Secure High Capacity Data Hiding Scheme based on Reference Matrix

Xiao-Shuang Li[1], Chin-Chen Chang[2], Ming-Xing He[1], and Chia-Chen Lin[3]
(Corresponding author: Chia-Chen Lin)

School of Computer and Software Engineering, Xi Hua University[1]
Department of Information Engineering and Computer Science, Feng Chia University[2]
Department of Computer Science and Information Management, Providence University[3]
Taichung 433, Taiwan
(Email: mhlin3@pu.edu.tw)

(Received July 17, 2018; Revised and Accepted Mar. 14, 2019; First Online Oct. 12, 2019)

Abstract

In this paper, a secure high-capacity data hiding scheme based on a reference matrix is proposed. With the help of the numbering reference matrix and a look-up table, each pixel pair of a cover image can conceal 6 secret bits, which offers 2 extra secret bits than Liu *et al.'s* scheme, while maintaining the average PSNR up to 41.97 dB. Experimental results confirm that our proposed scheme outperforms previous data hiding schemes in visual quality and hiding capacity. Moreover, statistical analysis confirms that the hidden data can be securely protected.

Keywords: Data Hiding; Hiding Capacity; Look Up Table; Reference Matrix

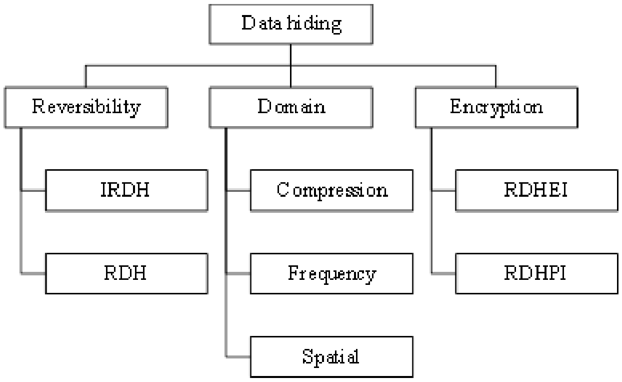


Figure 1: Taxonomy of data hiding

1 Introduction

With the development of information technologies, data hiding has attracted considerable researchers' attention in the field of information security because it can guarantee the security of the transmitted data over the Internet besides adopting the traditional cryptographic approaches, such as RSA [1], El Gamal [2], and DES [3]. The purpose of data hiding is to invisibly embed secret data into a cover medium, which can be audio, images, text, em etc. Similar to camouflage used by animals and insects to blend into the natural environment to protect themselves, the recognition of data that is hidden in cover images can be minimized when data hiding is adopted. To give a clear classification of the existing data hiding schemes, a taxonomy of data hiding is presented in Figure 1.

The first data hiding scheme was proposed by Bender *et al.* [1] in 1996. Over the next twenty years, many data hiding (DH) schemes have been proposed. DH schemes can be classified into three categories according to different criterion. For example, based on reversibility, DH schemes can be classified into reversible data hiding (RDH) [5, 9, 15, 16, 18, 25, 30] and irreversible data hiding (IRDH) [2, 3, 7, 11, 13, 17, 20, 21, 24, 27, 29]. The former reversible schemes are especially designed for military and medical applications and the original cover images can be restored after the hidden secret data is extracted. The latter irreversible schemes are considered conventional DH schemes, and the cover images can not be completely restored even the hidden secret data has been extracted.

The latest category is to apply data hiding to the encryption files, as first proposed in 2011 by Zhang [31]. The main idea of DH in an encryption file is to hide secret data into the encryption files so that data hiding applications can be expanded to areas such as healthcare, which emphasizes the confidentiality of patient information, and cloud applications that need to protect customer data which is stored at cloud service providers' side. Based on the encryption criteria, DH schemes can be classified into reversible data hiding in encrypted images (RDHEI) [5, 18, 25, 30] and reversible data hiding in public images (RDHPI) as shown in Figure 1. The former is to embed secret data into an encryption file and the latter is to embed secret data into a public image, such as a

Hello Kitty image.

Apart from reversibility and encryption criterion, DH schemes can also be classified into three subcategories: compression domain [5, 8, 9, 15], frequency domain [4, 14] and spatial domain [2, 3, 7, 11, 13, 17, 20, 21, 24, 27, 29], according to the domain where secret data is embedded. For the compression domain, a DH scheme can hide secret data into compression codes generated by various compression algorithms, such as BTC [28], VQ [12], SMVQ [10], *etc.* For the frequency domain, the secret data is embedded into the DCT [5, 9, 15] or DWT coefficients [14]. Take Chang *et al.'s* scheme [5] for example, they embedded secret data into the two successive zero DCT coefficients of the medium-frequency components in each block of the cover image.

For the spatial domain, secret data is directly embedded into the pixel value by modifying the pixel value according to the pre-determined hiding strategies [2,3,7,11, 13,17,20,21,24,27,29]. The most famous DH scheme for the spatial domain are LSB-based DH schemes. Among them, the first simple LSB DH scheme was proposed by Chan and Cheng in 2004 [2], which used secret bits to replace the least significant bits of pixels in a cover image. Following Chan and Cheng's idea, many LSB-based DH variants have been proposed. For example, Mielikainen defined a binary function of two cover pixels, which is assigned to a pre-determined value [21]. With their design, a cover pixel pair becomes a unit during data embedding. The LSB of the first pixel carries one secret bit, and a function of the two pixel values also carries another secret bit.

To reduce the distortion caused by data embedding, besides LSB approach various hiding strategies have been proposed. Take Wu and Tsai's scheme for example [27], they used pixel value difference (PVD) to design their embedding strategy. Later, various PVD-based DH schemes were designed [11, 20, 24, 29] to increase hiding capacity while reducing distortion. The latest PVD-based DH scheme was proposed by Mehdi *et al.* [20] in 2017, in which parity-bit PVD is adopted to offer a high payload and good visual quality.

No matter what kind of embedding strategy is designed, there are usually complex computations involved when a DH scheme offers a higher hiding capacity, secure protection of the hidden data, and reduced distortion of cover image. In 2008, Chang *et al.* tried to propose a DH utilizing a Sudoku matrix to embed secret data to meet the above three requirements while reducing the computation cost [3]. Unfortunately, the visual quality of stego-image provided by Chang *et al.'s* scheme was not high. In order to improve the weakness of Chang *et al.'* scheme, Hong *et al.* [13] proposed a novel DH scheme by using a search algorithm. Later, in 2014, another new secret data hiding approach based on a turtle-shell reference matrix was proposed by Chang *et al.* [7] to offer good visual quality and enhance the hiding capacity without a high computation cost. Subsequently, based on the reference matrix and turtle shell concept, Liu *et al.* [17] defined a look-up table to allow a pixel pair to carry one extra bit than Chang *et al.'s* scheme [7].

