EUROGRAPHICS Workshop on Multimedia (2004)
N. Correia, J. Jorge, T. Chambel, Z. Pan (Editors)

A Chinese Remainder Theorem Oriented Information Hiding Scheme

Chin-Chen Chang and Tzu-Chuen Lu

Department of Computer Science and Information Engineering,
National Chung Cheng University, Chiayi, Taiwan, R.O.C.

Abstract

Steganography is an information hiding technique that conveys secret information in a host signal using a secret method. Only the receivers and senders know the secret information. Many researchers have proposed their own steganographic techniques to hide information in various host signals, such as audios, videos, images, and so on. Nevertheless, most of the methods degrade the visual quality of the image when more information is hidden in the image. Therefore, this paper proposes a new steganographic scheme, which is based on the Chinese Remainder Theorem. The abbreviation of the scheme is CRTIH, and it not only conceals a larger amount of information in a hidden image but also upgrades the visual quality of the image.

Categories and Subject Descriptors (according to ACM CCS): I.5.4 [Applications]: Signal processing

1. Introduction

Recently, information hiding has become important in a number of application areas, such as digital images, audios, video protection, military communications, law investigation, digital elections, electronic cash, and so on [FAK99], [Sti95].

The goals of information hiding are to embed messages in other data and to send the message secretly. Generally speaking, classifications of information hiding include covert channels, steganography, anonymity, and copyright watermarking [Bau97]. Copyright marking or watermarking is a technique that embeds a copyright notice or watermark inside an image to protect intellectual property against possible attacks. For example, Jo and Kim proposed an image watermarking scheme based on vector quantization (VQ) to improve the degree for spreading watermarking information [JK02]. Cox et al. applied DCT to transform a host image and hide the watermark in perceptually significant areas, such as the 1000 largest DCT coefficients of the image [CKL97]. Delaigle et al. used the human visual system to improve the performance of watermarking [DVM98].

Steganography is a technique for hiding information in digital media. The hidden information is imperceptible. In addition, the purpose of steganography is to have covert communication between two parties whose existence is unknown to a possible attacker. The most important difference between watermarking and steganography is that steganography does not focus on the difficulty of removing secret data through image operations but rather focuses on the capacity for embedding. Many researchers have proposed their own steganographic techniques to hide information in various host signals, such as images, audios, videos, and so on. Nevertheless, most of the methods degrade the visual quality of the hidden signals in which more information is hidden. Therefore, this paper focuses on proposing a new steganographic scheme, which can preserve a very high visual quality of hidden images.

2. Related Works

Steganographic techniques have been studied for years, and different methods have been developed. For example, Koch and Zhao proposed an algorithm to embed a bit string in an image that provides a high-capacity channel for hiding information [KZ95]. Huang and Shi proposed an algorithm based on the DCT to

embed information bits in the DC and low frequency AC coefficients [HS02]. Amin described several steganographic techniques and developed one of the techniques to hide information [ASI03]. Tseng and Pan proposed an improved steganographic scheme for hiding a piece of critical information in a host binary image. The scheme of offers a good information-hiding ratio and ensures that for any bit that is modified in the host image, the bit is adjacent to another bit [TP02]. Chang et al. used the dynamic programming strategy to search an approximate optimal least significant bit (LSB) to embed the secret image in the host image [CHC03]. Tian employed difference expansion and generalized least significant bit embedding to embed data in digital images [Tia03]. Swanson et al. proposed two methods to embed information in an image. The first method employed spatial masking and data spreading to hide data by modified image coefficients. The second method applied frequency masking to modify image spectral components [SZT96].

One of the hiding methods uses VQ to embed secret information in the host image to form the hidden image [JK02], [LS00]. For example, Jo and Kim used VQ to compress the image and hide information in the compressed image. They partitioned the codewords of the codebook into three groups, G_{-1}, G_0, and G_1, and each block in a host image embeds one bit. Any codeword of G_0 or G_1 means that it embeds 0 or 1, while a G_{-1} codeword signifies that it cannot embed information [JK02]. Lu and Sum proposed a nonblind image watermarking method that carries watermark information codevector indices [LS00].

Since VQ is a lossy compression technique, the information hiding methods that are based on VQ will degrade the visual quality of the hidden images. This paper applies the Chinese Remainder Theorem (CRT) to correct the difference between the host image and the hidden image. CRT is one of the oldest theorems in theory of numbers and has been used in many areas, such as computing, coding, cryptography, and so on. Using CRT, we can find the relationship between the difference of the host image and hidden image and further upgrade the quality of the hidden image.

