$ at the start of the packet and increment θ for each covert subcarrier.

Even if the dirty constellation is improved, there is always a finite probability that the covert constellations are visible with all subcarriers transmitting covert signal. So only a part of subcarriers are chosen to transmit the covert signal. In order to avoid sudden changes in the modulation characteristics, the cover signal transmitted by noncovert subcarriers should always be distorted to mimic the influence of practical channels.

3. Countermeasures for Wireless Covert Channel Detection

As a covert communication technique aiming to deliver secret data via public wireless channel, the wireless covert channels should be secure against various detections. In other words, the covert transmissions in wireless channels have to be indistinguishable from normal transmissions. However, to the best of our knowledge, there still exist no specialized works concerning the detection of wireless covert channels. In the field of signal analysis, the frequency spectrum is always used to measure the difference between two signals. In [6], error vector magnitude (EVM) of constellations, peak to average power ratio (PAPR), and temporal variation of average signal power are used to measure the signal distortion in wireless covert communication. In fact, statistic-based detection should also be developed like covert timing channels [18–20]. In this paper, we propose Kolmogorov-Smirnov (KS) test [18] and regularity test [19] in wireless covert channel, which can be used to measure the difference between cover and covert signal from the aspect of frequency and regularity characteristic. The KS test is a shape test and the regularity test is a high order statistic-based test.

3.1. Kolmogorov-Smirnov Test in Wireless Covert Channel. In an OFDM system, we select the I/Q vectors and magnitudes in constellation of all the subcarriers as the detection objects, which can be captured by vector-signal analyzers or software defined radios. Denote $\mathbf{s}_a$ and $\mathbf{s}_b$ as the normal wireless signal and target wireless signal, respectively. The I/Q vectors and magnitudes corresponding to $\mathbf{s}_a$ are denoted by $\mathbf{a}^I = (a_1^I, \ldots, a_n^I)$, $\mathbf{a}^Q = (a_1^Q, \ldots, a_n^Q)$, and $\mathbf{a}^M = (a_1^M, \ldots, a_n^M)$, respectively. And the I/Q vectors and magnitudes corresponding to $\mathbf{s}_b$ are denoted by $\mathbf{b}^I = (b_1^I, \ldots, b_n^I)$, $\mathbf{b}^Q = (b_1^Q, \ldots, b_n^Q)$, and $\mathbf{b}^M = (b_1^M, \ldots, b_n^M)$, respectively.

The KS test statistic measures the maximum distance between distribution of $\mathbf{a}^{\delta}$ and $\mathbf{b}^{\delta}$ with $\delta \in \{I, Q, M\}$ to determine whether or not the distribution of $\mathbf{b}^{\delta}$ differs from that of $\mathbf{a}^{\delta}$. The histogram of the elements of $\mathbf{a}^{\delta}$ and $\mathbf{b}^{\delta}$ is made into K bins which are denoted by $B_1^{\delta}, \ldots, B_K^{\delta}$. The number of the elements of $\mathbf{a}^{\delta}$ and $\mathbf{b}^{\delta}$ in B_i^{δ} is denoted by $H_a^{\delta}(i)$ and $H_b^{\delta}(i)$ with $i \in \{1, 2, \ldots, K\}$, respectively. The cumulative distribution functions of $\mathbf{a}^{\delta}$ and $\mathbf{b}^{\delta}$ in B_i^{δ} are defined by

$$F_a^{\delta}(i) = \frac{\sum_{\alpha=1}^{i} H_a^{\delta}(\alpha)}{n}, \qquad F_b^{\delta}(i) = \frac{\sum_{\alpha=1}^{i} H_b^{\delta}(\alpha)}{n}. \tag{1}$$

The Kolmogorov-Smirnov distance between $\mathbf{a}^{\delta}$ and $\mathbf{b}^{\delta}$ is defined as

$$\text{KSTEST} = \max \frac{\left|\sum_{\alpha=1}^{i} H_b^{\delta}(\alpha) - \sum_{\alpha=1}^{i} H_b^{\delta}(\alpha)\right|}{n}, \quad \forall i \in \{1, 2, \ldots K\}. \tag{2}$$

3.2. Regularity Test in Wireless Covert Channel. For the regularity test in wireless covert channel, $\mathbf{a}^{\delta}$ and $\mathbf{b}^{\delta}$ with $\delta \in \{I, Q, M\}$ are divided into n/w sets $(\mathbf{a}^{\delta}_{\text{sub-1}}, \ldots, \mathbf{a}^{\delta}_{\text{sub-}n/w})$ and $(\mathbf{b}^{\delta}_{\text{sub-1}}, \ldots, \mathbf{b}^{\delta}_{\text{sub-}n/w})$, respectively. Each set contains w elements, respectively, which are denoted by $(\mathbf{a}^{\delta}_{\text{sub-1}}, \ldots, \mathbf{a}^{\delta}_{\text{sub-}n/w})$ and $(\mathbf{b}^{\delta}_{\text{sub-1}}, \ldots, \mathbf{b}^{\delta}_{\text{sub-}n/w})$ where $\mathbf{a}^{\delta}_{\text{sub-}i} = (a^{\delta}_{(i-1)\cdot w+1}, \ldots, a^{\delta}_{i\cdot w})$ and $\mathbf{b}^{\delta}_{\text{sub-}i} = (b^{\delta}_{(i-1)\cdot w+1}, \ldots, b^{\delta}_{i\cdot w})$. Then, for each set in $\mathbf{a}^{\delta}$, the standard deviation of the set $\mathbf{a}^{\delta}_{\text{sub-}i}$ is computed as

$$\sigma_i^{\delta} = \text{STDEV}\left(a^{\delta}_{(i-1)\cdot w+k},\ \forall k \in \{1, 2, \ldots w\}\right). \tag{3}$$

The regularity of $\mathbf{a}^{\delta}$ is the standard deviation of the pairwise differences between each σ_i^{δ} and σ_j^{δ} for all sets with $i < j$.

$$\text{regularity}_a^{\delta} = \text{STDEV}\left(\frac{\left|\sigma_i^{\delta} - \sigma_j^{\delta}\right|}{\sigma_i^{\delta}},\ i < j,\ \forall i, j\right). \tag{4}$$

The regularity of $\mathbf{b}^{\delta}$ which is denoted by $\text{regularity}_b^{\delta}$ can be obtained with (4). The regularity test in wireless covert channel determines whether or not the regularity of $\mathbf{a}^{\delta}$ and $\mathbf{b}^{\delta}$ is different. If the $|\text{regularity}_b^{\delta} - \text{regularity}_a^{\delta}| > \Delta$, the target signal $\mathbf{s}_b$ is determined to be the covert one.

3.3. Adversary Model. The KS test and regularity test in wireless covert channels require the samples for reference which can be captured from the normal constellation errors of the received signal. In practice, the transmitted signal in OFDM-based wireless communication is easily interfered by channel fading and noise. It is very hard to model the characteristics of the constellation errors of the received signal as it varies greatly with the power of the noise changing. Thus the referred normal constellation errors of the received signal on different channel conditions are required for the practical detection of the wireless covert channels. In this paper, we assume that the detector has the knowledge of the wireless covert channel schemes and has access to normal constellation errors of the received signal on different channel conditions. When the detector captures the signal, he can get the appropriate referenced signal for detection.

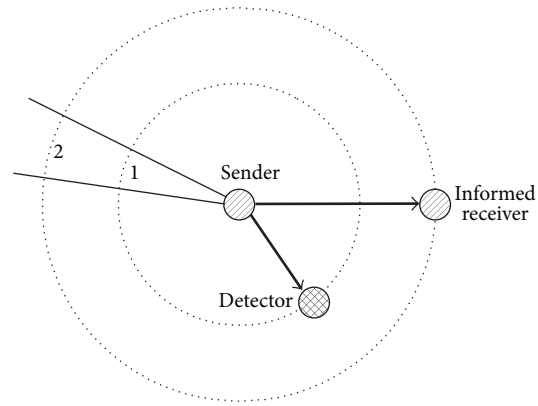


Figure 3: Adversary model on the condition that the detector is closer to the sender.

In [6], it is assumed that the informed receiver and the detector are located at the same place. However, the wireless channel conditions of the informed receiver and the detector are always different in real world. The detector may set more than one signal analyzer in some area. Due to the broadcast nature of the radio signal' transmission, one of the detector's signal analyzers may receive the radio signal with less interference and higher transmission SNR than the informed receiver which is illustrated in Figure 3. The undetectability of a wireless covert channel should be benchmarked under noisy channel with a range of noise SNRs. If the wireless covert channel can achieve well performance on detection under noisy channel with higher SNR, it can be proved undetectable and vice versa.

4. Wireless Covert Channel with Constellation Shaping Modulation

In this section, we propose a wireless covert channel with constellation shaping modulation (WCC-CSM). Its general framework for OFDM-based wireless communication system is demonstrated in Figure 4. In this paper, we assume that the common QPSK is the modulation scheme for each subcarrier of the OFDM-based wireless communication. All subcarrier can be used to establish wireless covert channel in the proposed scheme. The secret message bits are modulated into artificial noise signal by constellation shaping modulation. In every subcarrier, the artificial noise signal is added to the cover signal to generate the covert signal. The detailed description of the framework is given as follows.

In the framework, the cover data bits $\mathbf{m}_c$ are first modulated into the cover signal $\mathbf{s}_c$ by QPSK. With the distribution of the referred normal constellation errors $\mathbf{s}_{\text{normal}}$, the secret message bits $\mathbf{m}_s$ are modulated into the artificial noise $\mathbf{s}_s$

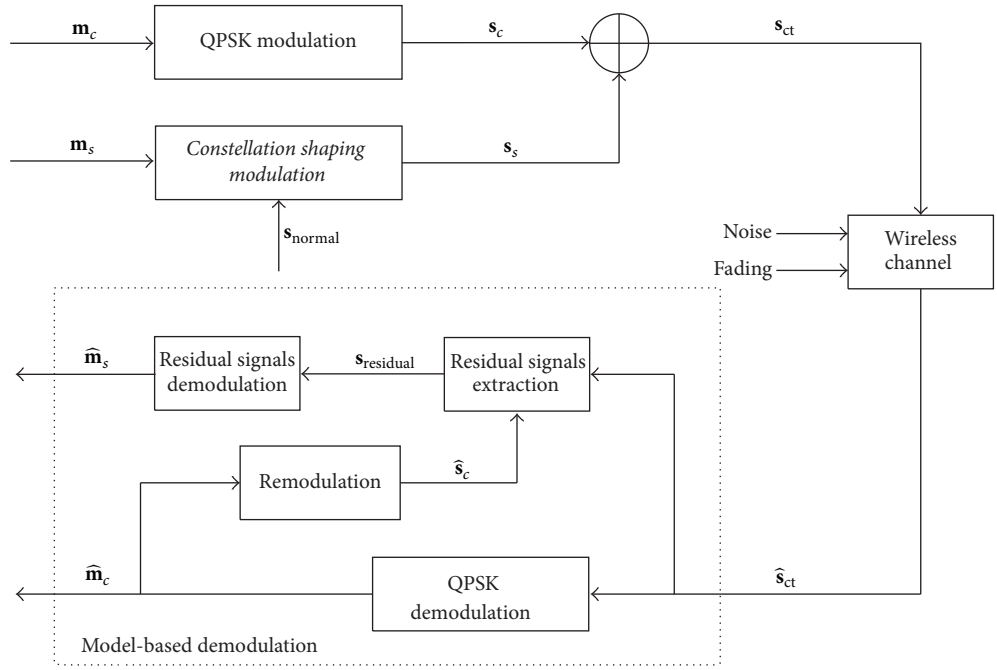


Figure 4: The framework of proposed wireless covert channel.

with constellation shaping modulation. The covert signal $\mathbf{s}_{ct}$ is generated by adding artificial noise signal $\mathbf{s}_s$ to the cover signal $\mathbf{s}_c$. The covert signal $\mathbf{s}_{ct}$ is translated into radio signal by a series of wireless communication processes, for example, IFFT and adding cyclic prefix, which are omitted in the framework. The receiver captures the covert signal $\widehat{\mathbf{s}}_{ct}$, which is the noisy version of the covert signal $\mathbf{s}_{ct}$ under the channel of various interferences. Then the received cover message bits $\widehat{\mathbf{m}}_c$ are demodulated by QPSK, and the residual signal $\mathbf{s}_{residual}$ are extracted by removing the reconstructed cover signal $\widehat{\mathbf{s}}_c$ from $\widehat{\mathbf{s}}_{ct}$. At last, the secret message bits $\widehat{\mathbf{m}}_s$ can be demodulated from $\mathbf{s}_{residual}$.

As shown in Figure 4, the proposed framework consists of constellation shaping modulation, the model-based demodulation. These are described in detail in the following subsections.

4.1. Constellation Shaping Modulation. The constellation shaping modulation is employed to generate the artificial noise signal $\mathbf{s}_s$ with secret message bits $\mathbf{m}_s$ embedded. The normal constellation errors $\mathbf{s}_{normal}$ are first divided into a certain amount of bins to capture the information about the distribution. Since the I vectors and Q vectors in constellation are orthogonal, the parameters of the distribution of $\mathbf{s}_{normal}$ can be obtained in different planes, respectively. Then, with the parameters of the distribution in each plane, the secret message bits $\mathbf{m}_s$ can be modulated into the I/Q vectors of artificial noise signal $\mathbf{s}_s$. The distribution of the I/Q vectors of $\mathbf{s}_s$ is kept the same as that of normal constellation errors $\mathbf{s}_{normal}$.

The secret message bits are denoted by $\mathbf{m}_s = (m_{s1}, \ldots, m_{sn})$. The I/Q vectors of artificial noise signal $\mathbf{s}_s$ are denoted by $\mathbf{x}_s^I + j \cdot \mathbf{x}_s^Q$. Here $\mathbf{x}_s^I, \mathbf{x}_s^Q$ are the I/Q vectors of the artificial noise signal at the sender satisfying $\mathbf{x}_s^I = (x_{s1}^I, \ldots, x_{sn}^I)$, $\mathbf{x}_s^Q = (x_{s1}^Q, \ldots, x_{sn}^Q)$. The constellation shaping modulation function is defined as

$$F_{\mathrm{CMS}}(\mathbf{m}_s) = \mathbf{x}_s^I + j \cdot \mathbf{x}_s^Q. \tag{5}$$

The I/Q vectors of the normal constellation errors captured from the actual communication are input to constellation shaping modulation for binning, which are denoted by $\mathbf{x}_{\mathrm{normal}}^I + j \cdot \mathbf{x}_{\mathrm{normal}}^Q$ when $\mathbf{x}_{\mathrm{normal}}^I = (x_{\mathrm{normal},1}^I, \ldots, x_{\mathrm{normal},N}^I)$, $\mathbf{x}_{\mathrm{normal}}^Q = (x_{\mathrm{normal},1}^Q, \ldots, x_{\mathrm{normal},N}^Q)$. Take I plane, for example, the histograms of $\mathbf{x}_{\mathrm{normal}}^I$ are divided by bins $[B_{L,1}, B_{U,1}]$, $\ldots$, $[B_{L,L}, B_{U,L}]$, where $B_{L,i}$ and $B_{U,i}$ are the low bound and up bound of the ith bins and the bounds satisfying $B_{L,i} = B_{U,i-1}$. The number of $\mathbf{x}_{\mathrm{normal}}^I$ in $[B_{L,i}, B_{U,i}]$ is denoted by $H_{\mathrm{normal}}^I(i)$. In this paper, the bins are divided with equiprobable area; it can be written as

$$H_{\mathrm{normal}}^I(i) = \frac{N}{L}, \quad \forall i \in \{1, \ldots, L\}. \tag{6}$$

The example of histogram of $\mathbf{x}_{\mathrm{normal}}^I$ with equal bins under Gaussian noise is illustrated in Figure 5. While the bins have different widths, the total area of each bin is equal.

It is assumed that there are two secret message bits embedded in one artificial noise signal in a subcarrier, so the element in $\mathbf{m}_s$ can be further written as $m_{si} = (m_{si,1}, m_{si,2}) \in \{00, 01, 11, 10\}$, $i = 1, \ldots, n$. The bit $m_{si,1}$ can be embedded in corresponding I vector x_{si}^I, and the bit $m_{si,2}$ can be embedded in corresponding Q vector x_{si}^Q. the center line α in I plane is obtained as the boundary $B_{U,L/2}$ or $B_{L,L/2+1}$.

$$\alpha = B_{L,L/2+1} = B_{U,L/2}. \tag{7}$$

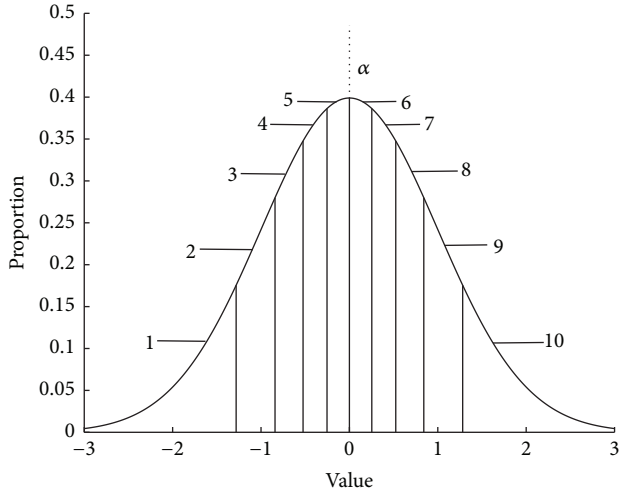


Figure 5: The equiprobable bins in I plane with $L = 10$.

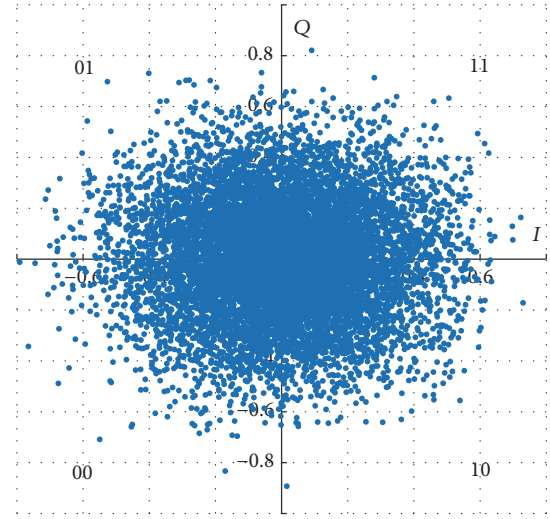


Figure 6: Extracted modulation error in constellation.

The corresponding center value of each bin are obtained as $\mathbf{c} = (c_1, c_2, \ldots, c_L)$ satisfying

$$c_i = \frac{1}{2}\left(B_{L,i} + B_{U,i}\right), \quad \forall i \in \{1, 2, \ldots, L\}. \tag{8}$$

The bit $m_{si,1}$ can be modulated into the corresponding I vector x_{si}^I according to the below equation.

$$x_{si}^I = \begin{cases} c_i, & m_{si,1} = 0,\ R \cdot \dfrac{N}{2} \in \left[\dfrac{N}{L} \cdot (i-1), \dfrac{N}{L} \cdot i\right], \\ c_j, & m_{si,1} = 1,\ (R+1) \cdot \dfrac{N}{2} \in \left[\dfrac{N}{L} \cdot (j-1), \dfrac{N}{L} \cdot j\right]. \end{cases} \tag{9}$$

Here, R is a random number with uniform distribution on $[0, 1]$. Equation (9) keeps that the distribution of the regenerated I vectors is the same as that of the normal constellation errors in the histogram with L bins meanings. The center lines α of histogram of $\mathbf{x}_{\text{normal}}^I$ should be shared with the informed receiver. The modulation in Q plane works in the same way. As shown in Figure 6, the I/Q vectors of artificial noise signal are the complex modulation vectors with the similar distribution to that of normal constellation errors in plain sight.

4.2. Model-Based Demodulation. The model-based demodulation is used to extract the residual signal and demodulate the secret message bits. The I/Q vectors of received covert signal $\widehat{\mathbf{s}}_{\text{ct}}$ are denoted by $\widehat{\mathbf{x}}_{\text{ct}}^I + j \cdot \widehat{\mathbf{x}}_{\text{ct}}^Q$; the received cover message bits denoted by $\widehat{\mathbf{m}}_c$ can be demodulated by QPSK with

$$F_{\text{de-QPSK}}\left(\widehat{\mathbf{x}}_{\text{ct}}^I + j \cdot \widehat{\mathbf{x}}_{\text{ct}}^Q\right) = \widehat{\mathbf{m}}_c. \tag{10}$$

Then the cover message bits $\widehat{\mathbf{m}}_c$ are remodulated by QPSK to acquire the ideal I/Q vectors of received cover signal $\widehat{\mathbf{s}}_c$ in each subcarrier.

$$F_{\text{QPSK}}\left(\widehat{\mathbf{m}}_c\right) = \widehat{\mathbf{x}}_c^I + j \cdot \widehat{\mathbf{x}}_c^Q. \tag{11}$$

Here $\widehat{\mathbf{x}}_c^I, \widehat{\mathbf{x}}_c^Q \in \{-1/\sqrt{2}, 1/\sqrt{2}, \}^n$ are the ideal normalized I/Q vectors of the received cover signal by QPSK modulation.

The informed receiver can extract the I/Q vectors of residual signal $\mathbf{s}_{\text{residual}}$ with

$$\mathbf{x}_{\text{residual}}^I + j \cdot \widehat{\mathbf{x}}_{\text{residual}}^Q = \left(\widehat{\mathbf{x}}_{\text{ct}}^I - \widehat{\mathbf{x}}_c^I\right) + j \cdot \left(\widehat{\mathbf{x}}_{\text{ct}}^Q - \widehat{\mathbf{x}}_c^Q\right). \tag{12}$$

Then the secret message bits can be extracted from the I/Q vectors $\mathbf{x}_{\text{residual}}^I + j \cdot \widehat{\mathbf{x}}_{\text{residual}}^Q$ of residual signal by the corresponding demodulation algorithm.

$$F_{\text{de-shaping}}\left(\mathbf{x}_{\text{residual}}^I + j \cdot \widehat{\mathbf{x}}_{\text{residual}}^Q\right) = \widehat{\mathbf{m}}_s. \tag{13}$$

Taking I plane, for example, with the I vector $x_{\text{residual},i}^I$ of residual signal and the corresponding center line value α shared between the sender and the receiver, the secret message bits can be demodulated with (14). The demodulation in Q plane works in the same way.

$$\widehat{m}_{si,1} = \begin{cases} 0, & x_{\text{residual},i}^I < \alpha, \\ 1, & x_{\text{residual},i}^I \geq \alpha. \end{cases} \tag{14}$$

Since the I/Q vectors of residual signal have the similar distribution to that of normal constellation errors in our proposed scheme directly, the undetectability of the proposed wireless covert channel is better than that of the wireless covert channels with dirty constellation which can be considered as the state-of-the-art existing method. At the same time, the constellation errors in each subcarrier can be used to transmit the secret data; the mapping sequence in the wireless covert channel with dirty constellation [6] is not necessary. No extra bandwidth is required for transmitting the shared mapping sequence.

5. Experimental Results

5.1. Experimental Setup. In this section, we benchmark the proposed scheme by examining the undetectability and

reliability. The cover message bits and the secret message bits are both provided by a pseudo-random bits generator. The wireless communication is set on an 802.11a/g PHY layer. The wireless covert channel is performed on all 100000 symbols. There are 48 subcarriers in a symbol in transmissions. TGn channel models are selected for the wireless channel models in simulation experiment [21]. The TGn channel models B and D chosen for the simulation experiment on 802.11a/g PHY layer are universal. The sender and informed receiver are kept stationary. So the Doppler shift of the wireless communication is negligible. The normal constellation errors with the size $N = 2000$ can be selected from the residual signal with the specific relative power captured from the actual communication. So the rest of the residual signal is used to be referenced normal constellation errors for detection. In the IEEE 802.11a/g standard [22], the modulation error at the sender for a QPSK modulation is mandated to be no more than 10 dB or 13 dB from an ideal modulation with different code rates. The relative powers of chosen normal constellation error are set to −10 dB and −13 dB. The wireless covert channel with dirty constellation (WCC-DC) is chosen for comparison in some simulation experiments. This covert channel with only 10% subcarriers carrying the secret message bits has been proved safe enough in [6]. The undetectability of the proposed wireless covert channels is measured by KS test and regularity test. In KS test, the KS distance is computed with 1920 constellation errors in subcarriers which are the constellation errors in 20 symbols. In regularity test, we compute the regularity measures for 1920 constellation errors under set size $w = 48$. The detection measures of the I vectors, Q vectors, and magnitudes of constellation errors are presented in the range of transmission SNR $= 10, \ldots, 40$. The reliability of the proposed wireless covert channels is measured by BER.

5.2. Undetectability. The relationship of undetectability and the number of the bins L in the proposed scheme is considered. The wireless channel models in experiments are all set to be TGn channel model B. The relative power of normal constellation errors is set to be $P_e = -10$ dB. The KS distances and regularity measures of I vectors, Q vectors, and magnitudes of constellation errors in the proposed wireless covert channels are presented in Figures 7 and 8 with $L = 50, 100, 200$. The detection measures of the wireless covert channels with dirty constellation (WCC-DC) are also presented for comparison. Each detection measure is obtained as an average over repeated experiments. In Figures 7(a) and 7(b), with the transmission SNR increasing, the KS distances between I vectors and Q vectors of constellation errors in the proposed wireless covert channels with different bin numbers remain almost unchanged, and the KS distances between I vectors and Q vectors in WCC-DC slightly increase. In Figure 7(c), with the transmission SNR increasing, the KS distances between magnitudes of constellation errors in all the proposed wireless covert channels are kept steady, but the KS distances between magnitudes in WCC-DC obviously increase. increase. In Figures 8(a) and 8(b), the regularity measures of I vectors and Q vectors of constellation errors in WCC-DC and proposed methods are only a little different from those of referenced normal constellation errors. In Figure 8(c), the regularity measures of magnitudes of constellation errors in all the proposed wireless covert channels are only a little different from those of referenced normal constellation errors, but the regularity measures of magnitudes in WCC-DC are a little higher than those of referenced normal constellation errors.

It is shown that the undetectability is not concerned with the bin numbers in proposed wireless covert channels. The undetectability of the proposed wireless covert channel is better than that of existing methods, especially with high transmission SNR.

Then, we concentrate on the relationship of undetectability and the wireless channel models. The bin number in the proposed wireless covert channels is set to be $L = 100$. The normal constellation errors are applied with the relative power $P_e = -10$ dB. The detection measures of I vectors, Q vectors, and magnitudes of constellation errors in the proposed wireless covert channels in TGn channel models B and D are presented in Figures 9 and 10. The detection measures of the wireless covert channels with dirty constellation (WCC-DC) are also presented for comparison. Each detection measure is obtained as an average over repeated experiments. In Figures 9(a)–9(c), the KS distances of I vectors, Q vectors, and magnitudes of constellation errors in proposed wireless covert channel and WCC-DC are almost equal in different channel models. With the transmission SNR increasing, the KS distances between I vectors, Q vectors, and magnitudes of constellation errors in the proposed wireless covert channels are kept steady in different channel models, and the KS distances between I vectors, Q vectors, and magnitudes in WCC-DC also increase in different channel models. In Figures 10(a)–10(c), the regularity measures of I vectors, Q vectors, and magnitudes of constellation errors in the proposed wireless covert channels and referenced normal constellation errors are almost equal, the regularity measures of I vectors and Q vectors of constellation errors in WCC-DC are only little different from those of referenced normal constellation errors, and the regularity measures of magnitudes in WCC-DC are a little higher than those of referenced normal constellation errors in different channel models.

It is shown that the undetectability of the proposed wireless covert channel is not related to wireless channel models. And the undetectability of the proposed wireless covert channel is better than that of existing methods in different channel models.

At last, relationship of the undetectability and the relative power of normal constellation errors is discussed. The bin number in the proposed wireless covert channels is set to be $L = 100$. The detection measures of I vectors, Q vectors, and magnitudes of constellation errors in the proposed wireless covert channels are presented in Figures 11 and 12 with the relative power of normal constellation errors $P_e = -10$ dB and $P_e = -13$ dB in TGn channel models B and D. Each detection measure is obtained as an average over repeated experiments. With the transmission SNR increasing, the KS distances between I vectors, Q vectors, and magnitudes of constellation errors in the proposed wireless covert channels are almost steady with different relative power of normal

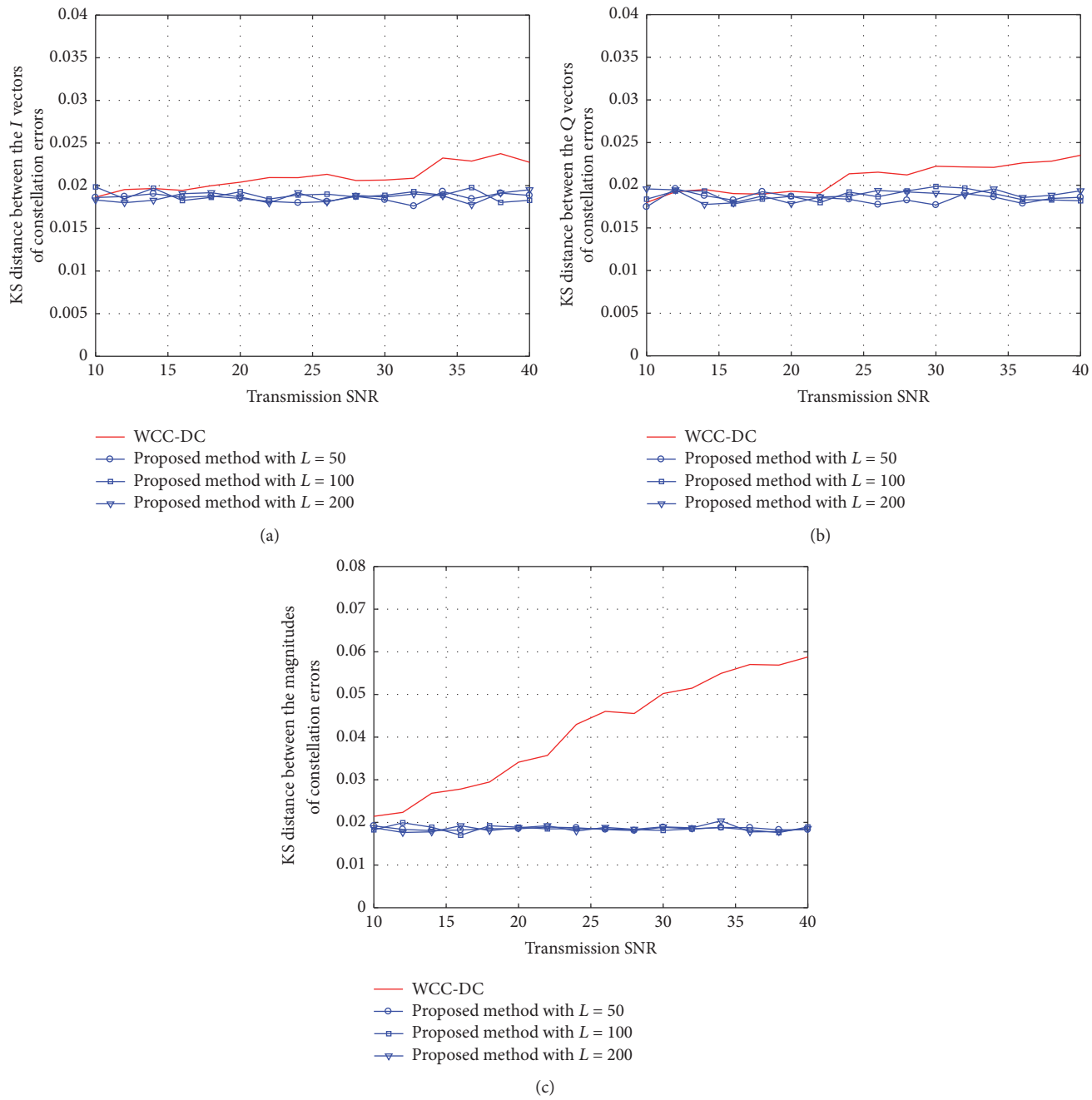


Figure 7: KS distances between (a) I vectors, (b) Q vectors, and (c) magnitudes of constellation errors with different bin numbers.

constellation errors and channel models, and the regularity measures of I vectors, Q vectors, and magnitudes in the proposed wireless covert channels are little different from those of referenced normal constellation errors.

It is shown that the undetectability in the proposed wireless covert channels is kept almost unchanged with different relative power of normal constellation errors.

5.3. Reliability. The relationship of reliability and the number of the bins L in the proposed scheme is considered. The wireless channel model in experiments is set to be TGn channel model B. The relative power of normal constellation errors is set to be $P_e = -10$ dB. The BERs of the proposed wireless covert channels are presented in Figure 15 with $L = 50, 100, 200$. The BER of WCC-DC is also presented for comparison. Each bit error rate is obtained as an average over repeated experiments. In Figure 13, the bit error rates of the proposed wireless covert channels are almost equal with different bin numbers. The bit error rates of WCC-DC are lower than those of all the proposed wireless covert channels. The degradation of reliability is considered as the cost for the improvement in undetectability. Given that the covert transmission rate of the proposed wireless covert channels is ten times as large as that in WCC-DC, the experimental

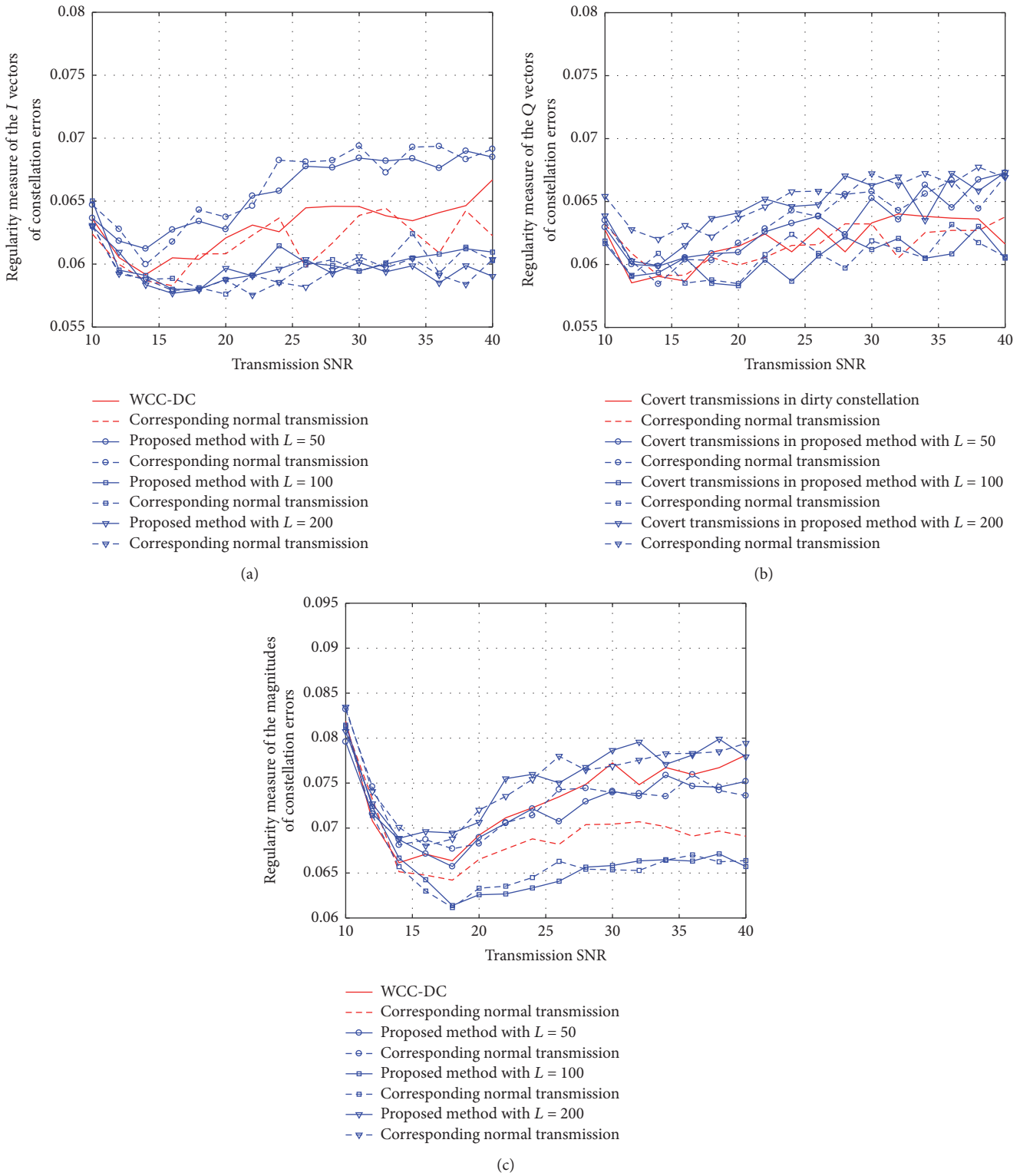


FIGURE 8: Regularity measures of (a) I vectors, (b) Q vectors, and (c) magnitudes of constellation errors with different bin numbers.

results of reliability with the same covert transmission rate are presented later.