Inspired by the schemes of Chang *et al.* [7] and Liu *et al.* [17], we aim to enhance both hiding capacity and visual quality while securely protecting the hidden data without a high computation cost. Later, experimental results will prove that in our method each pixel pair can conceal extra 2 secret bits compared to Liu *et al.'s* scheme and the visual quality of our proposed scheme is better than other previous schemes.

The rest of this paper is organized as follows. Section 2 briefly reviews related work. Section 3 explains our proposed scheme. Section 4 provides performance results and gives some discussions. Finally, a brief conclusion is given in Section 5.

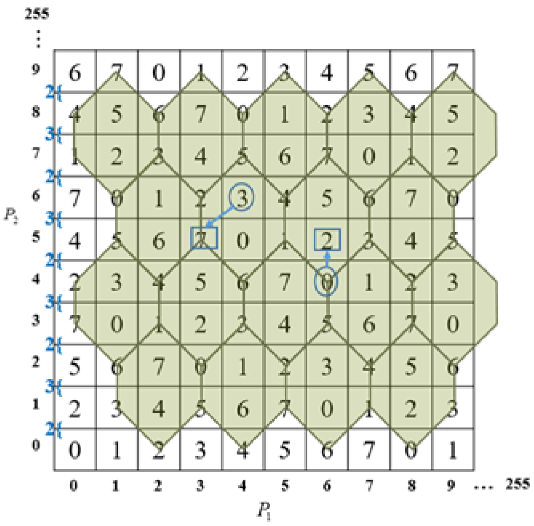
2 Related Work

We review Chang *et al.'s* scheme [7] and Liu *et al.'s* scheme [17] in Subsections 2.1 and 2.2, respectively, to provide insight into how these works specifically inspired our scheme.

2.1 Chang *et al.'s* Turtle Shell Based DH Scheme

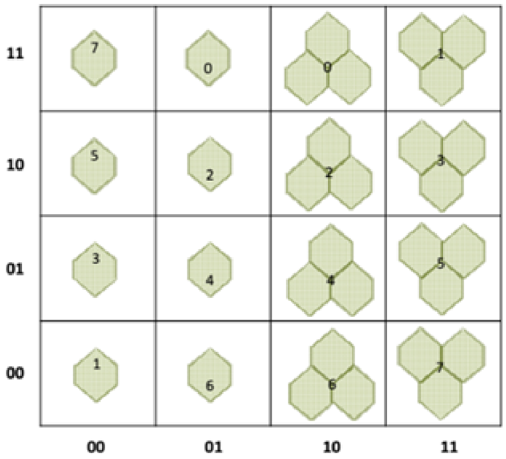
In 2014, a novel turtle-shell-based data hiding scheme was proposed by Chang *et al.* [7]. In Chang *et al.'s* scheme, a reference matrix M sized 256 × 256 digits, as shown in Figure 2, needs to be constructed first before the secret data is embedded into a cover image. Both the X and Y axes of reference matrix M represent the grayscale pixel values of an image and they are ranged from 0 to 255. Reference matrix M is composed of a large number of turtle shells. Each turtle shell is a hexagon shape and includes 6 edge elements and 2 back elements. Therefore, there are 8 different digits ranging from 0 to 7 in a turtle shell. In other words, three secret bits can be carried with a digit of the turtle shell.

Figure 2 illustrates an example of embedding secret data based on a reference matrix M. The location of each cover pixel pair (p_m, p_n) is mapped to (p_m, p_n) in reference matrix M and denoted as $M(p_m, p_n)$, where the p_m is the column value and the p_n is the row value. Assume the cover pixel pair is (4, 6) and the secret data is 7. $M(4, 6)$ belongs to the back element of a turtle shell, and its corresponding digit is 3, which is not equal to secret data 7. Therefore, the cover pixel pair $M(4, 6)$ is changed to $M(3, 5)$ because its corresponding digit is 7. In other words, the stego pixel pair is (3,5) to carry secret data 7. If the cover pixel pair is (6, 4) and the secret data is 2, then $M(6, 4)$ belongs to the edge element and its corresponding digit is 0 which is not equal to secret data 2. Since $M(6, 4)$ is the intersection point of three turtle shells, elements of three neighboring turtle shells need to be explored to find a pixel pair whose corresponding digit is equal to secret data 2. Finally, (6,4) is changed to (6,5)



P_2 \ P_1	0	1	2	3	4	5	6	7	8	9
9	6	7	0	1	2	3	4	5	6	7
8	4	5	6	7	0	1	2	3	4	5
7	1	2	3	4	5	6	7	0	1	2
6	7	0	1	2	3	4	5	6	7	0
5	4	5	6	7	0	1	2	3	4	5
4	2	3	4	5	6	7	0	1	2	3
3	7	0	1	2	3	4	5	6	7	0
2	5	6	7	0	1	2	3	4	5	6
1	2	3	4	5	6	7	0	1	2	3
0	0	1	2	3	4	5	6	7	0	1

Figure 2: Examples of reference matrix M



	00	01	10	11
11	7	0	0	1
10	5	2	2	3
01	3	4	4	5
00	1	6	6	7

Figure 3: Location table T

to carry secret data 2 and the stego pixel pair is set as (6,5). Chang *et al.'s* idea is simple and computation cost of data embedding and data extraction is few.

2.2 Liu *et al.'s* High Capacity Turtle Shell Based DH Scheme

In 2015, Liu *et al.* also proposed a high capacity DH scheme based on turtle shells [17]. In their scheme, the reference matrix M is the same as that defined in Chang *et al.'s* scheme [7]. And reference matrix M and a location table T must be constructed in advance and the secret data stream is divided to non-overlapping 2 bits pieces. Each pixel pair of a cover image can embed 4 bits of secret data with the assistance of reference matrix M and location table T. Location table T shown in Figure 3 guides the modification policy of pixel pairs' values of the cover image during the data embedding phase and plays a crucial role to enhance hiding capacity.

Location table T defines 16 elements of the turtle shells as shown in Figure 3; however, they can be concluded as two different back elements of the turtle shell and two different edge elements. The definition of edge element is the same as that given in Chang *et al.'s* scheme [7], which is at the intersection of three neighboring turtle shells. Each element presented in reference matrix M is only mapped to a specific location defined in location table T. According to the architecture presented in reference matrix M, the values of the elements defined in location table T are always found in a set of values. For example, the values of the front back element defined in location table T are always in the set of values $\{1, 3, 5, 7\}$. Each element defined in location table T is represented by two indicators $(p_j, p_j + 1)$, where p_j and $p_j + 1$ belong to $\{00, 01, 10, 11\}$, p_j is the column value, and $p_j + 1$ is the row value in the location table T. Note that p_j and $p_j + 1$ are two secret patterns.

During data embedding, 2-bit secret pieces are first mapped to location table T to find the specific pattern with a label. For example, (11, 01) is mapped to the edge element pattern with label 5. Once the pattern and its label are found, candidates with the same combination can be found from reference matrix M and be located. After that, the distance between elements which mapped to the original pixel pair and candidate can be calculated using Equation (1). The candidate with the minimal distance is then selected to carry the 2-bit secret pieces and its corresponding axes' values of reference matrix M is the pixel pair of the stego-image.

