A General Cryptanalysis of Permutation-Only Multimedia Encryption Algorithms

Shujun Li, Chengqing Li, Guanrong Chen, *Fellow, IEEE*, Dan Zhang and Nikolaos G. Bourbakis *Fellow, IEEE*

***Abstract*— Secrete permutations are widely used in multimedia encryption algorithms, and many permutation-only ciphers have been proposed in recent years for protection of different types of multimedia data, especially digital images and videos. Based on a normalized encryption/decryption model, this paper analyzes the security of permutation-only image ciphers working in spatial domain from a general perspective, taking a recently-proposed permutation-only image cipher called HCIE (hierarchical chaotic image encryption) as a typical example. It is pointed out that all permutation-only image ciphers are insecure against known/chosen-plaintext attacks in the sense that only $O\left(\log_L(MN)\right)$ known/chosen plain-images are enough to break the ciphers, where MN is the size of the image and L is the number of different pixel values. Also, it is found that the attack complexity is only $O(n \cdot (MN)^2)$, where n is the number of known/chosen plain-images used. Following the results, it is found that hierarchical permutation-only image ciphers such as HCIE are less secure than normal (i.e., non-hierarchical) permutation-only image ciphers. Experiments are shown to verify the feasibility of the known/chosen-plaintext attacks. The cryptanalysis result is then generalized to permutation-only image ciphers working in frequency domain and video ciphers. In conclusion, secret permutations have to be combined with other encryption techniques to design highly secure multimedia encryption systems. This paper provides for the first time theoretically and experimentally clarification of such a security principle on the design of multimedia encryption algorithms.**

***Index Terms*— permutation-only multimedia encryption, image, video, cryptanalysis, known-plaintext attack, chosen-plaintext attack, HCIE, chaos**

I. Introduction

With the rapid progress of digital network technologies today, the security of multimedia becomes more and more important, since multimedia data are transmitted over open networks more and more frequently. Typically, reliable security in storage and transmission of digital images and videos is needed in many real applications, such as pay-TV, medical imaging systems, military image databases as well as confidential video conferences. In recent years, some consumer electronic devices, such as mobile phones, have also started to provide the function of saving and exchanging digital images and video clips under the support of multimedia messaging services over wireless networks, which is another reason for demanding the security of digital images and videos.

To meet the above needs in practice, some encryption algorithms are required to offer a sufficient level of security for different multimedia applications. Apparently, the simplest way to encrypt an image (or a video) is to consider the 2-D image (or 3-D video) stream as a 1-D signal, and then to encrypt the 1-D signal with any available cipher [1]. In some multimedia applications, such a simple way may be

This research was partially supported by the AIIS Inc., NY, USA, the Applied R&D Centers of the City University of Hong Kong (under grants no. 9410011 and no. 9620004), and the National Natural Science Foundation of China (under grant no. 60202002).

Shujun Li and Guanrong Chen are with the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong, China. E-mails: hooklee@mail.com, eegchen@cityu.edu.hk.

Chengqing Li is with the Department of Mathematics, Zhejiang University, Hangzhou, Zhejiang 310027, China. E-mail: swiftsheep@hotmail.com.

Dan Zhang is with the College of Computer Science, Zhejiang University, Hangzhou, Zhejiang 310027, China.

Nikolaos G. Bourbakis is with the Information Technology Research Institute, College of Engineering and Computer Science, Wright State University, 3640 Glenn Hwy, Dayton, OH 45435, USA.

enough. However, in many applications, encryption schemes considering special features of digital images, such as bulky size and large redundancy in uncompressed images, are still required to achieve a better overall performance and to make the integration of the encryption scheme into the image processing easier. Since the 1990s, many different algorithms have been proposed to provide solutions to image encryption [2]–[24] and video encryption [19]–[36]. Meanwhile, cryptanalysis work has also been developed and some proposed image/video encryption schemes have been found to be insecure from the cryptographical point of view [20]–[22], [37]–[51]. For a comprehensive survey with detailed discussions on image and video encryption algorithms, see [52].

In image encryption community, secret permutations are widely used to shuffle the positions of pixels, which is an effective and easy way to make the cipher-image look “chaotic”. Similarly, in video encryption community, secret permutations are widely used to shuffle the DCT/wavelet coefficients, blocks or macroblocks. There are many image/video encryption algorithms that are based only on secret permutations [2]–[6], [9]–[14], [16], [22], [25], [27], [33], [35], which are called *permutation-only* (image/video) ciphers in this paper. The main advantages of using only secrete permutations in a cipher include easy implementation and the universality for most image/video formats (which work well in both spatial and frequency domains).

In most *permutation-only* ciphers, the security is analyzed for ciphertext-only attacks, i.e., brute-force attacks of exhaustively searching the secret key. However, from the cryptographical point of view, such a security analysis is not enough, since there exist other more powerful attacks, such as known/chosen-plaintext attacks and chosen-ciphertext attacks (see the next section for a brief introduction to different kinds of cryptographical attacks). In fact, it has been pointed out that permutation-only image/video ciphers are not secure against known/chosen-plaintext attacks [20]–[22], [39]–[45], but most previous cryptanalysis results are proposed for some specific permutation-only image/video ciphers and the essential security defects of these encryption algorithms that are based only on secret permutations have not been clarified in a general way. The lack of general cryptanalysis results makes it ambiguous in understanding whether or not the security of permutation-only multimedia encryption algorithms can be effectively enhanced by designing new methods to generate better secret permutations.

This paper gives a general cryptanalysis of most (if not all) permutation-only multimedia encryption algorithms, mainly focusing on the quantitative relation between the breaking performance and the number of required known/chosen plaintexts, as well as the estimation of the attack complexity. It will be pointed out that secret permutations alone cannot provide sufficient security against known/chosen-plaintext attacks, from both theoretical and experimental points of view. The cryptanalysis is performed on a general model of permutation-only image ciphers working in spatial domain, which then is generalized to permutation-only image working in frequency domain and also to permutation-only video ciphers. As a typical example of permutation-only image ciphers, a recently-proposed image encryption scheme called HCIE (hierarchical chaotic image encryption) [10]–[12][1] is investigated in detail, to show how the known/chosen-plaintext attacks work. It is shown that all permutation-only image ciphers are not secure

[1] The chaotic image encryption (CIE) scheme proposed in [9] is an initial version of HCIE.

against known/chosen-plaintext attacks, in the sense that only $O\left(\log_L(MN)\right)$ known/chosen plain-images are enough to break the ciphers, where MN is the size of the image (i.e., the number of pixels) and L is the number of different pixel values. An upper bound of the attack complexity is also derived to be $O(n\cdot(MN)^2)$, where n is the number of known/chosen plain-images. What's more, it is found that the hierarchical encryption structure suggested in HCIE cannot provide any higher security against known/chosen-plaintext attacks, but actually make the security weaker. As a conclusion, secure permutations must be used together with other encryption mechanisms to design a secure multimedia encryption scheme, as in some compound image/video ciphers [7], [8], [17], [18], [23], [24].