Then, we concentrate on the relationship of the reliability and the wireless channel models. The bin number in the proposed wireless covert channels is set to be $L = 100$. The relative power of normal constellation errors is set to be $P_e = -10$ dB. The BERs of the proposed wireless covert channels is presented in Figure 14 in TGn channel models B and D. The BER of WCC-DC is also presented for comparison. Each bit error rate is obtained as an average over repeated

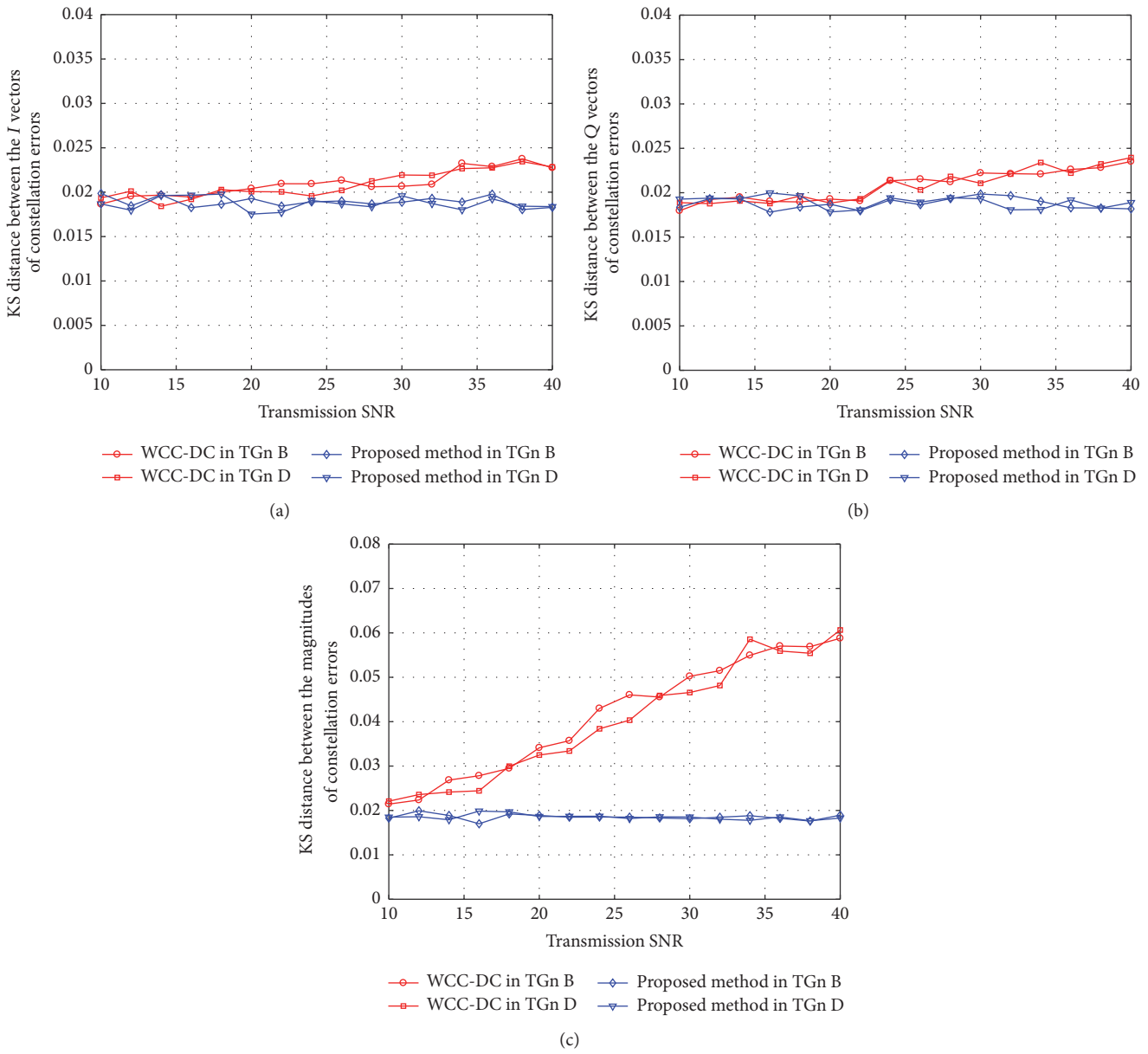


Figure 9: KS distances between (a) I vectors, (b) Q vectors, and (c) magnitudes of constellation errors in different channel models.

experiments. The bit error rates of two kinds of the wireless covert channels in TGn model B are all lower than those in TGn model D. The bit error rates of WCC-DC are lower than those of the proposed wireless covert channels in each channel model. So the reliability of the proposed wireless covert channel is better in the channel with less fading.

The relationship of the reliability and the relative power of normal constellation errors is discussed. The bin number in the proposed wireless covert channels is set to be $L = 100$. The BERs of the proposed wireless covert channels are presented in Figure 16 with the relative power of normal constellation errors $P_e = -10$ dB and $P_e = -13$ dB in TGn channel models B and D. Each bit error rate is obtained as an average over repeated experiments. In Figure 15, the bit error rates of the proposed wireless covert channels in TGn model B are lower than that in TGn model D with the equal relative power of normal constellation errors. The bit error rates of the proposed wireless covert channels with bigger relative power of normal constellation errors are lower than those with less relative power of normal constellation errors in each channel model. The reliability of the proposed wireless covert channels gets better with relative power of normal constellation errors increasing.

At last, the experiment of reliability of the two kinds of wireless covert channels with the same covert transmission rate is performed in Figure 16. The bin number in the proposed wireless covert channels is set to be $L = 100$. The wireless channel model in experiment is set to be TGn channel model B. The relative power of normal constellation errors is set to be $P_e = -10$ dB. The direct sequence spread

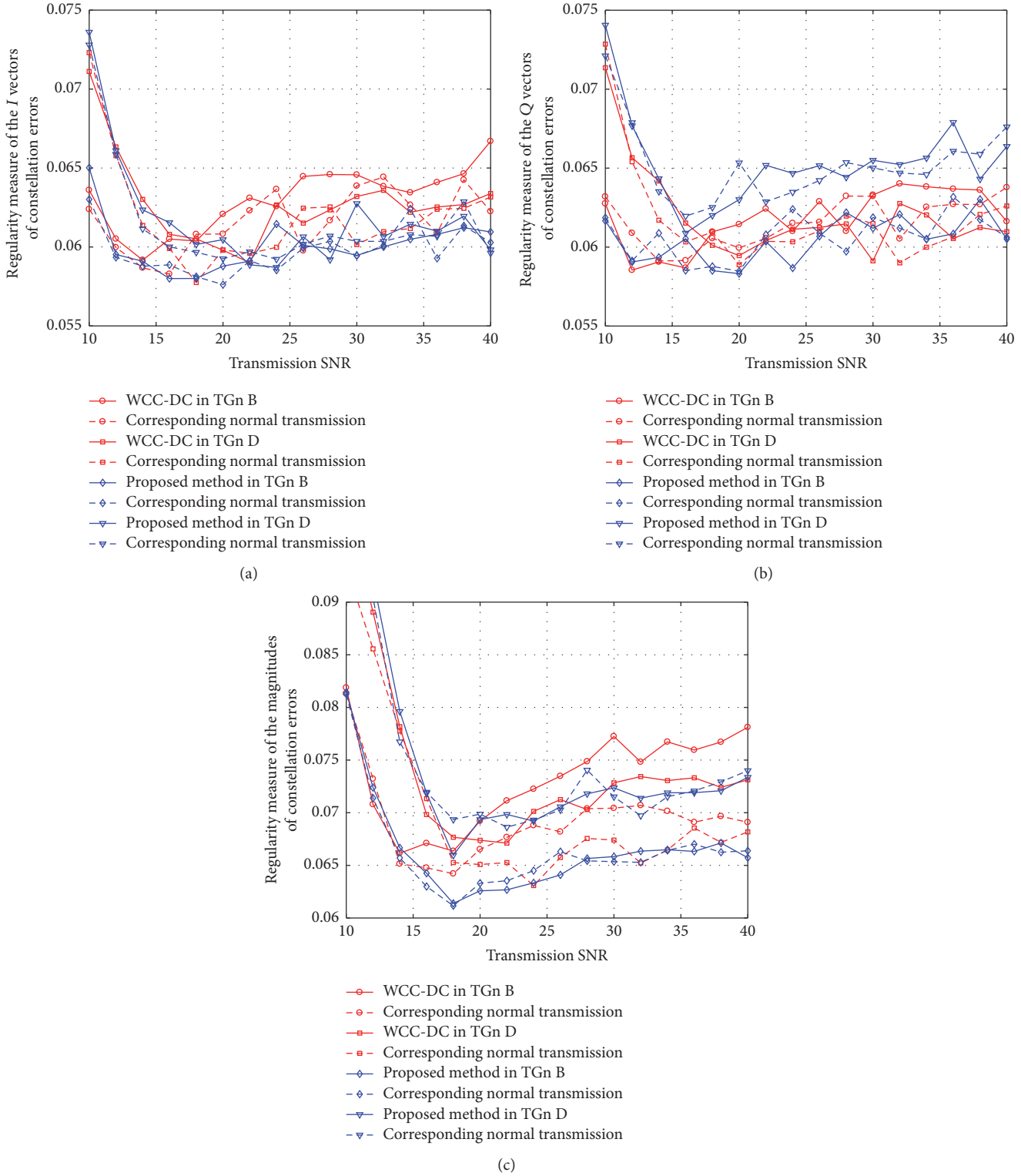


Figure 10: Regularity measures of (a) I vectors, (b) Q vectors, and (c) magnitudes of constellation errors in different channel models.

spectrum codes are applied in the secret message bits in the proposed wireless covert channels with $m = 10$. The BER of WCC-DC is presented for comparison. Each bit error rate is obtained as an average over repeated experiments. The bit error rates of the proposed wireless covert channels with the direct sequence spread spectrum codes are lower than those of WCC-DC.

It is proved that the proposed wireless covert channels are more reliable than WCC-DC with the same covert transmission rate.

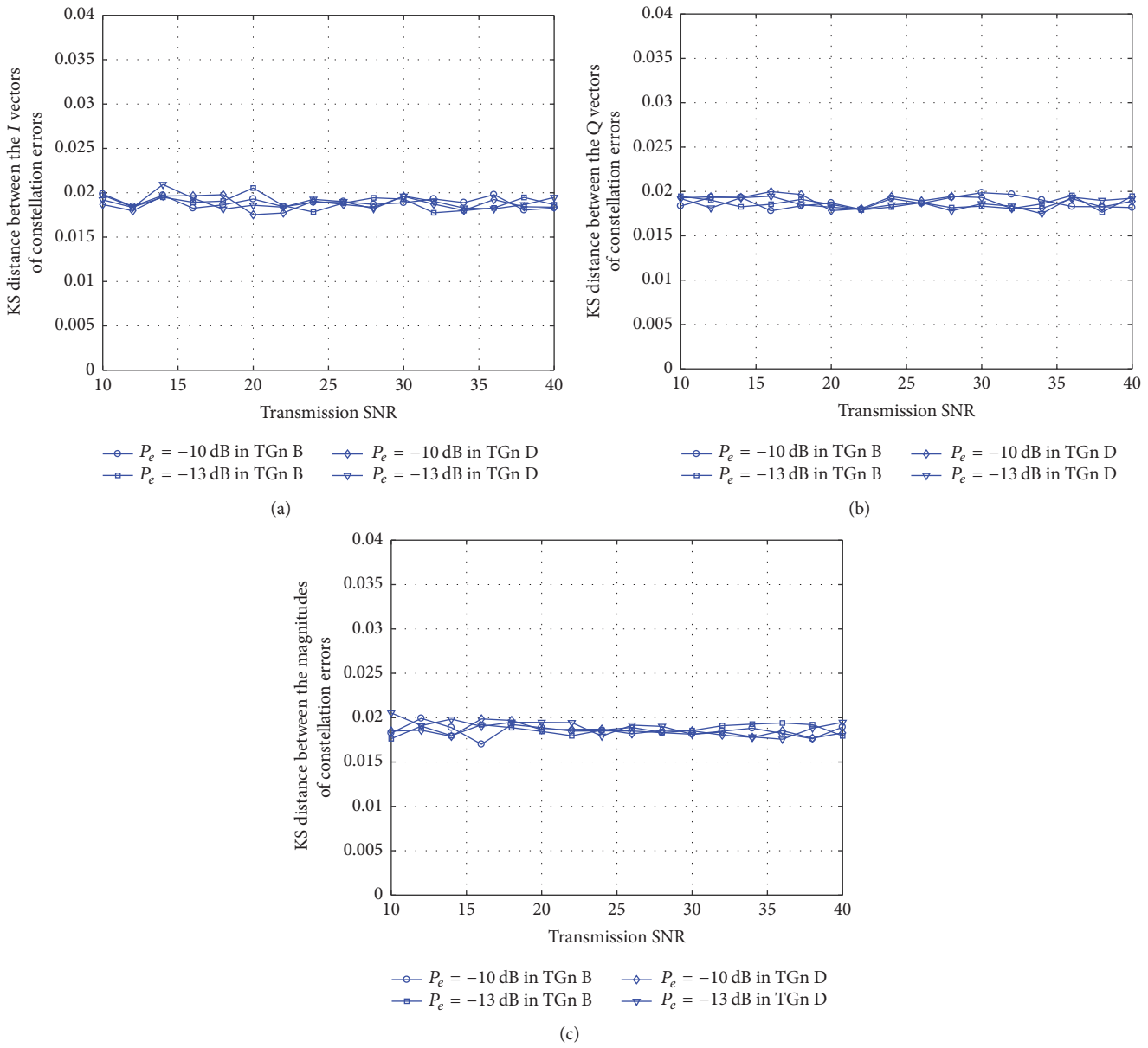


Figure 11: KS distances between (a) I vectors, (b) Q vectors, and (c) magnitudes of constellation errors with different relative powers.

6. Conclusions

Undetectability and reliability are the main aims of wireless covert channel. In this paper, we extend the detection of covert timing channel to the wireless covert channel. To improve undetectability, the wireless covert channel is proposed based on constellation shaping modulation; the constellation errors with the secret message bits embedded are distributed as normal constellation errors. The security against the detection and the reliability of the proposed wireless covert channel are improved with the same covert transmission rate.

Even if the proposed scheme can achieve high undetectability, the reliability of the proposed wireless channel needs to be strengthened. In the same framework, the bins near center line in I/Q plane are kept unused for guard band to improve the reliability and keep the undetectability of our future work.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grants nos. 61472188, 61602247,

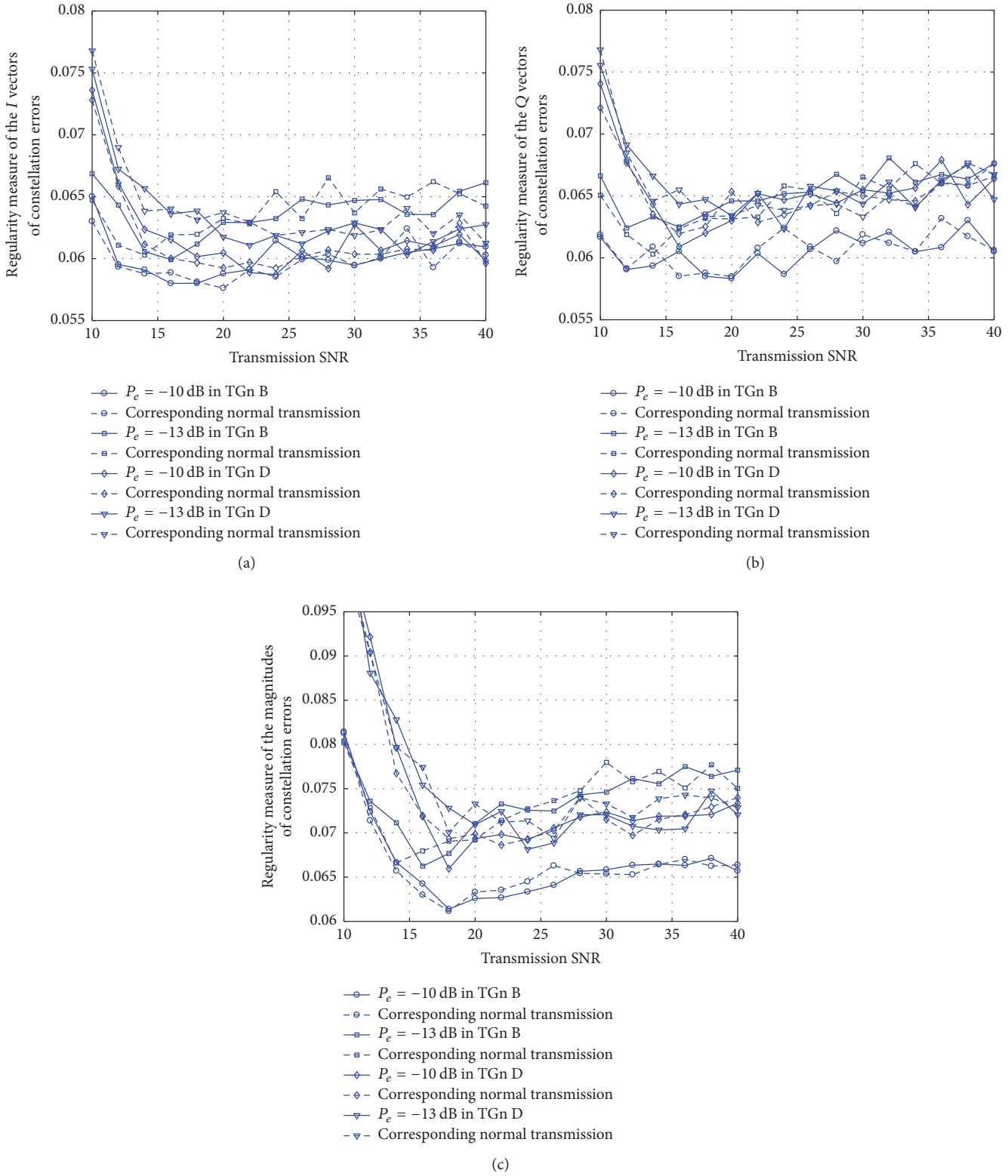


Figure 12: Regularity measures of (a) I vectors, (b) Q vectors, and (c) magnitudes of constellation errors with different relative powers.

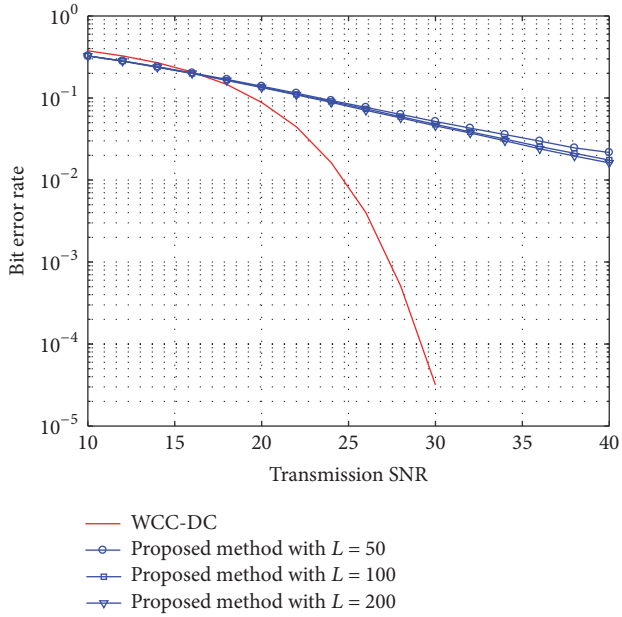


Figure 13: BERs of the proposed wireless covert channels with different bin numbers.

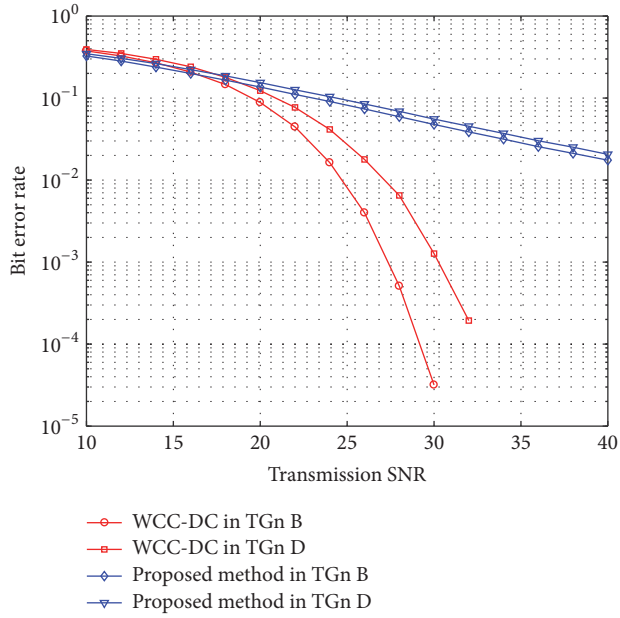


Figure 14: BERs of the proposed wireless covert channels in different channel models.

61702235, and U1636117), Natural Science Foundation of Jiangsu Province (Grants nos. BK20150472 and BK20160840), National Key Technology Research and Development Program of the Ministry of Science and Technology of China (Grant no. 2014BAH41B01), CCF-VENUSTECH Foundation (Grant no. 2016011), and Fundamental Research Funds for the Central Universities (30920140121006 and 30915012208).

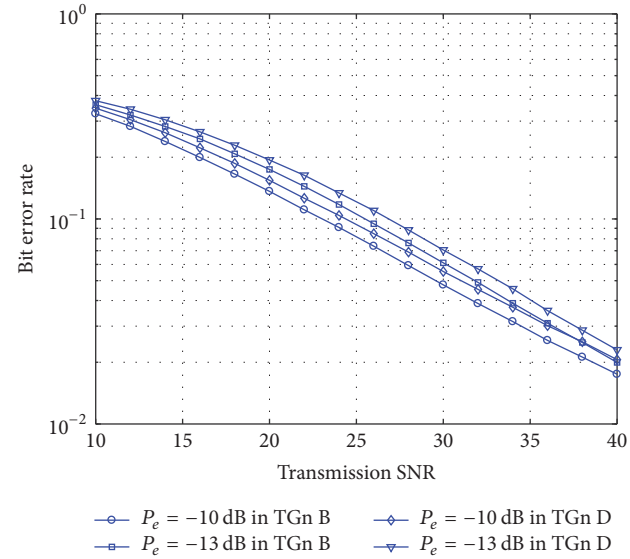


Figure 15: BERs of the proposed wireless covert channels with different relative powers.

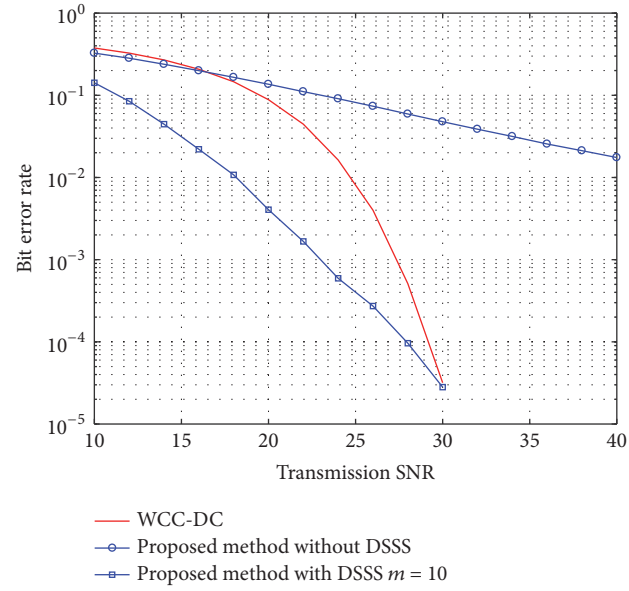


Figure 16: BERs of the proposed wireless covert channel with same covert transmission rate.

References

[1] S. Gianvecchio, H. Wang, D. Wijesekera, and S. Jajodia, "Model-Based Covert Timing Channels: Automated Modeling and Evasion," in *International Symposium on Recent Advances in Intrusion Detection*, pp. 211–230, 2008.

[2] K. Kothari and M. Wright, "Mimic: An active covert channel that evades regularity-based detection," *Computer Networks the International Journal of Computer & Telecommunications Networking*, vol. 57, pp. 647–657, 2010.

[3] R. J. Walls, K. Kothari, and M. Wright, "Liquid: A detection-resistant covert timing channel based on IPD shaping," *Computer Networks*, vol. 55, no. 6, pp. 1217–1228, 2011.

[4] G. Liu, J. Zhai, and Y. Dai, "Network covert timing channel with distribution matching," *Telecommunication Systems*, vol. 49, no. 2, pp. 199–205, 2012.

[5] A. Mileva and B. Panajotov, "Covert channels in TCP/IP protocol stack - extended version," *Central European Journal of Computer Science*, vol. 4, no. 2, pp. 45–66, 2014.

[6] A. Dutta, D. Saha, D. Grunwald, and D. Sicker, "Secret agent radio: Covert communication through dirty constellations," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 7692, pp. 160–175, 2013.

[7] I. Grabska and K. Szczypiorski, "Steganography in long term evolution systems," in *Proceedings of the 2014 IEEE Computer Society's Security and Privacy Workshops, SPW 2014*, pp. 92–99, May 2014.

[8] Z. Hijaz and V. S. Frost, "Exploiting OFDM systems for covert communication," in *Proceedings of the 2010 IEEE Military Communications Conference, MILCOM 2010*, pp. 2149–2155, November 2010.

[9] T. Kitano, H. Iwai, and H. Sasaoka, "A wireless Steganography technique by embedding DS-SS signal in digital mobile communication systems," *Science Engineering Review of Doshisha University*, vol. 52, pp. 127–134, 2011.

[10] T. Yucek and H. Arslan, *Covert OFDM transmission using cyclic prefix*, 2012.

[11] K. Hokai, H. Sasaoka, and H. Iwai, "Wireless steganography using MIMO system," in *Proceedings of the 5th IEEE International Conference on Communications and Electronics, IEEE ICCE 2014*, pp. 560–565, August 2014.

[12] J. Classen, M. Schulz, and M. Hollick, "Practical covert channels for WiFi systems," in *Proceedings of the 3rd IEEE International Conference on Communications and Network Security, CNS 2015*, pp. 209–217, September 2015.

[13] K. Szczypiorski and W. Mazurczyk, "Hiding data in OFDM symbols of IEEE 802.11 networks," in *Proceedings of the 2010 2nd International Conference on Multimedia Information Networking and Security, MINES 2010*, pp. 835–840, November 2010.

[14] J. Hu, S. Yan, X. Zhou, F. Shu, J. Li, and J. Wang, "Covert communication achieved by a greedy relay in wireless networks," *Information Theory*, 2017.

[15] J. Hu, S. Yan, X. Zhou, F. Shu, and J. Wang, "Covert communication in wireless relay networks," *Information Theory*, 2017.

[16] X. Wang, Y. Liu, X. Lu, S. Lv, Z. Shi, and L. Sun, "CovertMIMO: A covert uplink transmission scheme for MIMO systems," in *Proceedings of the ICC 2017 - 2017 IEEE International Conference on Communications*, pp. 1–6, Paris, France, May 2017.

[17] K. Szczypiorski, A. Janicki, and S. Wendzel, ""The good, the bad and the ugly": Evaluation of Wi-Fi steganography," *Journal of Communications*, vol. 10, no. 10, pp. 747–752, 2015.

[18] P. Peng, P. Ning, and D. S. Reeves, "On the secrecy of timing-based active watermarking trace-back techniques," in *Proceedings of the 2006 IEEE Symposium on Security and Privacy*, pp. 334–348, May 2006.

[19] S. Cabuk, C. E. Brodley, and C. Shields, "IP covert timing channels: Design and detection," in *Proceedings of the 11th ACM Conference on Computer and Communications Security, CCS 2004*, pp. 178–187, October 2004.

[20] S. Gianvecchio and H. Wang, "An Entropy-Based Approach to Detecting Covert Timing Channels," *IEEE Transactions on Dependable & Secure Computing*, vol. 8, pp. 307–316, 2011.

[21] E. Perahia and R. Stacey, *Next generation wireless LANs*, Cambridge University Press, 2008.

[22] IEEE, "Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) Specifications," in *The 5-Ghz Band*, pp. C1–C1184, IEEE, 2003.

Hindawi
Security and Communication Networks
Volume 2017, Article ID 7536381, 9 pages
https://doi.org/10.1155/2017/7536381

Research Article

Constructing APT Attack Scenarios Based on Intrusion Kill Chain and Fuzzy Clustering

Ru Zhang, Yanyu Huo, Jianyi Liu, and Fangyu Weng

Information Secure Center, Beijing University of Posts and Telecommunications, 10 West Tucheng Road, Haidian District, Beijing, China

Correspondence should be addressed to Ru Zhang; zhangru@bupt.edu.cn

Received 28 September 2017; Accepted 4 December 2017; Published 27 December 2017

Academic Editor: Zhenxing Qian

The APT attack on the Internet is becoming more serious, and most of intrusion detection systems can only generate alarms to some steps of APT attack and cannot identify the pattern of the APT attack. To detect APT attack, many researchers established attack models and then correlated IDS logs with the attack models. However, the accuracy of detection deeply relied on the integrity of models. In this paper, we propose a new method to construct APT attack scenarios by mining IDS security logs. These APT attack scenarios can be further used for the APT detection. First, we classify all the attack events by purpose of phase of the intrusion kill chain. Then we add the attack event dimension to fuzzy clustering, correlate IDS alarm logs with fuzzy clustering, and generate the attack sequence set. Next, we delete the bug attack sequences to clean the set. Finally, we use the nonaftereffect property of probability transfer matrix to construct attack scenarios by mining the attack sequence set. Experiments show that the proposed method can construct the APT attack scenarios by mining IDS alarm logs, and the constructed scenarios match the actual situation so that they can be used for APT attack detection.

1. Introduction

Nowadays, attacks on the network are becoming more and more complex, and, among them, APT attacks are increasingly frequent [1]. Unlike traditional attacks, APT attacks are not launched to interrupt services, but to steal intellectual property rights and sensitive data [2]. An APT attack has the stage and longevity characteristics and uncertain attack channel. Therefore, the Intrusion Detection System (hereinafter referred to as IDS) cannot detect an APT attack and can only generate alarms to certain steps in the attack. In 2012, Kabbah and Comodo companies' source codes were stolen [3]; in 2015, the OceanLotus Organization launched APT attacks on a number of essential institutions, including the Chinese government, certain research institutes, and maritime organizations in China [4]. Since then, APT attack has become a hot research topic. This paper focuses on how to correlate a large number of IDS security logs to dig out an APT attack scenario, and ultimately identifies an APT attack. Attack scenarios reflect the actual state of the network and can help defenders to take corresponding precautionary measures.

Correlating alarm logs is an important step to dig out attack scenarios. At present, researchers working on APT attack correlation built a full-scale attack model based on the phases of an APT attack and then correlated security logs with the attack model to generate the attack context. However, the establishment of APT attack model requires expert knowledge, and if the attack model is incomplete, some alarmed events will be unmatched and discarded, resulting in an incomplete attack route. In this paper, we propose a new method to solve this problem. We adopt fuzzy clustering correlation method to form clusters using multidimensional properties of alarm logs, so the correlated alarms are clustered to an APT attack route. Although each case is different, all APT attacks are phased, which conform to the feature of the intrusion kill chain model. According to this feature, we improve the fuzzy clustering algorithm by adding attack event property. We divide an APT attack process into several phases according to the intrusion kill chain model and categorize the attack events into different phases according to the characteristic of each phase, the behavior of each attack event, and the degree of harm. Then we compare the attack

events of two alarms in the process of clustering, and if the attack event of latter alarm is in the subsequent attack phase relative to the attack event of previous one, then the correlation of the two alarms is stronger. The merits of taking the attack event as a cluster dimension are that it improved the correlation of alarms in an attack sequence, and there is no need to establish the attack model beforehand, and the alarm will not be lost because its event cannot match. Finally, we analyze the clustering results, combine the repeated attacks, delete the incomplete attack fragments, and then establish the probability transfer matrix to mining the attack scenario.

2. Literature Review

An APT attack is targeted, camouflaged, and phased, and it cannot be identified effectively with traditional detection technologies [5]. Friedberg et al. used the whitelist method to detect APT attacks. This method studied normal system behaviors and reported those operations different from system normal model, to find out Zero-day Threats [2]. Choi et al. used the extraction of normal behavior and anomaly patterns to detect the anomalies of APT attacks and proposed a method to detect anomalies by mining unknown anomaly patterns [6]. The APT attack model is often used in security log-based APT attack detection [7]; Tankard [8] established an APT attack model to monitor the network to discover the rules of actual attacking process. And Zhang et al. [3] constructed the attack tree model based on the intrusion kill chain and analyzed the attack logs to form the attack route to predict an APT attack. Three security researchers from Lock Martin first proposed the intrusion kill chain model on the ICIW Conference in March 2011 [9]. From the perspective of intrusion detection, this model decomposes the attacking process into 7 steps of reconnaissance, weaponization, delivery, exploitation, installation, command and control, and actions on objectives, and this model meets the phase characteristic of APT attack. APT attack detecting methods based on attacking model rely on expert knowledge predefining model. If the attack model is incomplete, attack scenario will be disrupted. If an attacker does not attack by well-defined rules and bypasses a phase in other ways, then a complete attack scenario cannot be constructed. Therefore, in this paper, fuzzy clustering is used in correlation to resolve these problems.

In the context of using fuzzy clustering to correlate alarms, the alarms are correlated to form an attack sequence by calculating the similarity between the alarms [10]. In terms of alarms correlation, most papers studied in general multistep attacks and made no adjustment according to different complex attacks. Feng et al. [11] used the correlation of the IP address in clustering, but the causation of two alarms is not just reflected in IP addresses. In this paper, we divide the attack events using the intrusion kill chain model and use multidimensional properties including the IP address, the attack event, and the time stamp in fuzzy clustering. This method resolves problems such as inability in constructing the complete attack scenario using expert knowledge, and loose coupling of clustering using single property. Finally, the attack sequences are fused by the transfer matrix, which avoids small-frequency attack sequences being omitted when using frequently occurring item sets.

3. Mining an APT Attack Scenario

IDS alarm log is a kind of log generated when attack operations occur. It shows security situation of the entire network.

Definition 1 (alarm logs). We represent IDS alarm log as alarms $= \{a_1, a_2, a_3, \ldots, a_n\}$, where a_i indicates the ith alarm and is a six-tuple:

$$a_i = (\text{timestamp, sIP, dIP, sPort, dPort, alarm\_event}). \quad (1)$$

The meaning of each attribute is shown in Table 1.

Definition 2 (attack sequence). An attack sequence is a sequence of IDS alarms that is produced by an attacking process. We represent the attack sequence as AS $= \{a_1, a_2, a_3, \ldots, a_n\}$, where all the alarms are listed in temporal order.

Definition 3 (attack scenario). The attack scenario shows the intrusion process of many different attack actions according to a certain time and logical sequence, which can be described in the form of graphs. Therefore, it can be said that the attack scenario consists of many single attack steps, which are the attack alarm information detected by the safety device.

3.1. The Entire Process. As is shown in Figure 1, there are four steps in the entire process:

(1) Data preprocessing: the IDS alarm log is normalized to the six-tuple format as in Definition 1 after a simple elimination of false positives in the data.

(2) Attack event classification: the APT attack is divided into several phases based on the intrusion kill chain model, and the attack events are classified according to the purpose of each phase and the behaviors of each attack event.

(3) Fuzzy clustering: the similarity function of each property used in fuzzy clustering is defined so that fuzzy clustering can be conducted. The attack sequence set $\text{ASS} = \{\text{AS}_1, \text{AS}_2, \ldots, \text{AS}_q\}$ is formed after fuzzy clustering, where each attack sequence $\text{AS}_i = \langle a_1, a_2, a_3, \ldots, a_n \rangle$ represents a possible APT attacking process, where a_i is an alarm.

(4) Attack scenario mining: we analyze all attack sequences generated after fuzzy clustering and delete isolated attack sequences without subsequent data transmission. A probability transfer matrix is established using multiple attack sequences where each row and each column represents an attack event. And finally, the probability transfer matrix is converted into a probabilistic attack scenario graph that can be used to identify APT attacks in the network.

Table 1: The meaning of each attribute.

The attribute of an alarm	Meaning
timestamp	The time when the attack occurred
sIP	The source IP address
dIP	The destination IP address
sPort	The source port
dPort	The destination port
alarm_event	The IDS alarm event

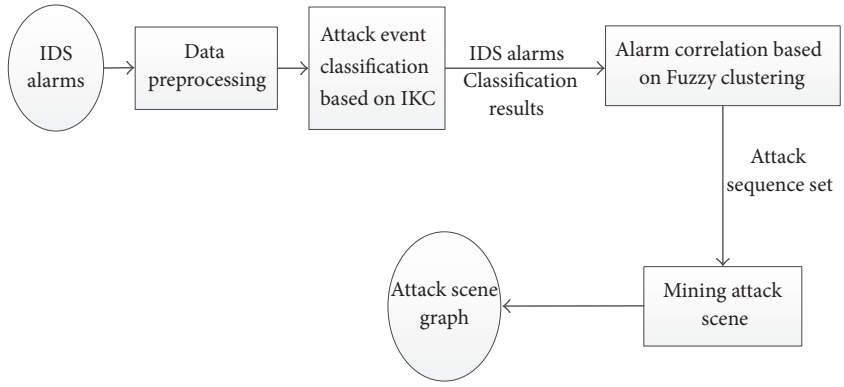


Figure 1: The entire process.

