An Efficient Smart Card Based Authentication Scheme Using Image Encryption

CHIN-CHEN CHANG1,2, HSIAO-LING WU1, ZHI-HUI WANG3, and QIAN MAO4

1Department of Information Engineering and Computer Science
Feng Chia University
Taichung, 407 Taiwan
2Department of Computer Science and Information Engineering
Asia University
Taichung, 413 Taiwan
3School of Software
Dalian University of Technology
Dalian City, 116 P.R. China
4Department of Optical-Electrical and Computer Engineering
University of Shanghai for Science and Technology
Shanghai, 200 P.R. China

To date, there are many authentication schemes have been proposed for smart cards. However, when a smart card is stolen or the authentication token is intercepted, most of these schemes will be insecure. In 2012, Chang et al. proposed an authentication scheme for smart card to avoid the aforementioned problem. They combined the smart card and image encryption technology for authentication. However, their scheme has lower quality for the restored image. Hence, we propose two image-encryption schemes in frequency domain. In the first scheme, we scramble the positions of the DCT coefficients. In the second scheme, we scramble the positions of the DCT coefficients; meanwhile, the values of the DCT coefficients are diffused. The experimental results showed that both of our proposed schemes have higher quality of the restored images with low storage.

Keywords: image encryption, torus automorphism, diffusion, authentication, smart card

1. INTRODUCTION

Due to the advantages of smart cards, such as their portability and ease of use, they are commonly used for many applications in our everyday lives. Since 1990, many smart card based authentication schemes have been proposed [1-6]. However, some researchers think that a pure smart card based authentication scheme exposes the user to potential risks. When a smart card is stolen, the thief can use the smart card if she or he is able to determine the authentication token. Therefore, a mixture of authentication schemes (i.e., smart card and image encryption) was proposed [7-9]. With regard to image encryption, many schemes have been proposed using various technologies, e.g., VQ-based [10], RSA-based [11], ECC-based [12, 13], AES-based [14, 15], and chaotic-based [16, 17] technologies. Since chaos-based technology provides high security and low computational cost, image encryption schemes based on this technology generally have better performance. Hence, Chang et al. used chaotic-based technology to propose a new au-
In [18], a legitimate user’s photograph was printed on the smart card, and the torus automorphism (one kind of chaotic technology) was used to scramble the positions of pixels for this photograph to produce a chaotic image for storage in the smart card. In addition, a secret sharing scheme was used to protect the initial parameters. If a user wants to use her/his smart card to make a transaction at a shop, she/he must provide the smart card and the shared secret to the shop for authentication. However, we found that Chang et al.’s scheme has low image quality when the storage is low. Hence, in this paper, we propose two improved schemes in frequency domain. Our schemes are similar in concept to Chang et al.’s scheme [18]. The original photograph is printed on the smart card and the cipher-image is stored in the smart card. However, the major difference between our schemes and [18] is in the image encryption. We perform a process that permutes and diffuses the DCT coefficients. Our first scheme provides lower computational cost and storage rate, but the use of smaller-sized key may have security concerns. Hence, the second scheme was proposed in order to achieve better security while maintaining the same advantages of the first scheme. The experimental results showed that both of our proposed schemes have higher quality with low storage rate.

The rest of this paper is organized as follows. In section 2, we briefly introduce torus automorphism, the diffusion algorithm, and Shamir’s secret sharing. Also, Chang et al.’s scheme is reviewed in this section. In sections 3 and 4, we propose the basic and three-dimensional image encryption schemes in frequency domain. Then, section 5 provides the experimental results and our analysis. The conclusions are given in section 6.

2. PRELIMINARIES

In this section, we briefly review the basic fundamentals used in our scheme. First, we introduce the torus automorphism [19], the diffusion algorithm and Shamir’s secret sharing [20] which are used in our scheme. The section is concluded with a review of Chang et al.’s scheme.