The rest of this paper is organized as follows. Section II gives some background knowledge of cryptography and cryptanalysis. In Sec. III, a normalized encryption/decryption model of permutation-only image ciphers working in spatial domain is described, and then HCIE is briefly introduced as a typical example to show how the secret pixel permutations are realized. Cryptanalysis on common permutation-only image ciphers and one special version, HCIE, are studied in detail in Sec. IV. Some experiments are shown in Sec. V to support the cryptanalysis. Sec. VI discusses the generalization of the cryptanalysis results to permutation-only image ciphers working in frequency domain and to permutation-only video ciphers. The last section concludes the paper.

II. Preliminaries of Cryptography and Cryptanalysis

To facilitate the following discussion, this section gives a brief introduction to the basic theory of modern cryptography and cryptanalysis, which compose the technology of encryption — cryptology [1]. Simply speaking, cryptography studies how to design good (secure and fast) encryption algorithms, and cryptanalysis tries to find security weaknesses of existing algorithms and studies whether or not they are vulnerable to some attacks.

An encryption scheme is called a *cipher*, or a *cryptosystem*. The message for encryption is called *plaintext*, and the encrypted message is called *ciphertext*, which are denoted here by P and C, respectively. The encryption procedure of a cipher can be described as $C = E_{K_e}(P)$, where K_e is the encryption key and $E(\cdot)$ is the encryption function. Similarly, the decryption procedure is $P = D_{K_d}(C)$, where K_d is the decryption key and $D(\cdot)$ is the decryption function. When $K_e = K_d$, the cipher is called a *private-key* cipher or a *symmetric* cipher. For private-key ciphers, the encryption-decryption key must be transmitted from the sender to the receiver via a separate secret channel. When $K_e \neq K_d$, the cipher is called a *public-key* cipher or an *asymmetric* cipher. For public-key ciphers, the encryption key K_e is published, and the decryption key K_d is kept secret, for which no additional secret channel is needed for key transfer.

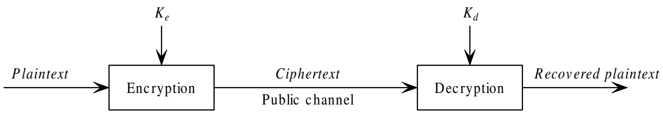


Fig. 1. The encryption and decryption procedures of a cipher.

Following the widely-acknowledged Kerckhoffs' principle in the cryptology community [1], it is assumed that all details of the encryption/decryption algorithms are known to attackers. This means that the security of a cipher relies on the decryption key K_d only. Thus, the main task of cryptanalysis is to reconstruct the key, or its equivalent form that can successfully decrypt all or partial contents of any plaintext encrypted by the cipher.

From the cryptographical point of view, a cryptographically strong cipher should be secure enough against all kinds of attacks. For most ciphers, the following four attacks corresponding to different scenarios should be checked:

- *the ciphertext-only attack* - attackers can only observe part of the ciphertexts;
- *the known-plaintext attack* - attackers can get some plaintexts and the corresponding ciphertexts;
- *the chosen-plaintext attack* - attackers can choose some plaintexts and get the corresponding ciphertexts;
- *the chosen-ciphertext attack* - attackers can choose some ciphertexts and get the corresponding plaintexts.

The last two attacks, which seem to seldom occur in practice, are feasible in some real applications [1, Sec. 1.1.7] and become more and more common in the digital world today. This paper mainly focuses on known-plaintext and chosen-plaintext attacks.

III. Permutation-Only Image Ciphers Working in Spatial Domain

A. A normalized model for encryption and decryption

When working in spatial domain, just as its name implies, *permutation-only* image ciphers encrypt images by permuting the positions of all pixels in a secret way. The secret permutations have to be invertible to make the decryption possible. This means that all permutation-only ciphers belong to symmetry ciphers, i.e., $K_e = K_d = K$, which is used to generate the secret permutations. Although many different methods have been proposed to realize secret key-dependent pixel permutations, for a given plain-image of size $M \times N$ ("height×width"), a permutation-only image cipher can be normalized with an *invertible key-dependent permutation matrix of size* $M \times N$, denoted by

$$\boldsymbol{W} = [w(i,j) = (i',j') \in \mathbb{M} \times \mathbb{N}]_{M\times N}\,, \tag{1}$$

where $\mathbb{M} = \{0, \cdots, M-1\}$ and $\mathbb{N} = \{0, \cdots, N-1\}$. With the permutation matrix $\boldsymbol{W}$ and its inverse $\boldsymbol{W}^{-1} = [w^{-1}(i,j)]_{M\times N}$, for a plain-image $f = [f(i,j)]_{M\times N}$ and its corresponding cipher-image $f' = [f'(i,j)]_{M\times N}$, the encryption and decryption procedures of a permutation-only image cipher can always be described as follows:

- *the encryption procedure*: for $i = 0 \sim (M-1)$ and $j = 0 \sim (N-1)$, $f'(w(i,j)) = f(i,j)$;
- *the decryption procedure*: for $i = 0 \sim (M-1)$ and $j = 0 \sim (N-1)$, $f(w^{-1}(i,j)) = f'(i,j)$.

In a short form, one can express the encryption procedure as $f'(\boldsymbol{W}(\boldsymbol{I})) = f(\boldsymbol{I})$ and the decryption procedure as $f(\boldsymbol{W}^{-1}(\boldsymbol{I})) = f'(\boldsymbol{I})$, where

$$\boldsymbol{I} = \begin{bmatrix} (0,0) & \cdots & (0,N-1) \\ \vdots & \ddots & \vdots \\ (M-1,0) & \cdots & (M-1,N-1) \end{bmatrix}_{M\times N}.$$

To ensure the invertibility of the permutation matrix, i.e., to make the decryption possible, the following should be satisfied: $\forall (i_1,j_1) \neq (i_2,j_2)$, $w(i_1,j_1) \neq w(i_2,j_2)$. This means that $\boldsymbol{W}$ determines a bijective (i.e., one-to-one) permutation mapping, $F_{\boldsymbol{W}} : \mathbb{M} \times \mathbb{N} \to \mathbb{M} \times \mathbb{N}$.