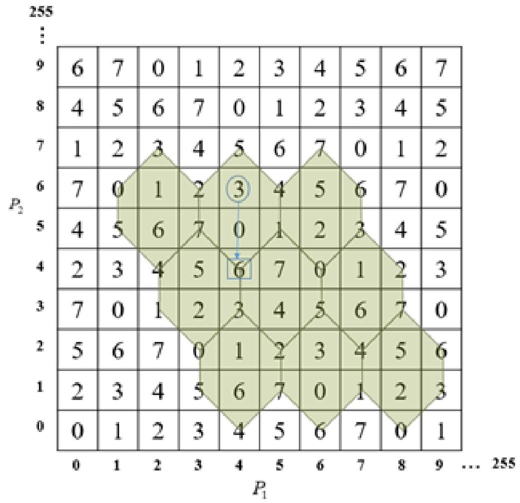
$$d(X, Y) = \sqrt{(X_i - Y_i)^2 + (X_j - Y_j)^2}, \qquad (1)$$

where the (X_i, X_j) is the selected candidate, and the (Y_i, Y_j) is the original cover pixel pair. Afterwards, according to reference matrix M and location table T, the secret data can be successfully extracted.

Assume the original cover pixel pair is (4, 6), and the binary secret data is $(10\ 00)_2$. According to location table T, (10, 00) is mapped to the edge element with label 6. As Figure 4 shows, there are pairs M(4, 4), M(7, 6), and M(9, 2) with the same pattern and same label. Among them, only M(4,4) has the minimal distance with the pixel pair of the cover image; therefore, pixel pair (4,6) of the cover image is changed to (4,4) of the stego-image to carry secret bits (10 00).

2.3 Discussions

The schemes of both Chang *et al.* and Liu *et al.* used the same reference matrix M. In their reference matrix, each turtle shell only covers 8 elements ranging from 0 to 7. Chang *et al.* directly used 8 elements mapped to a turtle shell to carry secret data; therefore, the hiding capacity is limited to 3 secret bits. From Liu *et al.'s* scheme [17], we found they defined a location table T to first specify a pattern with a label, then candidates



9	6	7	0	1	2	3	4	5	6	7
8	4	5	6	7	0	1	2	3	4	5
7	1	2	3	4	5	6	7	0	1	2
6	7	0	1	2	3	4	5	6	7	0
5	4	5	6	7	0	1	2	3	4	5
4	2	3	4	5	6	7	0	1	2	3
3	7	0	1	2	3	4	5	6	7	0
2	5	6	7	0	1	2	3	4	5	6
1	2	3	4	5	6	7	0	1	2	3
0	0	1	2	3	4	5	6	7	0	1
	0	1	2	3	4	5	6	7	8	9

Figure 4: Example of data embedding with Liu *et al.'s* scheme

with the same pattern and label are found from reference matrix M. Only the candidate with the minimal distance from the pixel pair of the cover image can be selected as he pixel pair of the stego-image. With the assistance of location table T, the hiding capacity of a pixel pair is up to 4 bits, which offers one extra bit than Chang *et al.'s* scheme [7]. Thus, location table T can be treated as a new grouping and it allows a pixel pair of cover image to carry one extra secret bit compared with the scheme by Chang *et al.* .

3 The Proposed Secure High Capacity Scheme

Inspired by Chang *et al.* and Liu *et al.*, we found there are two ways to increase hiding capacity: one is to redefine a reference matrix; and the other is to define a new look-up table to offer the similar function as location table T did in Liu *et al.'s* scheme [17]. In our proposed scheme, we first define a turtle shell matrix (TSM), which is an upgraded version of reference matrix M as defined in Section 2.1, and a numbering reference matrix (NRM). Then, we added a look-up table which plays the role of location table T to enhance the hiding capacity. Definitions of TSM and NRM and related discussions are given in Subsections 3.1 and 3.2. Construction of the look-up table is described in Subsection 3.3. The embedding phase and extraction phase are given in Subsections 3.4 and 3.5, respectively.

3.1 Definitions of the Turtle Shell Reference Matrix (TSM) and the Numbering Reference Matrix (NRM)

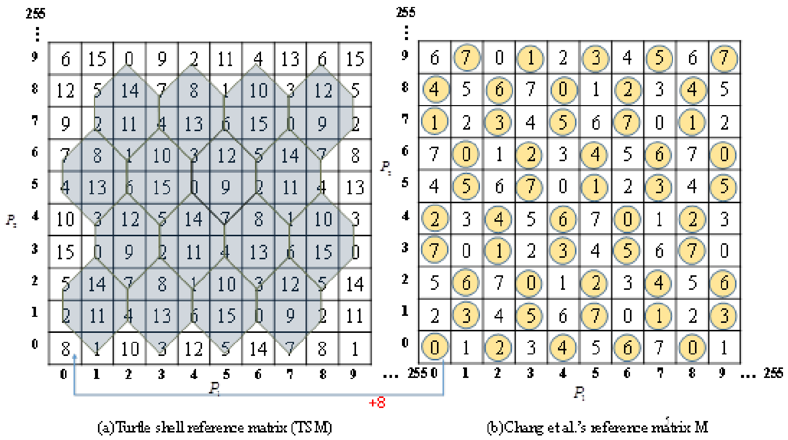
Chang *et al.* scheme [7] defined a turtle shell as a hexagon shape with 8 different digits, which ranges from 0 to 7 as shown in Figure 2. According to their definitions, a turtle shell contains 2 back elements and 6 edge elements. To enhance the hiding capacity, we add 8 to number on Chang *et al.'s* reference matrix based on our pre-determined patterns, which are yellow circles indicated in Figure 5(b). Finally, a new turtle shell reference matrix (TSM) can be found, as shown in Figure 5(a). Comparing with Figures 5(a) and 5(b), it is noted that digits of that three neighboring turtle shells are ranged from 0 to 15 in the TSM rather than 0 to 8, this is because certain elements' values have been added with 8.

Once TSM is constructed, values of X axis and Y axis TSM are relabeling with 0 and 1 to derive a new reference matrix called numbering reference matrix (NRM) as shown in Figure 6. It is noted that NRM is based on TSM; therefore, both are the same size of 256 × 256.