This paper shall propose a new steganographic scheme based on CRT. The scheme is called Chinese Remainder Theorem Oriented Information Hiding Scheme (CRTIH). The scheme not only hides greater amounts of information in an image, but it also does not influence the visual quality of the image.

3. The proposed method

The framework for our proposed method is shown in Fig. 1. Suppose a sender wants to share some information with a receiver, and only the sender and the receiver own the same host image and codebook. The sender hides the information in a host image using an encoder and delivers the hidden image to the receiver or publishes the hidden image on a Web site. When the receiver receives or downloads the hidden image, a decoder is used to decode the hidden image and to extract the information.

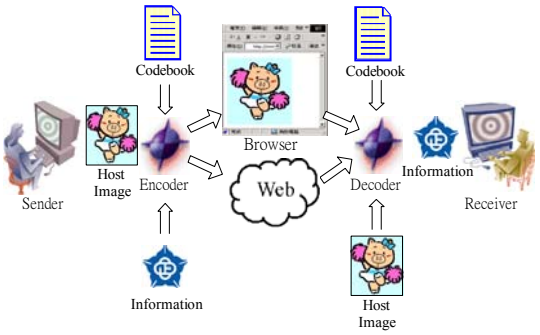

Figure 1: *The framework of CRTIH*

3.1 Encoder

In this subsection, we shall describe how to hide an information string in an image using an encoder. The diagram of the encoder in the framework is shown in Fig. 2.

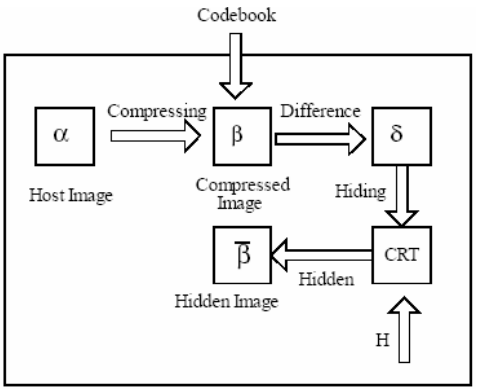

Figure 2: *The diagram of the encoder*

The symbol $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_{n*n})$ in Fig. 2 represents the host image with n*n pixels,

where $\alpha_i \in [0, 255]$ is the i^{th} pixel value. An example image with 4*4 pixels is shown in Fig. , where $\alpha = \{10, 12, 7, 9, 13, \ldots, 78\}$.

10	12	7	9
13	4	15	9
10	37	22	55
61	20	15	78

Figure 3: *The host image* α

Based on VQ compression, the host image α is compressed into a small size index table. The host image α is divided into several blocks, and each block searches the minimum distortion codeword from the codebook. For example, the host image α is divided into 2*2 blocks by the block with 2*2 pixels. Each block in α finds the closest corresponding codeword from the codebook, which is shown in Fig. 4. The index table of Fig. is shown in Fig. 5.

The symbol $\beta = (\beta_1, \beta_2, \ldots, \beta_{n*n})$ in Fig. 2 represents the compressed image, which is reconstructed from the index table. Each index in the index table is replaced by the codewords by a simple table-lookup operation. For example, the reconstructed image of the index table in Fig. 5 is shown in Fig. 6, where $\beta = \{9, 6, 9, 6, 9, \ldots, 75\}$.

Index	Codewords			
1	3	2	60	18
2	79	28	11	34
3	4	11	10	2
4	66	23	7	16
5	88	12	20	18
6	3	22	15	20
7	9	6	9	9
8	7	7	2	3
9	17	39	50	19
10	25	50	12	75

Figure 4: *An example codebook*

7	7
9	10

Figure 5: *The corresponding index table of Fig. 3*

9	6	9	6
9	9	9	9
17	39	25	50
50	19	12	75

Figure 6: *The compressed image, called* β *, of Fig. 5*

A secret information string is then hidden in the compressed image β. Let the information string be denoted by H = "$c_1\ c_2 \ldots c_m$," and $\|H\|$ is its length. H is obtained by encrypting a plaintext $M = m_1\ m_2 \ldots m_m$ using a DES-like method associated with the private key, where $c_j, m_j \in [0, 1]$ and $1 \le j \le m$. For example, H = "1110111000011001011," where $\|H\| = 19$. For each pixel in β, the hiding process based on CRT can be summarized as follows:

Step 1: Compute the difference between α and β.