From the above description, one can see that the design of a permutation-only image cipher focuses on two points: 1) what the secret key K is; 2) how the permutation matrix $\boldsymbol{W}$ and $\boldsymbol{W}^{-1}$ are derived from the secret key K. Generally speaking, each key defines a permutation matrix, and each permutation-only image cipher defines a finite set containing a number of permutation matrices selected from $(MN)!$ possible permutation matrices. In literature, many different methods have been proposed to derive a permutation matrix from a key, some of which are listed as follows:

- *SCAN language* based methods [2]–[4], [6], [23], [24]: define some different scan patterns of the 2-D image and combine these patterns to define a permutation matrix by scanning the whole image pixel by pixel;
- *quadtree* based methods [4], [6]: divide the image into multi-level quadtree and shuffle the order of four nodes in each level to realize a permutation matrix;
- *2-D chaotic maps* based methods [7], [8], [17], [18]: iterate a discretized 2-D chaotic map over the $M \times N$ image lattice for many times to realize a permutation matrix;
- *Fractal curves* based methods [5], [16]: use a fractal(-like) curve to replace the normal scan order to realize a permutation matrix;
- *pseudo-random rotations* based methods [9]–[12]: pseudo-randomly rotate pixels along some straight lines for many times to realize a permutation matrix;
- *matrix transformation based methods* [13]: use (integer) transformations of matrix, such as n-dimensional Arnold transformation and Fabonacci-Q transformation, to define permutation matrices;
- *composite methods* [14]: combine different methods to realize more complicated permutation matrices.

Although different types of secret keys are used in different permutation-only image ciphers to generate the permutation matrix, it is reasonable to consider the permutation matrix $\boldsymbol{W}$ itself as the equivalent encryption key and $\boldsymbol{W}^{-1}$ as the equivalent decryption key. From such a point of view, all permutation-only image ciphers can be considered the same. This is the base for the security analysis to be carried out below in this paper.

B. A typical permutation-only image cipher – HCIE [9]–[12]

HCIE is a two-level hierarchical permutation-only image cipher, and all involved permutation matrices are defined by pseudo-random combinations of four rotation mappings with pseudo-random parameters. For an image, $f = [f(i,j)]_{M\times N}$, the four mapping operations are described as follows, where $p < \min(M,N)$ holds for each mapping.

Definition 1: The mapping $f' = ROLR_b^{i,p}(f)$ $(0 \leq i \leq M-1)$ is defined to rotate the i-th row of f, in the left (when $b=0$) or right (when $b=1$) direction by p pixels.

Definition 2: The mapping $f' = ROUD_b^{j,p}(f)$ $(0 \leq j \leq N-1)$ is defined to rotate the j-th column of f, in the up (when $b=0$) or down (when $b=1$) direction by p pixels.

Definition 3: The mapping $f' = ROUR_b^{k,p}(f)$ $(0 \leq k \leq M+N-2)$ is defined to rotate all pixels satisfying $i+j=k$, in the lower-left (when $b=0$) or upper-right (when $b=1$) direction by p pixels.

Definition 4: The mapping $f' = ROUL_b^{l,p}(f)$ $(1-N \leq l \leq M-1)$ is defined to rotate all pixels satisfying $i-j=l$, in the upper-left (when $b=0$) or lower-right (when $b=1$) direction by p pixels.

Given a pseudo-random bit sequence $\{b(i)\}$ starting from i_0, the following `Sub_HCIE` function is used to permute an $S_M \times S_N$ image f_{sub} to be another $S_M \times S_N$ image f'_{sub}, where $(\alpha, \beta, \gamma, no)$ are control parameters. Note that all codes in this paper is described in C-language style.

```
for ($ite = 0$; $ite < no$; $ite++$) {
    $q = i_0 + (3S_M + 3S_N - 2) \times ite$;
    $p = \alpha + \beta \times b(q+0) + \gamma \times b(q+1)$;
    for ($i = 0$; $i \leq (S_M - 1)$; $i++$)
        $f'_{sub} = ROLR^{i,p}_{b(i+q)}(f_{sub})$;
        for ($j = 0$; $j \le (S_N - 1)$; $j++$)
            $f'_{sub} = ROUD^{j,p}_{b(j+q+S_M)}(f'_{sub})$;
        for ($k = 0$; $k \le (S_M + S_N - 2)$; $k++$)
            $f'_{sub} = ROUR^{k,p}_{b(k+q+S_M+S_N)}(f'_{sub})$;
        for ($l = (1 - S_N)$; $l \le (S_M - 1)$; $l++$)
            $f'_{sub} = ROUL^{l,p}_{b(l+q+2\times S_M+3\times S_N-2)}(f'_{sub})$;
    }
    $i_0 = i_0 + (3S_M + 3S_N - 2) \times no$;
```

One can see that the above Sub_HCIE function actually defines an $S_M \times S_N$ permutation matrix pseudo-randomly controlled by $(3S_M + 3S_N - 2) \times no$ bits in the bit sequence $\{b(i)\}$ from i_0. Based on this function, for an $M \times N$ image $f = [f(i,j)]_{M \times N}$, the encryption procedure of HCIE can be briefly described in two levels.

- *The secret key* is the initial condition $x(0)$ and the control parameter μ of the chaotic Logistic map, $f(x) = \mu x(1-x)$ [53], which is realized in L-bit finite precision.
- *Some public parameters*: S_M, S_N, α, β, γ and no, where $\sqrt{M} \le S_M \le M$, $M \bmod S_M = 0$, $\sqrt{N} \le S_N \le N$, and $N \bmod S_N = 0$.

  Note: *Although ($S_M, S_N, \alpha, \beta, \gamma, no$) can be all included in the secret key, they are not suitable for such a use due to the following reasons: 1) S_M, S_N are related to M, N; 2) α, β, γ are related to S_M, S_N (and then related to M, N, too); 3) S_M, S_N can be easily guessed from the mosaic effect of the cipher-image; 4) no cannot be too large to achieve an acceptable encryption speed.*
- *The initialization procedure* of generating the bit sequence used in the Sub_HCIE function: run the Logistic map from $x(0)$ to generate a chaotic sequence $\{x(i)\}_{i=0}^{\lceil L_b/8 \rceil - 1}$, and then extract the 8 bits following the decimal point of each chaotic state $x(i)$ to yield a bit sequence $\{b(i)\}_{i=0}^{L_b-1}$, where $L_b = \left(1 + \frac{M}{S_M} \cdot \frac{N}{S_N}\right) \cdot (3S_M + 3S_N - 2) \cdot no$; finally, set $i_0 = 0$ to let the Sub_HCIE function run from $b(0)$.
- *The two-level hierarchical encryption procedure*:
  - *The high-level encryption – permuting image blocks*: divide the plain-image f into blocks of size $S_M \times S_N$, which compose an $\frac{M}{S_M} \times \frac{N}{S_N}$ block-image $P_f = [P_f(i,j)]_{\frac{M}{S_M} \times \frac{N}{S_N}}$, where $P_f(i,j)$ is the block of size $S_M \times S_N$ at the position (i,j). Then, permute the positions of all blocks with the Sub_HCIE function in the following way:
    - create a pseudo-image $f_p = [f_p(i,j)]_{S_M \times S_N}$ containing $\left(\frac{M}{S_M} \cdot \frac{N}{S_N}\right)$ non-zero indices of all image blocks in P_f and $\left(M \cdot N - \frac{M}{S_M} \cdot \frac{N}{S_N}\right)$ zero-elements, and permute f_p with the Sub_HCIE function to get a shuffled pseudo-image f_p^*;
    - generate a permuted block-image P_{f^*} from P_f (i.e., permute f blockwise) using the shuffled indices contained in f_p^*.