3.2. Attack Events Classification Based on IKC. In some papers, the intrusion kill chain (IKC) model is widely used in constructing APT attack model. The attack event is an important property of an APT attack; therefore, the attack event dimension is included in fuzzy clustering, and all the attack events in the alarm log are classified by the purpose of each phase and the behavior of the attack event in the model, to calculate the correlation between two alarms in the fuzzy clustering. The two adjacent alarmed attack events are compared in clustering. If the latter alarmed attack event is in subsequent attack phase relative to the former alarmed attack event, the correction of the two alarms is higher.

We divide an APT attack into four phases based on the IKC model. Each phase has different purpose and different behavior.

(1) Information collection phase: it is the first step of an attack, including reconnaissance and information collecting, using some technical means such as scanning, probing, and social engineering.

(2) Intrusion phase: the attacker induces the target user to click on the phishing website or to download the malicious email attachment or install a backdoor through Trojan upload or loophole exploitation, to upgrade access permission to the target host.

(3) Latent expansion phase: the attacker maintains connection to the controlled host to obtain more valuable data and get ready for expansion. The attacker continues penetrating in the interior by using the host with permission as a stepping stone.

(4) Information theft phase: this is the confidential information transmission phase. The data will be transferred to the attacker's server after the attack has reached the host. The transport process often uses SSL or TLS secure transport protocol to encrypt data for camouflage. In addition to obtaining information, APT attackers can disrupt the facilities in the target network and interfere with the normal operation of the system.

We analyze the behavior and hazard of each attack event and classify all attack events into a certain phase. The classification process is shown in Figure 2.

3.3. Alarm Correlation Based on Fuzzy Clustering. Fuzzy clustering analysis generally constructs fuzzy matrix according to the property of the object and determines the clustering relationship according to the degree of membership. The properties of the alarm log are nonnumericand are typically measured in the following manner.

$x_i, x_j \in A$ where A is the alarm set, and the membership function of x_i and x_j in fuzzy clustering is defined as $S(x_i, x_j) = (\sum_{k=1}^{m} \alpha_k \cdot \delta(x_{ik}, x_{jk}))/m$, where m is the number of properties for an alarm, α_k is the weight of each property, and $\delta(x_{ik}, x_{jk})$ is the similarity function for each property, generated by the nature of property.

3.3.1. The Similarity Function. We define the similarity function of properties in fuzzy clustering according to different meanings of different properties. We define the similarity function of three properties including IP address, timestamp, and attack event as follows.

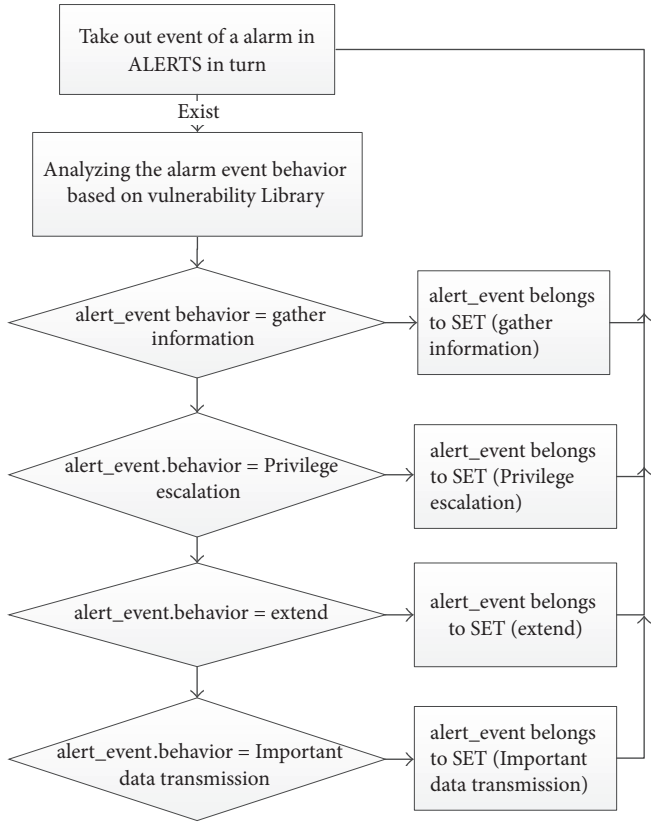


Figure 2: The classification process.

a_i is an original alarm, and a_j is a classified alarm using fuzzy clustering. We use similarity of a_i and a_j to measure a_i's membership of the class containing a_j, that is, $F(a_i, a_j) = \delta_k F_k(a_i, a_j)$, where δ_k is the weight of each property, and k refers to alarm event, IP address, and timestamp three properties.

(1) The Attack Event Similarity Function. In terms of the attack event dimension, the similarity function of a_i and a_j to an attack sequence is as follows:

$$F_{\text{alarm\_event}}\left(a_i, a_j\right) = \begin{cases} 1, & \Delta\alpha = 0 \text{ or } 1 \\ e^{-(\Delta\alpha - 1)}, & \Delta\alpha > 1 \\ 0, & \text{else,} \end{cases} \tag{2}$$

$$\Delta\alpha = \alpha\left(a_i.\text{alarm\_event}\right) - \alpha\left(a_j.\text{alarm\_event}\right).$$

$\alpha\left(a_i.\text{alarm\_event}\right)$ indicates the phase of a_i's attack event, and $\Delta\alpha$ is the difference between the phases of the two alarms. From the attacker's point of view, the attack of subsequent phase is more complex and purposeful and has higher access permission, so if $\Delta\alpha$ equals to 0 or 1, the degree of correlation between these two attack events is higher.

(2) The IP Similarity Function

$$F_{\text{IP}}\left(a_i, a_j\right) = \frac{N}{32}, \tag{3}$$

where $N = \max\{H(a_i.\text{sIP}, a_j.\text{dIP}), H(a_i.\text{sIP}, a_j.\text{sIP}), H(a_i.\text{dIP}, a_j.\text{sIP})\}$, sIP means the source IP, dIP means the destination IP, and $H(\text{IP1}, \text{IP2})$ is maximum same digits of the two IP from the high to low in binary. If two alarms have the same source IP or the same destination IP, or IPs of two alarms are in the same network domain, the two alarms may belong to an attack. Such as, if two alarms have different sIP, but the same dIP, then the attack is launched against the same host, for example, the alarm to an attack with a fake source IP address, such as Syn_flood.

(3) The Timestamp Similarity Function. APT attackers do not tend to profit in a short time, instead, they use the "controlled host" as a stepping stone for persistent searching until a thorough grasp of the target is achieved. In an attacking process, the time interval is relatively short when two attacks are in the same phase, and the time interval may be longer when two attacks occur in different phases, and when there is a long latency following the previous access. For this reason, we do not set time window for alarm logs. The similarity function of the timestamp property is as follows:

$$F_{\text{time}}\left(a_i, a_j\right) = e^{-\Delta t}, \quad \Delta t = a_i.\text{time} - a_j.\text{time}, \text{ the unit of } \Delta t \text{ is day.} \tag{4}$$

The complete similarity is calculated using the following function:

$$F\left(a_i, a_j\right) = \delta_{\text{alarm}_{\text{event}}} F_{\text{alarm}_{\text{event}}}\left(a_i, a_j\right) + \delta_{\text{IP}} F_{\text{IP}}\left(a_i, a_j\right) + \delta_{\text{time}} F_{\text{time}}\left(a_i, a_j\right). \quad (5)$$

IDS alarm logs are in ascending order by the timestamp, and the similarity of two alarms is calculated using the complete similarity function with multidimensional properties. When the similarity is greater than the threshold value, two alarms are considered triggered by the same attack.

3.3.2. Clustering Algorithm Process

Input: alarm log ALARMS = $\{a_1, a_2, a_3, \ldots, a_n\}$, and attack sequence set ASS = $\emptyset$.

Output: attack sequence set ASS = $\{\text{AS}_1, \text{AS}_2, \ldots, \text{AS}_q\}$, where each attack sequence $\text{AS}_i = \langle a_1, a_2, a_3, \ldots, a_n\rangle$ is a set of alarms and reflects a probable APT attack.

① For each original alarm a_i, calculate its membership to each attack sequence AS_i. If the attack sequence set ASS = $\{\text{AS}_1, \text{AS}_2, \ldots, \text{AS}_q\}$ is empty, then make $\text{AS}_1 = \{a_i\}$, and repeat step ①. If ASS is not empty, then use AS_1 in the ASS set in step ②.

② Scan attack sequence $\text{AS}_i = \langle a_1, a_2, a_3, \ldots, a_k\rangle$. First determine whether the phase of the alarm event a_i is equal to or later than the phase of AS_i (the phase in which the latest timestamp in AS_i occurs). If the answer is yes, go to step ③, and if the answer is no, then go to step ④.

③ Calculate the similarity between a_i and each element in AS_i separately using the similarity function and use the maximum value of the results as a membership degree of a_i to AS_i. If the membership degree is greater than or equal to the preset threshold value λ, then add a_i to attack sequence $\text{AS}_i = \{a_1, a_2, a_3, \ldots, a_k, a_i\}$ and go to step ④.

④ Take the next AS_i in ASS, if it exists, repeat step ②; if not, it means that all the attack sequences in the ASS have been scanned. If the membership degree of a_i to every attack sequence is less than λ, then create a new element $\text{AS}_r = \{a_i\}$ and add AS_r to ASS = $\{\text{AS}_1, \text{AS}_2, \ldots, \text{AS}_q, \text{AS}_r\}$, before going to step ⑤.

⑤ Repeat step ① to step ④ above until all ALARMS are analyzed.

3.4. Mining Out the Attack Scenario. We filter attack sequence set combining purpose and phase characteristic of APT attack and delete the incomplete attack sequence of all IP addresses not involving key assets. In the process of converting an attack sequence into a directed graph, alarms with timestamps approximate to the same attack event are merged into one attack event node. This is because the attacker would use different automation tools during the attack to make continuous malicious requests, generating alarms temporally approximate to the same attack event. Finally, the multiple directed graphs are converted to an attack scenario graph through the probability transfer matrix. The key steps of mining algorithm are shown in Figure 3.

Input: attack sequence set ASS = $\{\text{AS}_1, \text{AS}_2, \ldots, \text{AS}_n\}$ and the IP set IIP of key assets

Output: attack scenario graph

① Get a new $\text{AS}_i = \langle a_1, a_2, a_3, \ldots, a_n\rangle$ from ASS, $a_1, a_2, a_3, \ldots, a_n$ is sorted by timestamp. Determine whether the phase of the last alarm in AS_i is ahead of phase 3, and whether the length of AS_i equals 1. If one of the two conditions is met, and none of the IPs of AS_i is in IIP (key asset IP), then discard this attack sequence and repeat step ①; otherwise go to step ②.

② Convert the first alarm in AS_i to an event node that contains the alarmed attack event and scan from the second alarm, before going to step ③.

③ Take unspecified alarms in turn as a_i, and determine if its corresponding attack event is the same as the attack event of the previous alarm a_j. If the answer is yes then do not create a new node and repeat step ③, or, if the answer is no, convert a_i to a node that contains an attack event, and add a side from a_j node to a_i node. Repeat ③ until the last alarm in AS_i is processed. Then use the matrix to save the directed graph.

④ Go to step ①, if all attack sequences have been analyzed go to step ⑤.

⑤ Initialize a transfer matrix of an empty attack event, where each row and each column represents an attack event in the directed graph. Scan each directed graph, and if there is a directed side between the two attack event nodes A and B, add 1 to the value of location (A, B) in the matrix, and if a new attack event cannot be found in the matrix, add a row and a column in the transfer matrix to represent this attack event. Then each numeric in a row is converted to its proportion to the sum of all the numerical values in that row. The final matrix is expressed in the form of a directed graph.

4. Experiment Analysis

In order to prove the effectiveness of our method for mining out APT attacking scenario, we have used the IDS monitoring environment of a certain company and have simulated 10 advanced persistent attacking processes to steal data. Firstly, we use advanced transverse scan to probe an attack and exploit the vulnerabilities of the key hosts so as to increase access permission. In the process, attack tools such as Nmap, Sqlmap, and chopper are used in sending emails with malicious attachments and exploiting vulnerabilities. For example, Namp is used to scan multiple machines, and some of the vulnerable hosts are targeted with further attacks to extract permission. The entire process lasted one month, during which time IDS alarm logs were collected, and there are 1000 or so valid alarms after false positives were eliminated, some of which have similar timestamps.

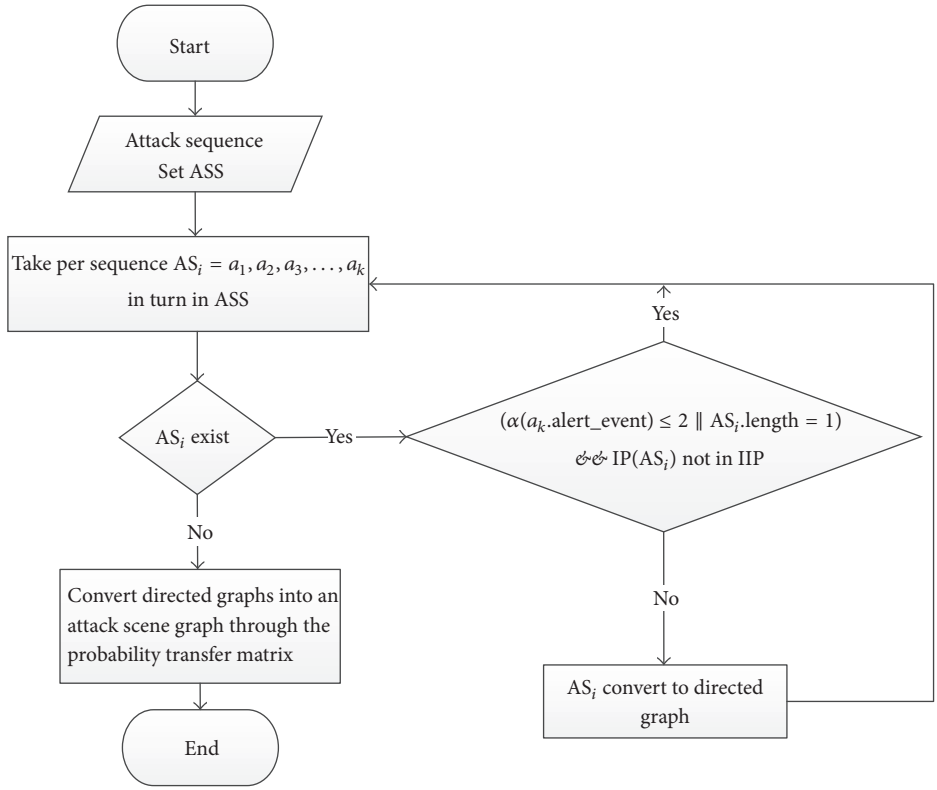


Figure 3: The process of mining out the attack scenario.

Different alarms to the same attack event are generated by the automated tools.

We then classify all the attack events in the experimental data into four phases of an APT attack before carrying out fuzzy clustering.

4.1. Clustering Algorithm Implementation. Alarms are sorted by timestamp in temporal order. Each alarm is normalized to the six-tuple format as in Definition 1.

The similarity function with multidimensional properties defined in Section 3.3.1 and the clustering algorithm defined in Section 3.3.2 are then used to cluster the IDS alarms ($\lambda = 0.65$, $\delta_{\text{alarm\_event}} = 0.4$, $\delta_{\text{IP}} = 0.4$, $\delta_{\text{time}} = 0.2$).

After clustering, 25 attack sequences are formed, some of which only have scanning and probing behaviors.

4.2. Attack Scenario Mining. We analyze the set of attack sequences generated above and discard any attack sequence where the last alarm is generated in the first or second phase of an attack event. There were 14 relatively complete attacks. Since attackers used different network hosts to launch attacks during the APT attack, attack alarms generated in one attack-planning process were distributed into different attack sequences, resulting in incomplete attack sequences. Eight attack sequences that conformed to the planning process were found after analyses were made. The eight attack sequences were then converted to directed graphs according to the algorithm in Section 3.4. Some directed graphs are shown in Figure 4.

The probability transfer matrix corresponding to all attack sequences is shown in Figure 5.

The probability transfer matrix is then converted into an attack scenario as shown in Figure 6.

Figure 6 shows that we can construct attack scenario by our proposed method. For an attack sequence, the attack means gradually changing from elementary to advanced, obtained permissions are getting more and more powerful, and suspicious files transmission or Trojan back door connection should happen in the end, which meet the phased characteristic of an APT attack. In order to verify the validity of the mined attack scenario, we analyze the attack scenario of an APT attack case named "Sea Lotus" detected by a certain organization. The attack event was unfolding when an intranet host user clicked the malicious mail attachment disguised as a normal file, resulting in the server's terminal virus infection and being controlled by an illegal APT organization, who then implanted the Trojan file qq.exe.bak in the folder c:\users\user\appdata\roaming\tencent, where communication was made with a hacker IP address and a small-amount data transmission was done. By analysis of the whole APT process, we find a series of events including alarms against a large number of malicious mail attachments, DNS requests from malicious domains, suspicious file transfers, and malicious domain connections. These events

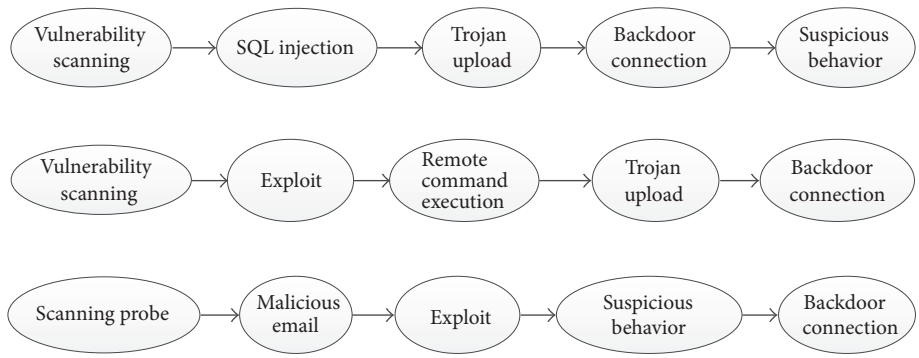


Figure 4: Part of the directed graph of an attack sequence.

	Scanning probe	CGI attack	Trojan upload	Exploit	Backdoor connection	Suspicious behavior	Remote command execution	Malicious email	Suspicious file directory access
Scanning probe		0.25		0.5				0.25	
CGI attack			0.33			0.67			
Trojan upload					0.5		0.2		0.3
Exploit						0.33	0.33		0.33
Backdoor connection						1			
Suspicious behavior					1				
Remote command execution			0.33						0.67
Malicious email				0.6		0.4			
Suspicious file directory access					1				

Figure 5: The probability transfer matrix.

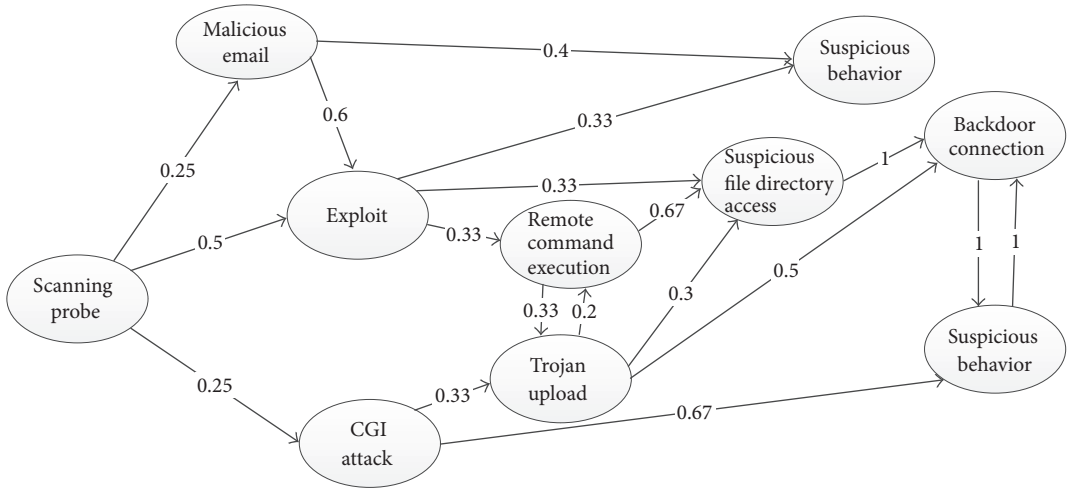


Figure 6: An attack scenario.

conform to the mined APT attack scenario, meaning that the attack scenario we mined reflects the true APT attack chain and is useful for the detection and defense of an APT attack.

Additionally, we use accuracy rate $R_r = N_c/N_n$ to evaluate the APT attack scenario mining method, where N_c is the effectively mined attack sequence by our mining method and N_n is the APT attack sequence that should be mined out. All the mined attack sequences include some attack sequences that do not match our attack strategy, and we delete such attack sequences, R_r= 80%.

Feng et al.'s paper [11] used alert clustering based on the correlation of IP addresses to produce alarm cluster sets. We

Table 2: Results of clustering algorithms with different dimensions.

Clustering algorithm	Dimensions of clustering algorithm	Result
Fuzzy Clustering which includes the attack-event dimension	alarm_event, IP address, timestamp	Escalating attack mechanism, and there is higher correlation in an attack sequence
Fuzzy Clustering which exclude the attack-event dimension	IP address, timestamp	Attack events intersect, and there is small correlation in an attack sequence

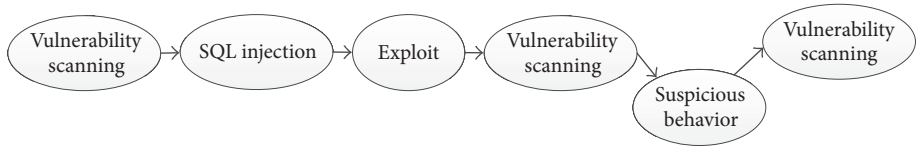


Figure 7: Attack sequence fragment.

use the clustering algorithm excluding the attack event dimension to process the same experimental data and do not classify attack events. We cluster with IP address and timestamp and analyze attack sequence set without considering attack events. We can get an attack sequence as shown in Figure 7.

In Figure 7, detection scanning occurs after either vulnerability exploitation or suspicious behaviors and attack events of different phases intersect. The clustering method that uses only two dimensions of the IP and the timestamp tends to correlate the attacking processes on the same asset by multiple attackers and/or certain misoperations to one attack sequence, resulting in decreased correlation between different alarms in an attack sequence. The results of clustering algorithms with different dimensions are shown in the Table 2.

By adding an extra dimension of the attack event, our proposed method can reduce the occurrence of decreased correlation. Thus, our method increases the degree of correlation between different alarms in an attack sequence, and it does not rely on any attack model built with expert knowledge.

5. Conclusion

In this paper, the attack events in an IDS log are classified based on the IKC model, the method of fuzzy clustering is used to correlate the alarm logs to produce the attack sequence set, and the nonaftereffect property of the probability transfer matrix is used to excavate the attack scenario from the attack sequence set. Based on the phased characteristic of an APT attack, in this paper, the purpose of an APT attack in each phase is analyzed and attack events are classified. In addition to the IP address and the timestamp, the use of the attack event as another key dimension in fuzzy clustering also improves the correlation degree of alarms in the same attack sequence. The effectiveness of this method has been proved by experiments. The method proposed in this paper can automatically construct attack scenario based on IDS logs and the attack scenario provides guidance for the detection and defense of APT attacks.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National key Research and Development Program of China (2016YFB0800903) and the NSF of China (U1636212, U1636112).

References

[1] P. Chen, L. Desmet, and C. Huygens, "A study on advanced persistent threats," in *Communications and Multimedia Security: 15th IFIP TC 6/TC 11 International Conference, CMS 2014, Aveiro, Portugal, September 25-26, 2014. Proceedings*, vol. 8735 of *Lecture Notes in Computer Science*, pp. 63–72, Springer, Berlin, Germany, 2014.

[2] I. Friedberg, F. Skopik, G. Settanni, and R. Fiedler, "Combating advanced persistent threats: from network event correlation to incident detection," *Computers & Security*, vol. 48, no. 7, pp. 35–57, 2015.

[3] X.-S. Zhang, W.-N. Niu, G.-W. Yang et al., "Method for APT prediction based on tree structure," *Journal of University of Electronic Science and Technology of China*, vol. 45, no. 4, pp. 582–588, 2016.

[4] SkyEye: OceanLotus APT Report [2015-05-29], https://ti.360.net/static/upload/report/file/OceanLotusReport.pdf.

[5] K. Munro, "Deconstructing flame: The limitations of traditional defences," *Computer Fraud and Security*, vol. 2012, no. 10, pp. 8–11, 2012.

[6] C. Choi, J. Choi, and P. Kim, "Abnormal behavior pattern mining for unknown threat detection," *Computer Systems Science & Engineering*, vol. 32, no. 2, pp. 171–177, 2017.

[7] Y. Fu, H. LI, X.-p. Wu, and J. Wang, "Detecting APT attacks: a survey from the perspective of big data analysis," *Journal on Communications*, vol. 36, no. 11, pp. 1–14, 2015.

[8] C. Tankard, "Advanced Persistent threats and how to monitor and deter them," *Network Security*, vol. 2011, no. 8, pp. 16–19, 2011.

[9] E. M. Hutchins, M. J. Cloppert, and R. M. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," in *Proceedings of the 6th International Conference on Information Warfare and Security (ICIW '11)*, pp. 113–125, Curran Associates Inc, Washington, Wash, USA, March 2011.

[10] H.-B. Mei, J. Gong, and M.-H. Zhang, "Research on discovering multi-step attack patterns based on clustering IDS alert sequences," *Journal on Communications*, vol. 32, no. 5, pp. 63–69, 2011.

[11] X. Feng, D. Wang, M. Huang, and J. Li, "A mining approach for causal knowledge in alert correlating based on the markov property," *Jisuanji Yanjiu yu Fazhan/Computer Research and Development*, vol. 51, no. 11, pp. 2493–2504, 2014.

Hindawi
Security and Communication Networks
Volume 2017, Article ID 6898617, 15 pages
https://doi.org/10.1155/2017/6898617

Review Article

A Survey on Breaking Technique of Text-Based CAPTCHA

Jun Chen,[1,2] Xiangyang Luo,[1] Yanqing Guo,[3] Yi Zhang,[1] and Daofu Gong[1]

[1]*State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450002, China*
[2]*Henan Institute of Science and Technology, Xinxiang 453003, China*
[3]*Dalian University of Technology, Dalian 116024, China*

Correspondence should be addressed to Xiangyang Luo; luoxy_ieu@sina.com

Received 25 September 2017; Accepted 27 November 2017; Published 24 December 2017

Academic Editor: Zhenxing Qian

The CAPTCHA has become an important issue in multimedia security. Aimed at a commonly used text-based CAPTCHA, this paper outlines some typical methods and summarizes the technological progress in text-based CAPTCHA breaking. First, the paper presents a comprehensive review of recent developments in the text-based CAPTCHA breaking field. Second, a framework of text-based CAPTCHA breaking technique is proposed. And the framework mainly consists of preprocessing, segmentation, combination, recognition, postprocessing, and other modules. Third, the research progress of the technique involved in each module is introduced, and some typical methods of segmentation and recognition are compared and analyzed. Lastly, the paper discusses some problems worth further research.

1. Introduction

As a multimedia security mechanism, CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart [1]) also called Human Interactive Proofs (HIP [2]), can protect multimedia privacy. Now, it has been successfully applied to Google, Yahoo, Microsoft, and other major websites. In order to verify security and reliability of CAPTCHA, the breaking technology came into being. It involves image processing, pattern recognition, image understanding, artificial intelligence, computer vision, and many other disciplines. The research on CAPTCHA breaking has great value in research and application. First of all, CAPTCHA breaking can verify the security of existing CAPTCHAs, and it can promote the development of CAPTCHA design technique. Secondly, the CAPTCHA is an integral part of artificial intelligence and an important prerequisite to actualize natural human-computer interaction. Finally, the research of breaking CAPTCHA not only constantly refreshes limits to Turing test, but also can be applied in other fields such as digital paper-based media, speech recognition, and image labeling.

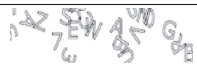
In recent decades, with the continuous development of CAPTCHA technology, relevant literature sources are abundant day by day. In 2013, [3] introduced CAPTCHAs of the time and attacks against them; the authors investigated the robustness and usability of CAPTCHAs at the time and discussed ideas to develop more robust and usable CAPTCHAs. Five years later, it is necessary to reorganize the emerging literature sources. Based on the research of text-based CAPTCHA breaking technique, this paper will review the relative research and prospect future trends.

The remainder of this paper is organized as follows: Section 2 briefly introduces the text-based CAPTCHA. Section 3 provides an overview of the text-based CAPTCHA breaking technique. Sections 4–8 describe main steps in the overall framework of the text-based CAPTCHA breaking technique. Section 9 points out some problems which can be further studied. Section 10 concludes up the full manuscript.

2. Overview on Text-Based CAPTCHA

In September 2000, the Carnegie Mellon University (CMU) research team designed the first commercial CAPTCHAs-Gimpy series text-based CAPTCHAs to resist malicious advertisements scattered by illegal scripting programs in the Yahoo chat room. At the same time, the research on CAPTCHA design and breaking also started. In 2002 and

Table 1: Typical types of text-based CAPTCHA and their features.

Type	Example	Source	Features
Solid CAPTCHA		Discuz!	Character independent, texture background, some interference
		Slashdot	A large number of interference lines and noise points
		Gimpy	Multiple strings, overlap, distortion
		Google	Unfixed length, distortion, adhesion
		Microsoft	Double-string, unfixed length, uneven thickness, tilting, adhesion
Hollow CAPTCHA		QQ	Hollow, shadows, interference shapes
		Sina	Hollow, adhesion, interference lines
		Yandex	Hollow, virtual contours, distortion, adhesion, interference lines
Three-dimensional CAPTCHA		Scihub	Hollow, shadows, interference lines, noise points
		Teabag	Grids, protrusion, distortion, background and character blending
		Parc	Colorful, shadow, rotation, zoom Special characters
Animation CAPTCHA		Program generating	Multiple characters jumping
		Hcaptcha	Multilayer character images blinking transformation

2005, the international seminars on HIP have been held, and a large number of related research results were published. In subsequent years, many research results were reported in international conferences including CVPR, NIPS, CCS, and NDSS. Many internationally renowned universities and research institutions have established research groups on CAPTCHA technology, such as CMU [1, 8–14], PARC [15–19], UCB [16, 17, 20, 21], Microsoft [2, 22–27], Google [28–30], Bell Laboratory [31, 32], Yan et al. [4, 33–42], Xidian University [41–47], and University of Science and Technology of China [48, 49]. In addition, many websites offer CAPTCHA services in public such as CAPTCHA [10], BotBlock [50], JCAPTCHA [51], and HCaptcha [52]. And some research groups focus on CAPTCHA recognition, such as PWNtcha [53], Captchacker [54], aiCaptcha [55], and Gery Mori [56].

The security of text-based CAPTCHA mainly depends on the visual interference effects [25], including rotation, twisting, adhesion, and overlap. The typical types of text-based CAPTCHA and their features are shown in Table 1.

To resist machine recognition, the text-based CAPTCHA's security is often protected by a series of technologies. From Table 1, we can sum up the following main features of the text-based CAPTCHA.

(1) A large enough character set. Only when a character set is large enough, the total number of CAPTCHA strings is large enough to resist violent breaking.

(2) The characters with distortion, adhesion, and overlap. Using characters with distortion, adhesion, and overlap, the breaking methods cannot easily segmented a CAPTCHA image into single characters.

(3) The characters are different in size, width, angle, location, and fonts. When comparing features of different characters, the various transformations may reduce recognition accuracy.

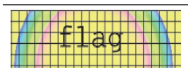
(4) The strings with unfixed length. In a CAPTCHA scheme, strings with unfixed length can increase breaking difficulty to a certain extent.

(5) Hollow characters and broken contours. Compared with the solid characters, hollow character's features are less, and broken contours can effectively resist the filling attack.

(6) The color and shape of complex backgrounds are similar to those of characters. If the images meet these conditions, the noise is difficult to remove. This may reduce recognition accuracy.

The above features effectively enhance text-based CAPTCHAs' security and bring great challenges to the CAPTCHA breaking research at the same time.

Table 2: Comparison of typical methods based on segmentation for breaking nonadherent CAPTCHA.

Example	Source	Success rate	Reference	Breaking method	Year
	Gimpy-r	78%	[57]	Segmentation: character gap Recognition: distortion evaluation	2004
	EZ-Gimpy	97.9%	[58]	Segmentation: connected region Recognition: distortion evaluation	2004
	Captcha-service	100%	[34]	Segmentation: vertical projection Recognition: statistical character pixels	2007
	Ego-share	92.2%	[5]	Segmentation: connected region Recognition: SVM	2009
	Ge-Captcha	100%	[59]	CW-SSIM	2010

Note. SVM: support vector machine, CW-SSIM: complex wavelet based structural similarity.

3. Research Progress of Breaking Text-Based CAPTCHA

For all kinds of text-based CAPTCHA schemes, the breaking methods are also various. According to whether there is segmentation or not, the existing breaking methods be contained in two categories.

3.1. Text-Based CAPTCHA Breaking Methods Based on Segmentation. The text-based CAPTCHA breaking based on segmentation has different processing methods for different objects and results. When there is no adherent character, individual characters are obtained using vertical projection and connected component with good effect. As shown in Table 2, the success rates of nonadherent character CAPTCHA range from 78% to 100%.

However, it had little success in adherent characters. Therefore, more complicated methods, such as different width, character features, and character contours, have been proposed one after another. With more and more antisegmentation technologies in CAPTCHA field, obtaining individual characters is becoming harder and harder. Then the researchers proposed the segmentation methods for obtaining character components by character structure, filters, and so forth. As can be seen from Table 3, the success rates of CAPTCHA breaking are generally low, with only a few higher than 80%.

3.2. Text-Based CAPTCHA Breaking Methods Based on Nonsegmentation. The text-based CAPTCHA breaking methods based on nonsegmentation can directly recognize preprocessed CAPTCHA images. The breaking method's success rate relies on recognition technique. In early stage, different pattern matching algorithms such as shape context [20] and similarity [57] are used for recognition. Later, with the improvement of the success rates of individual character recognition, researchers focus on the character segmentation technique. However, the text-based CAPTCHA design uses antisegmentation technique, which can prevent obtaining complete and individual characters. Nowadays with the advantage of deep learning, the breaking based on nonsegmentation will bounce back. The success rates of typical text-based CAPTCHA breaking methods based on nonsegmentation are as shown in Table 4.

3.3. The Framework of Text-Based CAPTCHA Breaking Technique. With the improvement of text-based CAPTCHA design, the breaking technique changes to meet it. The early text-based CAPTCHA contains nonadherent characters. The breaking technique is the traditional framework of "preprocessing + segmentation + recognition." In recent years, most of the text-based CAPTCHAs use CCT (Crowded Characters Together). Therefore, various breaking frameworks come into being, for example, "preprocessing + recognition," "preprocessing + recognition + postprocessing," "preprocessing + segmentation + combination + recognition," and "preprocessing + segmentation + combination + recognition + postprocessing."

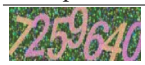
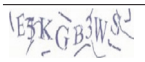
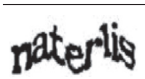
In this paper, the existing frameworks are integrated into an overall framework of text-based CAPTCHA breaking, as shown in Figure 1. The framework mainly consists of preprocessing, segmentation, combination, recognition, postprocessing, and other modules. The research progress of each module will be described in the following.

4. Preprocessing Methods of Breaking Text-Based CAPTCHA

The CAPTCHA preprocessing is the first step of CAPTCHA image processing before segmentation and recognition. Its main purpose is to highlight the information related to characters in a given image and to weaken or eliminate interfering information. The preprocessing of existing CAPTCHA breaking methods mainly includes image binarization, image thinning, denoising, and so on.

4.1. Image Binarization. Image binarization is to highlight interesting objects' contour and to remove noises in background. The key to binarization is to select an appropriate

Table 3: Comparison of typical methods based on segmentation for breaking adherent CAPTCHA.

Example	Source	Success rate	Reference	Breaking method	Year
	Google, Yahoo	4.89%–66.2%	[2]	Segmentation: width Recognition: CNN	2004
	Microsoft Google Yahoo	61% 8.7% 25.9%	[4]	Segmentation: color filling and projection Recognition: CNN	2008
	Hotmail	40%	[5]	Segmentation: change width Recognition: SVM Post-processing: DP search	2009
	MSN Yahoo	18% 45%	[6]	Segmentation: projection and central	2010
	Megaupload	78%	[36]	Segmentation: color filling Combination: nonredundancy Recognition: CNN	2010
	reCAPT-CHA Google	33% 46.75%	[38]	Segmentation: character structure feature Recognition: CNN	2011
	Yahoo	54.7%	[44]	Segmentation: projection and character feature Recognition: OCR	2012
	Yahoo	36%–89%	[41]	Segmentation: color filling Combination: redundancy Recognition: CNN Postprocessing: DFS	2013
	Microsoft	5.56% 57.05%	[60]	Different width/location segmenting and template matching	2015
	reCAPT-CHA	40.4%–94.3%	[61]	Segmentation: trichromatic code Recognition: SVM	2015
	Yahoo	57.3%–76.7%	[7]	Edge and fuzzy logic segmentation and recognition	2015
	Microsoft	5%–77.2%	[42]	Segmentation: Log-Gabor filter Combination: redundancy Recognition: KNN Postprocessing: DP search	2016
	MSN	27.1%–53.2%	[48]	Segmentation: different width Recognition: BPNN	2016

Note. CNN: convolutional neural network, DP: dynamic programming, OCR: optical character recognition, DFS: depth first search, KNN: k-nearest neighbor, BPNN: back-propagation neural network.

threshold. When the threshold is applied to the whole image, it is called the global threshold method; otherwise, it is called the local threshold method. If the threshold is not fixed during processing, it is called variable threshold method or dynamic threshold method. The common thresholding methods are Sauvola and Pietikainen's method [65], Otsu's method [66], and so on.