2.1 Gray Image Encryption Algorithm Based on Torus Automorphism

Without loss of generality, we assumed that the size of the gray plain-image $P$ is $M \times M$ and that each pixel $p_i$ has the coordinates $(x_i, y_i)$. In this paper, the two-dimensional torus automorphism was used as the chaotic system. Torus automorphism can be described as

$$
\begin{bmatrix}
x^n_i \\
y^n_i \\
\end{bmatrix} = A^n \begin{bmatrix}
x^0_i \\
y^0_i \\
\end{bmatrix} \pmod{N}, \text{ where } A = \begin{bmatrix}
1 & a \\
b & ab + 1 \\
\end{bmatrix}.
$$

The recurrence time $R$ depends on values of $a$, $b$, and $N$, where $R$, $a$, $b$, and $N$ are integers. The details concerning the generation of a strong chaos system is discussed in [21-23]. A torus automorphism encryption scheme consists of two algorithms, i.e., (1) an encryption algorithm and (2) a decryption algorithm. In the encryption algorithm, the
gray plain-image $P$ and the values of $a$, $b$, and $N$ are provided as inputs by the sender. Then, the gray cipher-image $C$ is generated as output by performing Eq. (1) $t$ times, where $t$ is smaller than $R$. In the decryption algorithm, the receiver inputs the cipher-image $C$ and the same values of $a$, $b$, and $N$. When Eq. (1) is performed $R - t$ times, the plain-image $P$ will be recovered.

If both of the following requirements are satisfied, the decrypted image is the same as the original plain-image:

- Both the encryption and decryption algorithms use the same matrix $A$.
- The sum of the iterations in encryption and decryption equals $R$.

2.2 The Diffusion Algorithm

In the paper, the diffusion algorithm of Mao et al.’s scheme [24] is used in our second scheme. The diffusion algorithm can be divided into two functions, i.e., the diffusion function and recovery function. First, we assume that the original image $P^0$ is performed the iterations $t$ times and the scrambled image $P^t$ is obtained. In this scrambling process, an arbitrary coefficient with coordinates of $(x^0_i, y^0_i)$ in $P^0$ is shifted to $(x^t_i, y^t_i)$ in $P^t$. In the diffusion function, the scrambled image $P^t$ is provided as input by sender. After that, the diffused image $P^t_d$ is generated as output. The diffusion function is shown as following:

$$L_d(x^t_i, y^t_i) = L_s(x^t_i, y^t_i) + \lambda L_d(x^t_i, y^t_i - 1) + (x^t_i - x^0_i) + (y^t_i - y^0_i) \mod 256,$$

where $L_s(x^t_i, y^t_i)$ and $L_d(x^t_i, y^t_i)$ denote the coefficients with coordinates of $(x^t_i, y^t_i)$ in scrambled image $P^t$ and diffused image $P^t_d$, respectively, and $\lambda$ is the speed of diffusion. In order to compute $L_d(1, 1)$, the initial value $L_d(1, 0)$ should be pre-determined.

When a receiver obtains this diffused image $P^t_d$ from the sender. She/he can execute the following recovery function to recover the scrambled image $P^t$:

$$L_s(x^t_i, y^t_i) = L_d(x^t_i, y^t_i) - \lambda L_d(x^t_i, y^t_i - 1) - (x^t_i - x^0_i) - (y^t_i - y^0_i) \mod 256.$$

After that, the scrambled matrix $P^t$ is obtained.

2.3 Shamir’s Secret Sharing

In 1979, Shamir proposed a secret sharing scheme [20] and named it Shamir’s $(t, n)$ SS scheme. Here, we only review this scheme based on a linear polynomial. There are $n$ shareholders, $\{U_1, U_2, ..., U_n\}$, and a sender, $S$. Shamir’s $(t, n)$ SS scheme consists of the following two procedures.