Let $\delta = \{\delta_1, \delta_2, \ldots, \delta_{n*n}\}$

$$= \{\alpha_1 - \beta_1, \alpha_2 - \beta_2, \ldots, \alpha_{n*n} - \beta_{n*n}\}$$

be the differences between α and β. For example, the differences between α in Fig. and β in Fig. 6 are shown in Fig. 7, where $\delta = \{10 - 9, 12 - 6, 7 - 9, 9 - 6, 13 - 9, \ldots, 78 - 75\} = \{1, 6, -2, 3, 4, \ldots, 3\}$.

1	6	-2	3
4	-5	6	0
-7	-2	-3	5
11	1	3	3

Figure 7: *The differences, called* δ *, between* α *and* β

Step 2: Find the closest prime numbers of δ.

The symbol $\delta' = \{\delta'_1, \delta'_2, \ldots, \delta'_{n*n}\}$ indicates a set of the prime numbers, where δ'_i is the closest prime number of $|\delta_i|$, and $|\delta_i|$ is the absolute value of δ_i. For example, δ' of δ is shown in Fig. 8, in which the closest prime numbers of $|\delta_2|$ and $|\delta_3|$ are 5 and 2, respectively, since $|\delta_2| = 6$ and $|\delta_3| = 2$.

1	5	2	3
3	5	5	0
7	2	3	5
11	1	3	3

Figure 8: *The set of prime numbers, called* δ'*, of* $|\delta|$

Step 3: Group the prime numbers.

Scan δ' in order and divide the prime numbers in δ' into several groups. Every prime number in a group is distinct. Let $\varphi = \{\varphi_1, \varphi_2, \ldots, \varphi_k\}$ be the set of groups, where $\varphi_t = (\varphi_{t0}, \varphi_{t1}, \ldots, \varphi_{tr}) = (\delta'_p, \delta'_{p+1}, \ldots, \delta'_{p+r})$, $\varphi_t \subseteq \delta'$. Each prime number φ_{tj} in φ_t is distinct, $1 \le t \le k$, $0 \le j \le r$, $0 \le p \le n*n$, and $0 \le r \le n*n-p$.

For example, $\varphi = \{\varphi_1, \varphi_2, \ldots, \varphi_5\}$

$= \{(\delta'_1, \delta'_2, \delta'_3, \delta'_4), (\delta'_5, \delta'_6), (\delta'_7, \delta'_8, \delta'_9, \delta'_{10}, \delta'_{11}), (\delta'_{12}, \delta'_{13}, \delta'_{14}, \delta'_{15}), (\delta'_{16})\}$ = {(1, 5, 2, 3), (3, 5), (5, 0, 7, 2, 3), (5, 11, 1, 3), (3)}.

The symbol φ'_t represents the ordered prime numbers of φ_t, where $\varphi'_{ti} > 1$ and φ'_t is ordered by ascending order. Based on the example, $\varphi'_1 = \{2, 3, 5\}$, $\varphi'_2 = \{3, 5\}$, $\varphi'_3 = \{2, 3, 5, 7\}$, $\varphi'_4 = \{3, 5, 11\}$, and $\varphi'_5 = \{3\}$.

Step 4: Based on CRT, compute the information load for each group.

Here, we describe the concept of CRT and apply it to compute the information load of each group. CRT assumes that there are k prime numbers $P_1, P_2, \ldots, P_k$ and k numbers, $a_1, a_2, \ldots, a_k$, where $P_1 < P_2 < \ldots < P_k$. Then there exists a natural number M that simultaneously satisfies $M = a_j \bmod P_j$, where $j = 1, 2, \ldots, k$. Here, mod is the modulus operation and $a_j \bmod P_j$ is the remainder when a_j is divided by P_j. For example, assume there are three prime numbers 2, 3 and 5, where $P_1 = 2$, $P_2 = 3$ and $P_3 = 5$. Use CRT to find an M such that M mod 2 = 0, M mod 3 = 1, and M mod 5 = 2, where $a_1 = 0$, $a_2 = 1$, and $a_3 = 2$.

The process of using CRT to find M is shown in the following.

(1) Compute N and N_i, where $N = P_1 * P_2 * \ldots * P_k$ and $N_i = N / P_i$.

$N = 2 * 3 * 5 = 30$, $N_1 = 15$, $N_2 = 10$, and $N_3 = 6$.

(2) Compute Q_j, where $Q_j \equiv N_j * b_j \equiv 1 \bmod P_j$.