    The above high-level encryption procedure can be considered as the permutation of the block-image: $P_f \xrightarrow{f_p^* = \texttt{Sub\_HCIE}(f_p)} P_{f^*}$, where f_p^* actually corresponds to an $\frac{M}{S_M} \times \frac{N}{S_N}$ permutation matrix.
  - *The low-level encryption – permuting pixels in each image block*: for $i = 0 \sim \left(\frac{M}{S_M} - 1\right)$ and $j = 0 \sim \left(\frac{N}{S_N} - 1\right)$, call the Sub_HCIE function to permute each block $P_{f^*}(i,j)$ to get the corresponding block of the cipher-image f': $P_{f'}(i,j) = \texttt{Sub\_HCIE}\,(P_{f^*}(i,j))$.

In HCIE, a total of $\left(1 + \frac{M}{S_M} \cdot \frac{N}{S_N}\right)$ permutation matrices are involved: 1) one high-level permutation matrix of size $\frac{M}{S_M} \times \frac{N}{S_N}$; 2) $\left(\frac{M}{S_M} \cdot \frac{N}{S_N}\right)$ low-level permutation matrices of size $S_M \times S_N$. With the above-mentioned representation of permutation-only image ciphers, the secret key $(\mu, x(0))$ of HCIE is equivalent to the $\left(1+\frac{M}{S_M}\cdot\frac{N}{S_N}\right)$ permutation matrices. To facilitate the following discussions, we use $\boldsymbol{W}_0 = [w_0(i,j)]_{\frac{M}{S_M}\times\frac{N}{S_N}}$ to denote the high-level permutation matrix, and use $\left\{\boldsymbol{W}_{(i,j)}\right\}_{i=0,j=0}^{\frac{M}{S_M}-1,\frac{N}{S_N}-1}$ to denote the $\left(\frac{M}{S_M}\times\frac{N}{S_N}\right)$ low-level permutation matrices, where $\boldsymbol{W}_{(i,j)} = \left[w_{(i,j)}(i',j')\right]_{S_M\times S_N}$. Apparently, the $\left(1+\frac{M}{S_M}\cdot\frac{N}{S_N}\right)$ permutation matrices can be easily transformed to an equivalent permutation matrix of size $M\times N$: $\boldsymbol{W} = [w(i,j)]_{M\times N}$.

When $S_M = M$ and $S_N = N$ (or $S_M = S_N = 1$), the two hierarchical encryption levels merge a single layer; the $\left(1+\frac{M}{S_M}\cdot\frac{N}{S_N}\right)$ permutation matrices become one permutation matrix of size $M\times N$; and HCIE is simplified to be CIE [9] – a typical permutation-only image cipher in which each pixel can be freely permuted to be any other positions in the whole image by a single $M\times N$ permutation matrix $\boldsymbol{W}$.

IV. Cryptanalysis of Permutation-Only Image Ciphers Working in Spatial Domain

In this section, we discuss the known/chosen-plaintext attacks to the above-normalized permutation-only image ciphers working in spatial domain and the typical example – HCIE. Also, we will point out in passing that the security of HCIE against brute-force attacks was much over-estimated in [10]–[12]. Note that HCIE has not been cryptanalyzed yet till now.

A. Cryptanalysis of general permutation-only image ciphers working in spatial domain

1) The known-plaintext attack: As shown above, when a *permutation-only* image cipher is used to encrypt images in spatial domain, a pixel at the position (i,j) will be secretly permuted to another fixed position (i',j') while the pixel value is unchanged. Therefore, by comparing a number of known plain-images and the corresponding cipher-images, it is possible for an attacker to (partially or even totally) reconstruct the secret permutations of all pixels, i.e., to derive the encryption/decryption keys – the permutation matrix $\boldsymbol{W}$ and its inverse $\boldsymbol{W}^{-1}$.

Given n known plain-images $f_1 \sim f_n$ and their cipher-images $f_1' \sim f_n'$, the deduction procedure of $\boldsymbol{W}$ and $\boldsymbol{W}^{-1}$ can be shown in the following `Get_Permutation_Matrix` function. With the input parameters ($f_1\sim f_n, f_1'\sim f_n', M, N$), this function returns an estimation of the permutation matrix $\boldsymbol{W}$ and its inverse $\boldsymbol{W}^{-1}$. Assuming the value of each pixel ranges in $\{0,\cdots,L-1\}$, the `Get_Permutation_Matrix` function is described as follows.

- *Step 1: compare pixel values within the n cipher-images $f_1' \sim f_n'$ to get $(n\cdot L)$ sets of pixel positions*:
$$\Lambda_1'(0)\sim\Lambda_1'(L-1),\cdots,\Lambda_n'(0)\sim\Lambda_n'(L-1),$$
where $\Lambda_m'(l)\subseteq\mathbb{M}\times\mathbb{N}$ denotes a set containing positions of all pixels in f_m' $(m=1\sim n)$ whose values are equal to $l\in\{0,\cdots,L-1\}$, i.e., $\forall(i',j')\in\Lambda_m'(l)$, $f_m'(i',j')=l$. Note that $\Lambda_m'(0)\sim\Lambda_m'(L-1)$ actually compose a partition of the set of all pixel positions: $\bigcup_{l=0}^{L-1}\Lambda_m'(l) = \mathbb{M}\times\mathbb{N} = \{(0,0),\cdots,(M-1,N-1)\}$, and $\forall l_1\neq l_2$, $\Lambda_m'(l_1)\cap\Lambda_m'(l_2)=\varnothing$;
- *Step 2: get a multi-valued permutation matrix, $\widehat{\boldsymbol{W}} = [\widehat{\boldsymbol{w}}(i,j)]_{M\times N}$,* where $\widehat{\boldsymbol{w}}(i,j) = \bigcap_{m=1}^{n}\Lambda_m'(f_m(i,j))$. Here, note that $\widehat{\boldsymbol{w}}(0,0)\sim\widehat{\boldsymbol{w}}(M-1,N-1)$ actually composes a new partition of the position set $\mathbb{M}\times\mathbb{N}$;
- *Step 3: determine a single-valued permutation matrix, $\widetilde{\boldsymbol{W}} = [\widetilde{w}(i,j)]_{M\times N}$ from $\widehat{\boldsymbol{W}}$,* where $\widetilde{w}(i,j)\in\widehat{\boldsymbol{w}}(i,j)$ and $\forall(i_1,j_1)\neq(i_2,j_2)$, $\widetilde{w}(i_1,j_1)\neq\widetilde{w}(i_2,j_2)$;
- *Step 4: output $\widetilde{\boldsymbol{W}}$ and its inverse $\widetilde{\boldsymbol{W}}^{-1} =$*

$[\widetilde{w}^{-1}(i,j)]_{M\times N}$ *as the estimations of* $\boldsymbol{W}$ *and* $\boldsymbol{W}^{-1}$.