Share generation

Sender $S$ picks a random polynomial $f(x)$ with degree $t - 1$: $f(x) = a_0 + a_1x + ... + a_{t-1}x^{t-1} \mod p$. The secret is $s = f(x) = a_0$, and all coefficients, $a_1, a_2, ..., a_{t-1}$, are in the finite field $GF(p)$ with $p > s$. $S$ computes $n$ shares $y_i = f(x_i)$, where $x_i$ is the public information of shareholder $U_i$ for $i = 1, 2, ..., n$. Then, the sender distributes each share $y_i$ to the corresponding shareholder, $U_i$, via a secure channel.
Secret reconstruction

Assume that \( t \) shareholders, \( \{U_1, U_2, \ldots, U_t\} \), want to recover the secret value, \( s \). They release their shares and use the following Lagrange interpolating formula to recover the secret, \( s \):

\[
s = f(0) = \sum_{i=1}^{t} f(x_i) \prod_{j=1, j \neq i}^{t} \frac{-x_j}{x_i - x_j} \mod p.
\]

2.4 Chang et al.’s Authentication Scheme Based on Spatial Domain

In this subsection, we briefly review the framework of Chang et al.’s authentication scheme [18]. There are three participants, (1) a trusted third party (such as a bank), (2) a user who has the smart card, and (3) a terminal (such as a shop). Their scheme consists of two phases, i.e., (1) the registration phase and (2) the authentication phase.

Registration phase

When the bank accepts the registration requisition from the user, the user takes a photograph of herself/himself and provides it to the bank via a secure channel. Chang et al. assumed that the original photograph \( P \) is a gray image with the size of \( M \times M \) pixels, and the pixels’ values ranged from 0 to 255. Each pixel can be denoted as an eight-bit byte. Due to the limited storage capacity of a smart card, there are only \( m_s \) bits to represent each pixel’s value in the secret image \( P_0 \), where \( m_s \) is smaller than eight.

First, the bank sets the initial parameters \( a \) and \( b \) to generate matrix \( A \), and the iteration times \( t \). After \( t \) times of iteration, the cipher-image \( C \) is generated and stored in the smart card; the original photograph \( P \) is printed on the smart card.

In order to protect the values of \( a \), \( b \), and \( R - t \), they used a second-degree polynomial function \( f(x) \) to share the three shares, \((x_1, y_1), (x_2, y_2)\) and \((x_3, y_3)\). Then, the first point, \((x_1, y_1)\), is kept by the bank, and the other points, \((x_2, y_2)\) and \((x_3, y_3)\), are kept as secrets by the user. Finally, the bank issues the smart card to the user.

Authentication phase

When the user uses the smart card in a shop, the authentication phase will be performed by the shop. First, the shop reads the cipher-image \( C \) from the smart card and obtains the three shares, \((x_1, y_1), (x_2, y_2)\) and \((x_3, y_3)\), from the user and the bank. Then, the cipher-image \( C \) is iterated \( R - t \) times using Eq. (1). If the right keys are provided, the secret image \( P_0 \) will be decrypted correctly. When the secret image \( P_0 \) looks the same as the original photograph \( P \), which is printed on the smart card, the authentication is successful; otherwise, this process is terminated.

3. BASIC IMAGE ENCRYPTION SCHEME

In this section, we propose a basic image encryption scheme based on the frequency domain. We only permute the positions of the DCT coefficients in the first scheme. Hence, it has the lower computational cost and storage rate. Our scheme consists of two phases: the registration phase and the authentication phase. There are three participants: a
bank, a user who has the smart card, and a shop. The bank issues the smart card to the user in the registration phase. Then, the legality of the user can be verified by the shop in the authentication phase. The details of the basic image encryption scheme are described below.

**Registration phase**

When a user $U$ wants to obtain a smart card from a trusted third party (such as a bank), she/he must take a photo of herself/himself and send it to the bank via a secure channel. We assumed that the original photograph $P$ is a gray image with the size of $M \times M$ pixels, and the pixels’ values ranged from 0 to 255. Then, the bank performs the following steps to generate the smart card for $U$:

**Step R1:** The original photograph $P$ is partitioned into many blocks with the size of $8 \times 8$ pixels, i.e., there are $M/8 \times M/8$ blocks. Each block is processed a DCT transform, and an $8 \times 8$ DCT block is obtained. The elements in the DCT block are the DCT coefficients, which are ranged from 0 to 255. In each DCT block, the coefficient that is located at the upper, left corner is the DC component, and the coefficients that are located at the lower, right corner are the high-frequency coefficients. In general, the DC component represents the main information of the image, and the high-frequency coefficients represent the details of the image. Therefore, an $m_f \times m_f$ ($m_f < 8$) sub-block is picked from the upper left corner in each DCT block. Finally, the compressed DCT matrix $L$ is obtained. Fig. 1 shows the compressing process.