3.2 NRM Numbering Rules

To further increase the hiding capacity of our proposed scheme, a new reference NRM must be generated and relabeled with two digits 0 and 1 as mentioned at the end of Subsection 3.1. However, there are many ways to label the values of the X axis and Y axis, such as, numbering the X axis in the order of (010101. . .) and the Y axis in the order of (11001100. . .) and so on. No matter what kind of numbering order is given, there is a crucial rule must be hold. That is, the numbering results must make sure the search scope is minimized. This is because different number order will lead the different size of search area. If the search area is larger, the distortion between the original pixel pair and the stego pixel pair will be larger, and it will lead to larger distortion of the stego-image. To maintain good visual quality, the numbering order which offers the minimal search area must be found. We experimented with various numbering strategies to find one offering a minimal search scope. For example, numbering the values of X axis in the order of (101010. . .) and numbering the values of Y axis in the order of (101010. . .), give the following numbering results as shown in Figure 7. We find with such a numbering strategy, taking digit 12 for example, only (01) and (11) mapped to digit 12 in the NRM. However, to form a new reference matrix, each digit presented in the NRM must map to four patterns (00), (01), (10), (11) so that each digit can carry 2 secret bits. If a numbering strategy such as numbering the values of X axis in the order of (101010. . .) and numbering the values of Y axis in the order of (101010. . .) is used, we can find that although the search scope has been expanded, digit 12 which maps to (00) and (10) patterns are still missing. As such, this numbering strategy cannot be used for the NRM.

After conducting many experiments, we found a numbering strategy which numbering the values of X axis in the order of (001100. . .) and numbering the values of Y axis in the order of (001100. . .) as shown in Figure 8. It is the best numbering result among all numbering strategies because it covers 4 combinations of 2 bits, and the



(a)Turtle shell reference matrix (TSM)

(b)Chang et al.'s reference matrix M

Figure 5: Turtle shell reference matrix (TSM)

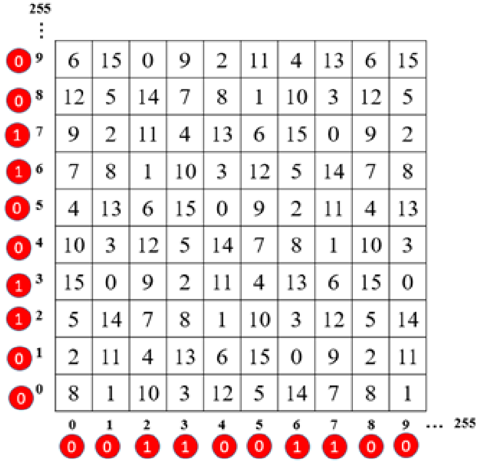


Figure 6: Numbering Reference Matrix (NRM)

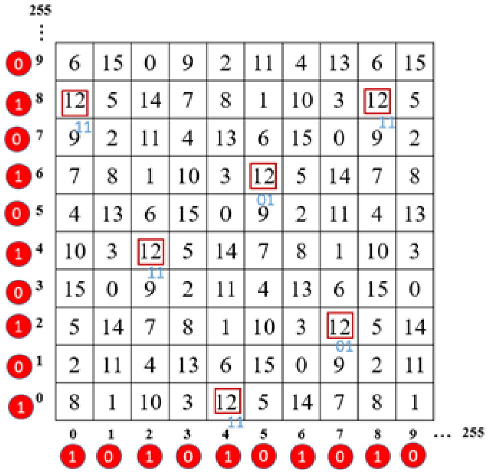


Figure 7: Example of NRM numbering strategy (101010...)

search area is always within the graphic area as shown in Figure 8.

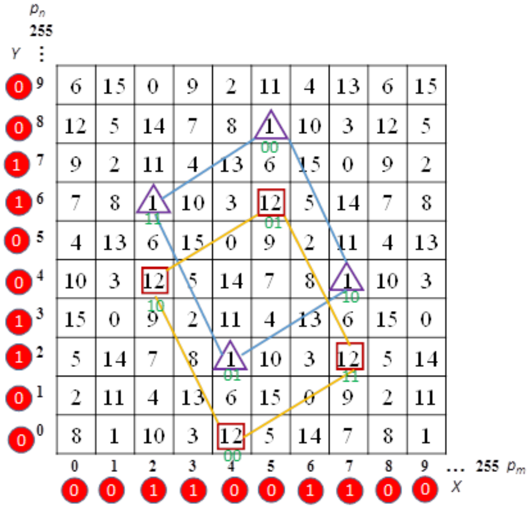


Figure 8: Optimal NRM numbering strategy (001100...)

3.3 Look-up Table Construction

To increase the embedding capacity, we designed a look-up table to carry 6 bits of secret data ranging from (000000) to (111111). As Figure 9 shows, each column contains 16 different digits of LU-values ranging from 0 to 15 and is listed from the bottom to the top. Each digit LU-value is transformed into a binary representation and presents in 4 bits and become the values of the Y axis of the look-up table as shown in Figure 9. For example, the LU-value located at (1,1) in the look-up table is equal to 15 and its binary stream is $(1111)_2$. The values of the X axis and Y axis of the numbering results from the NRM are combined as 2 bits and becomes the values of the X axis of the look-up table as shown in Figure 9.

Since the values of the Y axis of the look-up table are represented as 4 bits and the values of the X axis of look-up table are represented as 2 bits, there are 6 secret bits in total that can be represented by using our defined look-up table as shown in Figure 9. Certainly, multiple candidates can be found by combining our defined look-up table and NRM. To find a candidate which causes the least distortion between the original pixel pair and stego pixel pair, Equation (2) is defined to calculate the distance between the original cover pixel pair (A_m, A_n). and the candidate (B_m, B_n).

$$d(A,B) = \sqrt{(A_m - B_m)^2 + (A_n - B_n)^2}, \qquad (2)$$

where (A_m, A_n) is the cover pixel pair mapping to NRM and (B_m, B_n) is the stego pixel pair mapping to NRM.

3.4 Embedding Phase

As we mentioned in the above subsections, the TSM, NRM and look-up table must be constructed before data embedding. For a grayscale cover image sized $H \times W$ pixels, which is composed by $H \times W$ pixels $P = \{p_m | m = 1, 2, \cdots, (H \times M)\}$, the secret message is divided into non-overlapping 6 bits, where the first 2 bits are mapped to the numbering results of values of the X axis and Y axis of NRM. The last 4 bits are mapped to the Y axis of the look-up table. In order to embed 6 bits secret data into a cover pixel pair (p_m, p_n), cover pixel pair (p_m, p_n) is mapped into the NRM first and denoted as NRM (p_m, p_n), where the p_m is the column decimal value and the p_n is the row decimal value and both p_m and p_n are ranged from 0 to 255. Next, an LU-value of the look-up table can be determined according to the last 4 bits of the Y axis of the look-up table. Find NRM (p_m, p_n) whose corresponding digit is equal to a pre-determined LU-value, where its numbering results are equal to the value of the X axis of the look-up table and the distance between NRM (p'_m, p'_n) and cover pixel pair NRM (p_m, p_n) is minimal. Finally, the a stego pixel pair is determined as (p'_m, p'_n).

In order to explain our proposed embedding phase more clearly, two examples regarding data embedding are demonstrated in Figure 10.