Apparently, if and only if $\#(\widehat{\boldsymbol{w}}(0,0)) = \cdots = \#(\widehat{\boldsymbol{w}}(S_M-1,S_N-1)) = 1$, i.e., each element of $\widehat{\boldsymbol{W}}$ contains only one pixel position, it is true that $\widetilde{\boldsymbol{W}} = \boldsymbol{W}$ and the cipher is totally broken. However, because some elements of $\widehat{\boldsymbol{W}}$ contain more than one pixel position, generally $\widetilde{\boldsymbol{W}}$ is not an exact estimation of $\boldsymbol{W}$. Assume that there are $(\widehat{N} \leq MN)$ different elements in $\widehat{\boldsymbol{W}}$, and that the $\widehat{N}$ different elements are $\widehat{\boldsymbol{w}}_1 \sim \widehat{\boldsymbol{w}}_{\widehat{N}}$. Then, it can be easily verified that there are $\prod_{k=1}^{\widehat{N}} \#(\widehat{\boldsymbol{w}}_k)!$ possibilities of $\widetilde{\boldsymbol{W}}$. To make the estimation of $\widetilde{\boldsymbol{W}}$ as accurate as possible, some specific optimization algorithms can be used to choose a better position from $\widehat{\boldsymbol{w}}(i,j)$ as the value of $\widetilde{w}(i,j)$, such as genetic and simulated annealing algorithms. Our experiments show that even a simple algorithm can achieve a rather good estimation when $n \geq 3$ for 256×256 gray-scale images. The simple algorithm is called "taking-the-first" algorithm, which sets $\widetilde{w}(i,j)$ to be the first available element in $\widehat{\boldsymbol{w}}(i,j)$, where the term "available" refers to the constraint that $\forall (i_1,j_1) \neq (i_2,j_2)$, $\widetilde{w}(i_1,j_1) \neq \widetilde{w}(i_2,j_2)$.

Now, let us consider the decryption performance of the estimated permutation matrix $\widetilde{\boldsymbol{W}}$ when $\widetilde{\boldsymbol{W}} \neq \boldsymbol{W}$. Generally speaking, due to the large information redundancy existing in a digital image, only partially-recovered pixels are enough to reveal most visual information. Therefore, if there are enough correct elements in $\widetilde{\boldsymbol{W}}$, the decryption performance may be acceptable from a practical point of view. From the above discussions, one can see that correctly-recovered elements in $\widetilde{\boldsymbol{W}}$ belong to two different classes:

- *the absolutely correct elements*: derived from the single-valued elements of $\widehat{\boldsymbol{W}}$;
- *the probabilistically correct elements*: derived from the multi-valued elements of $\widehat{\boldsymbol{W}}$, and are correctly guessed by an optimization algorithm of selecting a proper position from each $\widehat{\boldsymbol{w}}(i,j)$.

Assuming that the number of single-valued elements of $\widehat{\boldsymbol{W}}$ is n_c and the successful probability of the optimization algorithm is p_s, the average number of correct elements in $\widetilde{\boldsymbol{W}}$ will be $n_c + p_s \cdot (MN - n_c)$. Because p_s is generally not fixed (tightly dependent on the employed optimization algorithm), only the absolutely correct elements are considered here (i.e., $p_s = 0$ is assumed) to perform a qualitative analysis. Now the problem of correct elements in $\widetilde{\boldsymbol{W}}$ is simplified to be the problem of singe-value elements in $\widehat{\boldsymbol{W}}$. Observing the `Get_Permutation_Matrix` function, one can see that the cardinality of $\widehat{\boldsymbol{w}}(i,j)$ is uniquely determined by $\Lambda_1'(f_1(i,j)) \sim \Lambda_n'(f_n(i,j))$. To further simplify the analysis, assume that the value of each pixel distributes uniformly in $\{0,\cdots,L-1\}$, and that the values of any two pixels (within the same image or in two different cipher-images[2]) are independent of each other. Then, one can consider the following two types of positions in $\widehat{\boldsymbol{w}}(i,j)$:

- *the only one real position* $w(i,j)$, which absolutely occurs in $\widehat{\boldsymbol{w}}(i,j)$;
- *other fake positions*, each of which occurs in each $\Lambda_m'(f_m(i,j))$ with a probability of $\frac{1}{L}$, i.e., each of which occurs in all the n sets, $\Lambda_1'(f_1(i,j)) \sim \Lambda_n'(f_n(i,j))$, with a probability of $\frac{1}{L^n}$.

Based on the above results, one can qualitatively deduce that the average cardinality of $\widehat{\boldsymbol{w}}(i,j)$ is $\overline{\#(\widehat{\boldsymbol{w}}(i,j))} = \left(1 + \frac{MN-1}{L^n}\right)$, which approaches 1 exponentially as n increases. Generally speaking, when $1 + \frac{MN-1}{L^n} < 1.5$, i.e.,

[2] Note that a plain-image and its cipher-image are totally related via the secret permutation matrix.

about half elements in $\widetilde{\boldsymbol{W}}$ are correct, the decryption performance will be acceptable. Solving this inequality, one has $n \geq \lceil \log_L(2(MN-1)) \rceil$. As an example, for 256×256 gray-scale images, $M = N = L = 256$, one has $n \geq \lceil \log_L(2(MN-1)) \rceil = \lceil 2.125 \rceil = 3$. The average cardinality is about 1.0039 when $n = 3$, so it is expected that the decryption performance for $n \geq 3$ will be rather good, which is verified by the experiments given in the next section. Here, note that the actual decryption performance is generally better than the above theoretical expectation for the following two reasons:

- human eyes have a powerful capability of suppressing image noises and extracting significant features: 10% noisy pixels cannot make much influence on the visual quality of a digital image, and it only needs 50% of pixels to reveal most visual information of the original image;
- due to the short-distance and long-distance relationships in natural images, two pixel values are close to each other with a non-negligible probability larger than the average probability; as a result, the wrongly-decrypted pixel are close to the right value with a probability larger than the average probability.