$Q_1 = 15 * 1 = 15$, since $Q_1 \equiv N_1 * b_1 \equiv 15 * b_1 \equiv 1 \bmod 2$, and $b_1 = 1$. Similarly,

$Q_2 = 10 * 1 = 10$, since $Q_2 \equiv N_2 * b_2 \equiv 10 * b_2 \equiv 1 \bmod 3$, $b_2 = 1$. And

$Q_3 = 6 * 1 = 6$, $Q_3 \equiv N_3 * b_3 \equiv 6 * b_3 \equiv 1 \bmod 5$, $b_3 = 1$.

(3) Compute M, where $M = a_1 * Q_1 + a_2 * Q_2 + \ldots + a_k * Q_k$.

The M for the prime numbers 2, 3 and 5 is 22, since $M = 0 * 15 + 1 * 10 + 2 * 6 = 22$, such that $22 \equiv 0 \bmod 2$, $22 \equiv 1 \bmod 3$, and $22 \equiv 2 \bmod 5$.

According to the CRT, for each φ'_t, there exists an M_t such that $M_t \equiv j \bmod \varphi'_{tj}$, where $0 \le j \le r$. In other words, there exists a solution M_t such that $M_t \equiv 0 \bmod \varphi'_{t0}$, $M_t \equiv 1 \bmod \varphi'_{t1}, \ldots, M_t \equiv r \bmod \varphi'_{tr}$. For example, for $\varphi'_1 = \{2, 3, 5\}$, the solution M_1 for the prime numbers 2, 3 and 5 is 22, such that

$22 \equiv 0 \bmod 2$, $22 \equiv 1 \bmod 3$, and $22 \equiv 2 \bmod 5$.

Similarly, $M_2 = 6$, since $6 \equiv 0 \bmod 3$, and $6 \equiv 1 \bmod 5$.

$M_3 = 52$, since $52 \equiv 0 \bmod 2$, $52 \equiv 1 \bmod 3$,

$52 \equiv 2 \bmod 5$, and $52 \equiv 3 \bmod 7$.

Therefore, $M = \{M_1, M_2, \ldots, M_r\} = \{M_1, M_2, \ldots, M_5\}$ = {22, 6, 52, 156, 3}. Each M_t represents the energy to hide the information in φ_t.

Step 5: Get the sub-information string from H for each group.

Let $L_t = \lfloor \log M_t \rfloor$ be the maximum length of information string to be hidden in φ_t. The sub-information string is denoted by h_t, where $h_t \subseteq H$, $<h_t> = L_t$, and $<h_t>$ is the total number of symbols in h_t. For example, assume H = "1110111000011001011" and M_1 = 22. The sub-information string that is hidden in φ_1 is h_1 = "c_1 c_2 c_3 c_4" = "1110," since the maximum length of the sub-information string is $L_1 = \lfloor \log M_1 \rfloor = \lfloor \log 22 \rfloor = 4$, and $<h_1> = 4$. Similarly, $L_2 = \lfloor \log 6 \rfloor = 2$ and h_2 = "11;" $L_3 = \lfloor \log 52 \rfloor = 5$ and h_3 = "10000;" $L_4 = \lfloor \log 156 \rfloor = 7$ and h_4 = "1100101;" $L_5 = \lfloor \log 3 \rfloor = 1$ and h_5 = "1."

Step 5: Hide information in each pixel.

The symbol $\overline{\beta} = \{\overline{\beta}_1, \overline{\beta}_2, \ldots, \overline{\beta}_{n*n}\}$ represents the hidden image. The value of $\overline{\beta}_i$ is computed by the following equation

$$\overline{\beta}_i = \begin{cases} \beta_i, \text{if } \delta_i = 0 \text{ or } \delta_i = 1 \text{ or } ((h_t)_{10} \bmod |\delta_i|) = 0, \\ \beta_i + ((h_t)_{10} \bmod \delta_i), \text{if } \delta_i > 1, \\ \beta_i - (|\delta_i| - (h_t)_{10} \bmod |\delta_i|), \text{otherwise.} \end{cases} \quad (1)$$