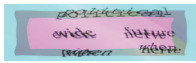
4.2. Image Thinning. Image thinning is to process the character's contour as skeleton. It must not change the character's adhesion. Its purpose is to highlight image contour and to simplify subsequent processing. The thinning algorithms contain two categories: noniterative algorithm and iterative algorithm. The common thinning algorithms include Hilditch algorithm [67] and Zhang and Suen algorithm [68].

4.3. Image Denoising. In order to resist breaking, there are noises and interference lines in CAPTCHA images. In addition, some noises are generated during grayscale and binarization. Therefore, we need to denoise CAPTCHA image. The typical methods are as shown in Table 5. We should choose the effective denoising method according to actual situation.

5. Segmentation Methods of Breaking Text-Based CAPTCHA

The segmentation aims to get individual characters or character components. There are the segmentation methods based on individual characters and the segmentation methods based on character components.

Table 4: Comparisons of typical methods based on nonsegmentation for breaking adherent CAPTCHA.

Example	Source	Success rate	Reference	Breaking method	Year
	EZ-Gimpy Gimpy	92% 33%	[20]	Shape context matching algorithm	2003
	EZ-Gimpy	99%	[57]	Correlation algorithm	2004
	Program generation	55%	[62]	RNN	2011
	Program generation	54.9%	[63]	2D LSTM-RNN	2013
	reCAPTCHA	99.8%	[30]	DCNN	2013
	reCAPTCHA	31.75%	[64]	HMM	2015

Note. RNN: recurrent neural network, 2D LSTM: 2-dimensional long short-term memory, DCNN: spatial displacement of the neutral network, HMM: Hidden Markov model.

5.1. Segmentation Methods Based on Individual Characters. The segmentation methods based on individual characters segment a CAPTCHA image to individual characters. For individual characters, we can use segmentation methods based on character projection and connected components. For CCT characters, we can use segmentation methods based on character width, connected feature, and character contour.

5.1.1. Segmentation Methods Based on Character Projection. The segmentation methods based on character projection determine the optimal segmentation position by analyzing the number of pixels projected under different conditions. This method applies to recognizing CAPTCHA characters without adhesion or slight adhesion. However, its effect is not obvious for the seriously adherent and distorted characters. The typical methods include vertical projection segmentation, horizontal projection segmentation, and guideline projection segmentation.

Using (1), [61] defines three-color bar code to segment reCAPTCHA images:

$$\text{Three-color Bar}\,(x)\begin{cases}\text{Blue}, & \text{for } H_{\Sigma}(x)=0,\\ \text{White}, & \text{for } H_{\Sigma}(x)=1,\\ \text{Black}, & \text{for } H_{\Sigma}(x)>1,\end{cases}\quad(1)$$

where $H_{\Sigma}(x)$ represents the total of object pixels in the xth column. In three-color bar a column is colored in blue if there is not any pixel that belongs to character in the column ($H_{\Sigma}(x) = 0$). If there is only one pixel in column ($H_{\Sigma}(x) = 1$), the column is encoded by white. Finally, the black corresponds to the column with more than one object pixel ($H_{\Sigma}(x) > 1$), as shown in Figure 2(a). After denoising, the optimal segmentation line is determined in the middle of blue bar or white bar, as shown in Figure 2(b).

5.1.2. Segmentation Methods Based on Connected Components. The segmentation methods based on connected components effectively segment individual characters using different connected components in an image. For slope and distortion characters, this method is effective. However, it is limited by adherent characters.

Reference [4] tried to segment Microsoft MSN CAPTCHA by combining connected components and vertical projection, as shown in Figure 3. First, different connected components are marked with different colors. And then the character blocks are generated according to different colors. Finally, strings are segmented to individual characters using the vertical projection feature, with a success rate of more than 90%.

5.1.3. Segmentation Methods Based on Character Width. The segmentation methods based on character width are suitable for CAPTCHA images which are not easily segmented to individual characters. [60] used different widths (the average width of 0.75 times, 1 time, 1.5 times, and 2 times) to segment an image. Thus, each character corresponds to four recognition results, from which to find an optimal segment as the final recognition result. In addition, [5] did not take the average width as standard; they gave a set of character segments between the minimum width and the maximum width and then determined the optimal segmentation scheme using dynamic programming, as shown in Figure 4.

5.1.4. Segmentation Methods Based on Character Feature. The segmentation method based on character features uses the features of CAPTCHA string, including inside features and outside features. Reference [38] classifies characters according to their own inside features, and each class contains the characters as shown in Table 6.

Reference [6] segments characters according to outside features among them. This paper proposes a new segmentation algorithm called middle-axis point separation

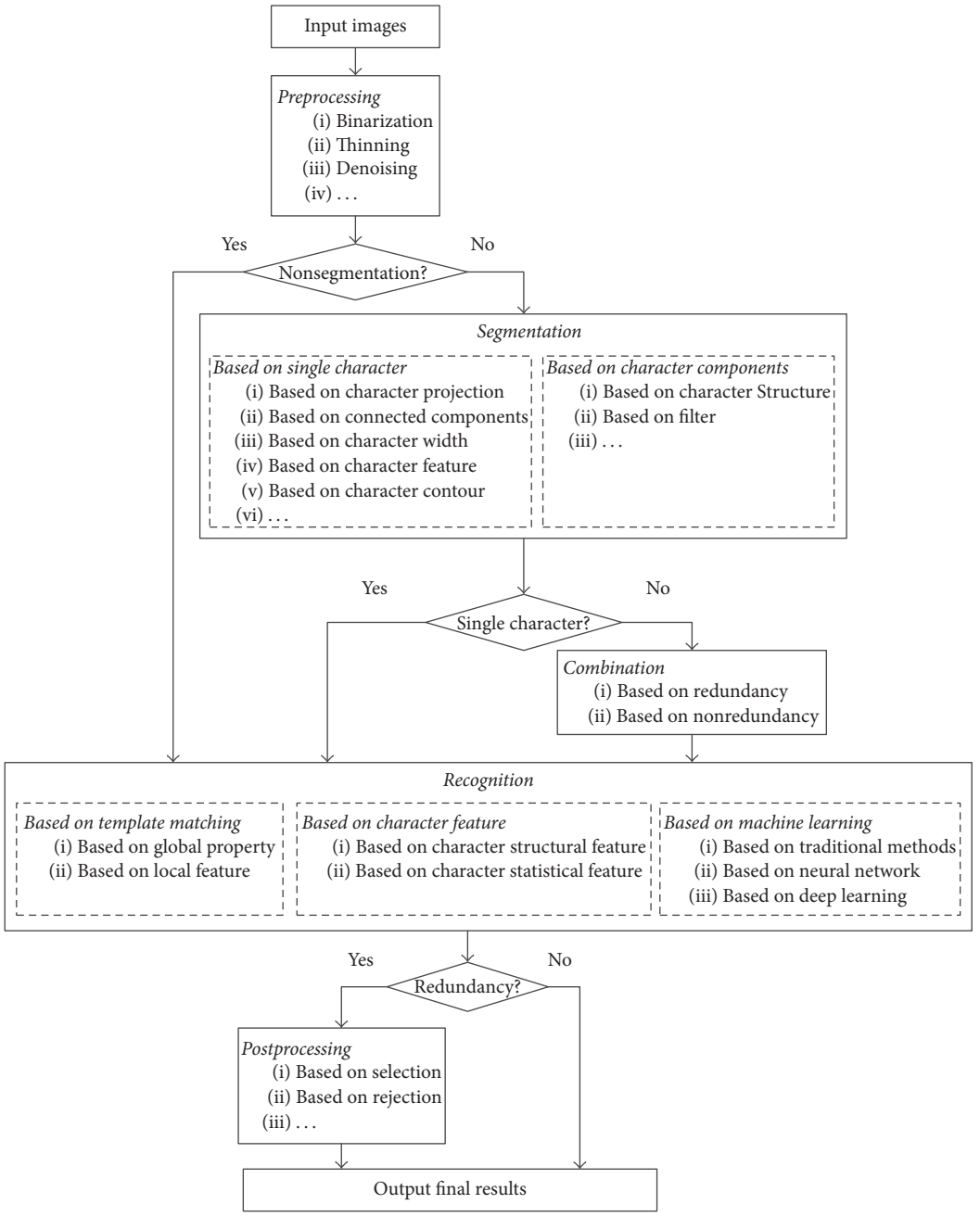
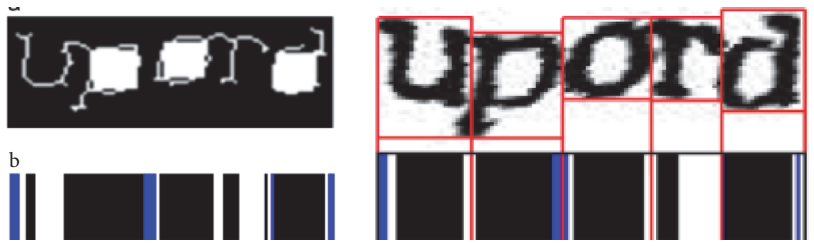


Figure 1: The framework of text-based CAPTCHA breaking technique.



(a) Original three-color bar (b) Denoised three-color bar

Figure 2: Three-color bar corresponding to CAPTCHA image.

Table 5: Comparisons of common denoising methods.

Denoising method	Typical algorithm	Implementation	Advantages	Disadvantages
Denoising method based on filter in the spatial domain	Average filter	The gray value of pixel is replaced by the mean of its neighboring pixels gray values.	The irrelevant details and gaps are removed.	The image is blurred.
	Median filter	The gray value of pixel is replaced by the median of its neighboring pixels gray values.	Remove effectively the salt and pepper noise, speckle noise.	Not applied to the image with many dots, lines, and spires.
	Wiener filter	The minimum mean square error criterion is used to adjust the filter effect.	Remove effectively the Gaussian noises.	Computation is complex.
Denoising method based on Gibbs and Hough transform	Gibbs	Markov random field theory.	Remove effectively noise points.	Not applied to irregular interference line.
	Hough transform	The straight line in the image is detected by using the point line duality of image space and Hough parameter space.	Remove effectively interference lines.	
Denoising method based on morphology	Open operation	First corrosion to expansion.	Smooth contours, cut off narrow lines, and eliminate fine.	The effect of denoising varies with operation mode and the size and shape of structural elements; the experiment needs to be repeated; the adaptability is poor.
	Close operation	First expansion to corrosion.	Smooth contour and fill holes, gaps, and fracture of contour line.	
Denoising method based on connected component	Connected component	The recursive method is used to find the connected domain to deal with pixel points, and then denoising based on gray features and morphological features of connected domain.	Remove effectively the noise interference, and the original details of the characters are generally not lost.	Need to analyze character's properties; hard to determine distinguish features.
Denoising method based on wavelet transform	Wavelet transform	Find the best mapping of original image in the wavelet transform domain to restore the original image.	Retain more image details.	Complex computation and it needs to adjust relative parameters.

Figure 3: Segmented CAPTCHA image in [4].

Table 6: Character class table.

Class	Dot	Circle	Cross	S	V
Characters	i, j	a, b, d, e, g, o, p, q	t, f	s, z	v, w, y

for CAPTCHAs. The algorithm utilizes the central pixel in background between two disconnected object pixels as segmentation points (see Figure 5).

5.1.5. Segmentation Methods Based on Character Contour. The segmentation method based on character contours is to analyze geometric features of character contours, so as to determine the appropriate segmentation lines. Reference [7] tried to connect connection edge points between two merged characters and determined the optimal segmentation line by confidence, as shown in Figure 6.

5.2. Segmentation Methods Based on Character Components. The segmentation methods based on character components produce multiple character components, rather than individual characters. The segmentation methods are mainly base on character structure or filter.

5.2.1. Segmentation Methods Based on Character Structure. Using structural feature of characters with black components and white components, [36] segmented a seriously overlapped string to multiple components. First, locate black components, as shown in Figure 7(b). And then, locate white components, as shown in Figure 7(c). Finally, identify black components of each character and the shared white components.

In [41], a CAPTCHA image contains several hollow characters, whose contours naturally form several closed regions (see Figure 8(a)). According to this structural feature, a character is segmented to several character components by color filling (see Figure 8(b)).

5.2.2. Segmentation Methods Based on Filter. Reference [42] is the first to apply Gabor filters for breaking CAPTCHAs, which extracts character components along four directions by convolving a CAPTCHA image with each of four filters, respectively, as shown in Figure 9. The segmentation method is not limited by adhesion, distortion, and overlap and is suitable for many kinds of characters.

In summary, the contrast among segmentation methods is given. As can be seen in Table 7, each segmentation method applies to different types of characters. It is only the individualized segmentation method that can obtain good results.

6. Combination Methods of Breaking Text-Based CAPTCHA

An individual character after segmentation can be recognized directly. But character components need to be combined into an individual character to be recognized. According to the number of generated candidate characters, combination technologies can be divided into two categories: the combination technique based on redundancy and the combination technique based on nonredundancy.

6.1. Combination Methods Based on Redundancy. The number of candidate characters generated by combination technique based on redundancy is more than the number of real characters. In [42], each character fragment is labeled in order from top to bottom and left to right, and then the components are combined on the idea of jigsaw puzzle to generate candidate characters.

6.2. Combination Methods Based on Nonredundancy. The number of candidate characters generated by combination technique based on nonredundancy is equal to the number of actual characters. In [36], the character components are nonredundant. The overlap area strokes may be reused to compose a complete character. Figure 7(a) shows a Megaupload CAPTCHA image. Figure 10 gives the combined four characters. The final success rate of combination is 78.25%.

7. Recognition Methods of Breaking Text-Based CAPTCHA

Nowadays, the recognition methods used in text-based CAPTCHA system include three categories: template matching, character feature, and machine learning.

7.1. Recognition Methods Based on Template Matching. Template matching is to compare similarity of each pixel between characters and every template and to find the highest similarity. According to matching range, there are the matching recognition methods based on global property and the matching recognition methods based on local feature.

7.1.1. Matching Recognition Methods Based on Global Property. The matching recognition methods based on global property is traverse scanning. Within search area, the optimal match point to each pixel is found by regional correlation matching calculation. Because many templates matching each pixel will be pretty slow, [45] proposes the second template matching algorithm to improve efficiency. Only if a rough matching is successful, an exact matching needs to be made.

7.1.2. Matching Recognition Methods Based on Local Feature. The shape context is a simple local feature shape descriptor. Its basic idea is to convert the matching problem of image into the matching problem of feature point set. In 2003, Mori and Malik [20] used shape context to break the CAPTCHA of Gimpy and EZ-Gimpy. For good robustness to image

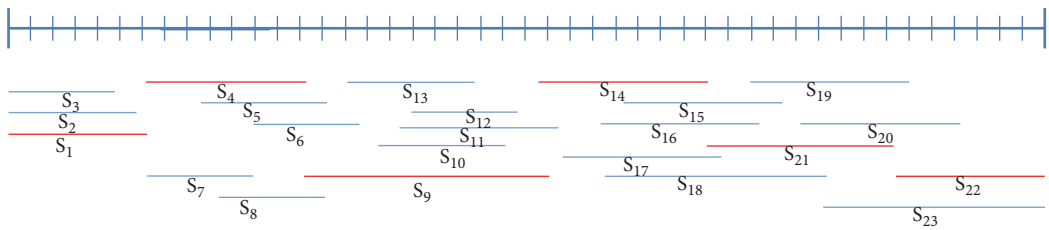


Figure 4: Character segments in [5].

Table 7: Comparisons of segmentation methods.

Segmentation methods	Main basis	Character features				Explanation
		Adhesion	Skew	Distortion	Overlap	
Segmentation based on individual character	Character projection	√	×	×	×	Suit strings with little adhesion rather than serious distorted, overlapped strings
	Connected components	×	√	√	×	Limited to overlapping rather than distortion
	Character width	√	×	×	×	Limited to severely distorted
	Character feature	√	×	×	×	Effective when character feature is notable and discriminant
	Character contour	√	√	√	×	Applied to the individual character with clear contour
Segmentation base on character components	Character structure	√	√	√	√	Only when character structure is easy to segment
	Filter	√	√	√	√	Wide application; the subsequent processing is complex

Figure 5: The middle-axis points in [6].

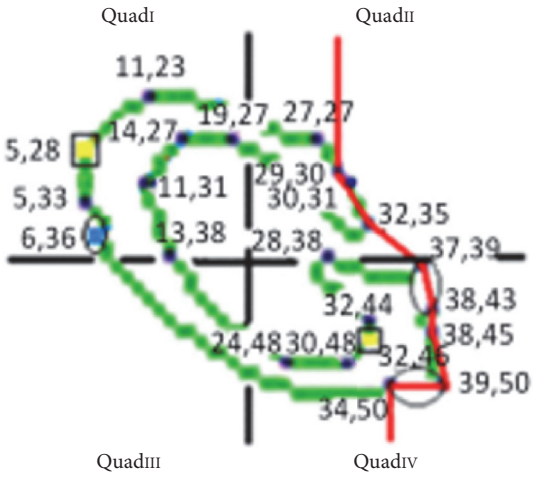


Figure 6: The optimal segmentation line in [7].

scaling and affine transformation, it is widely used in face recognition, CAPTCHA recognition, shape matching, and other fields.

7.2. Recognition Methods Based on Character Feature. Because the character of each CAPTCHA mechanism varies in design, we can define different methods according to the feature of characters, which is mainly based on character structural feature and character statistical feature.

7.2.1. Recognition Methods Based on Character Structural Feature. The structural feature can describe the details and structural information of characters, such as the number of loops, inflection point, convexo-concave degree, and cross points. For example, [46] uses the guidelines of characters (see Figure 11(a)) and closed loop detection (see Figure 11(b)) to break Yahoo CAPTCHA.

7.2.2. Recognition Methods Based on Character Statistical Feature. The recognition method based on character statistical feature uses commonly statistical features including pixel feature, projection feature, contour feature, and coarse mesh feature. This feature is robust to noise interference and is widely used in CAPTCHA recognition field. Reference [34] used the distinct pixel count for each of the letters A to Z (see Figure 12) to break captchaservice.org CAPTCHA with a near 100% success rate.

7.3. Recognition Methods Based on Machine Learning. The recognition methods based on machine learning is essentially using machine learning algorithms to correctly classify CAPTCHA characters. According to chronological order of

(a) Original image (b) Nonshared character components (c) Shared character components

Figure 7: An example of segmented CAPTCHA image in [36].

(a) Original image (b) Segmented image

Figure 8: An example of segmented CAPTCHA image in [41].

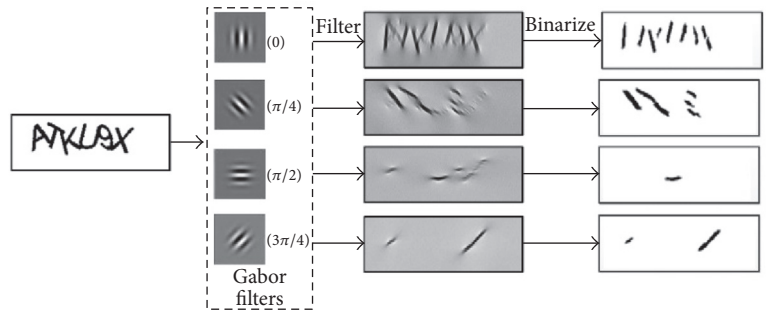


Figure 9: Segmentation CAPTCHA image by Gabor filters in [42].

Figure 10: Individual characters after combination in [36].

mainstream, it can be basically divided into three categories: traditional methods, neural network, and deep learning.

7.3.1. Recognition Methods Based on Traditional Methods. In the field of text-based CAPTCHA recognition, the most widely used traditional classifiers include SVM and KNN.

The idea of SVM is to separate classes via a hyperplane. The key is kernel function, which is responsible for mapping original features into high-dimensional space in a nonlinear way, thereby improving the separability for data. Reference [5] compared four kernel functions: RBF (Radius-Based Function), POLY (polynomial), LINEAR, and SIGMOID. The experimental results showed that the performance of the first two kernel functions was optimal.

KNN is based on the category of the nearest K samples to determine the category of a sample. Reference [42] tested SVM, BPNN (back-propagation neural network), template matching, CNN, and KNN. Among these classifiers, KNN achieved higher success rates on most of the schemes, but CNN was faster most of the time.

7.3.2. Recognition Methods Based on Neural Network. For the principle of parallel distributed operation in large number of neurons, the efficient learning algorithms, and the ability to imitate human cognitive systems, the neural network is very suitable to solve problems such as speech recognition and text recognition.

In [62], a BPNN used cross entropy for calculating the performance of a network with targets and outputs. Eventually, the system achieved an overall precision of 51.3%, 27.1%, and 53.2% for the CCT CAPTCHAs of Taobao, MSN, and eBay, respectively.

However, when applying neural network, we need to extract character features first. The quality of extracted features limits the final recognition rate to a certain extent.

7.3.3. Recognition Methods Based on Deep Learning. In recent years, deep learning has achieved remarkable achievements in recognition fields of text, image, audio, and so forth. The deep learning models commonly used in CAPTCHA recognition field are CNN, RNN, LSTM-RNN, and so forth.

CNN recognizes character images without feature extraction and has a certain degree of robustness in displacement, scale, and deformation. In the existing research results, a

(a) The guideline (b) The closed loop detection

Figure 11: The example of character structural features in [46].

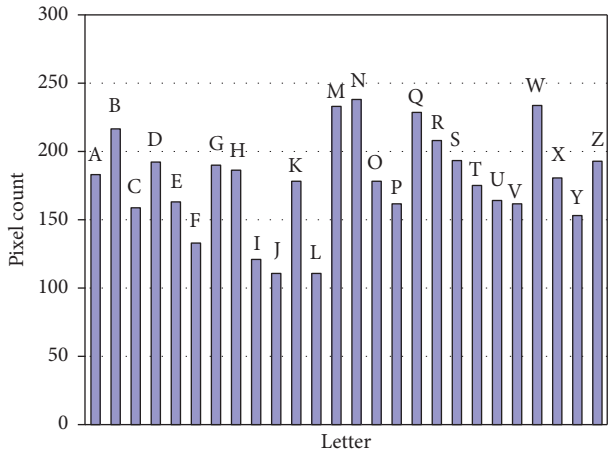


Figure 12: The pixel count for each of the letters A to Z in [34].

typical CNN is widely used [2, 4, 36, 38, 41] with a good recognition accuracy. Reference [30] trained large, distributed deep convolutional neural networks and achieved 99.8% accuracy in recognizing CAPTCHA images of reCAPTCHA.

However, due to lack of time dimension, CNN cannot combine context information in recognition. So RNN with feedback and time parameters was proposed to process time series data. Later, in order to solve vanishing gradient problem of RNN, LSTM was proposed in machine learning field. Reference [62] applied 2D LSTM-RNN in CCT CAPTCHAs recognition with a success rate of 55.2%. It innovatively obtained relative information not only in horizontal context, but also in vertical context.

In summary, a contrast among recognition methods is given, as shown in Table 8. According to the features of different networks, we should attempt to construct a new deep learning model by combining multiple networks. It will be a research trend in the field of text-based CAPTCHA recognition.

8. Postprocessing Methods of Breaking Text-Based CAPTCHA

In previous steps, some of character recognition results may be taken as final results directly, while others need to be further postprocessed. In postprocessing stage, the final result's reliability is ensured by simplification, selection, and optimization. According to different objects and methods, there are the postprocessing methods based on selection and the postprocessing methods based on rejection.

8.1. Postprocessing Methods Based on Selection. Usually, there are many redundant individual characters generated in previous steps. This requires selecting the most likely combined string as the final recognition result of CAPTCHA image. The selection strategies include the local optimization and the global optimization.

The local optimization selection only takes into account the recognition confidence optimality of an individual character. In [60], each character corresponds to several candidate characters with different widths. Therefore, the candidate character with the highest recognition confidence is selected as the final character.

The global optimization selection strives for the best results for all characters in an image. In [41], all candidate characters are found by the graph traversal, and then the string with the highest sum of characters recognition confidence values is taken as the final result, while in [5, 42], to avoid enumerating all candidate characters, a dynamic programming is used to determine the final result with the highest sum of characters' recognition confidence values directly. Compared with graph traversal, the dynamic programming is more effective and accurate.

8.2. Postprocessing Methods Based on Rejection. The purpose of postprocessing methods based on rejection is to determine whether the tested sample belongs to the types of training set by analyzing character recognition results. Therefore, the postprocessing methods based on rejection are a key to ensure high reliability of CAPTCHA recognition.

At present, the researchers have not been paid enough attention to the postprocessing methods based on rejection. To the best of our knowledge, there is only one paper [62] in CAPTCHA field. It considers multiple features, such as confidence, string length, character spaces, and the first and the last character of a string, to determine whether a candidate character should be rejected or not.

9. Some Problems Worth Further Research

As stated above, many achievements have been acquired. However, in view of the complexity of text-based CAPTCHA, there are still some issues worth exploring in depth in this field.

(1) Construction of Standard Test Database for Text-Based CAPTCHA. A rich and high quality text-based CAPTCHA image database is the necessary foundation for the research of text-based CAPTCHA breaking. At present, the researchers get CAPTCHA images mainly by web access and software

Table 8: Comparisons of recognition methods.

Recognition methods	Main basis	Typical methods	Advantages	Disadvantages
Recognition method based on template matching	Global property	Traversal search matching algorithm	The program is simple and suitable for standard character verification code.	The required template library is large; it depends on the choice of template matching.
	Local feature	Shape context matching algorithm	The image information is rich, and it is robust to image scaling and affine transformation.	Without rotation invariant.
Recognition method based on character feature	Character structure feature	Algorithm based on character structure feature	Sensitive to the details of characters; strong in distinguishing features.	The distortion is serious when there are noise interferences.
	Character statistical feature	Algorithm based on character statistical feature	Strong robustness against noise interference.	Targeted; application limited.
Recognition method based on machine learning	Template matching	SVM	Strong approximation ability and generalization ability; good adaptability and high accuracy for small sample space; suitable for two kinds of classification.	Not applied to infinite sample space.
		KNN	It is better to avoid the problem of imbalanced samples, which is suitable for overlapping samples of the same class.	Computation is complex; easy to misjudge in the domain with small sample size.
	Traditional method	BPNN	Flexible structure design, suitable for multiclass classification.	Slow convergence rate; depends on parameters.
	Deep learning	CNN	Accepts an input image directly; automatically extracts features; own robustness to displacement, scale, and deformation; high recognition accuracy.	Lack of time dimension; it could not identify using context information.
		RNN	Processes data in time series.	Time gradient may disappear.
		LSTM-RNN	Owns the time memory function; effective to prevent gradient disappear.	Unable to extract feature automatically.

generation. However, due to the diversity and timeliness of text-based CAPTCHA, it has not been possible to construct a common image database in the field of text-based CAPTCHA recognition. It is necessary to collect, classify, organize, and establish the text-based CAPTCHA images database. The database can provide the reliable training and testing data for research work and also provide the premise and basis of unified evaluation for various methods in this field.

(2) Multitype CAPTCHA Recognition. At present, only when training set and test set belong to the same type, the classifier can effectively recognize CAPTCHAs. In fact, there are a variety of character changes in a CAPTCHA. Therefore, it is an arduous and important task to design a reasonable classifier to recognize various types of CAPTCHAs.

(3) Segmentation-Free CAPTCHA Recognition. After more than ten years of development, the text-based CAPTCHA breaking has achieved a high success rate in individual character. However, the breaking success rate of the CAPTCHA string is generally low, and the results are less. With the wide application of CCT strings in text-based CAPTCHA, the problem of segmentation-free CAPTCHA recognition needs to be solved urgently. Now deep learning may provide new ideas and technical means to solve this problem.

(4) Application of Deep Learning Model. At present, in the field of CAPTCHA recognition, deep learning model can achieve better results than traditional methods. The most frequently used methods are based on CNN and its improved methods, while other deep learning models such as DBN (Deep Belief Networks), RNN, LSTM/BLSTM/MDLSTM, and DRL (Deep Reinforcement Learning) were not well used in text-based CAPTCHA recognition. Furthermore, the study of the interrelationships and fusion applications between the various deep learning models is not thorough. We hope that newer and better deep learning models are proposed to make a breakthrough in CAPTCHA recognition, which will certainly promote the development in this field.

(5) Rejection of Text-Based CAPTCHA. With the development of CAPTCHA breaking technique, the reliability of recognition results is also increasing. In this regard, on one hand, we should improve the correct rate of recognition; on the other hand, we should guarantee the correct rejection. In the field of CAPTCHA recognition, the concept of rejection has not been well known to the researchers. Therefore, this study has a potential development space.

(6) Misrecognition of Confusable Characters. When using the deep learning network to extract character features automatically, the characters with similar features are easily confused. It has practical significance to improve the precision of feature extraction and the training methods in the deep learning network.

10. Conclusions

Based on detailed investigation and in-depth analysis, this paper reviews the progress of text-based CAPTCHA breaking technique. First of all, this paper introduces various text-based CAPTCHAs and focuses on their features. Second, according to whether there is segmentation or not, we classify the existing breaking methods of text-based CAPTCHA and summarize their features. Meanwhile, we propose a framework of text-based CAPTCHA breaking technique and introduce the modules contained in the framework one by one. Next, we compare and analyze the basic principles, advantages, and disadvantages of the existing methods from five aspects: preprocessing, segmentation, combination, recognition, and postprocessing. Finally, some problems worth further research are discussed.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 61379151, 61401512, 61572052, and U1636219), the National Key R&D Program of China (nos. 2016YFB0801303 and 2016QY01W0105), and the Key Technologies R&D Program of Henan Province (no. 162102210032).

References

[1] L. Von Ahn, M. Blum, and J. Langford, "Telling humans and computers apart automatically," *Communications of the ACM*, vol. 47, no. 2, pp. 56–60, 2004.

[2] K. Chellapilla and P. Y. Simard, "Using Machine Learning to Break Visual Human Interaction Proofs (HIPs)," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 265–272, ofAdvances in Neural Information Processing Systems, 2004.

[3] N. Roshanbin and J. Miller, "A survey and analysis of current CAPTCHA approaches," *Journal of Web Engineering*, vol. 12, no. 1-2, pp. 001–040, 2013.

[4] J. Yan and A. S. E. Ahmad, "A low-cost attack on a microsoft CAPTCHA," in *Proceedings of the 15th ACM conference on Computer and Communications Security, CCS'08*, pp. 543–554, USA, October 2008.

[5] F. Jean-Baptiste and R. Paucher, "The Captchacker Project," 2009, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.800.3065&rep=rep1&type=pdf.

[6] S.-Y. Huang, Y.-K. Lee, G. Bell, and Z.-H. Ou, "An efficient segmentation algorithm for CAPTCHAs with line cluttering and character warping," *Multimedia Tools and Applications*, vol. 48, no. 2, pp. 267–289, 2010.

[7] R. A. Nachar, E. Inaty, P. J. Bonnin, and Y. Alayli, "Breaking down Captcha using edge corners and fuzzy logic segmentation/recognition technique," *Security and Communication Networks*, vol. 8, no. 18, pp. 3995–4012, 2015.

[8] L. von Ahn, M. Blum, N. J. Hopper, and J. Langford, "CAPTCHA: using hard AI problems for security," in *Advances in cryptology—EUROCRYPT 2003*, vol. 2656 of *Lecture Notes in Computer Science*, pp. 294–311, Springer, Berlin, Germany, 2003.

[9] https://www.google.com/recaptcha.

[10] http://captcha.net/.

[11] http://www.captcha.net/captchas/bongo.

[12] A. Schlaikjer and A. Dual, "Use Speech CAPTCHA: Aiding Visually Impaired Web Users while Providing Transcriptions of Audio Streams," Tech. Rep. LTI-CMU-07-014, Carnegie Mellon University, Pittsburgh, Pa, USA, 2007.

[13] J. Tam, J. Simsa et al., "Improving Audio CAPTCHAs," in *Proceedings of the Symposium on Usable Privacy and Security*, 2008.

[14] J. Tam, S. Hyde, J. Simsa, and L. Von Ahn, "Breaking audio CAPTCHAs," in *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems, NIPS 2008*, pp. 1625–1632, can, December 2008.

[15] H. S. Baird and K. Popat, "Human Interactive Proofs and Document Image Analysis," in *Proceedings of the International Workshop on Document Analysis Systems*, vol. 2423 of *Lecture Notes in Computer Science*, pp. 507–518, Springer, 2002.

[16] A. L. Coates, H. S. Baird, and R. J. Fateman, "Pessimal print: A reverse turing test," in *Proceedings of the 6th International Conference on Document Analysis and Recognition, ICDAR 2001*, pp. 1154–1158, usa, September 2001.

[17] M. Chew and H. S. Baird, "Baffletext: A human interactive proof," in *Proceedings of the Document Recognition and Retrieval X*, pp. 305–316, USA, January 2003.

[18] R. Chow, P. Golle, M. Jakobsson, L. Wang, and X. Wang, "Making CAPTCHAs clickable," in *Proceedings of the 9th Workshop on Mobile Computing Systems and Applications, HotMobile 2008*, pp. 91–94, USA, February 2008.

[19] P. Golle, "Machine learning attacks against the asirra CAPTCHA," in *Proceedings of the 15th ACM conference on Computer and Communications Security, CCS'08*, pp. 535–542, USA, October 2008.

[20] G. Mori and J. Malik, "Recognizing objects in adversarial clutter: breaking a visual CAPTCHA," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 134–144, June 2003.

[21] M. Chew and J. D. Tygar, "Image Recognition CAPTCHAs," in *Proceedings of the 7th International Information Security Conference*, vol. 3225 of *Lecture Notes in Computer Science*, pp. 268–279, Springer.

[22] K. Chellapilla, K. Larson, P. Simard, and M. Czerwinski, "Designing human friendly human interaction proofs (HIPs)," in *Proceedings of the the SIGCHI conference*, p. 711, Portland, Oregon, USA, April 2005.

[23] P. Y. Simard, R. Szeliski, J. Benaloh, J. Couvreur, and I. Calinov, "Using character recognition and segmentation to tell computer from humans," in *Proceedings of the 7th International Conference on Document Analysis and Recognition, ICDAR 2003*, pp. 418–423, UK, August 2003.

[24] K. Chellapilla, K. Larson, P. Y. Simard, and M. Czerwinski, "Building segmentation based human-friendly human interaction proofs (HIPs)," in *Proceedings of the Second International Workshop on Human Interactive Proofs, HIP 2005*, pp. 1–26, usa, May 2005.

[25] K. Chellapilla, K. Larson, P. Simard, and M. Czerwinski, "Computers beat humans at single character recognition in reading based human interaction proofs (HIPs)," in *Proceedings of the 2nd Conference on Email and Anti-Spam*, usa, July 2005.

[26] J. Elson, J. R. Douceur, J. Howell, and J. Saul, "Asirra: A CAPTCHA that exploits interest-aligned manual image categorization," in *Proceedings of the 14th ACM Conference on Computer and Communications Security, CCS'07*, pp. 366–374, USA, November 2007.

[27] Y. Rui and Z. Liu, "ARTiFACIAL: Automated reverse turing test using FACIAL features," *Multimedia Systems*, vol. 9, no. 6, pp. 493–502, 2004.

[28] K. A. Kluever and R. Zanibbi, "Balancing usability and security in a video CAPTCHA," in *Proceedings of the 5th Symposium On Usable Privacy and Security, SOUPS 2009*, USA, July 2009.

[29] R. Gossweiler, M. Kamvar, and S. Baluja, "What's up CAPTCHA? A CAPTCHA based on image orientation," in *Proceedings of the 18th International World Wide Web Conference, WWW 2009*, pp. 841–850, Spain, April 2009.

[30] I. J. Goodfellow, Y. Bulatov, J. Ibarz et al., "Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks," 2014, https://www.researchgate.net/publication/259399973_Multi-digit_Number_Recognition_from_Street_View_Imagery_using_Deep_Convolutional_Neural_Networks.

[31] T.-Y. Chan, "Using a test-to-speech synthesizer to generate a reverse Turing test," in *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence*, pp. 226–232, Sacramento, Calif, USA, 2003.

[32] G. Kochanski, D. Lopresti, and C. Shih, "A reverse turing test using speech," in *Proceedings of the 7th International Conference on Spoken Language Processing, ICSLP 2002*, pp. 1357–1360, September 2002.

[33] http://www.lancaster.ac.uk/people/yanj2/.

[34] J. Yan and A. S. El Ahmad, "Breaking visual CAPTCHAs with naïve pattern recognition algorithms," in *Proceedings of the 23rd Annual Computer Security Applications Conference, ACSAC 2007*, pp. 279–291, December 2007.

[35] J. Yan and A. S. El Ahmad, "Usability of CAPTCHAs or usability issues in CAPTCHA design," in *Proceedings of the 4th Symposium on Usable Privacy and Security, SOUPS 2008*, pp. 44–55, July 2008.