![Diagram of DCT Matrix Compression](image)

**Fig. 1. Compression of the DCT Matrix.**

**Step R2:** The bank sets the initial parameters $a$ and $b$ to generate matrix $A$. The modulus $N$ of the torus automorphism is equal to $(M \times m_f)/8$. The secret image $L^0$ (i.e., DCT matrix $L$) is iterated $t$ times by the chaotic system shown as Eq. (1), thereby the cipher-image $C$ (i.e., $L_t$) is generated. Image $C$ will be stored in the smart card. In addition, the original photograph $P$ is printed on the smart card.

**Step R3:** In order to protect the parameters $a$, $b$, and $R - t$, we modified Shamir’s secret sharing scheme to share the parameters $a$, $b$, and $R - t$. Our polynomial function was designed as follows:
\[ f(x) = a + bx + (R - t)x^2. \] (5)

According to Eq. (3), the bank computes three shares, \( y_1 = f(x_1), y_2 = f(x_2), \) and \( y_3 = f(x_3), \) where \( x_1 \) is the public information of the bank, and \( x_2 \) and \( x_3 \) are the public information of \( U. \)

**Step R4:** The bank keeps the first point, \((x_1, y_1)\), and user \( U \) keeps the other points, \((x_2, y_2)\) and \((x_3, y_3)\), secretly. The bank issued the smart card to \( U. \)

**Authentication phase**

When the user \( U \) wants to use the smart card in a shop, the shop must authenticate whether the user is a legitimate cardholder or not. First, user \( U \) and the bank must release their shares, \((x_1, y_1), (x_2, y_2),\) and \((x_3, y_3)\), to the shop. Then, the shop performs the following steps:

**Step A1:** The shop employs the Lagrange interpolating formula to recover Eq. (5), as shown:

\[
f(x) = \sum_{j=1}^{3} y_j \prod_{i=1,i\neq j}^{3} \frac{x - x_j}{x_i - x_j} \mod p.
\] (6)

Therefore, the shop obtains the parameters \( a, b, \) and \( R - t. \)

**Step A2:** The shop reads the cipher-image \( C \) (i.e., \( L^t \)) from the smart card. When the cipher-image \( C \) is iterated \( R - t \) times by Eq. (1), \( L^R \) is generated. If the right keys are used, \( L^R \) is equal to \( L^0 \) (i.e., DCT matrix \( L \)). Then, each \( m_f \times m_f \) sub-block is transferred back to the upper left corner in each \( 8 \times 8 \) DCT block, and the remaining coefficients in the DCT block are recovered as 0. After that, the inverse DCT transformation is processed on this rebuilt DCT matrix, and the recovered image \( P' \) is obtained. The recovery process is shown in Fig. 2.

![Fig. 2. Recovery of the DCT Matrix.](image)

**Step A3:** The shop employee visually compares the recovered image \( P' \) and the original photograph \( P \), which is printed on the smart card. If they look the same, authentication is successful; otherwise, this process is terminated.
The key space

The security of Chang et al.’s scheme and our scheme depends on parameters $a$, $b$, modulus $N$, and $R - t$. We used Chang et al.’s results, as shown in Table 1. When $N$ is small, the recurrence time $R$ is also small. The parameters $a$, $b$, and $N$ provide a key space with the size of $N^2$. The parameter $R - t$ is the number of iterations for decryption. Since $1 \leq R - t < R$, the size of the key space is $R - 1$. Therefore, the total key space $S$ between Chang et al.’s scheme and our scheme is:

$$S = (R - 1) \times N^2.$$  (7)

The storage rate of our first scheme is smaller than that of Chang et al.’s scheme. As a result, the total key space of our scheme is smaller than that of Chang et al.’s scheme. In order to increase the security of our first image encryption scheme, we propose the three-dimensional image encryption scheme in the next section.