Example 1. *Assume cover pixel pair is (4, 6), the secret data is 9 and its binary representation is (00 1001)2=9. Using the last 4 bits as the indicator, we find the column which maps to (1001) and LU-value, which is 9 in the look-up table. Then, we find digit 9 and its corresponding renumbering results are (0,0) from the NRM. Once multiple candidates are found, the minimal distance between candidate and cover pixel pair (4,6) is applied to determine a candidate which causes less distortion. Here, NRM (5,5) is determined because it is the closest to cover pixel pair (4,6). Finally, NRM (4, 6) is modified to NRM (5,5). In other words, the cover pixel pair (4,6) is changed to stego pixel pair (5,5).*

Example 2. *Assume cover pixel pair is (3, 3), the secret data is 6 and its binary representation is (00 0110)2=6. We use (0110) as the indicator to find the column of the look-up table, which is 6 in the look-up table. From Figure 10, we find NRM (0,9), and NRM (4,1) whose digits are equal to secret data 6, and their numbering results are the same as (0,0). Therefore, the distance between NRM (0,9) and NRT (3,3), and distance between NRM (4,1) and NRM (3,3) is computed, respectively. Finally, NMR (4,1) is selected because it is closer to NRM (3,3) compared with NRM (0,9). The cover pixel pair (3,3) is changed to stego pixel pair (4,1).*

3.5 Extracting Phase

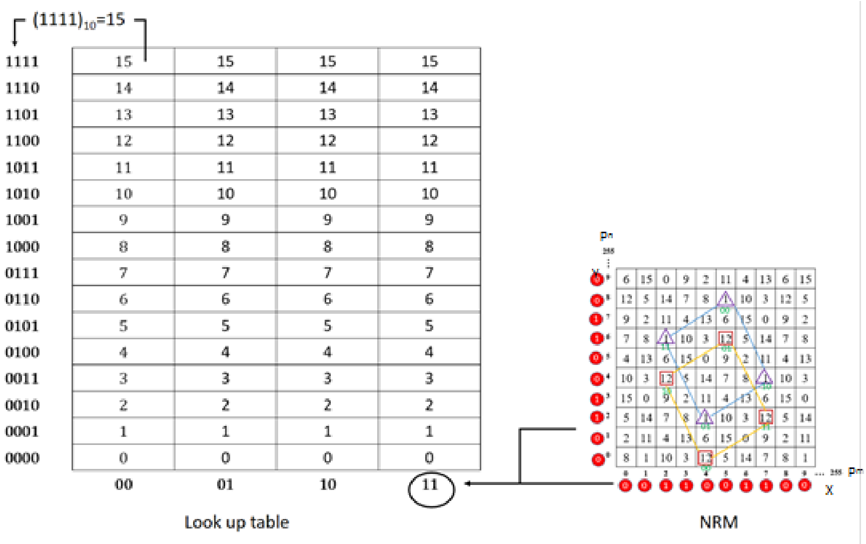
After embedding all of the secret message, the stego-image is generated. Once a receiver obtains the stego-image, the hidden secret data can be extracted with the assistance of NRM. To extract the hidden data, the receiver maps pixel pair of the stego-image into NRM of the NRM. Then, the receiver transforms the mapped digit into a binary representation and these bits are the last 4 bits of the hidden secret bits. According to the numbering results to which NRM maps, the receiver can get the first 2 secret bits. Finally, a secret unit can be derived by combining the above 2 bits and 4 extracted secret bits. The same operations are conducted continuously until all stego pixel pairs are processed and then the secret message can be extracted.

Example 3. *Take stego pixel pair (5, 5) as an example. The stego pixel pair maps to NRM (5,5) so that digit 9 and the numbering result (0,0) can be found. By transforming 9 into binary representation as (1001) and it means that the last 4 secret bits are (1001). As we mentioned, the numbering result (0,0) are the first 2 secret bits. Finally, the secret unit is derived as (00 1001)2 by combining above secret bits. Finally, a secret data 9 can be obtained after transforming (00 1001)2 into the decimal value. Take stego pixel pair (2, 7) for the other example. Stego pixel pair (2,7) maps to NRM (2,7) so that the digit 11 and the numbering result (1, 1) can be found from NRM. By transforming 11 into binary representation as (1011)2 and then combining (11) and (1011) as a new secret unit (11 1011)2. Transforming (11 1011)2 into the decimal value, receiver finally derives secret data as 59.*

4 Experimental Results

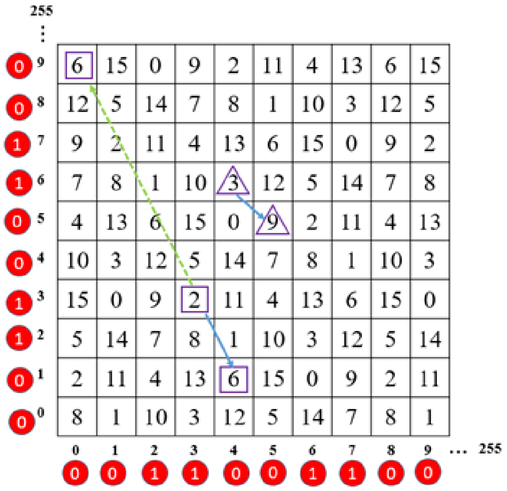
To prove the performance of the proposed scheme, several experiments are conducted. All experiments were implemented in Matlab 2012 on a PC with Intel(R) Core(TM) i7-3770 CPU 3.40 GHz, 8 GB RAM. Figure 11 shows the ten standard grayscale test images sized 512 × 512 that were used in our experiments: Wine, Lena, Harbour, Office, Airplane, Peppers, Baboon, Goldhill, Elaine, and Sailboat.

To estimate the visual quality of the stego-images we used the peak-signal-to-ratio (PSNR) as the measure-



	00	01	10	11
1111	15	15	15	15
1110	14	14	14	14
1101	13	13	13	13
1100	12	12	12	12
1011	11	11	11	11
1010	10	10	10	10
1001	9	9	9	9
1000	8	8	8	8
0111	7	7	7	7
0110	6	6	6	6
0101	5	5	5	5
0100	4	4	4	4
0011	3	3	3	3
0010	2	2	2	2
0001	1	1	1	1
0000	0	0	0	0

Figure 9: Look-up table



		0	1	2	3	4	5	6	7	8	9
		0	0	1	1	0	0	1	1	0	0
0	9	6	15	0	9	2	11	4	13	6	15
0	8	12	5	14	7	8	1	10	3	12	5
1	7	9	2	11	4	13	6	15	0	9	2
1	6	7	8	1	10	3	12	5	14	7	8
0	5	4	13	6	15	0	9	2	11	4	13
0	4	10	3	12	5	14	7	8	1	10	3
1	3	15	0	9	2	11	4	13	6	15	0
1	2	5	14	7	8	1	10	3	12	5	14
0	1	2	11	4	13	6	15	0	9	2	11
0	0	8	1	10	3	12	5	14	7	8	1

Figure 10: Example of our proposed embedding phase

ment. The definition for PSNR is given in Equation (3).

$$PSNR = 10\log_{10}(\frac{255^2}{MSE})(db), \tag{3}$$

where the mean square error (MSE) is between the original cover image and the stego-image, where for a grayscale cover image $H \times W$ pixels is defined as Equation (4):

$$MSE = \frac{1}{H \times M}\sum_{i=1}^{H\times M}(p_i - p_j)^2, \tag{4}$$

where p_i is the pixel value of the original cover image, and p_j is the pixel value of the corresponding stego-image. The higher the PSNR, the better the image quality. In general, if the PSNR is higher than 30 dB, it is difficult for the human eye to recognize any difference between the original cover image and stego-image. The stego-images generated with our proposed scheme carrying 786,432 secret bits are shown in Figure 12. The secret bits used in our experiments are randomly generated bitstream.