In the above equation, $(h_t)_{10}$ indicates a conversion of the sub-information string h_t into an integer in the decimal system. For example, if h_1 = "1110," then $(h_1)_{10}$ = 14. The first value of δ_1 is 1, which equals 1; therefore, the first pixel cannot be used to hide the information. The pixel value of $\overline{\beta}_1$ is equal to the original pixel value of β_1. The second value of δ_2 is 6, which is greater than 1; therefore, the pixel value of $\overline{\beta}_2$ is 8, since $\overline{\beta}_2 = \beta_2 + ((h_1)_{10} \bmod \delta_2) = 6 + ((1110)_{10} \bmod 6) = 6 + 2 = 8$, where $\overline{\delta}_2 \in \varphi_1$, M_1 = 22, and L_1 = 4. The third value of $\overline{\beta}_3$ is 9, since $((h_1)_{10} \bmod |\delta_3|) = ((110)_{10} \bmod |-2|) = 0$ and $\overline{\beta}_3 = \beta_3 = 9$, where $\overline{\delta}_3 \in \varphi_1$. The fourth value of $\overline{\beta}_4$ is 8, since $\overline{\beta}_4 = \beta_4 + ((h_1)_{10} \bmod \delta_4) = 6 + ((1110)_{10} \bmod 3) = 6 + 2 = 8$, where $\overline{\delta}_4 \in \varphi_1$. The fifth value of $\overline{\beta}_5$ is 12, since $\overline{\beta}_5 = \beta_5 + ((h_2)_{10} \bmod \delta_5) = 9 + ((11)_{10} \bmod 4) = 9 + 3 = 12$, where $\overline{\delta}_5 \in \varphi_2$. The final hidden image is shown in Fig. 9.

9	8	9	8
12	7	13	9
12	39	23	51
52	19	14	76

Figure 9: *The hidden image, called* $\overline{\beta}$*, of Fig. 5*

After the hiding process, the information string H is concealed in the compressed image β. The differences

$$\overline{\delta} = \{\overline{\delta}_1, \overline{\delta}_2, \ldots, \overline{\delta}_{n*n}\} = \{\alpha_1 - \overline{\beta}_1, \alpha_2 - \overline{\beta}_2, \ldots, \alpha_{n*n} - \overline{\beta}_{n*n}\}$$

between α and $\overline{\beta}$ are shown in Fig. 10.

The sender transmits the hidden image $\overline{\beta}$ to the receiver or publishes the hidden image on a Web site for information sharing. The encoding algorithm is shown in Algorithm 1.

1	4	-2	1
1	-2	2	0
-2	-2	-1	4
9	1	1	2

Figure 10: *The differences, called* $\overline{\delta}$*, between* α *and* $\overline{\beta}$

3.2 Decoder

When downloading or receiving the hidden image, the receiver can extract the hidden information using a decoder. In this paper, we assume the sender and the receiver both have the same host image α and codebook.

The extraction process can be stated as follows:

Step 1: Compute the differences, prime numbers, and information loads from α and β.

Since the receiver also has α and β, the decoder can compute the difference δ between α and β, find the closest prime numbers from δ, group prime numbers φ', compute the information load M of each group, and calculate the maximum length of sub-information string L.

For the example described in Subsection 3.1, δ = {1, 6, -2, 3, 4, -5, 6, 0, -7, -2, -3, 5, 11, 1, 3, 3}, φ' = {{2, 3, 5}, {3, 5}, {2, 3, 5, 7}, {3, 5, 11}, {3}} = {{ $\delta'_3, \delta'_4, \delta'_2$ }, { δ'_5, δ'_6 }, { $\delta'_{10}, \delta'_{11}, \delta'_7, \delta'_9$ }, { $\delta'_{15}, \delta'_{12}, \delta'_{14}$ }, { δ'_{16} }}, M = {22, 6, 52, 156, 3}, and L = {4, 2, 5, 7, 1}.

Step 2: Compute the hidden information.

The first group of φ' is φ'_1 = {2, 3, 5} = $\{\delta'_3, \delta'_4, \delta'_2\}$ and L_1 = 4. This means that the first sub-information with 4 bits is hidden in the third, fourth, and second pixels of $\bar{\beta}$. The information is obtained by Equation 1. Let us assume a hidden image, as shown in Fig. 9. The first sub-information is computed by

$$\begin{cases} \bar{\beta}_2 = 8 = 6 + ((h_1)_{10} \bmod 5), \\ \bar{\beta}_3 = 9 = 9, \\ \bar{\beta}_4 = 8 = 6 + ((h_1)_{10} \bmod 3). \end{cases} \quad (2)$$

The solution of $(h_1)_{10}$ is 14, since

$$\begin{cases} \bar{\beta}_2 = 8 = 6 + (14 \bmod 6), \\ \bar{\beta}_3 = 9 = 9, \\ \bar{\beta}_4 = 8 = 6 + (14 \bmod 3). \end{cases}$$