<table>
<thead>
<tr>
<th>Table 1. Examples of recurrence time $R$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modulus</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>$N = 128$</td>
</tr>
<tr>
<td>$N = 256$</td>
</tr>
<tr>
<td>$N = 512$</td>
</tr>
</tbody>
</table>

4. THREE-DIMENSIONAL IMAGE ENCRYPTION SCHEME

Image encryption in the spatial domain is simple, but the quality of image and its robustness are low. Hence, the image encryption scheme performed in DCT domain is proposed in section 3. The properties of less storage and high image quality are satisfied in the first scheme. But since we only permute the positions of the DCT coefficients, a small key space is resulted in, which decreases the security of the scheme. Therefore, we used the concept of [24] to diffuse the values of DCT coefficients in a three-dimensional image encryption scheme.

Similar to the first scheme, there are three participants, i.e., a bank, the user and a shop. And our scheme consists of two phases, i.e., (1) the registration phase and (2) the authentication phase. The details of our three-dimensional image encryption scheme are described below.

Registration phase

User $U$ takes a photograph $P$ and gives it to the bank. Then, the bank performs the following steps:

Step R1: The original photograph $P$, with the size of $M \times M$ pixels, is partitioned into $M/8 \times M/8$ blocks. Each block has $8 \times 8$ pixels. The DCT transform is proc-
essed to each block, and an $8 \times 8$ DCT block is obtained. The elements in the DCT block are the DCT coefficients, which are ranged from 0 to 255. In each DCT block, an $m_j \times m_j (m_j < 8)$ sub-block is chosen from the upper left corner. Then, the compressed DCT matrix $L$ is obtained. This step is the same as Step R1 of section 3.

**Step R2:** The bank generates matrix $A$ by choosing the initial parameters $a$ and $b$. The modulus $N$ of the torus automorphism is equal to $(M \times m_j)/8$. When the secret image $L^0$ (i.e., DCT matrix $L$) is iterated $t$ times by (1), the scrambled matrix $L'$ is obtained. In this scrambling process, an arbitrary coefficient with coordinates of $(x_0^i, y_0^i)$ in $L^0$ is shifted to $(x_t^i, y_t^i)$ in $L'$. After that, the scrambled matrix $L'$ is diffused by Eq. (2) of Subsection 2.2. In order to compute $L_d(1, 1)$, the initial value $L_d(1, 0)$ should be pre-determined. Fig. 3 shows the diffusion process, and Fig. 4 (a) shows the diffusion path. After the diffusion process, the cipher-image $C$ (i.e., $L_d$) is generated and it will be stored in the smart card. In addition, the original photograph $P$ is printed on the smart card.

![Diagram of DCT Matrix](image)

Fig. 3. Diffusion of the DCT Matrix.

![Diagram of Diffusion Path](image)

Fig. 4. Diffusion path in the encryption and decryption phases.

\[
\alpha_i = L_d(x_i^0, y_i^0), \quad \beta_i = L_d(x_t^i, y_t^i), \quad (a_i, b_i) = (x_0^i, y_0^i), \quad (c_i, d_i) = (x_t^i, y_t^i)
\]
Step R3: When the right keys, i.e., parameters $a$, $b$, $\lambda$, $L_d(1, 0)$, and $R - t$, are provided, the DCT matrix $L^0$ will be recovered correctly. In order to protect the parameters $a$, $b$, $\lambda$, $L_d(1, 0)$, and $R - t$, we modified Eq. (5) to share the parameters $a$, $b$, $\lambda$, $L_d(1, 0)$, and $R - t$. Our polynomial function was designed as follows:

$$f(x) = a + bx + (R - t)x^2 + \lambda x^3 + L_d(1, 0)x^4. \quad (8)$$

The bank computes five shares, $y_1 = f(x_1)$, $y_2 = f(x_2)$, $y_3 = f(x_3)$, $y_4 = f(x_4)$ and $y_5 = f(x_5)$, according to Eq. (8), where $x_1$, $x_2$, and $x_3$ are the public information of the bank, and $x_4$ and $x_5$ are the public information of $U$.  