[36] A. S. El Ahmad, J. Yan, and L. Marshall, "The robustness of a new CAPTCHA," in *Proceedings of the 3rd European Workshop on System Security, EUROSEC'10*, pp. 36–41, April 2010.

[37] B. B. Zhu, J. Yan, Q. Li et al., "Attacks and design of image recognition CAPTCHAs," in *Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS'10*, pp. 187–200, October 2010.

[38] A. S. E. Ahmad, J. Yan, and M. Tayara, "The Robustness of Google CAPTCHAs," Computing Science Technical Report CS-TR-1278, Newcastle University, 2011.

[39] A. S. El Ahmad, J. Yan, and W.-Y. Ng, "CAPTCHA design: Color, usability, and security," *IEEE Internet Computing*, vol. 16, no. 2, pp. 44–51, 2012.

[40] A. Algwil, D. Ciresan, B. Liu, and J. Yan, "A security analysis of automated Chinese turing tests," in *Proceedings of the 32nd Annual Computer Security Applications Conference, ACSAC 2016*, pp. 520–532, December 2016.

[41] H. Gao, W. Wang, J. Qi, X. Wang, X. Liu, and J. Yan, "The robustness of hollow CAPTCHAs," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS 2013*, pp. 1075–1085, November 2013.

[42] H. Gao, J. Yan, F. Cao et al., "A Simple Generic Attack on Text Captchas," in *Proceedings of the Network and Distributed System Security Symposium*, pp. 1–14, San Diego, Calif, USA, 2016.

[43] http://web.xidian.edu.cn/hchgao/paper.html.

[44] H. Gao, W. Wang, and Y. Fan, "Divide and conquer: An efficient attack on Yahoo! CAPTCHA," in *Proceedings of the 11th IEEE International Conference on Trust, Security and Privacy*

in Computing and Communications, TrustCom-2012, pp. 9–16, June 2012.

[45] F. Dai, H. Gao, and D. Liu, "Breaking CAPTCHAs with second template matching and BP neural network algorithms," *International Journal of Information Processing and Management*, vol. 4, no. 3, pp. 126–133, 2013.

[46] H. Gao, W. Wang, Y. Fan, J. Qi, and X. Liu, "The robustness of "connecting characters together" CAPTCHAs," *Journal of Information Science and Engineering*, vol. 30, no. 2, pp. 347–369, 2014.

[47] H. Gao, X. Wang, F. Cao et al., "Robustness of text-based completely automated public turing test to tell computers and humans apart," *IET Information Security*, vol. 10, no. 1, pp. 45–52, 2016.

[48] R. Hussain, H. Gao, and R. A. Shaikh, "Segmentation of connected characters in text-based CAPTCHAs for intelligent character recognition," *Multimedia Tools and Applications*, pp. 1–15, 2016.

[49] R. Hussain, H. Gao, R. A. Shaikh, and S. P. Soomro, "Recognition based segmentation of connected characters in text based CAPTCHAs," in *Proceedings of the 8th IEEE International Conference on Communication Software and Networks, ICCSN 2016*, pp. 673–676, June 2016.

[50] https://captcha.com/.

[51] http://jcaptcha.sourceforge.net/.

[52] http://www.hinsite.com.

[53] http://caca.zoy.org/wiki/PWNtcha.

[54] https://code.google.com/p/captchacker.

[55] http://www.brains-n-brawn.com/default.aspx?vDir=aicaptcha.

[56] http://www.cs.sfu.ca/~mori/research/gimpy/.

[57] G. Moy, N. Jones, C. Harkless, and R. Potter, "Distortion estimation techniques in solving visual CAPTCHAs," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004*, pp. II23–II28, July 2004.

[58] A. Bansal, D. Garg, and A. Gupta, "Breaking a Visual CAPTCHA: A Novel Approach using HMM," 2008, https://pdfs.semanticscholar.org/3c2c/9af1e9a3b7095edaf8f205dfbadc30f917fb.pdf.

[59] S. Li, S. A. H. Shah, M. A. U. Khan, S. A. Khayam, A.-R. Sadeghi, and R. Schmitz, "Breaking e-banking CAPTCHAs," in *Proceedings of the 26th Annual Computer Security Applications Conference, ACSAC 2010*, pp. 171–180, December 2010.

[60] C. Hong, B. Lopez-Pineda, K. Rajendran, and A. Recasens, "Breaking Microsoft's CAPTCHA," 2015, https://courses.csail.mit.edu/6.857/2016/files/hong-lopezpineda-rajendran-recansens.pdf.

[61] O. Starostenko, C. Cruz-Perez, F. Uceda-Ponga, and V. Alarcon-Aquino, "Breaking text-based CAPTCHAs with variable word and character orientation," *Pattern Recognition*, vol. 48, no. 4, pp. 1097–1108, 2015.

[62] L. Zhang, L. Zhang, S.-G. Huang, and Z.-X. Shi, "A highly reliable CAPTCHA recognition algorithm based on rejection," *Acta Automatica Sinica*, vol. 37, no. 7, pp. 891–900, 2011.

[63] R. Chen, J. Yang, R.-G. Hu, and S.-G. Huang, "A novel LSTM-RNN decoding algorithm in CAPTCHA recognition," in *Proceedings of the 3rd International Conference on Instrumentation and Measurement, Computer, Communication and Control, IMCCC 2013*, pp. 766–771, September 2013.

[64] S. Sano, T. Otsuka, K. Itoyama, and H. G. Okuno, "HMM-based attacks on Google's ReCAPTCHA with continuous visual and audio symbols," *Journal of Information Processing*, vol. 23, no. 6, pp. 814–826, 2015.

[65] J. Sauvola and M. Pietikäinen, "Adaptive document image binarization," *Pattern Recognition*, vol. 33, no. 2, pp. 225–236, 2000.

[66] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[67] C. J. Hilditch, "Linear Skeletons from Square Cupboards," *Machine Intelligence*, pp. 403–420, 1969.

[68] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Communications of the ACM*, vol. 27, no. 3, pp. 236–239, 1984.

Hindawi
Security and Communication Networks
Volume 2017, Article ID 4376282, 14 pages
https://doi.org/10.1155/2017/4376282

Research Article

A Cloud-User Protocol Based on Ciphertext Watermarking Technology

Keyang Liu, Weiming Zhang, and Xiaojuan Dong

School of Information Science and Technology, University of Science and Technology of China, Anhui, China

Correspondence should be addressed to Weiming Zhang; zhangwm@ustc.edu.cn

Received 14 August 2017; Accepted 9 November 2017; Published 11 December 2017

Academic Editor: Kim-Kwang Raymond Choo

With the growth of cloud computing technology, more and more Cloud Service Providers (CSPs) begin to provide cloud computing service to users and ask for users' permission of using their data to improve the quality of service (QoS). Since these data are stored in the form of plain text, they bring about users' worry for the risk of privacy leakage. However, the existing watermark embedding and encryption technology is not suitable for protecting the Right to Be Forgotten. Hence, we propose a new Cloud-User protocol as a solution for plain text outsourcing problem. We only allow users and CSPs to embed the ciphertext watermark, which is generated and embedded by Trusted Third Party (TTP), into the ciphertext data for transferring. Then, the receiver decrypts it and obtains the watermarked data in plain text. In the arbitration stage, feature extraction and the identity of user will be used to identify the data. The fixed Hamming distance code can help raise the system's capability for watermarks as much as possible. Extracted watermark can locate the unauthorized distributor and protect the right of honest CSP. The results of experiments demonstrate the security and validity of our protocol.

1. Introduction

1.1. Problem Background. Right to Be Forgotten (RTBF) is a kind of people's right that was proposed for protecting people's privacy and has been mentioned as early as 1995 in Data Protection Directive of EU [1]. The 17th article of General Data Protection Regulation (GDPR) [2], which was passed by EU in 2012 to strengthen data protection for individuals in EU, defined RTBF as the right that people deserve to obtain or erase the data expired or related to their privacy from the data controller. In 2013, Senate Bill 568 of California [3] was signed to protect the RTBF of children. In 2014, the European Court of Justice compelled Google to delete the links about a Spanish man's bankruptcy from its searching results, which confirmed that the RTBF is a basic right for people living in EU. Since then, Google, Facebook, and YouTube have erased tens of thousands of links based on the request of EU citizens [4]. However, the erasure of data cannot be technically confirmed by users if they do not believe their Cloud Service Providers (CSPs). Moreover, cloud computing becomes more and more powerful and economic. Companies like Amazon, Alibaba, and Microsoft have provided cloud computing service to help people manipulating their data more cheaply and easily. If users want to lodge their data in cloud servers to lower the expenses, they need to think carefully about the risk of data leakage. As a result, confirmed deletion and several related ideas can be introduced to deal with this problem, which is also the target of our protocol.

1.2. Related Works. There are two kinds of methods used in confirmed deletion. First comes the encryption. User (U) encrypts his data and transfers it to CSP for storing [5–7]. Once U wants to delete his data, he just needs to abandon the encryption key and inform CSP that related data are useless. The management of key can be authorized to several Third Parties and use secret sharing technology to prevent conspiracy [8]. Encryption can protect the privacy of data and RTBF in ideal circumstance though it destroys the value of data. When U uses encryption technology, he can only use the storage space of CSP while wasting their ability of computation.

To solve this dilemma, homomorphic encryption (HE) [9] was introduced into this field [10]. Once the data is encrypted by HE algorithm, CSP can calculate data as user

ordered while knowing nothing about it. However, HE has some other flaws. For example, it requires user to have the knowledge about what operations they want to do on data before knowing their results. What is more, the full-HE, which can do both addition and multiplication on ciphertext, is unbearably slow and costs a lot. The semi-HE, which is faster, faces the problem of restricted operations. In a word, it is not convenient and economic for using encryption to protect RTBF so far. In our solution, data will be stored as plain text in cloud servers so that U can use the ability of computation completely to manage U's data.

Other than confirming deletion, not deleted is more easy to be confirmed, which suggests the second way, tracing the unauthorized distribution of data. To the best of our knowledge, watermarking is used in copy deterrence and tracing down the distribution of illegal copies [11–13]. This fact indicates watermark can be used to protect RTBF by proving the crimes of CSP. As the successful cases have shown, Google and Facebook were forced to delete [4] those links infringing people's privacy once U reports them and proves the infringement. But this method faces a new problem that user can use his data to fraud CSP if he can get benefit from lawsuit like defaming the specific CSP or diddling indemnity. If CSP requires embedding another watermark so that he can identify whether the copy is stored in his server, CSP can leak the copy with both watermarks to avoid being charged. Once the embedding process is outsourced to a Third Party, it will raise the risk of information leak from TP. In a word, watermark technology cannot be used to protect RTBF directly.

In this paper, we design a new Cloud-User protocol as the solution based on the work of buyer-seller protocol [14, 15]. We generate and embed the watermark in ciphertext to make sure the watermark can be erased during downloading. By using only one watermark, we increase the SNR of data. Moreover, we introduce the idea of feature-extraction function (FEF), a fixed Hamming distance code into protocol to reduce the cost of searching and increase the capacity of system while maintaining the robustness of watermark system.

The rest of this paper is organized as follows. In Section 2, we will give a brief introduction to the problem models, design goals, and the threat models. Preliminaries will be introduced in Section 3. The proposed solution is described in Section 4 and the security of the scheme is analyzed in Section 5. Section 6 will explain the design of experiment as well as the subalgorithm we used for building demo. The results of each experiment will be analyzed in Section 7. The last section contains the conclusions and future work.

2. Problem Formulation

2.1. Problem Model. The problem model in this paper involves three parties: User (U), Cloud Service Provider (CSP), and the Trusted Third Party (TTP).

User. U possesses large quantities of data and wishes to store it in the CSP's server. In addition, those data are valuable and need enough computing power to dig out their value. As a result, U wants to store his data on CSP's server and requires CSP to do some complex operations for him. In our scheme, only U and CSP can touch and manipulate these data. According to RTBF, U has the right to retrieve his data and require CSP to delete it at any time. Once U finds his data that should be deleted, U can suspect that a CSP has distributed his data illegally for interests and require TTP to verify where it comes from. Once confirmed, U can sue CSP for being guilty and ask for a compensation.

Cloud Service Provider. CSP controls piles of servers which have large storage space and powerful computing ability. U can store, manipulate, and delete his data on CSP's servers only if he pays for it according to contract. Although CSP controls all the data in his servers, he does not have the ownership of data and should take responsibility for their security. CSP can never distribute data whether U cares or not and needs to backup it in case of servers' crash.

Trusted Third Party. TTP is an arbitration agency who is responsible for generating a valid watermark for every single trade between U and CSP. TTP should be trusty so that his verification can be used as evidence. Besides, TTP should know nothing about U's data unless U requires TTP to verify whether a specific copy has been marked to be deleted.

Our solution is designed to make sure any one of the three can only know what they allowed to know and do what they required to do. Whoever disobeys the contract will suffer loss.

2.2. Threat Model. In the proposed solution, we assume TTP is selected by U and CSP, so we do not consider TTP will conspire with any one and no one can get payment from him. So there is no conspiracy among our solutions. We should consider the threat that CSP or U can get benefit while offending the other one.

CSP's Attack. CSP controls all data stored in his servers; he should obey U's order to manipulate U's data according to contract. But it may copy U's data as a backup even after U requires CSP to delete it. Since CSP has a full access to these data in plain text, we can do nothing about his analysis on data and that should be considered in contract. On one hand, CSP may not delete the data as required, and those data are leaked for CSP's careless management. On the other hand, CSP may deliberately sell these data after U's delete requirement and even try to adjust it so that U and TTP cannot trace it.

U's Attack. U possesses the ownership of data. The benefit U that can be gained from CSP is the compensation. On one hand, if CSP is innocent, U can only use the retrieved data and original data to create a copy. On the other hand, once CSP has leaked a part of U's data files, U may use them to guess other data files' watermark and forge CSP's loss.

2.3. Design Goals. This paper aims to design a solution among CSP, U, and TTP that allows U to store his data in CSP's servers as plain text while providing the remote control according to contract. In particular, we formally detail the goals as follows.

Data Privacy. As claimed in problem model, TTP is responsible for generating watermarks for giving data while TTP should have no access to data's content. Hence, we carefully design our solution so that TTP can embed the identity watermark directly in the encrypted data.

Nonrepudiation. Any copy of unauthorized data must be identifiable to find the illegal distributor.

Fairness. The proposed solution is secure and fair to all parties. Nobody can frame an honest party.

The Right to Be Forgotten. Acceptable deletion requires no information about the data remaining in servers of CSP. Once CSP does not follow requirements and the bad behavior can be proved by U (i.e., the unauthorized copy is detected). U can require TTP to verify the watermark of leaked copy and provide it as evidence which cannot be denied based on contract.

3. Preliminaries

In our solution, there are four kinds of technology we will use. Each technical method can be adjusted to fit all kinds of data (D). To simplify the declaration, we use image data as an example to introduce our solution and complete our experiment. Here is a simple introduction to these technologies and the restriction our solution required.

3.1. Feature-Extraction Function (FEF). FEF is used to identify the content of data while getting no detail about it. FEF is an important part of our solution which is used to define the validity of data for U. FEF's input is data file and the output is a feature (Fea). Once a data file A and its adjusted copy B satisfied FEF(A) = FEF(B), we call B a derivative copy (DC) of A. The set of DC is derivative set (DS).

In our solution, FEF must fit the following requirement.

(1) One-Way Function. For B = FEF(A), no one can create a DC of A if he only has the knowledge about B. This is because Fea of the stored data is shared among all three parties in our solution. This property can make sure only U and CSP can distribute DC of data.

(2) Content-Based. For no digital watermark algorithm can promise that it can resist all attacks, we use FEF as a restriction of watermark extraction algorithm so that our solution can get balance between validity and security. In our solution, U should carefully select FEF to make sure all the valuable copies of original data belong to a DS.

(3) Equiprobability. The set of possible value for Fea must be large enough, and the possibility of each value is equal. This property protects the efficiency of searching process.

3.2. Digital Watermark. Digital watermark (W) is a signal embedded into data to identify some attributions of the data (i.e., ownership). According to the domain embedded, digital watermark embedding algorithms are divided into time-spatial embedding, which is fast and relatively easy to operate but is easy to be erased by geometrical attack, and transform domain embedding [16–19], which is good at resisting geometrical attack but is fragile facing filtering. Moreover, according to the preknowledge related to data before embedding, we classify the embedding method into preknowledge dependent embedding and preknowledge independent embedding. In most cases, dependent embedding is more robust than independent one. In our solution, we recommend to use the preknowledge dependent transform embedding method to enhance the security of our solution. Furthermore, our solution requires the following properties that digital watermark embedding algorithm should have.

(1) Markov Property. For a given $W = w_i \mid 1 < i < L$ of length L, the embedding and extracting process of w_i has no effect on the process of w_{i+1}.

(2) Predictability. Predictability means the embedding positions can be determined only by the length of embeddable positions and the bit length of watermark.

(3) Robustness. Based on the requirement of U, the watermark algorithm should guarantee that the watermark can be extracted from the DS of embedded data.

3.3. Homomorphic Encryption (HE). Encryption is the most famous method in information security. Homomorphic encryption [18] can translate some operations on plain text into other operations on ciphertext. In our solution, we require that data should be encrypted during transferring and embedding process. For full-HE is slow and costly, we decide to use semi-HE as a compromise that give the consideration to both efficiency and security. We list the requirement of our solution for the semi-HE as follows ($E()$ is encryption function, KEY is the encryption key, and S_i is the target information).

(1) Addition Homomorphism

$$E(KEY, S1 + S2) = E(KEY, S1) \odot E(KEY, S2). \quad (1)$$

(2) Multimap. The absolute value of each of the encryption results depends on the random number it used in different times:

$$E(KEY, S1 : t_1) \neq E(KEY, S1 : t_2). \quad (2)$$

4. Solution Framework

Our solution contains three protocols based on Public-Key Infrastructure (PKI) that is used for distributing public and private key pair combining to each registered ID. The notation used in protocols has been listed at the end of the paper.

4.1. Uploading Stage. In this subsection, we describe the details about uploading stage, including watermark's generating and embedding.

Input: D, L
Output: the threshold of watermark matching μ
(1) **if** $L < 2$ **then**
(2) **return** 100
(3) **end**
(4) **if** *embed* $\{1\}_L$ or $\{0\}_L$ *will change Fea* **then**
(5) **return** 100
(6) **end**
(7) $TESTD = EW(D, \{1 \mid 0\}_L)$
(8) $\mu \longleftarrow 0$
(9) **while** *there is attack method has not been tested* **do**
(10) AD = attacked TESTD
$$\mu = \max(\mu, 100 * \frac{sum(xor(DW(AD), DW(TSETD)))}{L})$$
(11) **end**

Algorithm 1: Generating μ.

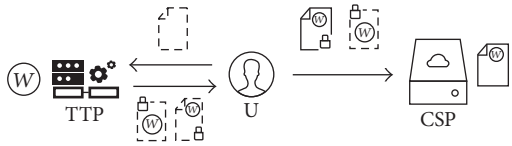


Figure 1: Data flow of updating process.

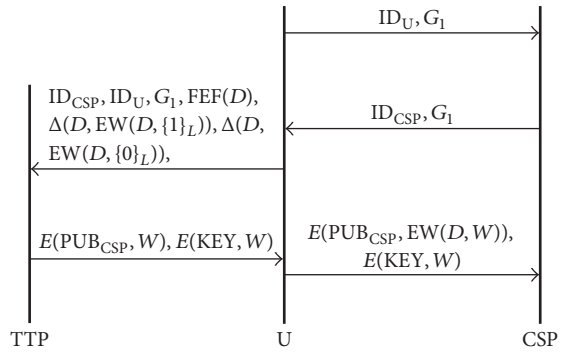


Figure 2: Details of updating process.

Before outsourcing data are transferred to CSP, U needs to embed watermark into his data as shown in Figure 1.

All transferred data are encrypted by CSP's public key or TTP's special key. The details of each process are presented by Figure 2 and introduced in the following steps.

Step 1. U sends CSP his ID and service contract to apply for storing and computing his data. The first contract (G_1) details the responsibilities and obligations of U and CSP and the subalgorithm, including parameters which CSP needs to know, used in the whole solution. The contract is signed by PRI_U to make sure of its integrity. Once CSP does not admit the contract, CSP can reject U's request and the protocol is finished.

Step 2. CSP sends ID_CSP and G_1 to U, which is signed by PRI_CSP. This step means U's request has been permitted. G_1 has been signed twice to make sure that its content has not been changed and will be used as an evidence in the future.

Step 3. U selects a watermark algorithm matching the requirements declared before and threshold μ according to Algorithm 1, which will be signed and attached behind G_1, to embed $\{1\}_L$ and $\{0\}_L$ into his data, where L is the watermark capacity of D and calculates the differences between original data and embedded data as δ_1 and δ_0 according to (3). TESTD is the data embedded with a random sequence of length L, which is used to test the robustness of watermark algorithm; the test round can be done more than 1 time for security purpose. U should make sure that the production of embedding process still belongs to DS(D).

$$\begin{aligned} \delta_1 &= \{\delta_{1i} \mid 1 < i < L\} = \Delta\left(D, E\left(D, \{1\}_L\right)\right) \\ \delta_0 &= \{\delta_{0i} \mid 1 < i < L\} = \Delta\left(D, E\left(D, \{0\}_L\right)\right). \end{aligned} \tag{3}$$

Then, U sends IDs, G_1, FEF(D), δ_1, and δ_0 to TTP for recording and generating watermark W.

Step 4. TTP generates W according to existing data of U that share the same Fea. We present Algorithm 2 as an example for generating watermark here. TTP creates δ_2 by δ_1, δ_0, and W like Algorithm 3.

According to Markov property and predictability, Algorithm 3 guarantees that TTP can create an additive watermark δ_2 based on δ_1 and δ_0. δ_2 is the same as the difference between original data and its copy embedded with W by the selected watermark algorithm. Then, TTP sends the encrypted δ_2 to U as well as signature. Here, we suggest that TTP use two keys to encrypt δ_2. PUB_CSP encrypted copy is for embedding, and TTP's KEY encrypted copy is for

Input: SW(Set of exist watermark of U with same Fea), μ, L(Capacity of File)
Output: A new watermark
(1) $flag = 1$
(2) **while** *flag* **do**
(3) randomly generate a sequence t shorter than $\frac{L}{2}$. **for** *All item x in SW* **do**
(4) **if** $sum(xor(x,t)) < \mu$ **then**
(5) $flag = 0$ break
(6) **end**
(7) **end**
(8) $flag = 1 - flag$
(9) **end**
(10) **return** t

Algorithm 2: Generating W.

Input: δ_1, δ_0, W
Output: δ_2
(1) n← length of W;
(2) $step \leftarrow round(L/n)$;
(3) embedding positions ←random sequence from[1,2,...step] of length $step/2$
(4) **for** $i = 1$ **to** $n + 1$ **do**
(5) **for** $j = 1$ **to** *step* **do**
(6) **if** $2(i-1)step + j > L$ **then**
(7) Break;
(8) **end**
(9) **else**
(10) **if** *j is in embedding positions* **then**
(11) **if** $w_i == 1$ **then**
(12) $\delta_{2((i-1)step+j)} = \delta_{1((i-1)step+j)}$
(13) **end**
(14) **else**
(15) $\delta_{2((i-1)step+j)} = \delta_{0((i-1)step+j)}$
(16) **end**
(17) **end**
(18) **else**
(19) $\delta_{2((i-1)step+j)}$ is equal to $\delta_{(1-w_i)((i-1)step+j)}$
(20) **end**
(21) **end**
(22) **end**
(23) **end**
(24) **return** δ_2

Algorithm 3: Generating $\delta 2$.

erasing in the future which will release the storage burden of TTP.

Step 5. U verifies TTP's signature to make sure that W is valid. Then U uses $\mathrm{PUB_{CSP}}$ to encrypt D and embeds W into D according to the addition homomorphism of encryption algorithm as the following proof has shown, which will get the encrypted file (ED) that contains W.

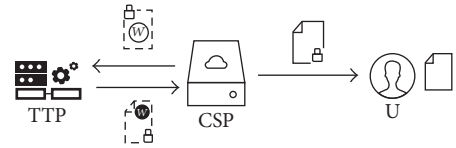


Figure 3: Data flow of downloading process.

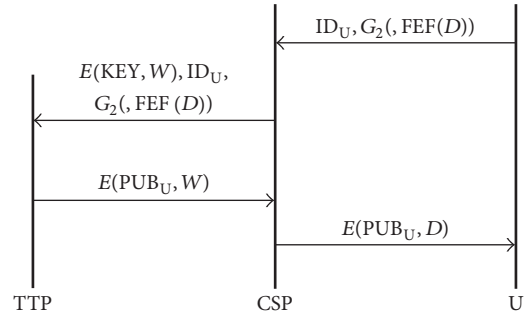


Figure 4: Details of downloading process.

Proof.

$$\begin{aligned}\mathrm{ED} &= E\left(\mathrm{PUB_{CSP}}, \delta_2\right) \odot E\left(\mathrm{PUB_{CSP}}, D\right)\\ &= E\left(\mathrm{PUB_{CSP}}, \delta_2 + D\right)\\ &\because \delta_2 = \Delta\left(D, \mathrm{EW}\left(D, W\right)\right)\\ &\therefore \mathrm{ED} = E\left(\mathrm{PUB_{CSP}}, \Delta\left(D, \mathrm{EW}\left(D, W\right) + D\right)\right)\\ &= E\left(\mathrm{PUB_{CSP}}, \mathrm{EW}\left(D, W\right)\right).\end{aligned} \tag{4}$$

□

U sends δ_2 encrypted by TTP's KEY and ED to CSP. CSP decrypts ED and stores it. Then, the uploading stage is finished.

4.2. Downloading Stage. The downloading stage is much simpler than the uploading protocol, for δ_2 has been stored encrypted in CSP's servers. The data flows are shown in Figure 3.

All data are still encrypted. Details of downloading stage are presented by Figure 4 and introduced as follows.

Step 1. U sends G_2 and $\mathrm{ID_U}$ to CSP. G_2 contains the requirement of retrieving or deletion which need erasing W from ED. U can use FEF(D) to help CSP and TTP search the exact data that he wants.

Step 2. After verifying U's signature, CSP sends G_2 and $\mathrm{ID_U}$ along with encrypted δ_2 to CSP so that CSP can create reversed watermark to erase W from ED.

Step 3. TTP verifies all the information stored in his database. If the information is correct, TTP first decrypts δ_2 and creates the reversed watermark $-\delta_2$. Then TTP encrypts it by $\mathrm{PUB_U}$ and sends it back to CSP.

Step 4. CSP embeds encrypted $-\delta_2$ into ED and then sends it to U. U decrypts receiving file to get his data according to the following proof.

Proof.

$$\begin{aligned} & E\left(\mathrm{PUB}_{\mathrm{U}}, \mathrm{EW}\left(D, W\right)\right) \odot E\left(\mathrm{PUB}_{\mathrm{U}}, -\delta_2\right) \\ & \quad \longrightarrow E\left(\mathrm{PUB}_{\mathrm{U}}, \mathrm{EW}\left(D, W\right) - \delta_2\right) \\ & \quad \longrightarrow E\left(\mathrm{PUB}_{\mathrm{U}}, \mathrm{EW}\left(D, W\right) - \Delta\left(D, \mathrm{EW}\left(D, W\right)\right)\right) \\ & \quad \longrightarrow E\left(\mathrm{PUB}_{\mathrm{U}}, D\right). \end{aligned} \tag{5}$$

□

Once G_2 requires CSP to delete U's data, U cannot download that data in future again, and TTP will create the log of this data and abandon the KEY of δ_2.

4.3. Arbitration Stage. When U finds an unauthorized file L that belongs to DS(D), U can identify the illegal distributor and bring a suit against it.

U should first execute FEF function to get Fea about the leaked data (L) and then provide the L, FEF(L) as well as ID_{U} to TTP. After verifying the information about U, TTP searches the data based on FEF(L) and ID_{U} to get logs of possible leaked data as set S. If S is empty, TTP tells U that this data is not recorded in his database. Otherwise, TTP executes DW(L) according to watermark algorithm and embedding positions of each item of S and calculates the bit error ratio (BER) of DW(L) and W as δ_3. If there is any δ_3 below μ declared in G_1 of that data, TTP believes the CSP signed G_1 violate U's RTBF or privacy. TTP will provide a proof with digital signature to U as a legal evidence.

5. Solution Analysis

Our solution proposed above can solve the problem we mentioned in problem model. The safety of our solution relies critically on the security of subalgorithms like watermarking and encryption algorithm. In this section, we will analyze properties we described in design goals and requirement to each party.

5.1. Effectiveness. Our solution can solve the problem of RTBF as we have mentioned. Once CSP want to violate U's RTBF, he needs to distribute DC(D) to others. If U finds that copy, he can send it to TTP and ask for arbitration, and CSP's crime will be proved. Once U wants to fraud an innocent CSP, U should create a copy that belongs to DS(D) and contains W. However, U has no information about W in plain text. It is technically impossible for CSP to do that if the encryption algorithm is secure enough.

5.2. Security. The security of our solution is based on the fact that U and CSP cannot get information about W. We assume all the subprotocols can satisfy the property we required.

CSP possesses embedded data (ED), δ_2 and $-\delta_2$ encrypted by KEY or $\mathrm{PUB}_{\mathrm{U}}$. CSP wants to create a copy of ED-δ_2, which is impossible if the encryption algorithm is strong enough. Besides, CSP can try to attack ED so that δ_3 are larger than μ. In this case, the robustness of watermark algorithm and FEF function is tested. With the help of μ and FEF, CSP cannot create a useful copy while maintaining the validity of data for distributing.

U possesses D, δ_1 and δ_0 in plain text, δ_2, ED encrypted by $\mathrm{PUB}_{\mathrm{CSP}}$, δ_2 encrypted by KEY. According to the multimap property, U cannot use δ_1 and δ_0 to create δ_2 in polynomial time. Besides, embedding positions will make it harder for both U and CSP to get information about W, though it sacrifices the robustness to some extent. Moreover, considering CSP may leak a part of data and be found by U, U can get a message containing W. U may try to use it to guess other watermarks. The watermark generation is completely random and each watermark shares different length and embedding positions. The possibility of creating a DC to match the watermark is P. Here, we neglect the possibility that a extract watermark can be recognized as two embedded water marks.

$$P \approx \frac{\sum_{i=1}^{\mu L} \binom{L}{i}}{2^L - \sum_{i=1}^{\mu L} \binom{L}{i}}. \tag{6}$$

In conclusion, our solution can make sure that U and CSP cannot get DC of the other one's copy. The robustness of watermark is controlled by U according to the FEF function and watermark algorithm.

5.3. Consumption

U. U outsources local data to CSP for reducing the local data storage space and the cost of complex computing. In our scheme, after uploading data, U can reserve FEF(D) for reducing the cost of searching. U should also do some computation for encrypting and decrypting data.

TTP. TTP has enough storage space for keeping the records of contracts, IDs, Fea, and watermarks for arbitration parts. In this paper, TTP is designed with memory and some necessary computing powers. TTP can take some fee for arbitration requirement so that it will not be annoyed by unsure request and balance the expenses.

CSP. CSP provides large storage space and strong computing power as service. It is reasonable to put the burden of storing outsourced data as well as encrypted δ_2 on CSP.

6. System Design

In this section, we will introduce the experiment we used for verifying the validity and security of our solution. We choose image as U's data to finish our experiment because it is the most popular kind of data used in outsourcing service. Before introducing experiment, we first clarify the subalgorithms we used in our solution.

6.1. Watermark Algorithm. The watermark scheme we used for experiment is Dither Modulation-Quantization Index Modulation (DM-QIM) [20]. It is a classical watermarking scheme and easy for use. Although it has been proved not safe enough [21, 22], it satisfies the requirements we proposed for watermark algorithm.

Generate n = q ∗ p where p and q are both random large prime;
Generate g as a random number of $\mathbb{Z}_n^*$
$\lambda = LCM(p-1, q-1)$
Public key ← (n,g), Private key ← (p,q)
#encryption:
randomly select r < n
$C = g^m * r^n \bmod n^2$
return C
#decryption:
$calculate\lambda$
$m = \dfrac{(c^\lambda \bmod n^2) - 1}{(g^\lambda \bmod n^2) - 1} \bmod n^2$
return m
#addition:
$C_3 = C_1 * C_2 = g^{m1+m2} * (r_1 r_2)^n$
return C_3

Algorithm 4: Paillier.

DM-QIM embeds watermark into transforming domain. It adjusts the value of some coefficients, which is the preknowledge, to embed the message according to (7) where step is the quantizer and d is the dither.

$$\mathrm{EW}(D_i, W_i) = \begin{cases} \mathrm{round}\left(\dfrac{D_i - d}{\mathrm{step}}\right) * \mathrm{step} + d & W_i == 1, \\ \mathrm{round}\left(\dfrac{D_i + d}{\mathrm{step}}\right) * \mathrm{step} - d & W_i == 0. \end{cases} \tag{7}$$

In the extracting process, we use ED_i to reprensent the ouput of $\mathrm{EW}(D_i, W_i)$. According to (8), we can find that different judgments (Jud), which are guessed result before extracting, will lead to different extracting processes and extract different values because of the quantizer. We can add up all $\mathrm{DW}(ED_i)$ that embed same bit of watermark to measure whether Jud is equal to W_i. In any case, we will get the watermark embedded in the picture.

$$\begin{aligned} \mathrm{DW}(ED_i) &= \begin{cases} \mathrm{round}\left(\dfrac{ED_i - d}{\mathrm{step}}\right) * \mathrm{step} + d - ED_i & \mathrm{Jud} == 1,\ W_i == 1, \\ \mathrm{round}\left(\dfrac{ED_i - d}{\mathrm{step}}\right) * \mathrm{step} + d - ED_i & \mathrm{Jud} == 1,\ W_i == 0, \\ \mathrm{round}\left(\dfrac{ED_i + d}{\mathrm{step}}\right) * \mathrm{step} - d - ED_i & \mathrm{Jud} == 0,\ W_i == 1, \\ \mathrm{round}\left(\dfrac{ED_i + d}{\mathrm{step}}\right) * \mathrm{step} - d - ED_i & \mathrm{Jud} == 0,\ W_i == 0, \end{cases} \\ &= \begin{cases} 0 & \mathrm{Jud} == 1,\ W_i == 1, \\ \left(\mathrm{round}\left(\dfrac{ED_i - d}{\mathrm{step}}\right) * \mathrm{step} + d\right) - ED_i & \mathrm{Jud} == 1,\ W_i == 0, \\ \left(\mathrm{round}\left(\dfrac{ED_i + d}{\mathrm{step}}\right) * \mathrm{step} - d\right) - ED_i & \mathrm{Jud} == 0,\ W_i == 1, \\ 0 & \mathrm{Jud} == 0,\ W_i == 0. \end{cases} \end{aligned} \tag{8}$$

This watermark scheme embeds 1 or 0 into each selected coefficient as Algorithm 3 which means it satisfies the Markov property. In our solution, we split image into several 8 × 8 nonoverlapping blocks firstly and use DCT to transform these blocks into transform domain, which means all the coefficients can be placed in a meaningful place so that TTP can determine which position to embed. Thus, this scheme has predictability.

In our solution, DCT coefficients below 0.4 are chosen for watermarks. Embedding positions are selected according to the value of $(i + j) \bmod 2$, where i and j are the coordinates of the coefficient. In our demo, the step is 100 and the dither is 25.

6.2. Encryption. In our system, we use AES and Paillier [23] as encryption algorithm that can fit solution's requirements. AES is a famous symmetric encryption algorithm [24] which is fast and safe. Paillier is a semi-HE that supports additive operations in ciphertext according to Algorithm 4 where $L(x) = (x - 1)/n$.

Although Paillier allows user to do addition, negative numbers and decimals are not allowed to calculate. Because δ_2 and $-\delta_2$ always need to encrypt negative numbers and decimals, we suggest U and CSP do as shown in Algorithms 5 and 6 which can solve this problem.

Besides, δ_0 and δ_1 may leak some information about the image; we recommend that U adds a mark to δ_0 and δ_1 which can be subtracted after decrypting δ_2.

6.3. Perceptual Hash Algorithm (PHA). We choose PHA as FEF function for it can reflect the content about image in its low frequency coefficients which is also used in searching engine [25]. Algorithm 7 shows pHash we used as FEF function.

(a) (b) (c)

Figure 5: The retrieved picture (a), embedded picture (b), and the original picture (c).

Input: message m, amp (integer amplifier which decides the accuracy of data)
Output: pretreated m, balance
(1) $m_1 = round(m * amp)$
(2) **if** m_1 *contains negative* **then**
(3) balance = round($\frac{\max(-\min(m_1), \max(m_1))}{10}$) $*$ 100
(4) **end**
(5) **else**
(6) balance = round($\frac{\max(m_1)}{10}$) $*$ 100
(7) **end**
(8) **for** $i = 1$ **to** L **do**
(9) $m_{2i} = m_{1i} + balance$
(10) **end**
(11) **return** m_2, *balance, amp*

Algorithm 5: Pretreatment: before encryption.

Input: decrypted dm, amp, balances
Output: message m
(1) FB=sum(balances)
(2) **for** $i = 1$ **to** L **do**
(3) $m_{1i} = dm_i - FB$
(4) **end**
(5) $m = round(m_1/amp)$ **return** m

Algorithm 6: Pretreatment: after encryption.