Hence, the first sub-information string is $(14)_{10}$ = (1110)2. The second sub-information is computed by

$$\begin{cases} \bar{\beta}_5 = 12 = 9 + ((h_2)_{10} \bmod 4), \\ \bar{\beta}_6 = 7 = 9 + (|-5| - (h_2)_{10} \bmod |-5|). \end{cases} \quad (3)$$

The solution of h_2 is 3. Therefore, the second sub-information string is $(3)_{10} = (11)_2$. Similarly, the third sub-information string is "10000", since $h_3 = (16)_{10} = (10000)_2$, and

$$\begin{cases} \bar{\beta}_7 = 13 = 9 + ((h_3)_{10} \bmod 6), \\ \bar{\beta}_9 = 12 = 17 - (|-7| - (h_3)_{10} \bmod |-7|), \\ \bar{\beta}_{10} = 39 = 39, \\ \bar{\beta}_{11} = 23 = 25 - (|-3| - (h_3)_{10} \bmod |-3|). \end{cases} \quad (4)$$

The fourth and fifth sub-information strings are "1100101" and "1," since $h_4 = (101)_{10} = (1100101)_2$ and $h_5 = (1)_{10} = (1)_2$, respectively. The complete information string is "1110111000011001011". The decoding algorithm is shown in Algorithm 2.

4. Experiments

The proposed method that has been developed is called the CRTIH system, and was run on a personal computer whose operating system is Windows 2000. The CPU of the personal computer is Pentium III, and its main memory is 256 megabytes. The test images used were Barbara, Boat, Lena, Pepper, Plane, Sailboat, Tiffany, Toys, GoldHil, Mandrill, Zelda, and Alan, called the host images. All the host images are 256 gray levels with 512*512 pixels. The images are shown in Fig. 11. The compressed images, which were compressed from the host images by the VQ-compressed technique, are shown in Fig. 12. The PSNR values of the compressed images with respect to the host image were 25.80 dB, 29.38 dB, 31.37 dB, 30.72 dB, 30.58 dB, 28.62 dB, 30.31 dB, 29.92 dB, 29.44 dB, 24.38 dB, 35.08 dB, and 26.99 dB.

In this paper, PSNR (peak signal-to-noise ratio) and (the number of hidden bits) are used to describe the performance of the proposed scheme. In order to test the performance of the CRTIH system, two experiments were conducted.

The first experiment embedded a secret image in the compressed images. The secret image used in these experiments was a binary image with 64*64 pixels, shown in Fig. 13. After the secret image was hidden in the compressed images using the CRTIH system, the hidden images were obtained. The PSNR values of the hidden images and the running times are shown in Table 1.

The second experiment tested the capacity of the compressed images. The CRTIH system randomly generated several various hidden strings and embedded them in the compressed images. The hidden images are shown in Fig. 14. Table 2 summarizes the results of the compressed images and the hidden images.

In addition, the compressed image Lena was used to compare the performance among Jo and Kim's method, Tian's method, and CRTIH [JK02], [Tia03]. The PSNR values of the image were 30.99 dB based on Jo and

Kim's method, 29.43 dB based on Tian's method, and 35.36 dB based on the CRTIH system. Meanwhile, the embedding capacity of Jo and Kim's method was 15,925 bits, of Tian's method was 260,018 bits, and of the CRTIH system was 269,030 bits. Therefore, the CRTIH system has greater embedding capacity than Jo and Kim's method [JK02] and Tian's method [Tia03]. Furthermore, a substantially upgraded visual quality of the image resulted.

Figure 11: *The host images*

Figure 12: The PSNR values of the compressed images

5. Analysis

In this section, we analyze the advantages of our proposed method CRTIH.

(1) Camouflage: The hidden information is invisible in the hidden image, and the hidden image is a meaningful image. Even though illegal users can steal the hidden image, it is not easy for the illegal users to attack the hidden image.

(2) Security: In this paper, we assume only the sender and receiver have the host images and codebook. In addition, the hidden information is encrypted by the DES-like system associated with the private key. Only legal users have the private key to decode the hidden information.

(3) Acceleration: The main factors that influence the performance of the CRTIH are VQ compression and CRT computing. Many researchers have proposed their own methods to speed up the compressing time of VQ compression. As for the computing time of CRT, it can be computed in real time. The experimental results show that the CRTIH system takes approximately 2 seconds to hide information for each image. Therefore, CRTIH can efficiently encode and decode images.