Step R4: The point $(x_1, y_1)$ is kept by the bank, the points $(x_2, y_2)$ and $(x_3, y_3)$ are kept by user $U$, and the points $(x_4, y_4)$ and $(x_5, y_5)$ are stored in the smart card. Finally, the bank issues the smart card to user $U$.

Authentication phase

When user $U$ wants to use the smart card in a shop, the shop must verify the identity of user $U$. First, user $U$ and the bank must release their shares, $(x_1, y_1)$, $(x_2, y_2)$, and $(x_3, y_3)$ to the shop, and the shop employee reads the points $(x_4, y_4)$ and $(x_5, y_5)$ from the smart card. Then, the following steps are performed by the shop:

Step A1: The following Lagrange interpolating formula can be used to recover Eq. (8):

$$f(x) = \sum_{i=1}^{5} y_i \prod_{j=1, j\neq i}^{5} \frac{x - x_j}{x_i - x_j} \mod p. \quad (9)$$

Therefore, the parameters $a$, $b$, $\lambda$, $L_d(1, 0)$, and $R - t$ are obtained.

Step A2: The shop employee reads the cipher-image $C$ (i.e. $L_d'$) from the smart card, she/he recovers the scrambled matrix $L'$ according to Eq. (3) of section 2.2. After that, the scrambled matrix $L'$ is obtained. Then, the scrambled matrix $L'$ is iterated $R - t$ times by Eq. (1).  

Fig. 5 shows the recovery process, and Fig. 4 (b) shows the recovery path. If the right keys are used, the shop employee can re-
CHIN-CHEN CHANG, HSIAO-LING WU, ZHI-HUI WANG AND QIAN MAO

Fig. 6. Results of image scrambling in the DCT domain.

retrieve compressed DCT matrix $L^{0}$. Each $m_f \times m_f (m_f < 8)$ sub-block is transferred back to the upper left corner in each $8 \times 8$ DCT block, and the remaining coefficients in the DCT block are recovered as 0. After that, the inverse DCT transformation is processed on this rebuilt DCT matrix, and the recovered image $P'$ is obtained.

**Step A3:** The shop employee visually compares the recovered image $P'$ and the original photograph $P$, which is printed on the smart card. If they look the same, authentication is successful; otherwise, this process is terminated.

5. EXPERIMENTAL RESULTS

In this section, we show the experimental results of encryption and decryption and discuss the security of the proposed scheme. Finally, we compare the storage rate of the smart card and the quality of the images between our scheme and Chang et al.’s scheme.

5.1 Encryption and Decryption Results

In this paper, we proposed two image encryption schemes in frequency domain. In the three-dimensional image encryption scheme, we scramble the positions of the DCT coefficients and diffused their values. Hence, we directly show the encryption and decryption results of the second scheme.

The image “Tiffany” with the size of $512 \times 512$ pixels was used for encryption. For each DCT block, only four, low-frequency coefficients were used. That is to say, we only stored a $2 \times 2 (m_f = 2)$ sub-block for each $8 \times 8$ DCT block. Therefore, the storage was 1/16, compared with the original plain-image. Here, we set the initial parameters $a = 7, b = 8$, and $N = 128$, and the initial values are $\lambda = 7$ and $L_d(1, 0) = 1$. According to Table 1, the recurrence time $R = 128$ for matrix $A_1$. Then, Fig. 6 (a) shows the plain-image. Fig. 6 (b) shows the compressed image. The encrypted image is shown in Figs. 6 (c)-(e), where 6 (c), 6 (d), and 6 (e) are iterated 1, 64, and 92 times, respectively. After the inverse process of diffusion and permutation, the images (c), (d), and (e) can be decrypted as (f). We can see that (f) is exactly the same as (b). Compared with (b), the PSNR of (c), (d), and (e) are all $-1.2$ dB, and the PSNR of (f) is 91.6 dB.