Input: I
Output: hash
(1) h_r = Resize(I, [32, 32])
(2) h_t=DCT(h_r)
(3) sum=0
(4) **for** $i, j = 1$ **to** 8 **do**
(5) $h_{lij} = h_{tij}$
(6) $sum = sum + h_{tij}$
(7) **end**
(8) mid=$\frac{sum}{64}$
(9) **for** $i, j = 1$ **to** 8 **do**
(10) $h_{lij} = sgn(h_{lij} - mid)$
(11) **end**
(12) **return** h_l

Algorithm 7: PHA.

7. Tolerance about System

We first evaluate the tolerance about system. We assume U that has uploaded a large number of images to CSP that have been registered in TTP. One of his images, which has been required to delete, is attractive that CSP wants to distribute it for benefits. CSP needs to erase the watermark embedded in image while he knows nothing about the watermarks' algorithm. So CSP could only use some basic function to attack it. Geometric attacks are not in considered for user that can get the information about watermark by recovering it in most cases, which is dangerous for CSP because U can use this information to create a copy of ED easily. We will consider three types of attack means: JPEG compression (JC), Gaussian filter (GF), and White Gaussian Noise (WGN) to represent the loss compression, filter, and noise attack in the following parts.

We use peak signal-to-noise ratio (PSNR) and bit error ratio (BER) as two indicators that evaluate the performance about our solution. In this section, we will evaluate DC of watermarked Lena provided by each attack mean of different parameters. To compare, the retrieved image's (Figure 5) PSNR maintains 313 dB in our solution.

7.1. JPEG Compression Test. JPEG compression is one of the most popular compression ways that is used for maintaining the main information in smaller size. We want to examine whether CSP can distribute a compressed version data illegally.

Table 1: Watermark's tolerance to GF.

Scale	σ 0.1	0.2	0.3	0.5	0.8	1	2	4	8	10
	BER									
2	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11
4	0.11	0.11	0.11	0.11	0.01	0.01	0.32	0.46	0.48	0.48
8	0.12	0.12	0.12	0.12	0.01	0.01	0.28	0.46	0.48	0.48
10	0.12	0.12	0.12	0.12	0.01	0	0.28	0.46	0.48	0.48

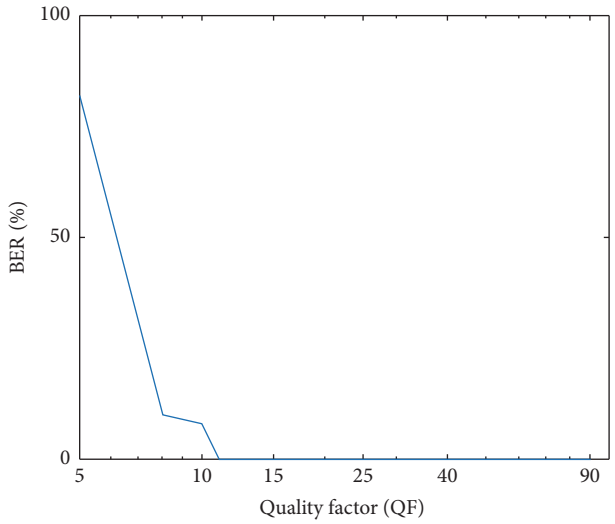


Figure 6: Watermark tolerance to JPEG compression.

Figure 6 shows that the BER decreased rapidly as the quality factor (QF) grows. When QF is 5, which is not a normal choice for compression, Fea of the attacked picture (Figure 8) has changed. This means that our solution can be against the JPEG compression if $\mu > 10\%$.

7.2. Gauss Filter. Filter is the riskiest attack for DM-QIM, since it erases the details within each block of selected scale by adjusting DCT coefficients. As σ grows, picture will become more and more smooth. The mid one in Figure 8 is attacked by GF with scale = 8 and $\sigma = 2$. It suggests that PHA we have used is not the best way to represent the content of image.

Table 1 and Figure 7 show the PSNR and BER affected by GF in different scales and σ. We can notice that GF with scale of 2 has no risk to our solution. When σ is close to 1, BER of attacked image decreases to nearly 0 and the PSNR grows. We consider this as a kind of tolerance to GF. As σ grows continually, BER grows rapidly and the watermark and the detail of picture are erased.

To be against these attacks means that we can change the watermark algorithm or amplify the step as well as dither, which will introduce more noise to embedded picture. This is completely a trade-off between security and the validity of data. The restriction to μ in this experiment is 30%.

7.3. White Gaussian Noise (WGN). Noise is another kind of attack, which will quickly decrease PSNR of image. We use

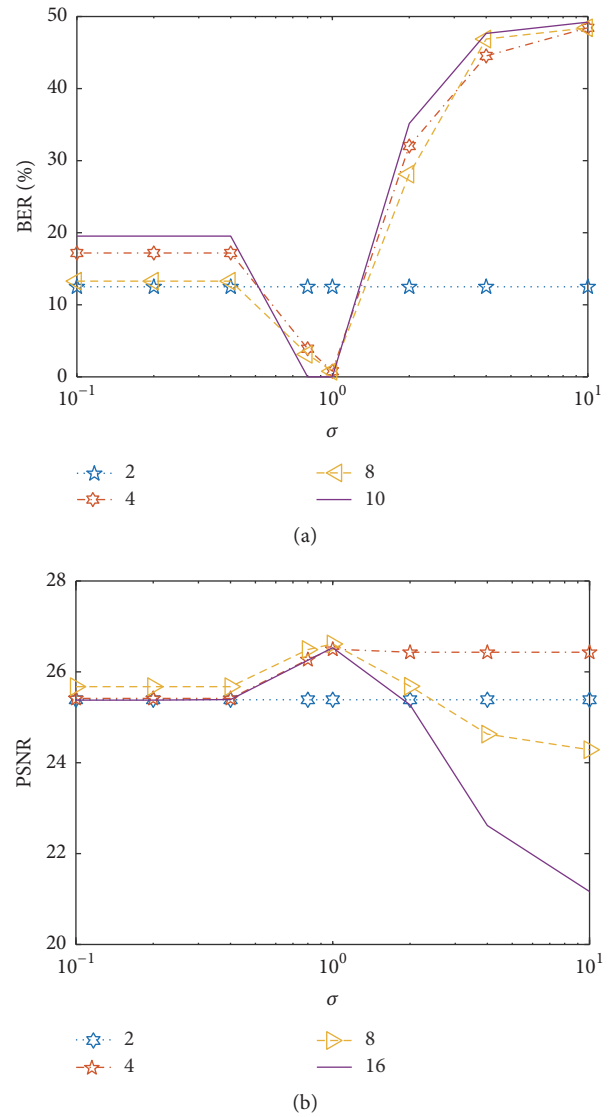


Figure 7: BER (a) and PSNR (b) change according to the watermark.

Gaussian noise to attack our picture. GF and compression will erase the details of images. This will help the attacker decrease the noise watermark introduced in and raise the PSNR of picture in some degree. However, WGN introduces more noise into picture (Figure 8) to cover the watermark which will decrease PSNR quickly and change the Fea.

(a) (b) (c)

Figure 8: Pictures attacked by JC (a), GF (b), and WGN (c).

Table 2: Watermark's tolerance to WGN.

PSNR	23.35	17.94	16.8	16.24	15.88	14.69	13.64	12.93
BER	0	0	0.09	0.26	0.39	0.51	0.52	0.52

Table 2 shows the change of BER according to WGN. The PSNR of Figure 8 is 15.88 which suggests that μ should be at least 26% so that they can defend WGN to some extent in our solution.

8. Capacity of System

In this section, we will take the arbitration stage into consideration. We assume that U finds a picture Y which may be a DC of his deleted data. For U may have not backed up his data, U uploads Y, FEF(Y), and ID_{U} to TTP so that TTP can determine which CSP may leak his data. In this experiment, we will test the capacity of our solution, which is the number of watermarks that are embedded into user's data with same Fea. There are several ways to generate a secure watermark [26]. For simplicity, we choose Algorithm 8 to generate our watermark easily. We named the result of Algorithm 8 as fixed Hamming distance codes, the Hamming distance of each element within answer is no less than the input limits μ. Fixed Hamming distance code allows us to identify the log about data as well as its contracts. And the watermark will be able to defend the attacks as long as U has to test μ according to Algorithm 1.

The final capacity of a TTP will be calculated by (9) where cap(D) means the span of D's value.

$$N = \left(\sum_{i=L_{\min}}^{L_{\max}} \mathrm{Len}\left(\mathrm{watermarks}^{i}\right) \right) * \mathrm{cap}\,(\mathrm{FEA}) * \mathrm{cap}\,(\mathrm{ID})\,. \tag{9}$$

8.1. Result. We first use 30% as threshold, 10 seconds as time limit and 128 bits as the length of watermark. We get at least 1000 watermarks. we select the 500th watermark as the embedded watermark and do attacks as Table 3 presents.

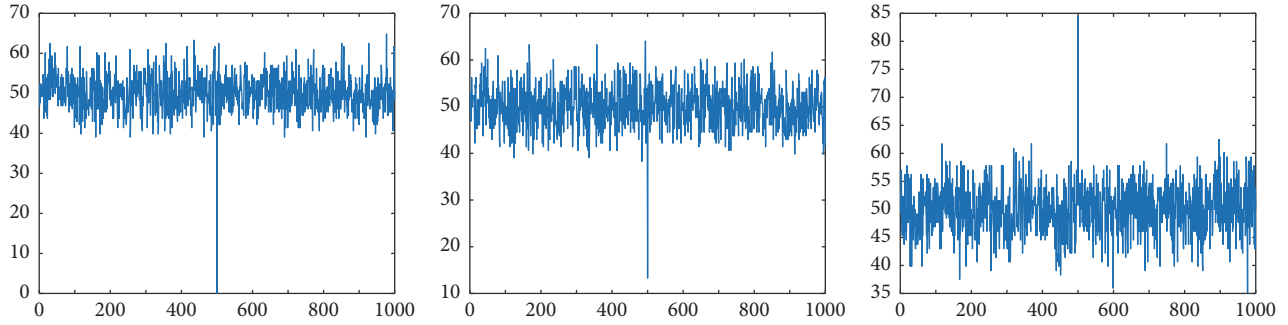
Input: μ, L
Output: $watermarks^L$
(1) $watermarks^L$ = empty set
(2) threshold=μL
(3) flag=1
(4) **while** *not reach time-limit* **do**
(5) random generate a temp watermark t of length L
(6) flag=1
(7) **for** each w in $watermarks^L$ **do**
(8) **if** *sum(XOR(w, t)<threshold)* **then**
(9) flag=0
(10) break
(11) **end**
(12) **end**
(13) **if** *flag* **then**
(14) Add t into $watermarks^L$
(15) **end**
(16) **end**
(17) **return** $watermarks^L$

Algorithm 8: Fixed Hamming distance coding.

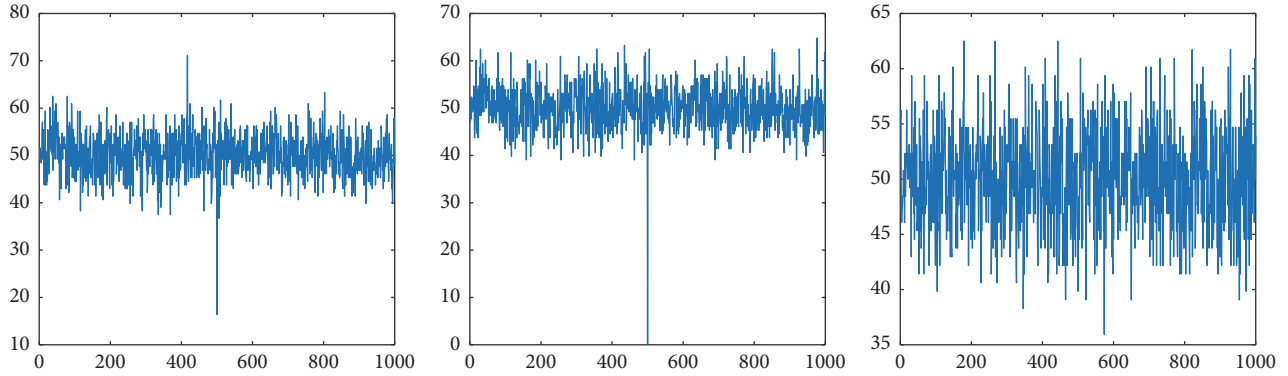
The results are presented by Figure 9 (The y-axis represents BER and x-axis represents sequence number of images). It suggests that our protocol can identify the certain data of it within our database and charge the CSP successfully under the predicted attacks. Fixed Hamming distance code makes sure that the robustness of this protocol is only determined by watermark algorithm and encryption method. The third picture in each line of Figure 9 shows that if the picture is overattacked, we cannot determine the source of the picture from watermark.

Table 3: Test attacks.

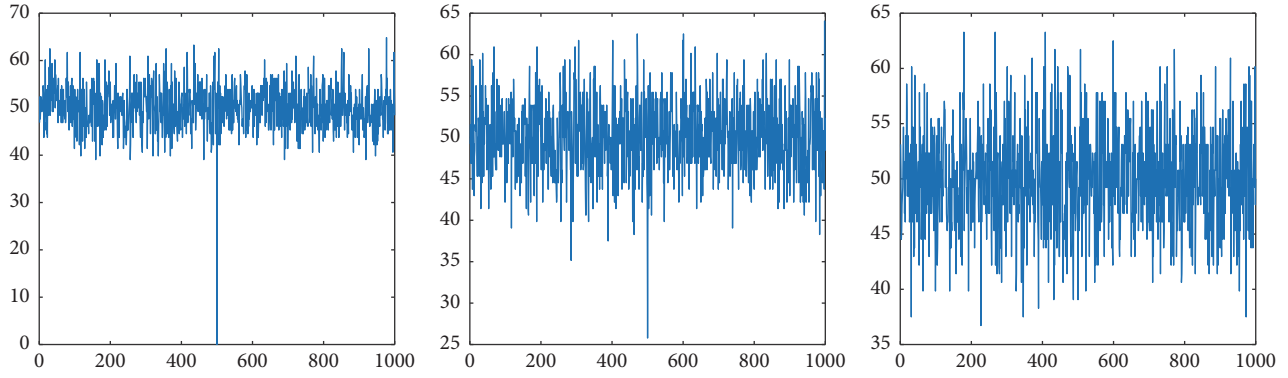
JPEG compression	QF = 25	QF = 8	QF = 5
GF	sigma = 0.3	sigma = 1	sigma = 4
WGN	PSNR = 18	PSNR = 16.75	PSNR = 16.41



(a) JPEG compression for QF of 25, 8, and 5



(b) GF for sigma of 0.3, 1, and 4



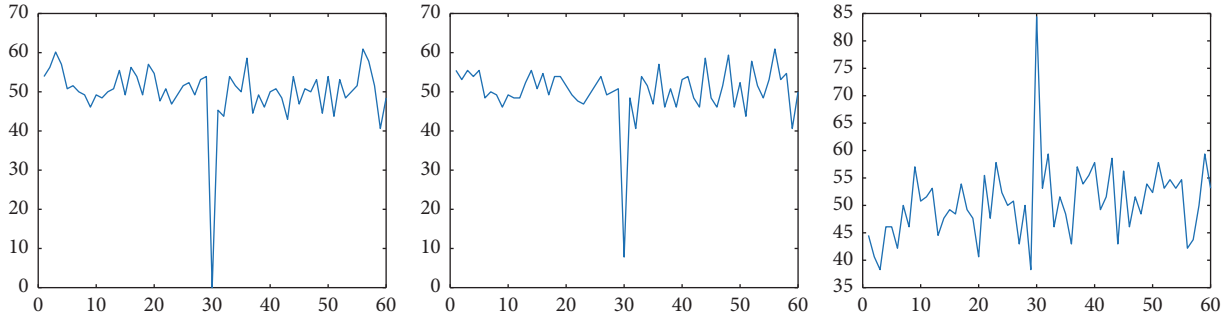
(c) WGN for PSNR of 18, 16.75, and 16.41

Figure 9: Test for $\mu = 0.3$.

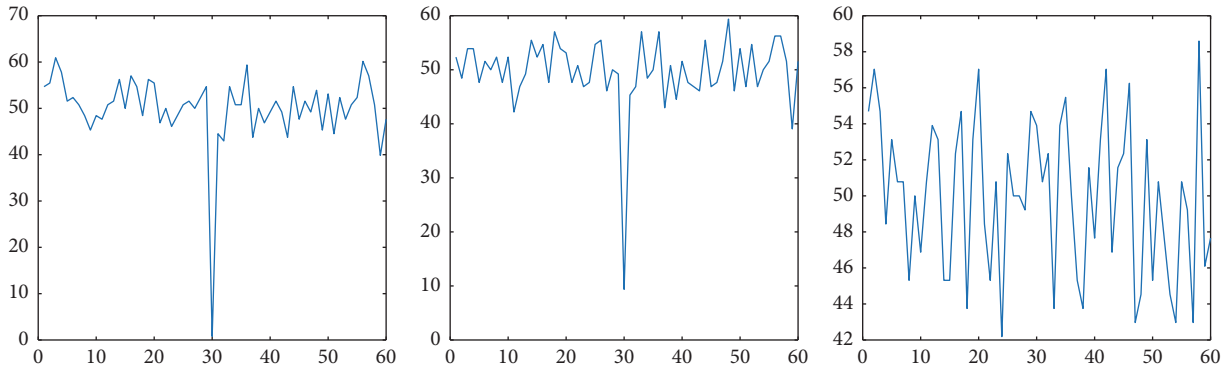
In addition, we raise the threshold to 40% which results in a quickly decreasing of capacity. We can only get 60 watermarks within 10 seconds. We select the 30th watermark for embedding and do the same tests. The results are shown in Figure 10 (The y-axis represents BER and x-axis represents sequence number of images). It suggested that raising up threshold is not economic to increase the robustness of watermark algorithm for it decreases the number of watermarks largely.

9. Conclusion

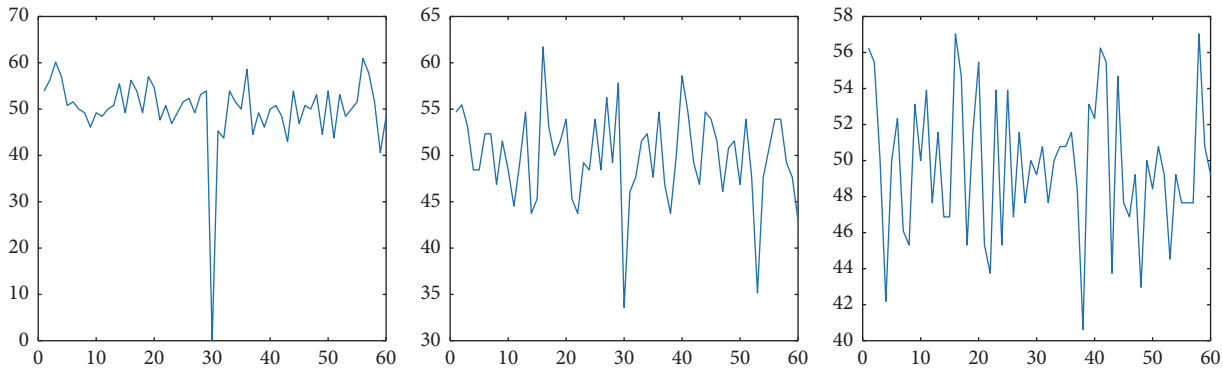
In this paper, we propose a Cloud-User protocol as a solution to solve the Right to Be Forgotten problem technically. Our solution supports confirmed deletion of plain data that is stored in CSP's servers. To achieve security goals, our solution combines the existing homomorphic cryptography, watermark techniques, minimum Hamming coding, and the content-based feature extraction so that the innocent party



(a) JPEG compression for QF of 25, 8, and 5



(b) GF for sigma of 0.3, 1, and 4



(c) WGN for PSNR of 18, 16.75, and 16.41

Figure 10: Test for $\mu = 0.4$.

will not suffer losses by the other one's attack. We implement a prototype of our solution to demonstrate its availability and practicality.

10. Future Work

For future work, there are still some aspects worth thinking. Firstly, the algorithms we used in prototype are not the best ones that fit our solution. Choosing a better encryption algorithm and watermark scheme may decrease the cost of communication and computation for U and TTP.

Secondly, a better FEF can help protecting U's right and raise the robustness of our protocol. We treat the combination of FEF and watermark scheme as the most challenging question for our solution.

Thirdly, every time user retrieves his data will cost a lot for all three parties. Designing a better drawing back protocol can raise the efficiency of our solution.

Finally, as a large pile of data is plain text in CSP's server, how to provide preview of data base on its content like existing systems [27] in low cost while not leaking the information of watermarks is waiting to be solved.

Notations

CSP: Cloud Service Provider
U: User
TTP: Trusted Third Party
D: Data
W: Digital watermark

DC(D):	Derivative copy of D
DS(D):	Derivative set of D
DW(D):	Watermark extraction function that extracts watermark from D
EW(D, M):	Watermark embedding function that embeds M into D
FEF(D):	Feature of D
G_i:	The ith round contract
ID_x:	Identity of x
KEY:	Key for symmetric cryptography
PUB_x:	Public key of x
PRI_x:	Private key of x
a_n:	A string generated by combination of a that has length n
D(KEY/PUB, D/M):	Decryption function for D/M while key is KEY/PUB
E(KEY/PUB, D/M):	Encryption function for D/M while key is KEY/PUB
$\Delta(D_1, D_2)$:	The difference between D_1 and D_2 as $\{D_{1i} - D_{2i} \mid 1 < i < \mathrm{Len}(D_1)\}$.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the Natural Science Foundation of China under Grants U1636201 and 61572452.

References

[1] Directive, EU,95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data,Official Journal of the EC, 23(6), 1995.

[2] Proposal for a Regulation of the European Parliament and of the Council on the Protection of Individuals with Regard to the Processing of Personal Data and on the Free movement of Such Data (General Data Protection Regulation), 2012.

[3] Steinberg, An act to add Chapter 22.1 (commencing with Section 22580) to Division 8 of the Business and Professions Code, relating to the Internet,California senate, 2013.

[4] Google, *European privacy requests for search removals*, 2017, https://www.google.com/transparencyreport/removals/europeprivacy.

[5] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2594–2608, 2016.

[6] Y. Tang, P. P. C. Lee, J. C. S. Lui, and R. Perlman, "Secure overlay cloud storage with access control and assured deletion," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 6, pp. 903–916, 2012.

[7] R. Geambasu, T. Kohno, A. A. Levy et al., "Vanish: Increasing Data Privacy with Self-Destructing Data," in *Proceedings of the USENIX Security Symposium*, pp. 299–316, 2009.

[8] G. Roxana, K. Tadayoshi, A. Amit et al., "Increasing data privacy with self-destructing data," in *Proceedings of the USENIX Security09*, pp. 299–316, Berkeley, CA, USA, 2009.

[9] C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC '09*, pp. 169–178, June 2009.

[10] R. Rivest L, L. Adleman, and L. Dertouzos M, "On data banks and privacy homomorphisms," *Foundations of Secure Computation*, vol. 4, no. 11, pp. 169–180, 1978.

[11] Z. Fu and X. Cao, "An Outsourcing Data Storage Scheme Supporting Privacy Preserving and Data Hiding Based on Digital Watermarking," in *Proceedings of the International Conference on Cloud Computing and Security*, pp. 468–474, Springer, 2016.

[12] J. Long, D. Zhang, C. Zuo, J. Duan, and W. Huang, "A robust low-overheadwatermarking for field authentication of intellectual property cores," *Computer Science and Information Systems*, vol. 13, no. 2, pp. 609–622, 2016.

[13] T. Bianchi and A. Piva, "Secure watermarking for multimedia content protection: a review of its benefits and open issues," *IEEE Signal Processing Magazine*, vol. 30, no. 2, pp. 87–96, 2013.

[14] N. Memon and P. W. Wong, "A buyer-seller watermarking protocol," *IEEE Transactions on Image Processing*, vol. 10, no. 4, pp. 643–649, 2001.

[15] C.-L. Lei, P.-L. Yu, P.-L. Tsai, and M.-H. Chan, "An efficient and anonymous buyer-seller watermarking protocol," *IEEE Transactions on Image Processing*, vol. 13, no. 12, pp. 1618–1626, 2004.

[16] B. Chen and G. W. Wornell, "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 47, no. 4, pp. 1423–1443, 2001.

[17] P. Tao and A. M. Eskicioglu, "A robust multiple watermarking scheme in the Discrete Wavelet Transform domain," in *Proceedings of the Internet Multimedia Management Systems V*, pp. 133–144, October 2004.

[18] J. Molina-Garcia, R. Reyes-Reyes, V. Ponomaryov, and C. Cruz-Ramos, "Watermarking algorithm for authentication and self-recovery of tampered images using DWT," in *Proceedings of the 9th International Kharkiv Symposium on Physics and Engineering of Microwaves, Millimeter and Submillimeter Waves, MSMW 2016*, pp. 1–4, June 2016.

[19] O. Wahballa, A. Abdalla, K. Hamdnaalla, M. Ramadan, and C. Xu, "An efficient and secure certificateless public key watermarking scheme based on 1VD-DWT," in *Proceedings of the 2016 IEEE International Conference on Cloud Computing and Big Data Analysis, ICCCBDA 2016*, pp. 183–188, July 2016.

[20] B. Chen and G. Wornell, "Digital watermarking and information embedding using dither modulation," in *Proceedings of the 1998 IEEE Second Workshop on Multimedia Signal Processing*, pp. 273–278, Redondo Beach, CA, USA.

[21] P. Bas and J. Hurri, "Security of DM quantization watermarking schemes: a practical study for digital images," in *Proceedings of the International Workshop on Digital Watermarking*, pp. 186–200, Springer, 2005.

[22] B. Matam and D. Lowe, "Watermark-only security attack on DM-QIM watermarking: Vulnerability to guided key guessing," in *Proceedings of the Crime Prevention Technologies and Applications for Advancing Criminal Investigation*, pp. 85–106, IGI Global, 2012.

[23] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proceedings of the International*

Conference on the Theory and Applications of Cryptographic Techniques, pp. 223–238, 1999.

[24] C.-C. Lu and S.-Y. Tseng, "Integrated design of AES (Advanced Encryption Standard) encrypter and decrypter," in *Proceedings of the IEEE International Conference on Application-Specific Systems, Architectures, and Processors, ASAP 2002*, pp. 277–285, July 2002.

[25] E. Klinger, "pHash The open source perceptual hash library," 2017, http://www.phash.org/.

[26] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 44, no. 5, pp. 1897–1905, 1998.

[27] S. Pandey, P. Khanna, and H. Yokota, "A semantics and image retrieval system for hierarchical image databases," *Information Processing & Management*, vol. 52, no. 4, pp. 571–591, 2016.

Hindawi
Security and Communication Networks
Volume 2017, Article ID 2314860, 9 pages
https://doi.org/10.1155/2017/2314860

Research Article

Adaptive Steganalysis Based on Selection Region and Combined Convolutional Neural Networks

Donghui Hu,[1] Qiang Shen,[1] Shengnan Zhou,[1] Xueliang Liu,[1] Yuqi Fan,[1] and Lina Wang[2]

[1]*School of Computer and Information, Hefei University of Technology, Hefei 230009, China*
[2]*School of Computer and Information, Wuhan University, Wuhan 430072, China*

Correspondence should be addressed to Donghui Hu; hudh@hfut.edu.cn

Received 19 September 2017; Accepted 23 October 2017; Published 12 November 2017

Academic Editor: Zhenxing Qian

Digital image steganalysis is the art of detecting the presence of information hiding in carrier images. When detecting recently developed adaptive image steganography methods, state-of-art steganalysis methods cannot achieve satisfactory detection accuracy, because the adaptive steganography methods can adaptively embed information into regions with rich textures via the guidance of distortion function and thus make the effective steganalysis features hard to be extracted. Inspired by the promising success which convolutional neural network (CNN) has achieved in the fields of digital image analysis, increasing researchers are devoted to designing CNN based steganalysis methods. But as for detecting adaptive steganography methods, the results achieved by CNN based methods are still far from expected. In this paper, we propose a hybrid approach by designing a region selection method and a new CNN framework. In order to make the CNN focus on the regions with complex textures, we design a region selection method by finding a region with the maximal sum of the embedding probabilities. To evolve more diverse and effective steganalysis features, we design a new CNN framework consisting of three separate subnets with independent structure and configuration parameters and then merge and split the three subnets repeatedly. Experimental results indicate that our approach can lead to performance improvement in detecting adaptive steganography.

1. Introduction

Steganography is a technique for embedding confidential information into multimedia data, which can be used for concealed transmission or copyright protection. Steganalysis is an opposite art to detect the existence of steganography. In past years, researchers have developed a variety of information steganography techniques without affecting image quality. LSB (Least Significant Bit) [1] is a nonadaptive method which does not take into account the contribution of each pixel within an image when embedding information, so that it is proved to be defective with the development of the detection technology.

Some adaptive steganography techniques have been proposed to improve the antidetection ability by adjusting the embedding locations on the basis of the embedding costs. Many currently available adaptive steganography algorithms, such as HUGO BD (Highly Undetectable Stego Bounding Distortion) [2], WOW (Wavelet Obtained Weights) [3], S-UNIWARD (Spatial-Universal Wavelet Relative Distortion) [4], and HILL (High-pass, Low-pass, and Low-pass) [5], have a high antidetection capability, and most of them are designed under the framework of minimizing a distortion function, in which each pixel of an image suitable for embedding information is firstly assigned a high embedding cost. With the cost we can calculate the value of distortion function and obtain the stego via minimizing the distortion function using some coding techniques, such as STCs (Syndrome-Trellis Codes) [6]. To detect content-adaptive schemes, some researchers handcraft various high-dimensional features such as the spatial rich model (SRM) [7–9] and selection channel aware maxSRM [10] and maxSRMd2 [10]. Some other work focuses on designing efficient convolutional neural network (CNN) architectures [11] to extract features directly from the input images. Qian et al. [12] proposed a CNN based steganalysis method using Gaussian activation function and

in [13] they transfer features generated from a pretrained model to regularize CNN model. In literatures [14, 15], Xu et al. proposed a new-type network structure, including absolute value layer and batch normalization layer. In [16, 17], hybrid networks composing many subnetworks are designed to fit the rich-model features set. However, these CNN based steganalysis methods do not consider the characteristics of adaptive steganography and hence have limitation in evolving diverse and effective steganalysis features due to the inflexible CNN frameworks. Recently there are some other articles on steganalysis based on deep learning. Ni et al. [18] proposed a CNN which has a quite different structure from the ones used in conventional computer vision tasks. Besides, a new activation function called truncated linear unit (TLU) is adopted in the CNN model and it incorporates the selection channel inside the network architecture. In [19], Yang et al. proposed another approach towards using selection channel inside the network architecture which can improve the detection accuracy. Compared to the existing CNN model for steganalysis, our model can evolve effective steganalysis features for the detection of adaptive steganography by merging and separating subnets of CNNs and focusing on a most effective region (MER), which can be the reference in designing the new CNN for steganalysis.

The contribution of this paper is as follows: (1) we design a region selection method to find the most effective region (MER) by calculating and comparing the sum of the embedding probabilities of each pixel in a region. The selected region is used as an input image of CNNs. (2) We propose a network consisting of three separate subnets, with each subnet possessing independent structures and diverse parameters, and the three subnets can be merged and separated repeatedly because some studies have shown that widening the network can significantly improve the performance [20]. Experimental results indicate that both the region selection method and the proposed CNN framework can lead to performance improvement in detecting adaptive steganography in some cases.

2. Proposed Method

Most of the state-of-the-art adaptive steganography methods first assign the distortion value for each pixel via a distortion function based on the embedding cost before embedding information, and then some advanced coding techniques, such as STCs, will be applied to minimize the expected distortion value for all pixels in texture areas. Obviously, since the degree of correlation between each pixel and the surrounding pixels is different, according to the adaptive steganography distortion function, pixels with different texture complexity will get diverse loss values. Hence some pixels, which may not be suitable for modification, in texture areas, are assigned with low costs and some other pixels of complex texture are assigned with high costs. We can also clearly study from [8] that the regions with high probability are substantially consistent with the embedding pixels. By large number of experiments (which will be described in Section 3.2.1), we also found that the embedding probability map for each adaptive steganography algorithm is approximately the same. Through this observation, we intend to roughly estimate the position of the modified pixels by embedding probability maps without knowing the specific modification point. Moreover, it is imprecise to use the embedding probability immediately to represent whether a pixel is embedded. Due to the fact that the optimal embedding region obtained by an embedding algorithm may be applicable to other embedding algorithms, in this paper, we use the most effective region instead of embedding probability of a pixel to steganalyze an image.

2.1. The Region Selection Method. In this section we first propose a method to predict embedded pixels using the embedding probability maps to find the most effective region. Our main idea is to first calculate the embedding probability of each pixel and then calculate and compare the sum of the embedding probability of all pixels in different regions to find the maximum one.

In the adaptive steganography, to determine whether a pixel is suitable for modification, a distortion metric [3] $D(X, Y)$ is designed to measure the embedding impact as follows:

$$D(X,Y) = \sum_{i=1,j=1}^{n,m} \rho_{i,j}\left(x_{i,j}, y_{i,j}\right), \tag{1}$$

where $\rho_{i,j}$ is the cost of pixel changes (from $x_{i,j}$ to $y_{i,j}$). Using the distortion function one can easily evaluate the expected distortion and compute the probabilities map. The probability of modification described in [2, 21] can be calculated by

$$\pi_\lambda(Y) = \prod_{i=1,j=1}^{n,m} \frac{\exp\left(-\lambda\rho_{i,j}\left(y_{i,j}\right)\right)}{\sum_{y_{i,j}\in\mathcal{F}_{i,j}} \exp\left(-\lambda\rho_{i,j}\left(y_{i,j}\right)\right)}. \tag{2}$$

where λ is used to satisfy the following distortion constraints:

$$D_\epsilon = \sum_{i=1,j=1}^{n,m} \sum_{y_{i,j}\in\mathcal{F}_{i,j}} \pi_\lambda\left(y_{i,j}\right)\rho_{i,j}\left(y_{i,j}\right), \tag{3}$$

or the payload constraint:

$$s = -\sum_{i=1,j=1}^{n,m} \sum_{y_{i,j}\in\mathcal{F}_{i,j}} \pi_\lambda\left(y_{i,j}\right)\log\pi_\lambda\left(y_{i,j}\right). \tag{4}$$

As shown in Algorithm 1, we estimate the embedding probability of each pixel in an image, enumerate all possible regions (with constant sizes), and calculate the sum of the probabilities of each region to find the region with the maximal value. This region is the most effective region for steganalysis. Figure 1 gives an experimental result of the method. The method is constrained by the selection channel [22] elements of embedding algorithms and embedding rates.

Due to most of the adaptive steganography methods having similar embedding positions, the inaccuracies in the selection channel do not have an impact on the position of MERs. The main reason is probably the following few points. When calculating the embedding probabilities of an image

Input: input parameters: an input image I, the width W and the height H of the region.
Output: output result: the MER O, save as PGM format.
(1) Initialize the probability map (matrix) P with random weights θ;
(2) //Select a pixel from the input image and calculate the probability.
(3) **for** $i = 1$, Rows **do**
(4) **for** $j = 1$, Column **do**
(5) Calculate change costs $D(X, Y)$ using Eq. (1);
(6) Compute λ using Eq. (4);
(7) Set the probability $\pi_\lambda(X, Y) = f(D(X, Y), \lambda)$ using Eq. (2);
(8) Store $\pi_\lambda(X, Y)$ in P;
(9) Initialize the MER O with 0;
(10) //Select the upper left corner coordinates of the area with size of $W \times H$.
(11) **for** $i = 1$, Rows-W **do**
(12) **for** $j = 1$, Column-H **do**
(13) Calculate the sum of the probability in an matrix ($W \times H$) with top-left corner of (i, j);
(14) Statistics out all of the sum and its corresponding i, j;
(15) Select the maximal value of those sums;
(16) Cut the $W \times H$ area in an input image according to i and j;
(17) Save this area O as a PGM format image;

Algorithm 1: Finding the most effective region ($W \times H$) for steganalysis.

Table 1: The ratios of the distance between two MERs extracted by different algorithms.