(4) Lossless information: The hidden information that is extracted from the hidden image by the decoder is exactly the same as the original secret information.

(5) Higher embedding capacity: Based on CRTIH, the average embedding capacity of a pixel in a host image is about 1.3 (bpp, bits per pixel) of the image, where 1.3 = (443,327 + 315,159 + 269,030 + 285,325 + 257,883 + 362,555 + 299,215 + 300,894 + 369,013 + 538,925 + 216,833 + 441,767) (bits) / (512 * 512) (pixels) * 12 (images).

(6) Higher visual quality: From the experimental results, we can see that CRTIH significantly upgrades the PSNR values of the compressed images.

Algorithm 1: Encoding
Input: a host image (α) with n*n pixels,
a compressed image (β) with n*n pixels,
and a secret information string (H)
Output: the hidden image ($\overline{\beta}$)
Initial:
Let P be a set of distinct prime numbers, where P = $\{P_0, P_1, \ldots, P_k\}$.
Let PL be a set of indices pointing to the locations of all prime numbers in P, where PL = $\{PL_0, PL_1, \ldots, PL_k\}$.
For i = 0 to n*n
Compute the difference between α and β, where $\delta_i = \alpha_i - \beta_i$.
If $\delta_i \leqq 1$ **Then**
The i^{th} pixel value of $\overline{\beta}$ equals to the i^{th} pixel value of α, where $\overline{\beta}_i = \beta_i$.
Else
Find the closest prime numbers of δ_i, called δ_i'.
If δ_i' not exist in P **Then**
Put δ_i' into the set of distinct prime numbers, where $P = P \cup \delta_i'$.
Put i into the set of indices, where $PL = PL \cup i$.
Else
Use CRT to compute M, where $P_j \equiv j \pmod M$ and $0 \leq j \leq k$.
Compute the length of sub-information, where $L = \lfloor \log M \rfloor$.
Get the sub-information h from H, where <h> = L.
Upgrade the pixel values of $\overline{\beta}$, where $\overline{\beta}_j$ is computed by Equation 1, $PL_0 \leqq j \leqq PL_k$, and k is number of elements in PL.
Remove all elements from P and PL.
End If
End If
Next
Return $\overline{\beta}$

Algorithm 2: Decoding
Input: a host image (α) with n*n pixels, a compressed image (β) with n*n pixels,
and a hidden image ($\overline{\beta}$)
Output: the hidden information (H)
Initial:
Let P be a set of distinct prime numbers, where P = $\{P_0, P_1, \ldots, P_k\}$.
Let PL be a set of indices pointing to the locations of all prime numbers in P, where PL = $\{PL_0, PL_1, \ldots, PL_k\}$.
For i = 0 to n*n
Compute the difference between α and β, where $\delta_i = \alpha_i - \beta_i$.
If $\delta_i \leq 1$ **Then**
The i^{th} pixel value of $\overline{\beta}$ equals to the i^{th} pixel value of α, where $\overline{\beta}_i = \beta_i$.
Else
Find the closest prime numbers of δ_i, called δ_i'.
If δ_i' does not exist in P **Then**
Put δ_i' into the set of distinct prime numbers, where $P = P \bigcup \delta_i'$.
Put i into the set of indices, where $PL = PL \bigcup i$.
Else
Use CRT to compute M, where $P_j \equiv j \pmod M$ and $0 \leq j \leq k$.
Compute the length of sub-information, where $L = \lfloor \log M \rfloor$.
Use CRT to compute the value of $(h)_{10}$, where $\overline{\beta}_j$ is computed by Equation 1, $PL_0 \leq j \leq PL_k$, and k is number of elements in PL.
Concatenate H and h to get a new H; that is, H = H||h.
End If
End If
Next
Return H

Table 1: *The PSNR values of the compressed images and hidden images*

File name	PSNR of Compressed images	PSNR of Hidden images	Time (seconds)
Barbara	25.80	25.87	1
Boat	29.38	29.38	1
Lenna	31.37	31.37	1
Pepper	30.72	30.75	0
Jet	30.58	30.60	1
Sailboat	28.62	28.71	1
Tiffany	30.31	30.32	1
Toys	29.92	29.97	0
GoldHil	29.44	29.46	1
Mandrill	24.38	24.40	0
Zelda	35.08	35.11	1
Alan	26.99	26.99	1