The second point implies that the decryption performance of natural images will be better than the performance of noise-like images, from the point of view of decryption error ratio. For experimental verification and more explanations, see Sec. V-A, Figs. 4 and 5.

Next, let us consider the time complexity of the above-discussed known-plaintext attack, i.e., the time complexity of the `Get_Permutation_Matrix` function. Note that the time complexity depends on the implementation details of this function. This paper only gives a conservative estimation, i.e., an upper bound, of the time complexity. The time complexity of each step is as follows:

- *Step 1*: The L sets of each cipher-image f'_l are obtained by scanning f'_l once: for $i = 0 \sim (M-1)$ and $j = 0 \sim (N-1)$, add (i,j) into the set $\Lambda'_m(f'_l(i,j))$. Thus, the time complexity of this step is $O(n \cdot MN)$.
- *Step 2*: Without loss of generality, assume all cipher-pixels satisfy uniform distributions. Then, the average cardinality of $\Lambda_m(l)$ is $\frac{MN}{L}$ and an upper bound of the time complexity of this step is $MN \cdot \left(\frac{MN}{L} \cdot \left(\frac{1}{2} \cdot \frac{MN}{L} \right)^{n-1} \right) = 2MN \cdot \left(\frac{MN}{2L} \right)^n$, which exponentially increase as n increases if $MN > 2L$. However, in practice, the real complexity is much smaller due to the optimization of the calculation process. Here, we consider the so-called halving algorithm, which calculates the intersection of n sets $A_1 \sim A_n$ by dividing them into multi-level groups of $(2, 4, \cdots, 2^i, \cdots)$ sets. For example, when $n = 11$, the calculation process is described by

$$((A_1 \overset{1}{\cap} A_2) \overset{3}{\cap} (A_3 \overset{2}{\cap} A_4)) \overset{7}{\cap} ((A_5 \overset{4}{\cap} A_6) \overset{6}{\cap} (A_7 \overset{5}{\cap} A_8)) \overset{10}{\cap} ((A_9 \overset{8}{\cap} A_{10}) \overset{9}{\cap} A_{11}), \tag{2}$$

where $\overset{i}{\cap}$ denotes the i-th intersection operation. The goal of this halving algorithm is to minimize the cardinalities of the two sets involved in each intersection operation so as to reduce the global complexity. To make the estimation of the complexity easier, let us consider the case of $n = 2^d$, where d is an integer. In this case, the overall complexity is shown in Eq. (3). As two typical examples, when $M = N = 256$ and $L = 2$ (monotonic images), the complexity is about $\left(2^{29.2} \cdot n\right)$; when $M = N = 256$ and $L = 256$ (gray-scale images), the complexity is only $\left(2^{15} \cdot n\right)$. One can see that now the complexity is always much smaller than $2MN \cdot \left(\frac{MN}{L}\right)^n$. When n is not a power of 2, the complexity will be

$$\begin{aligned}\sum_{k=d-1}^{0} 2^k \cdot \left(\frac{MN}{L^{d-k}}\right)^2 &= \sum_{k'=1}^{d} 2^{d-k'} \cdot \left(\frac{MN}{L^{k'}}\right)^2 = 2^d \cdot (MN)^2 \cdot \left(\sum_{k'=0}^{d} \frac{1}{(2\cdot L^2)^{k'}} - 1\right)\\ &= n \cdot (MN)^2 \cdot \left(\frac{1-((2L^2)^{-1})^{d+1}}{1-(2L^2)^{-1}} - 1\right)\\ &< n \cdot (MN)^2 \cdot \left(\frac{1}{1-(2L^2)^{-1}} - 1\right) = \frac{n}{2L^2-1} \cdot (MN)^2. \end{aligned} \tag{3}$$

smaller than $\frac{2^{\lceil \log_2 n \rceil}}{2L^2-1} \cdot (MN)^2 \leq \frac{2n}{2L^2-1} \cdot (MN)^2$.

- *Step 3*: The time complexity of this step is determined by the details of the involved optimization algorithm. For the “taking-the-first” algorithm, the complexity is $MN \cdot \left(1 + \frac{MN-1}{L^n}\right) \approx MN + \frac{(MN)^2}{L^n}$.
- *Step 4*: The time complexity of this step is MN.

Combing the above discussions, the final time complexity of the `Get_Permutation_Matrix` function is always of order $n \cdot (MN)^2$, which is practically small even for a PC.

From the above analysis, one can see that the time complexity is mainly determined by Step 2. When the “taking-the-first” algorithm is adopted in the `Get_Permutation_Matrix` function, Step 2 can be skipped so that the total complexity will still be of order $O\left(n \cdot (MN)^2\right)$, even without using the halving algorithm to calculate the intersections. In this case, Step 3 can be described as follows:

- *Step 3’*: For $i = 0 \sim (M-1)$ and $j = 0 \sim (N-1)$, do the following operations:
  - *Step 3’a*: find the first element satisfying $f_1(i,j) = f_1'(i',j'), \cdots, f_n(i,j) = f_n'(i',j')$ by searching each element in $\Lambda_1'(f_1(i,j))$ and checking whether it occurs in $\Lambda_2'(f_2(i,j)) \sim \Lambda_n'(f_n(i,j))$;
  - *Step 3’b*: set $\widetilde{w}(i,j) = (i',j')$ and then delete (i',j') from $\Lambda_1'(f_1(i,j)) \sim \Lambda_n'(f_m(i,j))$.

It is obvious that the time complexity of Step 3’a is always less than $n \cdot (MN)$ and averagely is $O\left(n \cdot \frac{MN}{L}\right)$, so the time complexity of Step 3’ is always less than $n \cdot (MN)^2$ and averagely is $O\left(n \cdot \frac{(MN)^2}{L}\right)$.

2) The chosen-plaintext attack: The chosen-plaintext attack works in the same way as the known-plaintext attack, but the plain-images can be deliberately chosen to optimize the estimation of $\widetilde{\boldsymbol{W}}$ (i.e., to maximize the decryption performance). The following two rules are useful in the creation of the n chosen plain-images $f_1 \sim f_n$:

- the histogram of each chosen plain-image should be as uniform as possible;
- the i-dimensional ($2 \leq i \leq n$) histogram of any i chosen plain-images should be as uniform as possible, which is a generalization of the above rule.

The goal of the above two rules is to minimize the average cardinality of the elements in $\widehat{\boldsymbol{W}}$, and then to maximize the number of correct elements in the estimated permutation matrix $\widetilde{\boldsymbol{W}}$.