![Fig. 6. Results of image scrambling in the DCT domain.](image-url)
5.2 Security Analysis

In this paper, a two-factor authentication scheme was proposed, i.e., smart card and image encryption. We assumed that Eve is masquerading as a legitimate user and obtains service from the shop. Eve stole the smart card from Alice, who is the legitimate user. To evaluate the security of smart card based image encryption schemes, we assumed that Eve may have the following capabilities:

1. Eve has the capability of modifying the photograph. That is, Eve can replace Alice’s photograph on the smart card with herself photograph.
2. Eve may either (i) obtain Alice’s shares \((x_2, y_2)\) and \((x_3, y_3)\), or (ii) extract the secret information of the smart card, but she cannot achieve both (i) and (ii).

According to capability (2), the following cases are discussed:

Case 1. When Eve obtains the shares \((x_2, y_2)\) and \((x_3, y_3)\), she can present those shares to the shop for authentication. The shop will decrypt the cipher-image and recover the original photograph, which is a picture of Alice. Even Eve is able to obtain the correct shares, she cannot impersonate a legitimate user.

Case 2. If Eve extracts the secret information (i.e., the cipher-image), she may attempt to obtain the parameters \(a, b, \lambda, L, (1, 0), \) and \(R - t\) and use them to generate a new cipher-image with her photograph. If it is assumed that she uses brute-force attacks to gain those parameters, she still cannot impersonate a legitimate user because she cannot replace Alice’s cipher-image in the smart card with the new cipher-image.

Hence, we believe that our scheme is secure under aforementioned assumptions.

5.3 Distribution of DCT Matrix Coefficients

In this analysis, the gray-scale image “Tiffany” (512 × 512 pixels) was used in our second scheme and in Chang et al.’s scheme. We calculated the histograms of the plain-image and the cipher-image. The results are shown in Figs. 7 and 8. Figs. 7 (a1) and 8 (a1) show the gray plain-images. Figs. 7 (a2) and 8 (a2) show the compressed images. The encrypted images are shown as Figs. 7 (a3)-(a5) and 8 (a3)-(a5), and the decrypted images are shown as Figs. 7 (a6) and 8 (a6). The histograms of Figs. 7 (a1)-(a6) and 8 (a1)-(a6) are shown as Figs. 7 (b1)-(b6) and 8 (b1)-(b6).

We can see that the histograms of the cipher-images are fairly uniform and are significantly different from the plain-image in our second scheme. In Chang et al.’s scheme, the histograms of the cipher-images are the same as the histograms of the plain-image. Therefore, their scheme cannot resist statistical attacks, while our second scheme can resist statistical attacks.

5.4 Performance Comparison

In this section, we compare the performances of Chang et al.’s scheme and our proposed three-dimensional image encryption scheme. We used the image “Tiffany,” with
the size of $512 \times 512$ pixels, i.e., $M = 512$, for the comparisons. In order to compare performance, different storages were used, i.e., $(8 \times 512)^2$ bits, $(4 \times 512)^2$ bits, $(3 \times 512)^2$ bits, $(2 \times 512)^2$ bits, and $(512)^2$ bits. Chang et al.’s scheme requires 8, 4, 3, 2, and 1 bits to represent each pixel, respectively. Our scheme requires $8 \times 8$, $4 \times 4$, $3 \times 3$, $2 \times 2$ and $1 \times 1$ sub-blocks to represent each $8 \times 8$ DCT block, respectively, all of which are located at the upper, left corner. Table 2 shows the results of the comparisons.