Table 2: *The PSNR values of the hidden images with various hidden strings*

File name	$\|H\|$ (bits)	PSNR of Compressed images	PSNR of Hidden images	Time (seconds)
Barbara	443,327	25.80	30.10	3
Boat	315,159	29.38	33.53	3
Lenna	269,030	31.37	35.40	2
Pepper	285,325	30.72	34.68	2
Jet	257,883	30.58	34.64	2
Sailboat	362,555	28.62	32.73	3
Tiffany	299,215	30.31	33.77	3
Toys	300,894	29.92	33.92	3
GoldHil	369,013	29.44	33.32	3
Mandrill	538,925	24.38	28.74	3
Zelda	216,833	35.08	38.48	2
Alan	441,767	26.99	30.43	3

6. Conclusions

In this paper, we proposed a high embedding capacity and high visual quality information-hiding scheme that is based on CRT. The experimental results show the scheme efficiently hides a great amount of secret information. In addition, the visual quality of a hidden image is even better than that of the VQ-compressed image.

Figure 13: *The secret image with 64*64 pixels*

***s*Figure 14:** *The PSNR values of the hidden images*

References

[ASI03] AMIN M. M., SALLEH M., IBRAHIM S., KATMIN M. R. AND SHAMSUDIN M. Z. I.: Information Hiding Using Steganography. *Proceedings of the 4th National Conference on Telecommunication Technology*, Shah Alam, Malaysia, (2003), pp. 21-25.

[Bau97] BAUER F. L.: Decrypted Secrets-methods and Maxims of Cryptology. Berlin. Heidelberg. Germany. Springer-Verlag. 1997. *Decrypted Secrets-methods and Maxims of Cryptology*, Berlin, Heidelberg, Germany, Springer-Verlag, (1997).

[CHC03] CHANG C. C., HSIAO J. Y. AND CHAN C. S.: Finding Optimal Least-significant-bit Substitution in Image Hiding by Dynamic Programming Strategy. *Pattern Recognition*, Vol. 36, (2003), pp. 1583-1595.

[CKL97] COX I., KILIAN J., LEIGHTON F. AND SHAMOON T.: Secure Spread Spectrum Watermarking for Multimedia. *IEEE Transactions on Image Processing*, Vol. 6, No. 12, (1997), pp. 1673-1687.

[DVM98] DELAIGLE J., VLEESCHOUWER C. AND MACQ B. DE: Psychovisual Approach to Digital Picture Watermarking. *Journal of Electronic Imaging*, Vol. 7, No. 3, (1998), pp. 628-640.

[FAK99] FABIEN A. P., ANDERSON R. J. AND KUHN, M. G.: Information Hiding - A

Survey. *Proceedings of the IEEE Special Issue on Protection of Multimedia Content*, Vol. 87, No. 7, (1999), pp. 1062-1078.

[HS02] HUANG J. AND SHI, Y. Q.: Reliable Information Bit Hiding. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 12, No. 10, (2002), pp. 916-920.

[JK02] JO M. AND KIM, H.: A Digital Image Watermarking Scheme Based on Vector Quantization. *IEICE Transactions on Information and System*, Vol. E85-D, No. 6, (2002), pp. 1054-1056.

[KZ95] KOCH E. AND ZHAO J.: Embedding Robust Labels into Images for Copyright Protection. *Proceedings of the International Congress on Intellectual Property Rights for Specialized Information, Knowledge and New Technologies, Vienna, Austria*, (1995), pp. 242-251.

[LS00] LU Z. M. AND SUN S. H.: Digital Image Watermarking Technique Based on Vector Quantization. *IEE Electronics Letters*, Vol. 36, No. 4, (2000), pp. 303-305.

[Sti95] STINSON D. R.: Cryptography: Theory and Practice. CRC Press. Inc.. Boca Raton. Florida. 1995. *Cryptography: Theory and Practice*, CRC Press, Inc., Boca Raton, Florida, (1995).

[SZT96] SWANSON M. D., ZHU B., TEWFIK A. H.: Robust Data Hiding for Images. *Proceedings of the IEEE 7th Digital Signal Processing Workshop, Loen, Norway*, (1996), pp. 37-40.

[Tia03] TIAN J.: Reversible Data Embedding and Content Authentication Using Difference Expansion. *IEEE Transactions on Circuits and Systems for Video Technology*, (2003).

[TP02] TSENG Y. C. AND PAN H. K.: Data Hiding in 2-color Images. *IEEE Transactions on Computers*, Vol. 51, No. 7, (2002). pp, 873-878.