As an example of the two rules, consider the condition when $M = N = L = 256$ (256-valued gray-scale images of size 256×256). In this case, the following two chosen plain-images are enough to ensure a perfect estimation of the permutation matrix $\boldsymbol{W}$: $f_1 = [f_1(i,j) = i]_{256\times256}$ and $f_2 = [f_2(i,j) =$

$j]_{256\times 256}$, i.e.,

$$f_1 = f_2^T = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ i & \cdots & i \\ \vdots & \ddots & \vdots \\ 255 & \cdots & 255 \end{bmatrix}_{256\times 256} \tag{5}$$

and

$$f_2 = f_1^T = \begin{bmatrix} 0 & \cdots & j & \cdots & 255 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & j & \cdots & 255 \end{bmatrix}_{256\times 256} . \tag{6}$$

For the above two chosen plain-images, it is true that $\forall (i_1, j_1) \neq (i_2, j_2)$, $(f_1(i_1, j_1), f_1(i_2, j_2)) \neq (f_2(i_1, j_1), f_2(i_2, j_2))$. This can ensure that $\forall l_1, l_2 \in \{0, \cdots, L-1\}$, $\#\left(\Lambda_1'(l_1) \cap \Lambda_2'(l_2)\right) = 1$. For n images satisfying this constraint, we say that they compose *an orthogonal image set*. This concept is introduced to facilitate the following discussion on the chosen-plaintext attack to HCIE.

In general cases, it can be easily deduced that $n = \lceil \log_L(MN) \rceil$ orthogonal images[3] have to be created to carry out a successful chosen-plaintext attack. Apparently, it will never be larger than $\lceil \log_L(2(MN-1)) \rceil$ – the number of required plain-images in the known-plaintext attack with a good breaking performance (recall the above sub-subsection). This means the chosen-plaintext attack is a little (but not so much) stronger than the chosen-plaintext attack in the present case of discussion.

B. Cryptanalysis of HCIE

1) The known-plaintext attack: Since HCIE is a permutation-only image cipher, given n known plain-images $f_1 \sim f_n$ of size $M \times N$ and the corresponding cipher-images $f_1' \sim f_n'$, one can simply call the above `Get_Permutation_Matrix` function with the input parameter $(f_1 \sim f_n, f_1' \sim f_n', M, N)$ to estimate an $M \times N$ permutation matrix $\boldsymbol{W}$, which is equivalent to the $\left(1 + \frac{M}{S_M} \cdot \frac{N}{S_N}\right)$ smaller permutation matrices. However, if the hierarchical structure of HCIE is considered, the known-plaintext attack may be quicker and the estimation will be more effective, as demonstrated later in the next section. Thus, the following hierarchical procedure of known-plaintext attacks to HCIE is suggested[4]:

- *Reconstruct the high-level permutation matrix $\boldsymbol{W}_0$:*
  - for $i = 0 \sim \left(\frac{M}{S_M} - 1\right)$ and $j = 0 \sim \left(\frac{N}{S_N} - 1\right)$, calculate the mean values of the $2n$ blocks $P_{f_1}(i,j) \sim P_{f_n}(i,j)$, $P_{f_1'}(i,j) \sim P_{f_n'}(i,j)$ and denote them by $\overline{P_{f_1}(i,j)} \sim \overline{P_{f_n}(i,j)}$ and $\overline{P_{f_1'}(i,j)} \sim \overline{P_{f_n'}(i,j)}$;
  - generate $2n$ images $\overline{P}_{f_1} \sim \overline{P}_{f_n}$ and $\overline{P}_{f_1'} \sim \overline{P}_{f_n'}$ of size $\frac{M}{S_M} \times \frac{N}{S_N}$ as follows: $\forall m = 1 \sim n$,

    $$\overline{P}_{f_m} = \left[\overline{P_{f_m}(i,j)}\right]_{\frac{M}{S_M} \times \frac{N}{S_N}} \tag{7}$$

    and

    $$\overline{P}_{f_m'} = \left[\overline{P_{f_m'}(i,j)}\right]_{\frac{M}{S_M} \times \frac{N}{S_N}} , \tag{8}$$

    and call the `Get_Permutation_Matrix` function with the input parameters

    $$\left(\overline{P}_{f_1} \sim \overline{P}_{f_n}, \overline{P}_{f_1'} \sim \overline{P}_{f_n'}, \frac{M}{S_M}, \frac{N}{S_N}\right)$$

    to get an estimated permutation matrix $\widetilde{\boldsymbol{W}}_0 = [\widetilde{w}_0(i,j)]_{\frac{M}{S_M} \times \frac{N}{S_N}}$ and its inverse $\widetilde{\boldsymbol{W}}_0^{-1} = \left[\widetilde{w}_0^{-1}(i,j)\right]_{\frac{M}{S_M} \times \frac{N}{S_N}}$.

[3] When $MN \leq L$, only one chosen plain-image is enough, if each pixel value is different from the others.

[4] For HCIE, the permutation matrices also depend on the values of the public parameters. To simplify the following description, without loss of generality, it is assumed that all public parameters are fixed for all known plain-images.

- *Reconstruct the* $\left(\frac{M}{S_M}\cdot\frac{N}{S_N}\right)$ *low-level permutation matrices* $\left\{\boldsymbol{W}_{(i,j)}\right\}_{i=0,j=0}^{\frac{M}{S_M}-1,\frac{N}{S_N}-1}$:
  - for $i=0\sim\left(\frac{M}{S_M}-1\right)$ and $j=0\sim\left(\frac{N}{S_N}-1\right)$, call the `Get_Permutation_Matrix` function with the input parameters $(P_{f_1}(i,j)\sim P_{f_n}(i,j), P_{f_1'}(i',j')\sim P_{f_n'}(i',j'), S_M, S_N)$, where $(i',j')=W_0(i,j)$, to determine an estimated permutation matrix $\widetilde{\boldsymbol{W}}_{(i,j)}$ and its inverse $\widetilde{\boldsymbol{W}}_{(i,j)}^{-1}$.