Table 2 shows that when the storage is higher than 50% of the original image, Chang et al.’s scheme had a better quality of the recovered image. This is because that the DCT transform losses some information of the original image, which decreases the quality of the recovered image. When the storage is low, the quality of the image recovered by Chang et al.’s scheme is apparently lower than that of our scheme. Taking the limitation of the storage of smart card into consideration, low storage is a very important factor. Therefore, our schemes are suitable for the authentication of smart cards. Also, our three-dimensional scheme provides higher security.
Table 2. Comparison of results achieved by Chang et al.’s scheme and our scheme.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Storage bits/rate</th>
<th>PSNR (dB)</th>
<th>Recovered Image</th>
<th>Recurrence Time $R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chang et al.’s</td>
<td>$(8 \times 512)^2$/100%</td>
<td>100.3</td>
<td>![Recovered Image]</td>
<td>512</td>
</tr>
<tr>
<td>Our Scheme</td>
<td></td>
<td>29.1</td>
<td>![Recovered Image]</td>
<td>512</td>
</tr>
<tr>
<td>Chang et al.’s</td>
<td>$(4 \times 512)^2$/50%</td>
<td>29.8</td>
<td>![Recovered Image]</td>
<td>512</td>
</tr>
<tr>
<td>Our Scheme</td>
<td></td>
<td>27.5</td>
<td>![Recovered Image]</td>
<td>256</td>
</tr>
<tr>
<td>Chang et al.’s</td>
<td>$(3 \times 512)^2$/37.5%</td>
<td>23.8</td>
<td>![Recovered Image]</td>
<td>512</td>
</tr>
<tr>
<td>Our Scheme</td>
<td></td>
<td>26.1</td>
<td>![Recovered Image]</td>
<td>192</td>
</tr>
<tr>
<td>Chang et al.’s</td>
<td>$(2 \times 512)^2$/25%</td>
<td>17.8</td>
<td>![Recovered Image]</td>
<td>512</td>
</tr>
<tr>
<td>Our Scheme</td>
<td></td>
<td>24.1</td>
<td>![Recovered Image]</td>
<td>128</td>
</tr>
<tr>
<td>Chang et al.’s</td>
<td>$(512)^2$/12.5%</td>
<td>12.0</td>
<td>![Recovered Image]</td>
<td>512</td>
</tr>
<tr>
<td>Our Scheme</td>
<td></td>
<td>21.2</td>
<td>![Recovered Image]</td>
<td>64</td>
</tr>
</tbody>
</table>

6. CONCLUSIONS

In this paper, we proposed two image encryption schemes in frequency domain. In the first scheme, we scramble the positions of the compressed DCT coefficients. In the second scheme, we scramble the positions of the compressed DCT coefficients and diffuse their values. Due to the advantages of DCT transform, we were able to obtain higher robustness and better image quality at lower storage rates. Therefore, our schemes are more applicable for smart cards.

REFERENCES

1. C. Guo, C. C. Chang, and C. Y. Sun, “Chaotic maps-based mutual authentication and


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.

Hsiao-Ling Wu received the BS degree in applied mathematics from Feng Chia University, Taichung, Taiwan in 2007. She is currently pursuing her Ph.D. degree in information engineering and computer science from Feng Chia University, Taichung, Taiwan. Her current research interests include electronic commerce, information security, cryptography, and mobile communications.
Zhi-Hui Wang received the BS degree in software engineering in 2004 from the North Eastern University, Shenyang, China. She received her MS degree in software engineering in 2007 and the PhD degree in software and theory of computer in 2010, both from the Dalian University of Technology, Dalian, China. Since November 2011, she has been a visiting scholar of University of Washington. Her current research interests include information hiding and image compression.

Qian Mao was born in Shanxi Province, China, in 1978. She received the B.S. degree in Mechanical Engineering and Automation Science from Nanjing University of Aeronautics and Astronautics, Jiangsu, China, in 2000, and M.E. degrees in Traffic Information Engineering and Control from Shanghai Ship and Shipping Research Institute, Shanghai, China, in 2003, and the Ph.D. degree in Traffic Information Engineering and Control from Tongji University, Shanghai, China, in 2006. Since 2006, she has been with the faculty of the School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, where she is currently a Lecturer. She is also a post-doctoral researcher of Asia University, Taiwan. Her research interests include information security, image processing, and information theory and coding.