In addition to the PSNR, the Structural Similarity Index Metric (SSIM) is measured the degradation in the quality, which is based on structural information. And the value of SSIM is between -1 to 1. The value of 1 means the two images are identically the same. SSIM between the original image I and the corresponding stego-image C is defined as Equation (5):

$$SSIM(I,C) = \frac{(2u_I u_C + c_1)(2\sigma_{IC} + c_2)}{(u_I^2 + u_C^2 + c_1)(\sigma_I^2 + \sigma_C^2 + c_2)}, \tag{5}$$

where u_I, u_C, σ_I^2, σ_C^2 are the averages and variances of I and C respectively, σ_{IC} is the covariance between I and C, c_1 and c_2 are as follows:

$$c_1 = (k_1L)^2; \quad \text{where } k_1 \ll 1 \text{ (small constant)}, \tag{6}$$
$$c_2 = (k_2L)^2; \quad \text{where } k_2 \ll 1 \text{ (small constant)}, \tag{7}$$

where L is defined as the dynamic range of the pixel values.

In addition, we also calculate the Normal Cross Correlation (NCC) between the original image I and the corresponding stego-image C as defined in as Equation (8):

$$NCC = \frac{\sum_i\sum_j I_{ij}C_{ij}}{\sum_i\sum_j (I_{ij})^2} \tag{8}$$

where the I_{ij} and C_{ij} are the original and stego-image bits at $(i,j)^{th}$ position. When the value of NCC is 1, it indicates two images are identically the same, and vice versa.

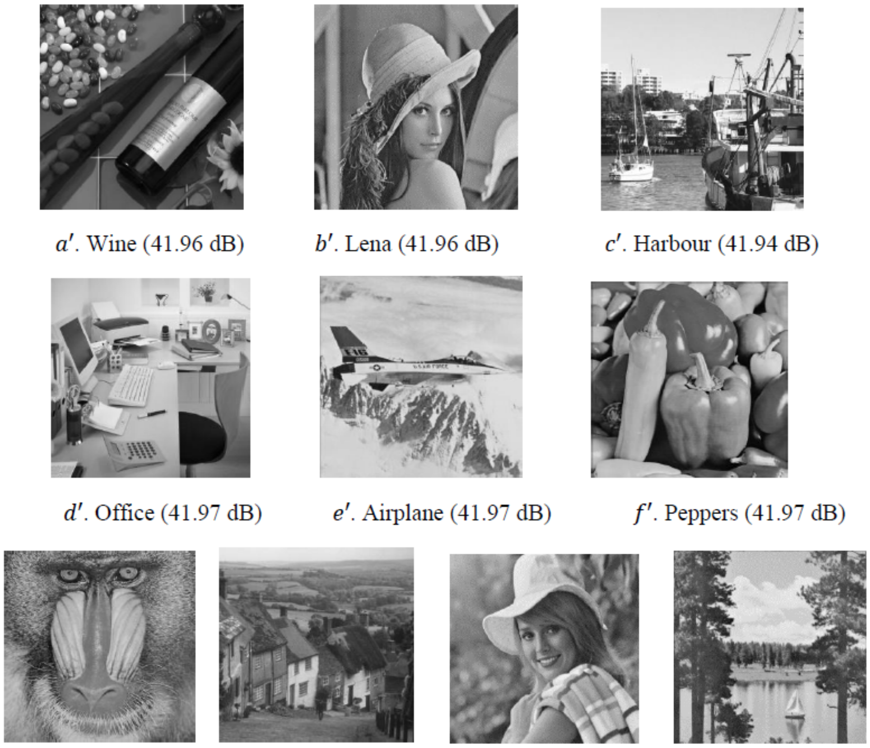
Figure 11 shows ten original cover images $a, b, c, d, e, f, g, h, i, j$, and Figure 12 shows the corresponding stego-images $a', b', c', d', e', f', g', h', i', j'$. Even though the

a. Wine b. Lena c. Harbour

d. Office e. Airplane f. Peppers

g. Baboon h. Goldhill i. Elaine j. Sailboat

Figure 11: Ten 512 × 512 graysacle test images



a'. Wine (41.96 dB) b'. Lena (41.96 dB) c'. Harbour (41.94 dB)

d'. Office (41.97 dB) e'. Airplane (41.97 dB) f'. Peppers (41.97 dB)

g'. Baboon (41.95 dB) h'. Goldhill (41.96 dB) i'. Elaine (41.96 dB) j'. Sailboat (41.97 dB)

Figure 12: The stego-images corresponding the ten test grascale images

amount of the secret message is up to 786,432 bits, the average visual quality of the stego-images is higher than 41 dB, and the average of the SSIM is 0.9342, and the least of the NCC is 0.9991 (See Table 1).

To better demonstrate the advantages of our proposed scheme, we also compared our scheme with previous schemes, such as those by Yang *et al.* [29], Liu *et al.* [17] and Shen and Huang [24] and Mehdi *et al.* [20]. The comparison results are listed in Table 2. Obviously, the maximum embedding capacity of our proposed scheme is much higher than the other schemes, and moreover the visual quality is higher than that of Yang *et al.'s* [29], Shen and Huang's [24] and Mehdi *et al.'s* [20]. Especially, the average embedding capacity of our proposed scheme surpasses the 356,209 bits of the Yang *et al.'s* scheme, exceeds the embedding capacity of the scheme of Shen and Huang by 372,943 bits, and also exceeds the 551,491 bits of the Mehdi *et al.'s* scheme. Although Liu *et al.'s* average PSNR is higher by 3.59 dB than our scheme, the hiding capacity is still 524,288 bits, which is 262,144 fewer secret bits compared to our scheme.

To demonstrate the visual quality performance of our proposed scheme, the third experiment was conducted and the comparisons among our proposed scheme and three previous schemes which claimed they can offer better image quality of setgo image or provide high hiding capacity [11, 17, 20] are shown in Table 3. Here, all schemes carried the similar amount of secret data to derive the PSNR values. Table 3 shows that the visual quality of our proposed scheme is better than the other three previous schemes. Note that the average PSNR of Liu *et al.'s* scheme is 45.55 dB, which is lower than that of our scheme 46.82 dB when the hiding capacity is set as 524,288 bits. By combining Tables 2 and 3, it is confirmed that our proposed scheme has a higher hiding capacity and offers better visual quality in the stego-image with the same hiding capacity.