Algorithm	Distance (the unit for location is pixel).				
	$D = 0$	$0 < D \leq 30$	$30 < D \leq 70$	$70 < D \leq 100$	$D > 100$
WOW 0.2 & HILL 0.2	0.2575	0.5025	0.1182	0.0472	0.0746
WOW 0.2 & HUGO 0.2	0.2544	0.4900	0.1271	0.0489	0.0796
HILL 0.2 & HUGO 0.2	0.2284	0.4137	0.1553	0.0680	0.1346
WOW 0.1 & 0.2	0.4544	0.4534	0.0525	0.0169	0.0228
HILL 0.1 & 0.2	0.4879	0.4421	0.0372	0.0127	0.0201
HUGO 0.1 & 0.2	0.4642	0.4282	0.0603	0.0200	0.0273

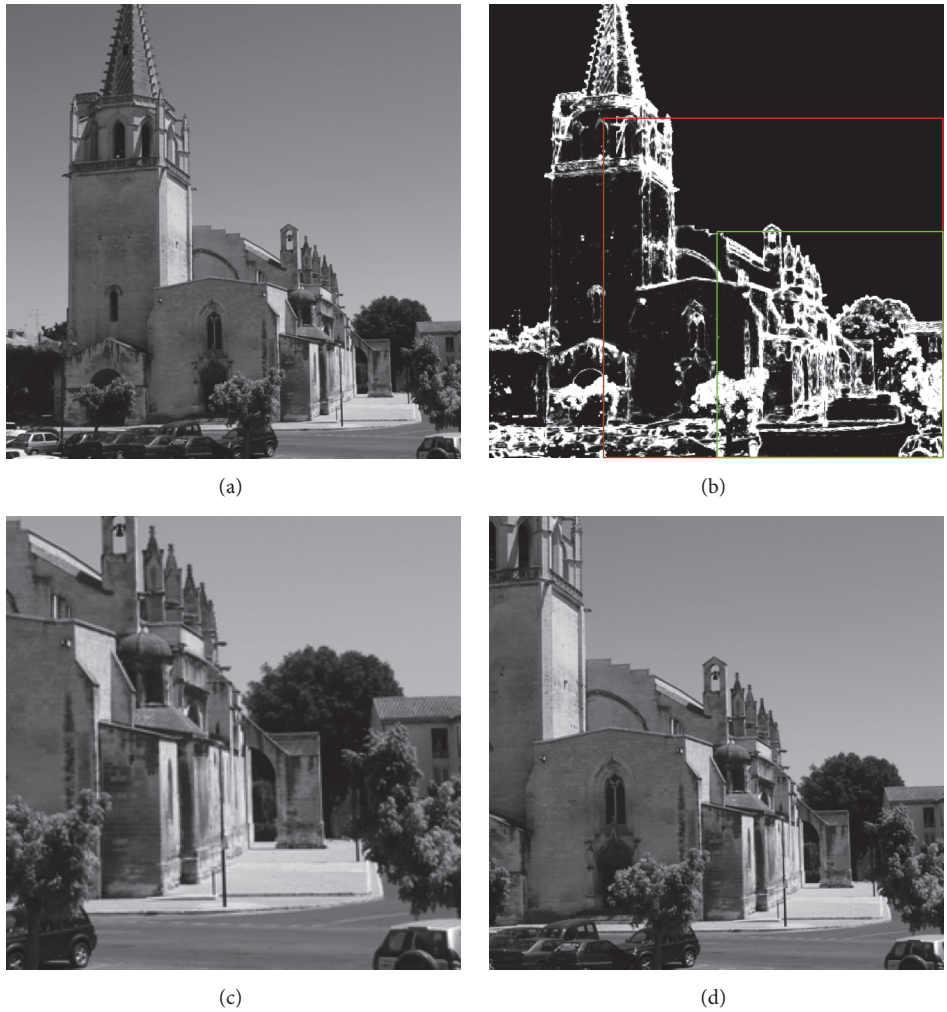
by our region selection method as well as in the extraction process of maxSRM or maxSRMd2, we may need to know the embedding algorithm and embedding rate in advance. Having a priori knowledge can obviously make the model have some limitations when we test the unknown image sets. So we test the robustness of our proposed method to reduce the impact of prior knowledge and we introduce the distance D to denote the distance between the two MERs of images embedding with different algorithms or rates and use the coordinates of the points $P_1(x_1, y_1)$, $P_2(x_2, y_2)$ in the upper left corner of different MERs to mark their position in the original images. D is calculated as follows:

$$|D| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \quad (5)$$

Our experiments given in Section 3 also show that when using different embedding algorithms and embedding rates to estimate the embedding probability, the positions of MERs selected by Algorithm 1 are very close. We would like to use Table 1 to explain that our method is robust and universality. One can even use some kind of embedding algorithm (HILL) to roughly estimate other algorithms (WOW, HUGO BD, S-UNIWARD, etc.), which does not affect the performance. Finally, the selected MERs are used as the input of the CNN model in the following subsection so that the CNN model can focus on the regions with high embedding probabilities.

2.2. The Combined Network

2.2.1. The Overall Structure. Figure 2 shows the network model designed in this paper, including preprocessing module, the convolution and downsampling module, and classification module. Our method broadens the network proposed in [14] with three separate subnets and lengthens the network with "depthconcat" layers, fully connected layers, and dropout layers. In the preprocessing module, the high-pass filter layer uses the KV filter kernel [7, 12], "SQUARE5x5," to obtain the image residuals. After kernel filtering, the data flow into the combined network comprising three independent subnets where the kernel sizes of each convolution layer in each subnet are 1, 3, and 5, respectively. After several convolution and downsampling layers, the data of the three subnets are merged together through the concat layer *DepthConcat* and flow into a new combined network comprising three subnets, where kernel sizes of the first convolution layer of the three subnets are 1, 3, and 1, respectively. Finally the features are merged for the second time. After two merging operations, the network becomes a single one, including two



(a) (b)

(c) (d)

Figure 1: The MER for images embedded by HUGO with payload of 0.2 bits per pixel. (a) Cover image. (b) The embedding probability map. (c) The MER with size 256 × 256 pixels (the region with green line in (b)). (d) The MER with size 384 × 384 pixels (the region with red line in (b)).

convolution layers, two pooling layers, three full connection layers, two dropout layers, and four activation layers, and so forth. The configuration parameters of each layer are given in detail in Figure 2. Taking the first convolution layer as an example, the value 8 outside the parentheses is the number of the kernels, 1×1 is the size of the kernel, $1(S)$ is the stride, and the rest of the configuration is entirely the same. It should be noticed that the kernel size of the last pooling layer is not fixed. When training the areas with different sizes, the dimensions are different in the pooling layer so that we need to dynamically modify the kernel size of the last pooling layer to ensure that the data size is 1×1 before feeding the data to the fully connected layer.

2.2.2. Learning Features from the Network. It is widely recognized that when most of the neural networks are employed for generating the final features, original images are trained by the convolution layer and downsampling layer, and so forth. The purpose of convolution process is to model a correlation between a pixel and its surrounding pixels. For instance, if you set the convolution kernel size to 3, it means that you want to associate a pixel with the 8 points around it (to generate steganalysis feature). Through the establishment of this link, after layers of dimensionality reduction, the final classification can be achieved. At present, the kernel size of each layer is fixed in the design of most neural networks. The large size of the convolution kernel may cause information redundancy; that is, a pixel may establish connections with less relevant pixels. The small size of the convolution kernel may lead to missing some significant information (pixels). Many neural network framework needs to set up a configuration file. Once these parameters have been set, they cannot be arbitrarily modified after training begins. Accordingly, considering the limitations of the single model, we employ three subnets with independent structures to model the feature map with different filter sizes. As described in Section 2.2.1, the two "separating-merging" procedures make the learned features further diversified. In this way, features with different grain sizes can be generated. At the bottom of the structure we add three fully connected layers, among these layers, and we

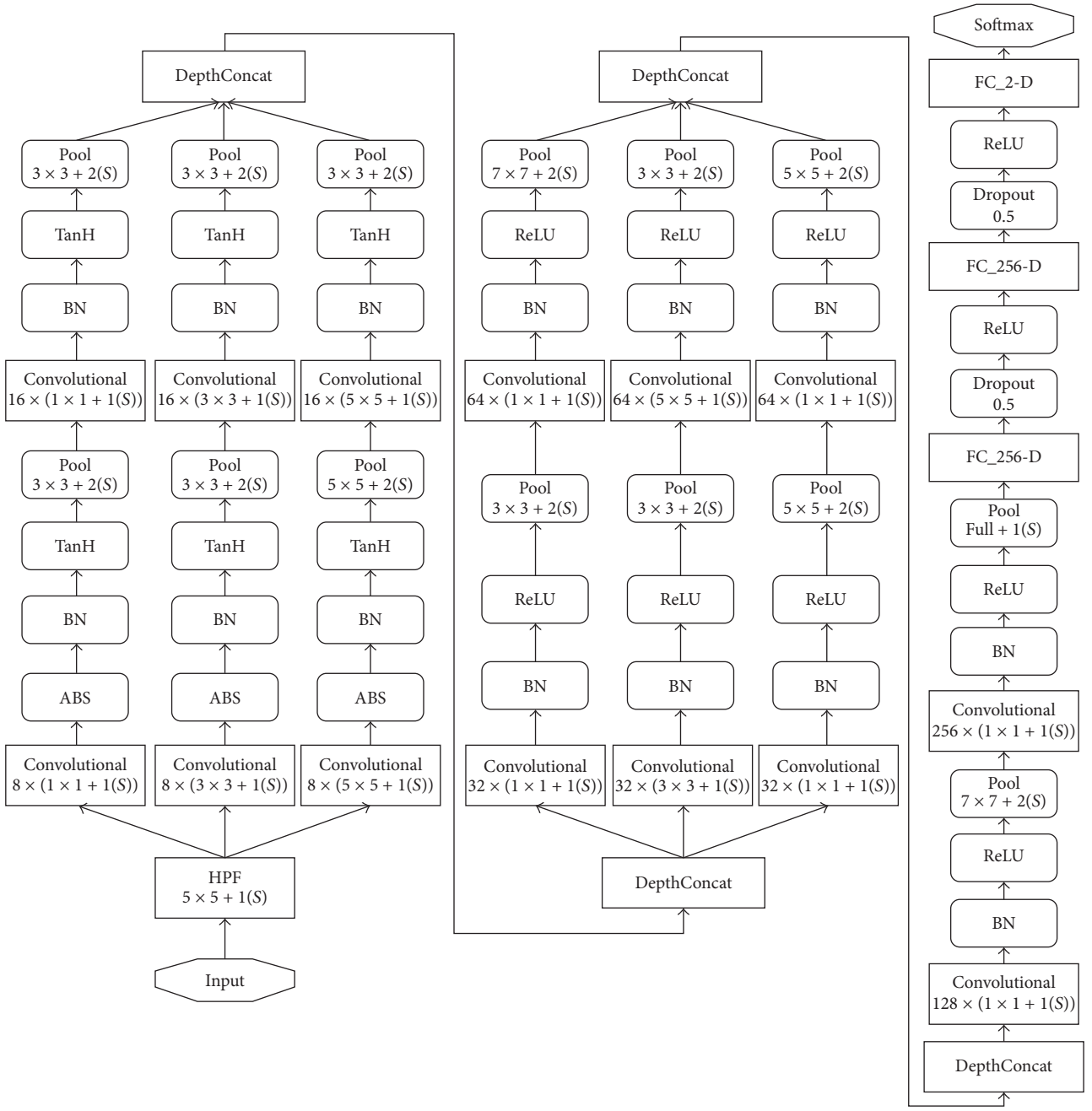


Figure 2: An improved model structure consisting of three separate subnets.

use the dropout layer [23] to sparse the coefficients of the network. "Dropout" means that the weights of some random nodes of the hidden layer are not allowed to work when the model is trained but may work in the next iteration. The mathematical formula is shown as follows:

$$Y = \begin{cases} W|_p * X, & \text{training} \\ W * pX, & \text{testing}, \end{cases} \tag{6}$$

where X, Y are the input and output, respectively, parameter p is the retaining probability of the dropout layer, and $W|_p$ is the subset of weight parameters directly sampled by probability p. During the training phase and testing phase, the output is slightly different as shown in (6).

In training, we calculate the loss and gradient according to the existing label and use the gradient to update the network parameters. In this paper, we select the stochastic gradient descent (SGD) to optimize the parameters of the proposed network architecture, which has several advantages such as high efficiency, fast speed, and simplifying the problem complexity. Deep learning has a limitation that we need to do the parameters initialization manually, which is crucial

to the final results, such as the learning rate, weight decay, the initialization strategy of the parameters, and the ratio of the dropout layer, so that adjusting these parameters is time-consuming. The batch normalization layers [24] are adopted in this model to solve this problem. The normalization process is performed via (7) and (8):

$$\widehat{x}_i = \frac{x_i - E[X]}{\sqrt{\sigma_X^2 + \epsilon}}, \tag{7}$$

where values x_i is an input over a minibatch of $X = \{x_1, x_2, \ldots, x_m\}$. $E[X]$ and σ_X^2 are the mean and variance of X, respectively. The output values are calculated by scaling and shifting the normalized x_i as shown in

$$\widehat{y}_i = \gamma \widehat{x}_i + \beta, \tag{8}$$

where γ and β are learned by the network. Finally the last softmax layer is used for classification.

3. Experiments

3.1. Dataset and Settings. The dataset used in this paper is BOSSbase v1.01 [25] containing 10,000 grayscale images with size of 512 × 512. We test four adaptive steganographic algorithms: HOGU BD [2], S-UNIWARD [4], WOW [3], and HILL [5] with default parameters during embedding information. We also use these algorithms to get the probability maps. For comparison, we use SRM [7] and maxSRMd2 [10] to extract 34,671-dimension feature and classify them with an ensemble classifier [26]. We use the open source framework Caffe [27] to implement the proposed model. In our approach, among which 5000 pairs of images are selected for training, 2500 pairs are selected as the validation set used to optimize the parameters of the model during the training process, and the remaining 2500 pairs are used as test set to evaluate the efficiencies of classification. We use two high performance graphics cards, NVIDIA Geforce GTX TITAN X and NVIDIA Quadro K5200, to speed up the computation and optimization, because the GPU platform is primarily based on the CUDA framework [28] of NVIDIA company which is widely used in scientific computing and can extremely improve the speed of operation at present effectivly.

When training the model, considering the sensitivity of ReLU activation function to the learning rate, we set the initial learning rate to 0.001. For every 50,000 iterations, the learning rate is modified as one-tenth of the current learning rate. The maximum number of iterations is set to 150,000. The momentum used in gradient descent and the weight decay used to prevent overfitting are set to 0.9 and 0.0005, respectively. Note that when the high-pass filter layer in the model is used to obtain the image residuals, it does not need backpropagation process. Consequently, both the learning rate and the weight decay of this layer are set as 0. Except the fully connected layers, the weight decay functions of the other layers are prohibited. For the sake of maximizing the retention of the original image information, we adopt the mean-pooling strategy in all pooling layers. The ratio of dropout layer between each fully connected layer is set to 0.5. Apart from this exception, the learning rate *lr_mult* (a parameter in Caffe) of weight in all layers is set to 1 (the real learning rate is *lr_mult* multiplied by initial learning rate 0.001) and the learning rate *decay_mult* of bias is twice of the above value *lr_mult*.

3.2. Results and Analysis. Table 1 shows the ratios of approximation of the MERs extracted from 10,000 grayscale images with the size of 512 × 512, and the size of the MERs is 384×384. We compute distance D using (5). Taking the value 0.2575 in the upper left corner as an example, it means that 25.75% of the regions extracted by WOW (with embedding rate 0.2 bpp) and HILL (with embedding rate 0.2 bpp) are the same (D is 0). The first three rows are the results when using different steganographic algorithms with the same embedding rates, and the last three rows are the results when using the same embedding algorithms with different embedding rates. Table 2 shows the performance comparison of multiple methods based on the testing error performance. We directly use the output value of the accuracy layer as the detection rates of all experimental results. We test the four adaptive steganography algorithms described in Section 3.1 with embedding rates of 0.1 to 0.4 bpp. Three kinds of MERs with sizes of 256 × 256, 384 × 384, and 512 × 512 are used. The experimental results of the three different sizes of images are shown in columns 3–5. We compare our method with the single network proposed in [14], SRM [7], and maxSRMd2 [10], and the corresponding experimental results are shown in columns 6–8. The best detection result of our proposed method in each row is shown in bold font.

3.2.1. The Effectiveness of Selection of MER. First, we conduct experiments to show that when using different embedding algorithms and embedding rates to estimate the embedding probability for the same image, the MER method and Algorithm 1 we proposed are robustness. From Table 1, we can find most of the distances D are within the range of [0, 30]. When D is greater than 70, the ratio is very small which means that the embedding probability maps calculated by different adaptive steganographic algorithms and different rates are approximately the same, and the positions of MERs selected by Algorithm 1 using different steganographic algorithms and embedding rates are very close. Secondly, from Table 2, we can clearly find that the region selection strategy might be more applicable in the case of low embedding rates for the adaptive, algorithms including HUGO BD [2], WOW [3], and HILL [5], but plays a trivial role in detecting S-UNIWARD [4], because the changed pixels will not be limited only to the textured regions of the image as the payload increases. As the changed pixels start to spread all over the image, the MER based method will become less effective and may cause serious information loss in detecting S-UNIWARD due to lack of considering all of the differences between covers and stegos. In addition, the MERs with size of 384 × 384 are more competitive for detection than those of 256 × 256. We further compare the detection performance on regions with the same size (384 × 384) selected by our method and randomly selected method and show the comparison

Table 2: Performance comparison of multiple methods based on the testing error.

Algorithm	bpp	Proposed method			Method [14]	SRM	maxSRMd2
		256	384	512	512		
WOW	0.1	**0.4144**	0.4349	0.4577	0.4249	0.4056	0.3001
	0.2	0.3235	0.3150	**0.2917**	0.3625	0.3204	0.2353
	0.3	0.2375	0.2202	**0.2160**	0.2473	0.2501	0.1905
	0.4	0.1900	**0.1814**	0.1847	0.2072	0.2092	0.1593
HUGO	0.1	0.4083	**0.3688**	0.3711	0.3957	0.3601	0.3103
	0.2	0.2979	**0.2731**	0.2816	0.3070	0.2834	0.2396
	0.3	0.2533	0.2304	**0.2194**	0.2316	0.2259	0.1990
	0.4	0.1899	0.1741	**0.1517**	0.1746	0.1852	0.1641
HILL	0.1	0.4422	**0.4403**	0.4646	0.4500	0.4354	0.3776
	0.2	0.3464	**0.3344**	0.3494	0.3642	0.3628	0.3151
	0.3	0.2653	0.2581	**0.2491**	0.2683	0.3017	0.2655
	0.4	0.2107	0.1937	**0.1822**	0.2083	0.2503	0.2192
S-UNIWARD	0.1	0.4680	0.4566	**0.4511**	0.4525	0.4075	0.3661
	0.2	0.3624	0.3952	**0.3319**	0.3364	0.3247	0.2912
	0.3	0.2781	0.2601	**0.2511**	0.2675	0.2584	0.2383
	0.4	0.2348	0.2027	**0.1845**	0.2074	0.2052	0.1959

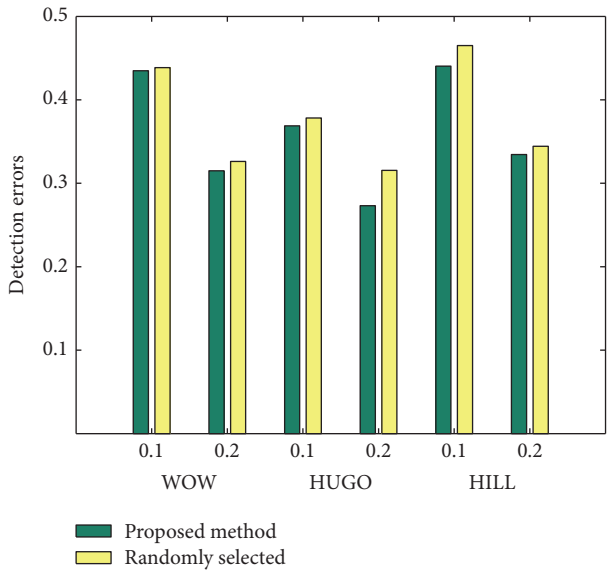


Figure 3: The detection errors of different ways to obtain the MER.

results in Figure 3. The histograms in yellow are the detection results on randomly selected regions and the green ones are produced by our method. We can conclude that, for the three kinds of adaptive steganography algorithms with low embedding rates of 0.1 bpp and 0.2 bpp, our method effectively outperforms by 1%~4% in terms of the detection accuracy compared with the random selection method. More specifically, our proposed method, the region selection, may not be as good as some state-of-the-art methods such as maxSRMd2 [10] because our method is not strictly following the idea of selection channel awareness. At present some of the commonly Selection-Channel-Aware method first estimates the embedding probability of an image and then selects points suitable for feature extraction. As for our method, we estimate the embedding probability to select an area suitable for embedding. However, we treat the points in the MER without distinction when extracting features. In Table 2, MERs with improper size may not be suitable for steganalysis of adaptive steganography algorithms, especially for HUGO and HILL. The total amount of pixels of size 384×384 is about half of the original picture and 256×256 is about a quarter of the original. In the case of appropriate size of the region, the MER selection method achieves better results.

3.2.2. The Effectiveness of the Proposed Network. We also evaluate the effectiveness of the network when it is used without considering the impact of selection of MERs. In Table 2, the 5th column shows the experimental results of our proposed network with original images, and the 6th–8th columns show the performance of all of the compared methods. We can find that in general our network achieves better performance than the method in [14] and SRM [7]. When the embedding rates are high, the detection accuracy is higher than that of the state-of-the-art method maxSRMd2 [10] except for detecting WOW [3]. MaxSRMd2 [10] must first estimate the embedding probability when extracting features. However, a potential disadvantage of maxSRMd2 is that the point where the embedding probability is high does not necessarily correspond to the real embedded point, which may cause a certain amount of information loss. Our proposed method relies on all pixel points when using the combined networks to extract features from 512×512 images. Therefore, our method is more suitable to detect 512×512 images when the embedding rate is high.

4. Conclusion

In this paper, a new region selection method is proposed to find the effective region for CNN to detect adaptive steganographic methods. We also design a combined network consisting of three separate subnets with independent structures. By repeatedly separating and merging the independent subnets with different configuration parameters, the extracted features are more diverse and effective. Experimental results show that our approach has advantages and disadvantages in different situations. At relatively high embedding rates, the proposed combined CNN model outperforms the state-of-the-art steganalysis methods maxSRMd2 except WOW. The region selection strategy might be more applicable in the case of low embedding rates for several adaptive algorithms including HUGO BD, WOW, and HILL. In the future, we will consider to design new method which can choose the MER more accurately and propose new network structures that can evolve more diverse and effective features for steganalysis.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants nos. U1536204 and 61672386 and the Natural Science Research Project of Colleges and Universities in Anhui Province under Grant no. KJ2017A734.

References

[1] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding—a survey," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062–1078, 1999.

[2] T. Filler and J. Fridrich, "Gibbs construction in steganography," *IEEE Transactions on Information Forensics Security*, vol. 5, no. 4, pp. 705–720, 2010.

[3] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in *Proceedings of the IEEE International Workshop on Information Forensics and Security, WIFS 2012*, pp. 234–239, December 2012.

[4] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP Journal on Information Security*, vol. 2014, article 1, 2014.

[5] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in *Proceedings of the IEEE International Conference on Image Processing*, pp. 4206–4210, 2014.

[6] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Transactions on Information Forensics Security*, vol. 6, no. 3, pp. 920–935, 2011.

[7] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics Security*, vol. 7, no. 3, pp. 868–882, 2012.

[8] W. Tang, H. Li, W. Luo, and J. Huang, "Adaptive steganalysis against WOW embedding algorithm," in *Proceedings of the the 2nd ACM workshop*, pp. 91–96, Salzburg, Austria, June 2014.

[9] W. Tang, H. Li, W. Luo, and J. Huang, "Adaptive steganalysis based on embedding probabilities of pixels," *IEEE Transactions on Information Forensics Security*, vol. 11, no. 4, pp. 734–745, 2016.

[10] T. Denemark, V. Sedighi, V. Holub, R. Cogranne, and J. Fridrich, "Selection-channel-aware rich model for Steganalysis of digital images," in *Proceedings of the 2014 IEEE International Workshop on Information Forensics and Security, WIFS 2014*, pp. 48–53, December 2014.

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.

[12] Y. Qian, J. Dong, W. Wang, and T. Tan, "Deep learning for steganalysis via convolutional neural networks," in *Proceedings of the IS&T/SPIE Electronic Imaging*, p. 94090J, San Francisco, Calif, USA.

[13] Y. Qian, J. Dong, W. Wang, and T. Tan, "Learning and transferring representations for image steganalysis using convolutional neural network," in *Proceedings of the 23rd IEEE International Conference on Image Processing, ICIP 2016*, pp. 2752–2756, September 2016.

[14] G. Xu, H. Wu, and Y. Q. Shi, "Ensemble of CNNs for Steganalysis: An empirical study," in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, pp. 103–107, ACM, June 2016.

[15] G. Xu, H.-Z. Wu, and Y.-Q. Shi, "Structural design of convolutional neural networks for steganalysis," *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 708–712, 2016.

[16] J. Zeng, S. Tan, B. Li, and J. Huang, "Large-scale jpeg steganalysis using hybrid deep-learning framework," 2017.

[17] J. Zeng, S. Tan, and B. Li, "Pre-training via fitting deep neural network to rich-model features extraction procedure and its effect on deep learning for steganalysis," in *Proceedings of the Media Watermarking, Security, and Forensics, Part of IS&T International Symposium on Electronic Imaging (EI '17)*, Burlingame, Calif, USA, Juanuary-February 2017.

[18] J. Ni, J. Ye, and Y. I. Yang, "Deep learning hierarchical representations for image steganalysis," *IEEE Transactions on Information Forensics Security*, 2017.

[19] J. Yang, K. Liu, X. Kang, E. Wong, and Y. Shi, "Steganalysis Based on Awareness of Selection-Channel and Deep Learning," in *Digital Forensics and Watermarking*, vol. 10431 of *Lecture Notes in Computer Science*, pp. 263–272, Springer International Publishing, 2017.

[20] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pp. 1–9, Boston, Mass, USA, June 2015.

[21] J. Fridrich and T. Filler, "Practical methods for minimizing embedding impact in steganography," in *Proceedings of the Electronic Imaging*, p. 650502, San Jose, Calif, USA, 2007.

[22] V. Sedighi and J. Fridrich, "Effect of imprecise knowledge of the selection channel on steganalysis," in *Proceedings of the 3rd ACM Information Hiding and Multimedia Security Workshop, IH and MMSec 2015*, pp. 33–42, June 2015.

[23] G. Hinton E, N. Srivastava, and A. Krizhevsky, "Improving neural networks by preventing co-adaptation of feature detectors," *Computer Science*, vol. 3, no. 4, pp. 212–223, 2012.

[24] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," https://arxiv.org/abs/1502.03167.

[25] P. Bas, T. Filler, and T. Pevný, ""Break Our Steganographic System": The Ins and Outs of Organizing BOSS," in *Information Hiding*, vol. 6958 of *Lecture Notes in Computer Science*, pp. 59–70, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[26] J. Kodovsky, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE Transactions on Information Forensics Security*, vol. 7, no. 2, pp. 432–444, 2012.

[27] Y. Jia, E. Shelhamer, J. Donahue et al., "Caffe: convolutional architecture for fast feature embedding," in *Proceedings of the ACM International Conference on Multimedia*, pp. 675–678, ACM, Orlando, Fla, USA, November 2014.

[28] J. Sanders and E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming*, Addison-Wesley Professional, 2010, Addison-Wesley Professional.

Hindawi
Security and Communication Networks
Volume 2017, Article ID 4892814, 9 pages
https://doi.org/10.1155/2017/4892814

Research Article

Noninteractive Verifiable Outsourcing Algorithm for Bilinear Pairing with Improved Checkability

Yanli Ren,[1] Min Dong,[1] Zhihua Niu,[2] and Xiaoni Du[3]

[1]*School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China*
[2]*School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China*
[3]*College of Mathematics and Information Science, Northwest Normal University, Lanzhou 730070, China*

Correspondence should be addressed to Yanli Ren; renyanli@shu.edu.cn

Received 19 June 2017; Revised 27 August 2017; Accepted 6 September 2017; Published 15 October 2017

Academic Editor: Rémi Cogranne

It is well known that the computation of bilinear pairing is the most expensive operation in pairing-based cryptography. In this paper, we propose a noninteractive verifiable outsourcing algorithm of bilinear pairing based on two servers in the one-malicious model. The outsourcer need not execute any expensive operation, such as scalar multiplication and modular exponentiation. Moreover, the outsourcer could detect any failure with a probability close to 1 if one of the servers misbehaves. Therefore, the proposed algorithm improves checkability and decreases communication cost compared with the previous ones. Finally, we utilize the proposed algorithm as a subroutine to achieve an anonymous identity-based encryption (AIBE) scheme with outsourced decryption and an identity-based signature (IBS) scheme with outsourced verification.

1. Introduction

Outsourcing computation has received widespread attention with the development of cloud computing and the proliferation of mobile devices [1]. Despite of the huge benefits, it also encounters some security concerns and challenges. Firstly, the computation tasks often include some private information that should not be disclosed to the cloud servers, since the servers are not fully trusted. Secondly, the cloud servers may return an invalid result, but the outsourcer fails to detect the error [1]. Therefore, two main security challenges of the outsourcing computation are privacy and checkability: (1) the cloud servers cannot learn anything about the private inputs and the outputs of the computation outsourced to them; (2) the outsourcer can detect any failure if the cloud servers return a wrong computation result.

Verifiable computation (VC) allows a client with limited computation capability to outsource evaluation of a function on some inputs to a powerful but semitrusted server [2, 3]. The client in this model first executes a lot of off-line computation and encrypts the function which will be evaluated and then sends the encrypted function to the server. The server then performs the computation on the encoded function and responds with a result and a proof that the result is correct. Finally, the client verifies whether the computation has been carried out honestly based on the server's proof. During the whole process, the computation cost of the client is less than computing the function directly itself.

Our Contributions. In this paper, we propose a noninteractive verifiable outsourcing algorithm of bilinear pairing in the one-malicious model of two untrusted servers, which improves the checkability of the outsourcer without any interactive operation between the outsourcer and the server. In the proposed algorithm, the outsourcer could detect any failure with a probability close to 1 if one of the servers returns the false result. The proposed algorithm improves the checkability at the expense of only a little efficiency when compared with previous algorithms. Finally, we utilize the proposed algorithm as a subroutine to achieve an AIBE scheme with outsourced decryption and an IBS scheme with outsourced verification.

1.1. Related Works. In the cryptographic community, outsourcing expensive operations to a semitrusted device is widely studied. Chaum and Pedersen [4] introduced the concept of "wallets with observers" that allows installing a piece of hardware on the client's device to execute some operations for each transaction. Hohenberger and Lysyanskaya formalized this model [5] and presented algorithms for the computation of modular exponentiations (MExps) based on two noncolluding servers. Further, Chen et al. [1] proposed a new outsourcing algorithm for MExps with improved efficiency and checkability based on the same model as [5]. However, it is still possible for the outsourcer to be cheated by the server. Ren et al. then constructed a verifiable outsourcing scheme of MExps, where the outsourcer can detect the error with a probability of 1 if the server misbehaves [6]. Lai et al. [7] proposed an attribute-based encryption (ABE) scheme with verifiable outsourcing decryption, which guaranteed that an outsourcer can efficiently detect the wrong results. Qin et al. [8] then proposed new ABE scheme with outsourced decryption, where the outsourcer could verify the outsourcing results with a high efficiency at the expense of minimal overhead. Chen et al. first considered the problem of outsourcing computation in attribute-based signature (ABS) schemes and delegated the verification of signature to an untrusted server [9]. Yu et al. [10] proposed a secure and efficient cloud storage auditing scheme with verifiable outsourcing of key updates. The process of key updates is outsourced to the third party auditor (TPA), and the TPA only knows the encrypted secret key. Meanwhile, the outsourcer could verify the effectiveness of encrypted secret keys when uploading new files to the cloud server. Also, Wang et al. [11] proposed a privacy-preserving public auditing system for data storage security and extended it to handle the problem of multiple auditing, where the TPA could learn nothing about data and the integrity of data could be verified publicly. Other works target specific classes of functions, such as revocable identity-based encryption [12], solution of linear equations [13], and image features extraction [14].

In recent years, bilinear pairings have various applications in constructing new cryptographic primitive, for example, identity-based encryption [15], short signature [16], and key agreement protocol [17]. In pairing-based cryptography, the computation of bilinear pairing is the most expensive operation and it has important effects on efficiency of these schemes or protocols. Thus, a lot of research work has been done to compute bilinear pairing efficiently [18, 19].

Chevallier-Mames et al. [20] presented the first algorithm for secure outsourcing of bilinear pairings based on an untrusted server, where the outsourcer could detect any failure with probability of 1 if the server returns an incorrect result. However, the outsourcer must execute some other expensive operations such as scalar multiplications and modular exponentiations, where these computations are even comparable to those of bilinear pairings in some scenarios [19, 21]. Subsequently, other works on delegation of bilinear pairings [22, 23] also suffer from the same problems. Chen et al. proposed the first efficient outsourcing algorithm of bilinear pairing in the one-malicious version of two untrusted program models [24], where the outsourcer only carried out 5 point additions and 4 multiplications without any expensive operations, which is suitable for the computation-limited client. However, the checkability of the algorithm in [24] is only 1/2, and the outsourcer may accept a false result returned by a malicious server with probability of 1/2. Tian et al. presented two outsourcing algorithms of bilinear pairing based on two servers [25]. One is more efficient than the algorithm of [24], and the outsourcer needs to execute 4 point additions and 3 multiplications with the same checkability. The other algorithm is more flexible based on two untrusted servers with improved checkability. As we know, it is also possible for the outsourcer to be cheated by the server and the error cannot be detected successfully. Recently, Ren et al. presented a new outsourcing algorithm of bilinear pairing, which improves the checkability of the outsourcer to 1, and it is impossible for the server to cheat the outsourcer to accept a false outsourcing result [26]. However, it needs two interactive rounds between the outsourcer and the server and increases the communication cost, though the checkability is improved to 1.

1.2. Organization. The rest of this paper is organized as follows. In Section 2, we introduce the definition of bilinear pairing and security model of the outsourcing scheme. A noninteractive verifiable outsourcing algorithm of bilinear pairing is presented and its security analysis is given in Section 3. In Section 4, we introduce two applications of the proposed outsourcing scheme: an AIBE scheme with outsourced decryption and an IBS scheme with outsourced verification. The performance evaluation of the proposed scheme is presented in Section 5. In Section 6, we conclude the paper.

2. Definitions

In this section, we introduce the properties of bilinear pairings, security definition, and model of the proposed outsourcing algorithms.

2.1. Bilinear Pairings. Let q be a large prime, $G, \widehat{G}$ are two cyclic addition groups of order q, and G_T is a cyclic multiplicative group of order q. P, Q are generators of $G, \widehat{G}$, respectively. $e : G \times \widehat{G} \to G_T$ is a bilinear map with the following properties [15, 16, 21]:

1. Bilinear: $e(a_0 R, b_0 V) = e(R, V)^{a_0 b_0}$ for any $R \in G,\ V \in \widehat{G}$, and $a_0, b_0 \in {Z_q}^*$
2. Nondegenerate: there exist $R_0 \in G,\ V_0 \in \widehat{G}$ such that $e(R_0, V_0) \neq 1$
3. Computable: there is an efficient algorithm to compute $e(R, V)$ for any $R \in G,\ V \in \widehat{G}$

The bilinear map and the bilinear pairing can be realized by supersingular elliptic curves or hyperelliptic curves over finite groups and Weil or Tate pairings, respectively [15, 16, 21].

2.2. Security Definition and Model. Now we review the formal security definition of an outsourcing algorithm introduced by [5]. An algorithm *Alg* includes a trusted part T and an untrusted program U, and T^U denotes the works carried out by T invoking U. An adversary A is simulated by a pair of algorithms (E, U'), where E denotes the adversarial environment that submits adversarial inputs for *Alg* and U' represents adversarial software written by E. As described in [5], we assume that the two adversaries (E, U') can only make direct communication before the execution of T^U, and, in other cases, they can only communicate with each other by passing messages through the outsourcer T.

Definition 1 (algorithm with outsource *I/O*). An algorithm *Alg* takes five inputs and generates three outputs. The first three inputs are chosen by an honest party, and the last two inputs are generated by the environment E. The first input is honest and secret, which is unknown for both E and U'; the second is honest and protected, which may be public for E but is private for U'; the third is honest and unprotected, which may be public for both E and U'; the fourth is adversarial and protected, which is public for E but is protected from U'; and the last one is adversarial and unprotected, which is public for E and U'. Similarly, the first output is secret, which is protected from E and U'; the second is protected, which may be public for E but not U'; and the third is unprotected, which may be public for both E and U'.

The following security definition ensures that both E and U' cannot obtain any information about the private inputs and outputs of T^U, even if T uses the malicious software U' written by E.

Definition 2 (outsource-security). Let *Alg* be an algorithm with outsource *I/O*. T^U is called an outsource-secure implementation of *Alg* if the following conditions hold:

(1) *Correctness*: T^U is a correct implementation of *Alg*

(2) *Security*: for all probabilistic polynomial-time (PPT) adversaries $A = (E, U')$, there exist two PPT simulators (S_1, S_2) such that the following pairs of random variables are computationally indistinguishable

Pair One. $EVIEW_{\text{real}} \sim EVIEW_{\text{ideal}}$, which means that the malicious environment E cannot gain anything interesting about the private inputs and outputs during the execution of T^U. The detailed definitions of the real process and the ideal process are omitted because of limited space; please see [5] for the details.

Pair Two. $UVIEW_{\text{real}} \sim UVIEW_{\text{ideal}}$, which means that the untrusted software U' written by E learns nothing about the inputs and outputs during the execution of T^U. Please also see [5] for the detailed definitions.

Assume that T^U is a correct implementation of *Alg*; we have the following definitions.

Definition 3 (α-efficient, secure outsourcing). A pair of algorithms (T, U) are α-efficient if the running time of $T$is not more than an α-multiplicative factor of that of *Alg* for any input x.