To further prove the safety of our proposed scheme, we examined the pixel value difference (PVD) histograms of the original cover images and corresponding stego-images, where both are at their maximum embedding capacity as shown in Figure 13. The PVD histogram is calculated by computing the difference in the neighboring pixels between the original cover image and the stego-image. The smaller the gap between the two curves, the smaller the image changes, which confirms that the stego-image is more secure. Using the test images 'Baboon' and 'Peppers' for example, we show their PVD histograms after completely embedding secret data. As Figure 13 shows, the gap between two curves is small for the two test images. This confirms that our proposed scheme offers a relatively high visual quality and also guarantees the security of the hidden data.

To prove the computation cost is still low even the distance between the original cover pixel pair and multiple candidates pixel pairs need to be computed to reduce the potential distortion caused during data embedding. The computation time of data embedding and data extracting phases are listed in Table 4. From Table 4, we can see the proposed scheme is quite efficient and suitable for real-time applications.

5 Conclusions

In this paper, a novel data hiding scheme based on reference matrix and look-up table is proposed. The use of a NRM and look-up table not only allows 6 secret bits to be conceled in a pixel pair of the cover image, but also successfully reduces the caused distortion during data embedding. During extracting phase, only NRM is required; therfore, the extraction phase is also quite efficient. Lastly, the experimental results confirmed that our proposed scheme offers higher embedding capacity than other existing schemes while maintaining good visual quality and guaranteeing the security of the hidden data.

Acknowledgments

This research is funded by: (1) National Natural Science Foundation of China (nos.U143310218); (2) Chunhui Project of Education Ministry of China (nos.Z2014045); (3) Science and Education and Technology Bureau Project of Cheng-du Municipality (nos.2016-XT00-00015-GX); (4) Graduate Innovation Fund Project of Xi Hua University (nos.ycjj2018003).

References

[1] W. Bender, D. Gruhl, N. Morimoto, "Techniques for data hiding," *IBM Systems Journal*, vol. 35, no. 3, pp. 313–336, 1996.

[2] C. K. Chan, L. M. Cheng, "Hiding data in images by simple LSB substitution," *Pattern Recognition*, vol. 37, pp. 469–474, 2004.

[3] C. C. Chang , Y. C. Cho, T. D. Kieu, "An information hiding scheme using sudoku," in *Proceedings of Third International Conference of Innovative Computing, Information and Control*, pp. 17–22, 2008.

[4] C. C. Chang, T. D. Kieu, Y. C. Chou, "A lossless data embedding technique by joint neighboring coding," *Pattern Recognition*, vol. 42, no. 7, pp. 1597–1603, 2009.

[5] C. C. Chang, C. C. Lin, C. S. Tseng, W. L. Tai. "Reversible hiding in DCT-based compressed images," *Information Sciences*, vol. 177, no. 13, pp. 2768–2786, 2007.

[6] C. C. Chang, P. Y. Lin, Z. H. Wang, and M. C. Li, "A sudoku-based secret image sharing scheme with reversibility," *Journal of Communications*, vol. 5, no. 1, pp. 5–12, 2010.

[7] C. C. Chang, Y. J. Liu, T. S. Nguyen, "A novel turtle shell based scheme for data hiding," in *Proceedings*

Table 1: Experimental results of the proposed scheme

Images	EC (bits)	PSNR (dB)	BER	SSIM	NCC
Wine	786,432	41.96	0	0.9221	0.9989
Lena	786,432	41.96	0	0.9210	0.9991
Harbour	786,432	41.94	0	0.9258	0.9993
Office	786,432	41.97	0	0.9125	0.9993
Airplane	786,432	41.97	0	0.9206	0.9994
Peppers	786,432	41.97	0	0.9334	0.9991
Baboon	786,432	41.95	0	0.9657	0.9991
Goldhill	786,432	41.96	0	0.9531	0.9991
Elaine	786,432	41.96	0	0.9550	0.9992
Sailboat	786,432	41.97	0	0.9327	0.9992
Average	786,432	41.96	0	0.9342	0.9992

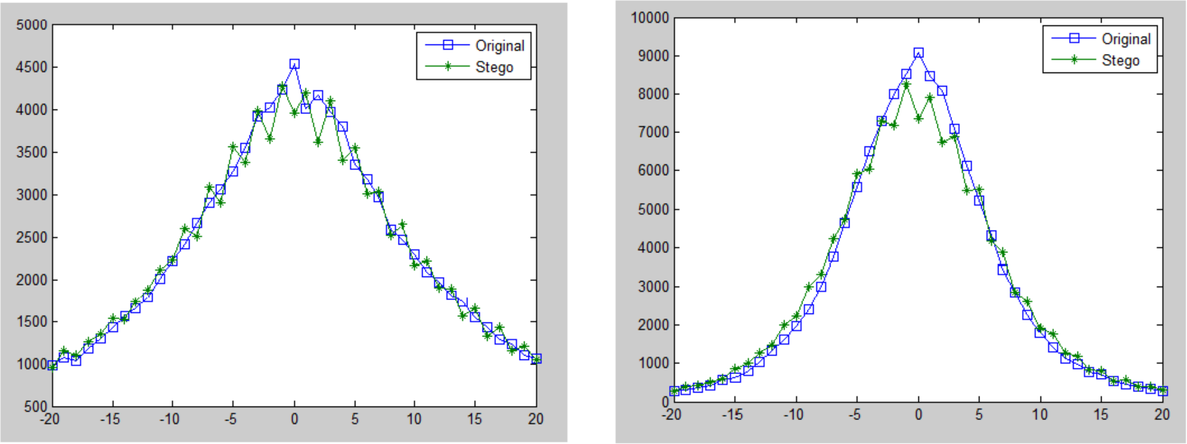
Note: EC= hiding capacity

Table 2: Comparisons the maximum EC and PSNR of proposed scheme with four existing schemes

	[29]		**[20]**		**[17]**		**[24]**		**Proposed scheme**	
Images	EC	PSNR	EC	PSNR	EC	PSNR	EC	PSNR	EC	PSNR
Baboon	482,515	34.67	540,850	40.66	524,288	45.55	443,472	38.88	786,432	41.95
Peppers	408,281	40.47	587,971	41.14	524,288	45.54	404,226	41.25	786,432	41.97
Goldhill	418,575	40.25	536,210	41.10	524,288	45.58	405,956	41.81	786,432	41.96
Sailboat	430,888	38.11	539,652	41.45	524,288	45.54	411,306	41.29	786,432	41.97
Lena	410,854	40.54	552,773	39.44	524,288	45.55	402,485	42.46	786,432	41.96
Average	430,223	38.81	551,491	40.76	524,288	45.55	413,489	40.94	786,432	41.96

Table 3: PSNR comparison of proposed scheme with three schemes

	[17]		**[20]**		**[11]**		**Proposed scheme**	
Images	EC	PSNR	EC	PSNR	EC	PSNR	EC	PSNR
Lena	524,288	45.55	540,850	40.66	512,384	43.01	524,288	46.83
Baboon	524,288	45.55	587,971	41.14	512,516	43.58	524,288	46.82
Peppers	524,288	45.54	536,210	41.10	512,392	43.62	524,288	46.82
Elaine	524,288	45.54	539,652	41.45	493,520	43.41	524,288	46.80
Sailboat	524,288	45.55	552,773	39.44	524,508	42.86	524,288	46.83
Average	524,288	45.55	551,491	40.76	511,064	43.29	524,288	46.82



a. Baboon

b. Peppers

Figure 13: PVD histogram of the original cover image and corresponding stego-image

Table 4: Execution time (s) of data embedding and data extracting phase

Images	Embedding Time (s)	Extraction Time (s)
Wine	1.0213	1.0123
Lena	1.0345	1.0325
Harbour	1.0246	1.0432
Office	1.0436	1.0256
Airplane	1.0325	1.0364
Peppers	1.0245	1.0532
Baboon	1.0325	1.0267
Goldhill	1.0248	1.0365
Elaine	1.0356	1.0245
Sailboat	1.0267	1.0542
Average	1.0301	1.0345

of Tenth International Conference of Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP'14), pp. 89–93, 2014.

[8] C. C. Chang, T. C. Lu, "A difference expansion oriented data hiding scheme for restoring the original host image," *Journal of Systems and Software*, vol. 79, no. 12, pp. 1754–1766, 2006.

[9] C. C. Chang, R. Tang, C. C. Lin, W. L. Lyu, "High-capacity reversible data Hiding method for JPEG images," *Journal of Software*, vol. 13, no. 1, pp. 1–17, 2018.

[10] C. C. Chen and C. C. Chang, "High capacity SMVQ-based hiding scheme using adaptive index," *Signal Processing*, vol. 90, no. 7, pp. 2141–2149, 2010.

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

[12] H. M. Feng and J. H. Horng, "VQ-based fuzzy compression systems designs through bacterial foraging particle swarm optimization algorithm," in *Proceedings of the 5th International Conference on Genetic and Evolutionary Computing*, pp. 256–259, 2011.

[13] W. Hong, T. S. Chen, C. W. Shiu, "A minimal Euclidean distance searching technique for Sudoku steganography," in *Proceedings of International Symposium of Information Science and Engineering*, pp. 515–518, Dec. 2008.

[14] S. Lee, C. D. Yoo, 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.

[15] Y. K. Lin, "A data hiding scheme based upon DCT coefficient modification," *Computer Standards & Interfaces*, vol. 36, no. 5, pp. 855–862, Sept. 2014.

[16] Y. J. Liu, C. C. Chang, "A turtle shell-based visual secret sharing scheme with reversibility and authentication," *Multimedia Tools and Applications*, vol. 77, no. 19, pp. 1–16, 2018.

[17] Y. J. Liu, C. C. Chang, T. S. Nguye, "High capacity turtle shell-based data hiding," *IET Image Processing*, vol. 10, no. 2, pp. 130–137, 2015.

[18] 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 Security*, vol. 8, no. 3, pp. 553–562, Mar. 2013.

[19] M. Matsui, "Linear cryptanalysis method for DES cipher," in *Advances in Cryptology (Eurocrypt'93)*, LNCS 765, pp. 386–397, Springer, 1993.

[20] H. Mehdi., W. A. W. Ainuddin, T. S. Anthony Ho, J. Noman, "A data hiding scheme using parity-bit pixel value differencing and improved rightmost digit replacement," *Signal Processing Image Communication*, vol. 50, pp. 44–57, 2017.

[21] J. Mielikainen, "LSB matching revisited," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 285–287, May 2006.

[22] C. Qin, C. C. Chang, and T. J. Hsu, "Reversible data hiding scheme based on exploiting modification direction with two steganographic images," *Multimedia Tools and Applications*, vol. 74, no. 15, pp. 5861–5872, 2015.

[23] R. L. Rivest, A. Shamir and L. Adleman, "A method of obtaining digital signatures and public key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

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

[25] A. Smita, K. Manoj, "Mean value based reversible data hiding in encrypted images," *Optik*, vol. 130, pp. 922–934, 2017.

[26] Y. Tsiounis, M. Yung, "On the security of El Gamal based encryption," in *International Workshop on Public Key Cryptography*, LNCS 1431, pp. 117–134, Springer, 1998.

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

[28] Y. G. Wu, S. C. Tai, "An efficient BTC image compression technique," *IEEE Transactions on Consumer Electronics*, vol. 44, no. 2, pp. 317–325, 1998.

[29] H. Yang, C. Y. Weng, H. K. Tso, *et al.*, "A data hiding scheme using the variet ies of pixel-value differencing in multimedia images," *Journal of Systems and Software*, vol. 84, no. 4, pp. 669–678, 2011.

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

[31] X. P. Zhang, D. Schonberg, and K. Ramchandran, "Reversible data hiding in encrypted image," *IEEE Signal Processing Letters*, vol. 18, no. 14, pp. 255–258, Feb. 2011.

Xiao-Shuang Li received her BS and MS degree in applied Information and Computing Science in 2016 and 2019 respectively from Xihua University, Chengdu, Sichuan, China. Her current research interests include data hiding, modern cryptographic algorithms, cloud computing security technologies and visual cryptography.

Chin-Chen Chang received his BS degree in applied mathematics in 1977 and the MS degree in computer and decision sciences in 1979, both from the National Tsing Hua University, Hsinchu, Taiwan. He received his Ph.D in computer engineering in 1982 from the National Chiao Tung University, Hsinchu, Taiwan. Since February 2005, he has been a Chair Professor of Feng Chia University. In addition, he has served as a consultant to several research institutes and government departments. His current research interests include database design, computer cryptography, image compression and data structures.

Ming-Xing He received the M.Sc Degree from Chongqing University and the Ph.D from Southwest Jiaotong University in 1990, 2003 respectively. He is a full professor of the School of Computer and Software Engineering, Xihua University, Chengdu, P.R.China. His current research interests include cryptography and information security. He has co-authored five books and has published over 100 papers in refereed professional journals and international conferences. He received the DAAD scholarship reward of Germany in 2002, the Excellent Ph.D. Dissertation Award in Southwest Jiaotong University in 2003, and the grant of National Science Foundation of China (NSFC) in 2004, 2007 and 2015. He is a Senior Member of CACR and member of the ACM.

Chia-Chen Lin (also known as Min-Hui Lin) received her Ph.D degree in information management in 1998 from the National Chiao Tung University. Dr. Lin is currently a professor of the Department of Computer Science and Information Management, Providence University. Since 2018, she is the Fellow of IET. In additions, she serves Associate Editor and Editor for several representative EI, SCIE journals. Her research interests include image and signal processing, information hiding, mobile agent, and electronic commerce.