Definition 4 (β-checkable, secure outsourcing). A pair of algorithms (T, U) are β-checkable if T detects any deviation of U' from its advertised functionality during the implementation of $T^{U'(x)}$ with probability not less than β for any input x.

Definition 5 ((α, β)-outsource-security). A pair of algorithms (T, U) are called an (α, β)-outsource-secure implementation of *Alg* if they are both α-efficient and β-checkable.

The proposed algorithms are executed based on two untrusted program models introduced by [5]. In this model, the adversarial environment E writes two programs $U' = (U_1', U_2')$, and T installs these programs in a manner such that all subsequent communication between any two of E, U_1', and U_2' must pass through T. The new adversary attacking T is $A = (E, U_1', U_2')$. We assume that at most one of the programs misbehaves, but we do not know which one. It is named as the one-malicious version of two untrusted models. In the real world, it is equivalent to buying two copies of the untrusted software from different vendors and achieving the outsource security as long as one of them is honest [1].

3. Verifiable Secure Outsourcing of Bilinear Pairing

As [5], a subroutine named Rand is used to speed up the computations. The inputs for Rand are a prime q, two cyclic addition groups $G, \widehat{G}$ of order q, and a bilinear map $e : G \times \widehat{G} \to G_T$, where G_T is a cyclic multiplicative group of order q and the output for each invocation is a random, independent vector of the following form:

$$\begin{aligned}
&\big(t_1, t_2, a_1P + a_2P, a_3P, a_4P, b_1P + b_2P, b_3P,\\
&\quad -(a_1P + a_2P + b_3P), -(t_2a_1P + a_2P),\\
&\quad -(a_1P + t_2a_2P), a_1Q + a_2Q, a_3Q, b_1Q + b_2Q, b_3Q,\\
&\quad b_4Q, -(b_1Q + b_2Q + a_3Q), -(t_1b_1Q + b_2Q),\\
&\quad -(b_1Q + t_1b_2Q), e(a_3P, a_3Q), e(b_3P, b_3Q),\\
&\quad e(a_4P, b_1Q + b_2Q)^{t_1+1}, e(a_1P + a_2P, b_4Q)^{t_2+1},\\
&\quad e(a_1P + a_2P, b_1Q + b_2Q)^{-1}\big),
\end{aligned} \tag{1}$$

where $a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4 \in_R Z_q^*$, $t_1, t_2 \in [2, 3, \ldots, s]$, and s is a small number.

We can use the table-lookup method to implement this functionality. First, a trusted server computes a table of random, independent vectors in advance and then stores it into the memory of T. For each invocation of i, T needs to retrieves a new vector in the table.

3.1. Verifiable Outsourcing Algorithm. We propose a noninteractive verifiable outsourcing algorithm *NIVBP* for bilinear

pairing in the one-malicious model. In *NIVBP* algorithm, T outsources its bilinear pairing computations to U_1 and U_2 by invoking the subroutine Rand. A requirement for *NIVBP* is that the adversary A cannot know any useful information about the inputs and outputs of *NIVBP*.

Let q be a large prime. The input of *NIVBP* is $A \in G$ and $B \in \widehat{G}$, and the output is $e(A, B)$. A and B are both computationally blinded to U_1 and U_2. The proposed *NIVBP* algorithm is described as follows:

(1) T firstly runs Rand one time to create a blinding vector as (1).

(2) T queries U_1 in random order as follows:

$$\begin{aligned}
&U_1\left(A + a_1P + a_2P, B + b_1Q + b_2Q\right) \longrightarrow \\
&\alpha_{11} = e\left(A + a_1P + a_2P, B + b_1Q + b_2Q\right), \\
&U_1\left(A + b_1P + b_2P, a_3Q\right) \longrightarrow \\
&\alpha_{12} = e\left(A + b_1P + b_2P, a_3Q\right), \\
&U_1\left(-b_3P - a_1P - a_2P, B + b_3Q\right) \longrightarrow \\
&\alpha_{13} = e\left(-b_3P - a_1P - a_2P, B + b_3Q\right), \\
&U_1\left(A + a_4P, -t_1b_1Q - b_2Q\right) \longrightarrow \\
&\alpha_{14} = e\left(A + a_4P, -t_1b_1Q - b_2Q\right), \\
&U_1\left(-t_2a_1P - a_2P, B + b_4Q\right) \longrightarrow \\
&\alpha_{15} = e\left(-t_2a_1P - a_2P, B + b_4Q\right).
\end{aligned} \tag{2}$$

Similarly, T queries U_2 in random order as follows:

$$\begin{aligned}
&U_2\left(A + a_1P + a_2P, B + b_1Q + b_2Q\right) \longrightarrow \\
&\alpha_{21} = e\left(A + a_1P + a_2P, B + b_1Q + b_2Q\right), \\
&U_2\left(A + a_3P, -a_3Q - b_1Q - b_2Q\right) \longrightarrow \\
&\alpha_{22} = e\left(A + a_3P, -a_3Q - b_1Q - b_2Q\right), \\
&U_2\left(b_3P, B + a_1Q + a_2Q\right) \longrightarrow \\
&\alpha_{23} = e\left(b_3P, B + a_1Q + a_2Q\right), \\
&U_2\left(A + a_4P, -b_1Q - t_1b_2Q\right) \longrightarrow \\
&\alpha_{24} = e\left(A + a_4P, -b_1Q - t_1b_2Q\right), \\
&U_2\left(-a_1P - t_2a_2P, B + b_4Q\right) \longrightarrow \\
&\alpha_{25} = e\left(-a_1P - t_2a_2P, B + b_4Q\right).
\end{aligned} \tag{3}$$

(3) T verifies whether U_1 and U_2 generate the correct outputs, which means that (4)–(6) hold.

(a)

$$\alpha_{11} = \alpha_{21} \tag{4}$$

(b)

$$\begin{aligned}
&\left(\alpha_{12} \cdot \alpha_{22} \cdot e\left(a_3P, a_3Q\right)\right)^{t_1+1} \\
&\quad = \alpha_{14} \cdot \alpha_{24} \cdot e\left(a_4P, b_1Q + b_2Q\right)^{t_1+1}
\end{aligned} \tag{5}$$

(c)

$$\begin{aligned}
&\left(\alpha_{13} \cdot \alpha_{23} \cdot e\left(b_3P, b_3Q\right)\right)^{t_2+1} \\
&\quad = \alpha_{15} \cdot \alpha_{25} \cdot e\left(a_1P + a_2P, b_4Q\right)^{t_2+1}.
\end{aligned} \tag{6}$$

If not, T outputs "error"; otherwise, T outputs

$$\begin{aligned}
&\alpha_{12} \cdot \alpha_{22} \cdot e\left(a_3P, a_3Q\right) = e\left(A, b_1Q + b_2Q\right)^{-1}, \\
&\alpha_{13} \cdot \alpha_{23} \cdot e\left(b_3P, b_3Q\right) = e\left(a_1P + a_2P, B\right)^{-1},
\end{aligned} \tag{7}$$

$$\begin{aligned}
&e(A, B) \\
&\quad = \alpha_{11} \cdot e\left(A, b_1Q + b_2Q\right)^{-1} \cdot e\left(a_1P + a_2P, B\right)^{-1} \\
&\qquad \cdot e\left(a_1P + a_2P, b_1Q + b_2Q\right)^{-1}.
\end{aligned} \tag{8}$$

Correctness. It is obvious that formula (4) holds if two servers are all honest. In addition,

$$\begin{aligned}
&\alpha_{12} \cdot \alpha_{22} \cdot e\left(a_3P, a_3Q\right) \\
&\quad = e\left(b_1P + b_2P, a_3Q\right) e\left(A + a_3P, -b_1Q - b_2Q\right) \\
&\quad = e\left(A, b_1Q + b_2Q\right)^{-1},
\end{aligned} \tag{9}$$

$$\begin{aligned}
&\alpha_{13} \cdot \alpha_{23} \cdot e\left(b_3P, b_3Q\right) \\
&\quad = e\left(-a_1P - a_2P, B + b_3Q\right) e\left(b_3P, a_1Q + a_2Q\right) \\
&\quad = e\left(a_1P + a_2P, B\right)^{-1},
\end{aligned} \tag{10}$$

$$\begin{aligned}
&\left(\alpha_{12}\alpha_{22}e\left(a_3P, a_3Q\right)\right)^{t_1+1} \\
&\quad = \alpha_{14}\alpha_{24}e\left(a_4P, b_1Q + b_2Q\right)^{t_1+1} \\
&\quad = e\left(A + a_4P, b_1Q + b_2Q\right)^{-(t_1+1)} e\left(a_4P, b_1Q + b_2Q\right)^{t_1+1} \\
&\quad = e\left(A, b_1Q + b_2Q\right)^{-(t_1+1)},
\end{aligned} \tag{11}$$

$$\begin{aligned}
&\left(\alpha_{13}\alpha_{23}e\left(b_3P, b_3Q\right)\right)^{t_2+1} \\
&\quad = \alpha_{15}\alpha_{25}e\left(a_1P + a_2P, b_4Q\right)^{t_2+1} \\
&\quad = e\left(a_1P + a_2P, B + b_4Q\right)^{-(t_2+1)} e\left(a_1P + a_2P, b_4Q\right)^{t_2+1} \\
&\quad = e\left(a_1P + a_2P, B\right)^{-(t_2+1)}.
\end{aligned} \tag{12}$$

Therefore, formulas (4)–(6) hold according to the above analysis. Finally, T obtains $e(A, B)$ as (8).

Remark 6. If one of the servers is dishonest, the results could be verified successfully with a probability close to 1 except that

the dishonest server knows the values of $\alpha_{11}, \alpha_{12}, \alpha_{13}, \alpha_{14}, \alpha_{15}$ (or $\alpha_{21}, \alpha_{22}, \alpha_{23}, \alpha_{24}, \alpha_{25}$) and t_1, t_2. As we know, five queries sent to U_1 and U_2 are submitted in random order and $t_1, t_2 \in [2, 3, \ldots, s]$. So, the dishonest server could guess the values of $\alpha_{11}, \alpha_{12}, \alpha_{13}, \alpha_{14}, \alpha_{15}$ (or $\alpha_{21}, \alpha_{22}, \alpha_{23}, \alpha_{24}, \alpha_{25}$) and t_1, t_2 with the probabilities of $1/5!$ and $1/(s-1)^2$, respectively. Therefore, the checkability of the *NIVBP* algorithm is

$$1 - \frac{1}{5!\,(s-1)^2} = 1 - \frac{1}{120\,(s-1)^2} \approx 1. \tag{13}$$

Remark 7. The proposed algorithm *NIVBP* is also applicative in the condition where $G, \widehat{G}$ are two cyclic multiplication groups. Let $g, \hat{g}$ be generators of $G, \widehat{G}$, respectively. $e : G \times \widehat{G} \to G_T$ is a bilinear map. In this case, the inputs of *NIVBP* are also $A \in G$ and $B \in \widehat{G}$, and the output is also $e(A, B)$. The details are also omitted because of limited space.

3.2. Security Analysis

Theorem 8. *In the one-malicious model, the proposed algorithm $(T, (U_1, U_2))$ is an outsource-secure implementation of NIVBP, where the input (A, B) may be honest and secret or honest and protected or adversarial and protected.*

Proof. Let $A = (E, U_1', U_2')$ be a PPT adversary that interacts with a PPT algorithm T in the one-malicious model.

First, we prove that $\text{EVIEW}_{\text{real}} \sim \text{EVIEW}_{\text{ideal}}$, which means that the environment E learns nothing during the execution of $(T, (U_1, U_2))$. If the input (A, B) is honest and protected or adversarial and protected, it is obvious that the simulator S_1 behaves the same as in the real execution. Therefore, we only need to prove the case where (A, B) is an honest, secret input.

So, suppose that (A, B) is an honest, secret input. The simulator S_1 in the ideal experiment behaves as follows. On receiving the input on round i, S_1 ignores it and instead makes five random queries of the form (α_j, β_j) to both U_1' and U_2' Finally, S_1 randomly checks one output $e(\alpha_j, \beta_j)$ from each program. If an error is detected, S_1 saves all states and outputs $Y_p^i =$ "error," $Y_u^i = \varnothing$, $\text{rep}^i = 1$, and thus the final output for ideal process is $(\text{estate}^i, \text{"error,"}\ \varnothing)$. If no error is detected, S_1 checks the remaining four outputs. If all checks pass, S_1 outputs $Y_p^i = \varnothing$, $Y_u^i = \varnothing$, $\text{rep}^i = 0$; that is, the final output for ideal process is $(\text{estate}^i, y_p^i, y_u^i)$; otherwise, S_1 selects a random element r and outputs $Y_p^i = r, Y_u^i = \varnothing, \text{rep}^i = 1$, and the output for ideal process is $(\text{estate}^i, r, \varnothing)$. □

In addition, we need to show that the inputs to (U_1', U_2') in the real experiment are computationally indistinguishable from those in the ideal one. In the ideal experiment, the inputs are selected uniformly at random. In the real one, each part of all five queries that T makes to any program is generated by invoking the subroutine Rand and thus is computationally indistinguishable from random numbers. Therefore, we consider three possible conditions. If (U_1', U_2') both are honest in round i, $\text{EVIEW}_{\text{real}}^i \sim \text{EVIEW}_{\text{ideal}}^i$, since the outputs of *NIVBP* are not replaced and $\text{rep}^i = 0$; if one of (U_1', U_2') is dishonest in round i, the fault must be detected by both T and S_1 with a probability close to 1, resulting in an output of "error"; otherwise, the output of *NIVBP* is corrupted with a probability of $1/120(s-1)^2$. In the real experiment, the five outputs generated by (U_1', U_2') are multiplied together along with a random value. Thus, $\text{EVIEW}_{\text{real}}^i \sim \text{EVIEW}_{\text{ideal}}^i$ even when one of (U_1', U_2') misbehaves, so we conclude that $\text{EVIEW}_{\text{real}} \sim \text{EVIEW}_{\text{ideal}}$.

Second, we prove that $\text{UVIEW}_{\text{real}} \sim \text{UVIEW}_{\text{ideal}}$, which means that the untrusted software (U_1', U_2') learns nothing during the execution of $(T, (U_1', U_2'))$. In the ideal experiment, the simulator S_2 always behaves as follows: when receiving the input on round i, S_2 ignores it but submits five random queries of the form (α_j, β_j) to U_1' and U_2'. Then S_2 saves its states and those of (U_1', U_2'). Since the honest, secret or honest, protected or adversarial, protected inputs are all private for (U_1', U_2'), the simulator S_2 is applicable to all those conditions. As shown in Pair One, the inputs to (U_1', U_2') in the real experiment are computationally indistinguishable from those in the ideal one randomly chosen by S_2. Thus, $\text{UVIEW}_{\text{real}}^i \sim \text{UVIEW}_{\text{ideal}}^i$ for each round i, and so $\text{UVIEW}_{\text{real}} \sim \text{UVIEW}_{\text{ideal}}$.

Theorem 9. *In the one-malicious model, the proposed algorithm $(T, (U_1, U_2))$ in Section 3.1 is verifiable; that is, the outsourcer can test the error with a probability close to 1 if one of the servers outputs the false result.*

Proof. Assume that U_1 is an honest server and U_2 is a malicious server. At the end of the algorithm, the outsourcer verifies the results by formulas (4)–(6). It is obvious that U_2 must generate the correct value of α_{21}; otherwise, formula (4) cannot pass the verification with a probability of 1. Thus, the only possibility of U_2 cheating T is returning the false value of $\alpha_{21}, \alpha_{22}, \alpha_{23}, \alpha_{24}, \alpha_{25}$, which is denoted by $\overline{\alpha_{21}}, \overline{\alpha_{22}}, \overline{\alpha_{23}}, \overline{\alpha_{24}}, \overline{\alpha_{25}}$, respectively.

Assume that $\overline{\alpha_{21}}, \overline{\alpha_{22}}, \overline{\alpha_{23}}, \overline{\alpha_{24}}, \overline{\alpha_{25}}$ could pass the verification of formulas (5) and (6); that is,

$$\begin{aligned}
&\left(\alpha_{12} \cdot \overline{\alpha_{22}} \cdot e\left(a_3 P, a_3 Q\right)\right)^{t_1+1} \\
&\quad = \alpha_{14} \cdot \overline{\alpha_{24}} \cdot e\left(a_4 P, b_1 Q + b_2 Q\right)^{t_1+1}, \\
&\left(\alpha_{13} \cdot \overline{\alpha_{23}} \cdot e\left(b_3 P, b_3 Q\right)\right)^{t_2+1} \\
&\quad = \alpha_{15} \cdot \overline{\alpha_{25}} \cdot e\left(a_1 P + a_2 P, b_4 Q\right)^{t_2+1},
\end{aligned} \tag{14}$$

which means that

$$\begin{aligned}
\frac{\overline{\left(\alpha_{22}\right)}^{t_1+1}}{\overline{\alpha_{24}}} &= \frac{\alpha_{14} \cdot e\left(a_4 P, b_1 Q + b_2 Q\right)^{t_1+1}}{{\alpha_{12}}^{t_1+1} \cdot e\left(a_3 P, a_3 Q\right)^{t_1+1}}, \\
\frac{\overline{\left(\alpha_{23}\right)}^{t_1+1}}{\overline{\alpha_{25}}} &= \frac{\alpha_{15} \cdot e\left(a_1 P + a_2 P, b_4 Q\right)^{t_2+1}}{{\alpha_{13}}^{t_2+1} \cdot e\left(b_3 P, b_3 Q\right)^{t_2+1}}.
\end{aligned} \tag{15}$$

Since U_1 is an honest server, $\alpha_{11}, \alpha_{12}, \alpha_{13}, \alpha_{14}, \alpha_{15}$ must be correct. In addition, $e(a_3 P, a_3 Q)$, $e(b_3 P, b_3 Q)$, $e(a_4 P, b_1 Q + b_2 Q)^{t_1+1}$, $e(a_1 P + a_2 P, b_4 Q)^{t_2+1}$ are generated randomly by Rand subroutine, and so these values must be true. Thus, the

Table 1: Comparison of the outsourcing algorithms for bilinear pairing.

Algorithm	Pair [24]	TZR1 [25]	TZR2 [25] ($s = 4$)	VBP [26]	NIVBP ($s = 4$)
PA (T)	5	4	11	8	8
M (T)	4	3	9	11	19
Invoke (Rand)	3	1	2	2	1
Pair (U)	8	6	6	6	10
MExp (U)	0	0	0	4	0
Interactive	No	No	No	Yes	No
Servers	Two	Two	Two	Two	Two
Checkability	0.5	0.5	0.84	1	0.999

values of $\overline{(\alpha_{22})}^{t_1+1}/\overline{\alpha_{24}}$ and $\overline{(\alpha_{23})}^{t_2+1}/\overline{\alpha_{25}}$ should be true even if $\overline{\alpha_{21}}, \overline{\alpha_{22}}, \overline{\alpha_{23}}, \overline{\alpha_{24}}, \overline{\alpha_{25}}$ are incorrect; otherwise, they could not pass the verification of formulas (5) and (6).

In order to obtain the true values of $\overline{(\alpha_{22})}^{t_1+1}/\overline{\alpha_{24}}$ and $\overline{(\alpha_{23})}^{t_2+1}/\overline{\alpha_{25}}$, U_2 must guess the values of $\alpha_{21}, \alpha_{22}, \alpha_{23}, \alpha_{24}, \alpha_{25}$ and t_1, t_2. As shown in Section 3.1, the probabilities of guessing the true values of $\alpha_{21}, \alpha_{22}, \alpha_{23}, \alpha_{24}, \alpha_{25}$ and t_1, t_2 are $1/5!$ and $1/(s-1)^2$, respectively. Therefore, the outsourcer can test the error with a probability of $1 - 1/5!(s-1)^2 = 1 - 1/120(s-1)^2 \approx 1$. □

Theorem 10. *In the one-malicious model, the proposed algorithm $(T, (U_1, U_2))$ is an $(O(s/m), \approx 1)$-outsource-secure implementation of NIVBP, where s is a small positive integer and m is the bit length of q and q is the order of $G, \widehat{G}$.*

Proof. The proposed algorithm *NIVBP* makes one call to Rand and 8 point additions (PA) in G or $\widehat{G}$ and $O(s)$ multiplication in G_T in order to compute $e(A, B)$. As shown in [24], it takes roughly $O(m)$ multiplications in resulting finite field to compute the bilinear pairing, where m is the bit length of q. Thus, the proposed algorithm is an $O(s/m)$-efficient implementation of *NIVBP*. On the other hand, it must be detected with a probability close to 1 if U_1 or U_2 fails during any execution of *NIVBP* from Theorem 9. □

3.3. Comparison. We compare the outsourcing algorithms for bilinear pairing with input privacy in Table 1, where s is a small positive integer and "PA" and "M" denote the operation of point addition in G or $\widehat{G}$ and multiplication in G_T, respectively.

From Table 1, we conclude that the *NIVBP* algorithm increases checkability of the outsourcer, though a little computation cost is appended compared with Pair and TZR1 algorithms. In addition, the *NIVBP* algorithm improves computation efficiency and checkability of the outsourcer simultaneously compared with TZR2 algorithm for the same parameter: $s = 4$. The efficiency and checkability of the *NIVBP* algorithm are nearly the same as those of VBP algorithm, but it decreases the communication cost, since it is noninteractive while the VBP algorithm is interactive. Therefore, the *NIVBP* algorithm increases checkability and decreases communication cost of the outsourcer, although a little computation cost is appended.

4. Applications

In this section, we introduce two applications of the proposed *NIVBP* algorithm: anonymous identity-based encryption (AIBE) scheme [27] and identity-based signature (IBS) scheme [28].

Let $G, \widehat{G}, G_T$ be three cyclic multiplication groups of order q, and let $g, \widehat{g}$ be generators of $G, \widehat{G}$, respectively. $e : G \times \widehat{G} \to G_T$ is a bilinear map. In the following schemes, $G = \widehat{G}$.

4.1. Boyen-Waters AIBE Scheme with Outsourcing Decryption. The proposed outsource-secure AIBE scheme consists of the following algorithms.

Setup. It chooses a random generator $g \in G$, random group elements $g_0, g_1 \in G$, and random exponents $\omega, t_1, t_2, t_3, t_4 \in Z_q$. The master key MSK $= \{\omega, t_1, t_2, t_3, t_4\}$, and the public parameters PK are as follows:

$$\left\{e(g,g)^{t_1 t_2 \omega}, g, g_0, g_1, v_1 = g^{t_1}, v_2 = g^{t_2}, v_3 = g^{t_3}, v_4 = g^{t_4}\right\}. \tag{16}$$

Extract (MSK, ID). To issue a private key for identity ID, it chooses two random exponents $r_1, r_2 \in Z_q$ and computes the private key $\mathrm{SK}_{\mathrm{ID}} = \{d_0, d_1, d_2, d_3, d_4\}$ as follows:

$$\begin{aligned} d_0 &= g^{r_1 t_1 t_2 + r_2 t_3 t_4}, \\ d_1 &= g^{-\omega t_2}\left(g_0 g_1^{\mathrm{ID}}\right)^{-r_1 t_2}, \\ d_2 &= g^{-\omega t_1}\left(g_0 g_1^{\mathrm{ID}}\right)^{-r_1 t_1}, \\ d_3 &= \left(g_0 g_1^{\mathrm{ID}}\right)^{-r_2 t_4}, \\ d_4 &= \left(g_0 g_1^{\mathrm{ID}}\right)^{-r_2 t_3}. \end{aligned} \tag{17}$$

Encrypt (PK, ID, M). To encrypt a message $M \in G_T$ for an identity ID, it chooses random $s, s_1, s_2 \in Z_q$ and creates the ciphertext $\mathrm{CT} = \{C', C_0, C_1, C_2, C_3, C_4\}$ as follows:

$$\left\{M e(g,g)^{t_1 t_2 \omega s}, \left(g_0 g_1^{\mathrm{ID}}\right)^{s}, v_1^{s-s_1}, v_2^{s_1}, v_3^{s-s_2}, v_4^{s_2}\right\}. \tag{18}$$

Decrypt (PK, ID, CT). The outsourcer T executes the *NIVBP* algorithm for five times and obtains

$$e(C_i, d_i) = NIVBP(C_i, d_i), \quad i = 0, 1, 2, 3, 4 \tag{19}$$

and then computes $C' \prod_{i=0}^{4} e(C_i, d_i) = M$.

4.2. Paterson-Schuldt IBS Scheme with Outsourcing Verification. The detailed scheme is shown as follows.

Setup. It picks $\alpha \in Z_q$, $g_2 \in G$, and computes $g_1 = g^{\alpha}$. Further, choose $u', m' \in G$ and vectors $U = (u_i)$, $M = (m_i)$ of length n_u and n_m, respectively, where u_i, m_i are random elements from G. The public parameters are $\mathrm{PK} = \{g, g_1, g_2, u', U, m', M, e(g_2, g_1)\}$ and the master secret key is ${g_2}^{\alpha}$.

Extract. Let u be a bit string of length n_u representing an identity and let $u[i]$ be the i-th bit of u. Set $U \subset \{1, 2, \ldots, n_u\}$ as the set of index i such that $u[i] = 1$. To construct the private key d_u of the identity u, pick $r_u \in Z_q$ and compute

$$d_u = \left({g_2}^{\alpha} \left(u' \prod_{i \in U} u_i \right)^{r_u}, g^{r_u} \right). \tag{20}$$

Sign. Let $M \subset \{1, \ldots, n_m\}$ be the set of index j such that $m[j] = 1$, where m is a message and $m[j]$ is the j-th bit of m. To generate a signature σ for the message m, randomly choose $r_m \in Z_q$ and compute

$$\sigma = \left({g_2}^{\alpha} \left(u' \prod_{i \in U} u_i \right)^{r_u} \left(m' \prod_{j \in M} m_j \right)^{r_m}, g^{r_u}, g^{r_m} \right). \tag{21}$$

Verify. Given a signature $\sigma = (V, R_u, R_m)$ of an identity u for a message m, the outsourcer T executes the *NIVBP* algorithm and obtains

$$\begin{aligned} e(V, g) &= NIVBP(V, g), \\ e\left(u' \prod_{i \in U} u_i, R_u \right) &= NIVBP\left(u' \prod_{i \in U} u_i, R_u \right), \\ e\left(u' \prod_{i \in U} u_i, R_u \right) &= NIVBP\left(m' \prod_{j \in M} m_j, R_m \right). \end{aligned} \tag{22}$$

And verify

$$\begin{aligned} & e(V, g) \\ & = e(g_2, g_1)\, e\left(u' \prod_{i \in U} u_i, R_u \right) e\left(u' \prod_{i \in U} u_i, R_u \right). \end{aligned} \tag{23}$$

It is obvious that the two outsourcing schemes are verifiable and secure, since the *NIVBP* algorithm is verifiable with input privacy as described in Section 3.

5. Performance Evaluation

In this section, we provide an experimental evaluation of the proposed outsourcing algorithms. Our experiment is simulated on two machines with Intel Xeon Processor running at 3.4 GHz with 32 G memory (cloud server) and Intel Celeron Processor running at 1.2 GHz with 2 G memory (the outsourcer), respectively. The programming language is Java, using Java Pairing-Based Cryptography (JPBC) Library. The parameter q is a 160-bit prime that is randomly generated.

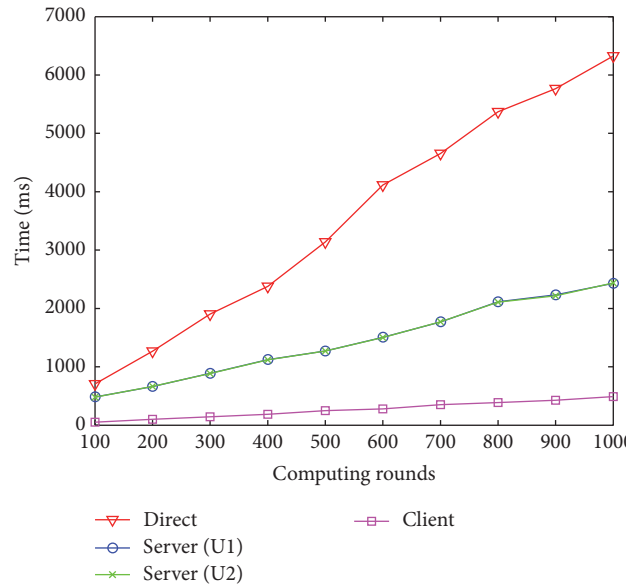


Figure 1: Simulation for the NIVBP algorithm.

In Figure 1, we provide the simulation of *NIVBP* algorithm, which means that the fault can be found with a probability close to 1 if one of the servers misbehaves. It is obvious that the time cost for the outsourcer T is much smaller than that for directly computing bilinear pairing, since a number of computations have been delegated to two servers. Therefore, the proposed *NIVBP* algorithm is the implementation of secure and verifiable outsourcing for bilinear pairing.

In Figure 2, we compare the evaluation times of the outsourcing algorithms for bilinear pairing proposed in [24–26] and this paper, respectively. From Figure 2, we conclude that, for the outsourcer T, the *NIVBP* algorithm is superior to TZR2 algorithm in efficiency, and it appends small computation cost to improve the checkability compared with Pair and TZR1 algorithms. In addition, the *NIVBP* algorithm is nearly the same as VBP algorithm in efficiency, but it is noninteractive and decreases the communication cost of the outsourcer. Thus, the proposed *NIVBP* algorithm improves the checkability and decreases communication cost for the outsourcer simultaneously based on two servers in the one-malicious model.

6. Conclusions

In this paper, we propose a noninteractive verifiable outsource-secure algorithm for bilinear pairing. The security model of our proposed algorithm is based on two noncolluding servers, and the outsourcer can detect any failure with a probability close to 1 if one of the servers misbehaves. Compared with the previous ones, the proposed algorithm improves the checkability and communication efficiency simultaneously for the outsourcer.

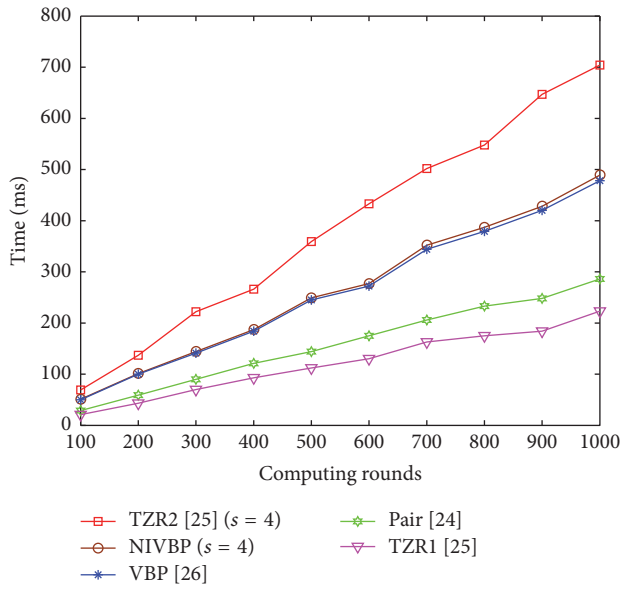


FIGURE 2: Efficiency comparison of the outsourcing algorithms for bilinear pairing.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work described in this paper was supported by the National Natural Science Foundation of China (Grant no. 61572309), Natural Science Foundation of Shanghai (no. 16ZR1411200), and Program for New Century Excellent Talents in University (NCET-12-0620).

References

[1] X. F. Chen, J. Li, J. Ma, Q. Tang, and W. Lou, "New algorithms for secure outsourcing of modular exponentiations," *IEEE Transactions On Parallel And Distributed Systems*, vol. 25, no. 9, pp. 2386–2396, 2014.

[2] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: outsourcing computation to untrusted workers," in *Advances in cryptology—CRYPTO 2010*, vol. 6223 of *Lecture Notes in Comput. Sci.*, pp. 465–482, Springer, Berlin, Germany, 2010.

[3] K.-M. Chung, Y. Kalai, and S. Vadhan, "Improved delegation of computation using fully homomorphic encryption," in *Advances in cryptology—CRYPTO 2010*, vol. 6223 of *Lecture Notes in Comput. Sci.*, pp. 483–501, Springer, Berlin, Germany, 2010.

[4] D. Chaum and T. Pedersen, "Wallet databases with observers," in *Advances in Cryptology—CRYPTO' 92*, vol. 740 of *Lecture Notes in Computer Science*, pp. 89–105, Springer, Berlin, Germany, 1993.

[5] S. Hohenberger and A. Lysyanskaya, "How to securely outsource cryptographic computations," in *Proceedings of the TCC 2005*, vol. 3378 of *Lecture Notes in Computer Science*, pp. 264–282, Springer.

[6] Y. Ren, N. Dingy, X. Zhang, H. Lu, and D. Gu, "Verifiable outsourcing algorithms for modular exponentiations with improved checkability," in *Proceedings of the 11th ACM Asia Conference on Computer and Communications Security, ASIA CCS 2016*, pp. 293–303, ACM, June 2016.

[7] J.-Z. Lai, R. H. Deng, C. Guan, and J. Weng, "Attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 8, pp. 1343–1354, 2013.

[8] B. Qin, R. H. Deng, S. Liu, and S. Ma, "Attribute-based encryption with efficient verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 7, pp. 1384–1393, 2015.

[9] X. Chen, J. Li, X. Huang, J. Li, Y. Xiang, and D. S. Wong, "Secure outsourced attribute-based signatures," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3284–3294, 2014.

[10] J. Yu, K. Ren, and C. Wang, "Enabling cloud storage auditing with verifiable outsourcing of key updates," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1362–1375, 2016.

[11] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *Proceedings of the IEEE INFO-COM*, pp. 525–533, San Diego, Calif, USA, March 2010.

[12] Y. Ren, N. Ding, X. Zhang, H. Lu, and D. Gu, "Identity-based encryption with verifiable outsourced revocation," *Computer Journal*, vol. 59, no. 11, pp. 1659–1668, 2016.

[13] X. Chen, X. Huang, J. Li, J. Ma, W. Lou, and D. S. Wong, "New algorithms for secure outsourcing of large-scale systems of linear equations," *IEEE Transactions on Information and Forensics Security*, vol. 10, no. 1, pp. 69–78, 2015.

[14] Y. Ren, X. Zhang, G. Feng, Z. Qian, and F. Li, "How to Extract Image Features based on Co-occurrence Matrix Securely and Efficiently in Cloud Computing," *IEEE Transactions on Cloud Computing*, 2017.

[15] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Advances in Cryptology—CRYPTO 2001*, vol. 2139 of *Lecture Notes in Computer Science*, pp. 213–229, Springer, Berlin, Germany, 2001.

[16] J. C. Cha and J. H. Cheon, "An identity-based signature from gap Diffie-Hellman groups," in *Proceedings of the PKC*, vol. 2567 of *Lecture Notes in Computer Science*, pp. 18–30, Springer.

[17] A. Joux, "A one round protocol for tripartite Diffie-Hellman," in *Proceedings of the ANTS*, vol. 1838 of *Lecture Notes in Computer Science*, pp. 385–393, Springer, 2000.

[18] M. Scott, N. Costigan, and W. Abdulwahab, "Implementing cryptographic pairings on smartcards," in *Proceedings of the CHES*, vol. 4249 of *LNCS*, pp. 134–147, 2006.

[19] P. S. Barreto, S. D. Galbraith, C. Heigeartaigh, and M. Scott, "Efficient pairing computation on supersingular abelian varieties," *Designs, Codes and Cryptography*, vol. 42, no. 3, pp. 239–271, 2007.

[20] B. Chevallier-Mames, J.-S. Coron, N. McCullagh, D. Naccache, and M. Scott, "Secure delegation of elliptic-curve pairing," in *Proceedings of the CARDIS 2010*, vol. 6035 of *LNCS*, pp. 24–35, 2010.

[21] S. D. Galbraith, K. G. Paterson, and N. . Smart, "Pairings for cryptographers," *Discrete Applied Mathematics*, vol. 156, no. 16, pp. 3113–3121, 2008.

[22] P. Tsang, S. Chow, and S. Smith, "Batch pairing delegation," in *Proceedings of the IWSEC 2007*, 90, 74 pages, 2007.

[23] S. S. M. Chow, M. H. Au, and W. Susilo, "Server-aided signatures verification secure against collusion attack," in *Proceedings of the 6th International Symposium on Information, Computer and Communications Security, ASIACCS 2011*, pp. 401–405, March 2011.

[24] X. Chen, W. Susilo, J. Li et al., "Efficient algorithms for secure outsourcing of bilinear pairings," *Theoretical Computer Science*, vol. 562, pp. 112–121, 2015.

[25] H. Tian, F. Zhang, and K. Ren, "Secure bilinear pairing outsourcing made more efficient and flexible," in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, ASIACCS 2015*, pp. 417–426, April 2015.

[26] Y. Ren, N. Ding, T. Wang, H. Lu, and D. Gu, "New algorithms for verifiable outsourcing of bilinear pairings," *Science China Information Sciences*, vol. 59, no. 9, Article ID 99103, 2016.

[27] X. Boyen and B. Waters, "Anonymous hierarchical identity-based encryption (without random oracles)," in *Advances in cryptology—CRYPTO 2006*, vol. 4117 of *Lecture Notes in Computer Science*, pp. 290–307, Springer, Berlin, Germany, 2006.

[28] K. G. Paterson and J. C. N. Schuldt, "Efficient identity-based signatures secure in the standard model," in *Proceedings of the ACISP 2006*, vol. 4058 of *Lecture Notes in Computer Science*, pp. 207–222, Springer.