With the $\left(1+\frac{M}{S_M}\cdot\frac{N}{S_N}\right)$ inverse matrices $\boldsymbol{W}_0^{-1}$ and $\left\{\boldsymbol{W}_{(i,j)}\right\}_{i=0,j=0}^{\frac{M}{S_M}-1,\frac{N}{S_N}-1}$, one can decrypt a new cipher-image f_{n+1}' as follows to get an estimated plain-image f_{n+1}^*:

```
for ($i=0$; $i\leq(M/S_M)-1$; $i++$)
    for ($j=0$; $j\leq(N/S_N)-1$; $j++$) {
        $f_{temp}=P_{f_{n+1}'}(w_0^{-1}(i,j))$;
        for ($ii=0$; $ii\leq S_M-1$; $ii++$)
            for ($jj=0$; $jj\leq S_N-1$; $jj++$)
                $f_{temp}^*(ii,jj)=f_{temp}\left(w_{(i,j)}^{-1}(ii,jj)\right)$;
        $P_{f_{n+1}^*}(i,j)=f_{temp}^*$;
    }
```

In fact, in the above procedure, any measure keeping invariant in the block permutations can be used instead of the mean value. A typical measure is the histogram of each $S_M\times S_N$ block. Although the mean value is less precise than the histogram, it works well in most cases and is useful to reduce the time complexity. When L and $S_M\times S_N$ are both too small, the efficiency of the mean value will become low, and the histogram or the array of all pixel values can be used as a replacement. Apparently, in most cases it is easier to get the high-level permutation matrix $\boldsymbol{W}_0$ than the low-level permutation matrices.

Finally, let us see whether the hierarchical structure used in HCIE is helpful to enhance the security against the known-plaintext attack to the common permutation image ciphers. As discussed above, $n\geq\lceil\log_L(2(MN-1))\rceil$ known plain-images are needed to achieve an acceptable breaking performance. Since the hierarchical structure makes it possible for an attacker to work on permutation matrices of size $S_M\times S_N$ or $\frac{M}{S_M}\times\frac{N}{S_N}$ (both smaller than $M\times N$), it is obvious that for HCIE the number of required known plain-image will be smaller than $\lceil\log_L(2(MN-1))\rceil$. Also, the attack complexity will become less, since it is proportional to the square of the matrix sizes. In such a sense, hierarchical permutation-only image ciphers are less secure than non-hierarchical ones, which discourages the use of HCIE. This result has been confirmed by our experiments (see the next section).

2) The chosen-plaintext attack: Following the same way introduced in the chosen-plaintext attack to common permutation-only image ciphers, one can choose $n=\lceil\log_L(MN)\rceil$ plain-images to carry out a chosen-plaintext attack to HCIE. Similar to the known-plaintext attack, the use of a hierarchical structure in HCIE can also make the construction of chosen plain-images easier. Accordingly, an attacker can also work hierarchically to construct n chosen plain-images, $f_1,\cdots,f_n$, as follows:

- *high-level*: $\overline{P}_{f_1}\sim\overline{P}_{f_n}$, which are defined in Eq. (7), compose an orthogonal image set;
- *low-level*: $\forall(i,j)$, $P_{f_1}(i,j)\sim P_{f_n}(i,j)$ compose an orthogonal image set.

In this case, the minimal number of required chosen plain-image becomes

$$\begin{aligned} n &= \max\left(\lceil\log_L(S_M\cdot S_N)\rceil, \left\lceil\log_L\left(\frac{M}{S_M}\cdot\frac{N}{S_N}\right)\right\rceil\right) \\ &\leq \lceil\log_L(MN)\rceil, \end{aligned} \tag{9}$$

where the equality holds if and only if the hierarchical encryption structure is disabled, i.e., when $(S_M=M, S_N=N)$ or

$(S_M = S_N = 1)$.

3) The brute-force attack: In [10]–[12], it was claimed that the complexity of brute-force attacks to HCIE is $O\left(2^{L_b}\right)$, since there are $L_b = \left(1+\frac{M}{S_M}\cdot\frac{N}{S_N}\right)\cdot(3S_M+3S_N-2)\cdot no$ secret chaotic bits in $\{b(i)\}_{i=0}^{L_b-1}$ that are unknown to attackers. However, this statement is not true due to the following fact: the L_b bits are uniquely determined by the secret key, i.e., the initial condition $x(0)$ and the control parameter μ, which have only $2L$ secret bits. This means that there are only 2^{2L} different chaotic bit sequences. Now, let us study the real complexity of brute-force attacks. For each pair of guessed values of $x(0)$ and μ, the following operations are needed:

- generating the chaotic bit sequence: $L_b/8$ chaotic iterations;
- creating the pseudo-image f_p: the complexity is $S_M \cdot S_N$;
- shuffling the pseudo-image f_p: running the `Sub_HCIE` function once;
- generating P_{f^*}: the complexity is $M\cdot N$;
- shuffling the partition image P_{f^*}: running the `Sub_HCIE` function for $\left(\frac{M}{S_M}\cdot\frac{N}{S_N}\right)$ times.

Assume that the computing complexity of the `Sub_HCIE` function is $(4S_M + 4S_N)\cdot no$. Then, the total complexity of brute-force attacks to HCIE can be calculated about $O\left(2^{2L}\cdot(L_b+MN)\right)$, which is much smaller than $O\left(2^{L_b/8}\right)$ when L_b is not too small. Additionally, considering the fact that the Logistic map can exhibit a sufficiently strong chaotic behavior only when μ is close to 4 [53], the complexity should be even smaller. The above analysis shows that the security of HCIE was much over-estimated by the authors in [10]–[12], even under brute-force attacks.

V. Experiments

To verify the decryption performance of the above-discussed known-plaintext attack to general permutation-only image ciphers working in spatial domain and particularly to HCIE, some experiments are performed using the six 256×256 test images with 256 gray scales shown in Fig. 2. Assume that the first $n = 1\sim 5$ test images are known to an attacker, the cipher-image of the last test image is decrypted with the estimated permutation matrices to see the breaking performance. In the experiments, the "taking-the-first" algorithm is used to generate $\widetilde{\boldsymbol{W}}$ from $\widehat{\boldsymbol{W}}$ in the `Get_Permutation_Matrix` function. It turns out that such a simple algorithm is enough to achieve a considerable performance in real attacks.

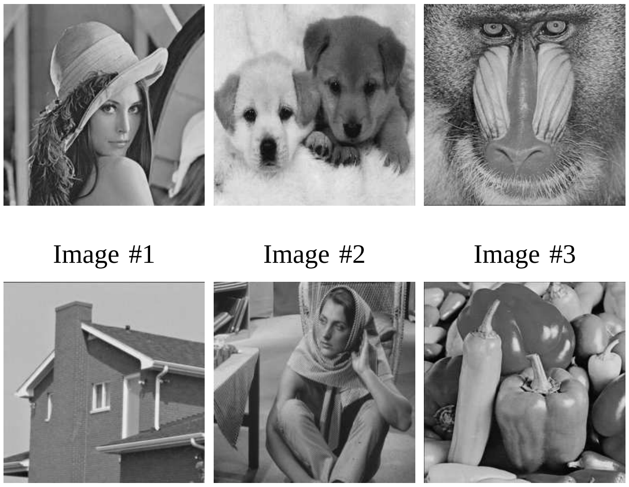


Image #1 Image #2 Image #3

Image #4 Image #5 Image #6

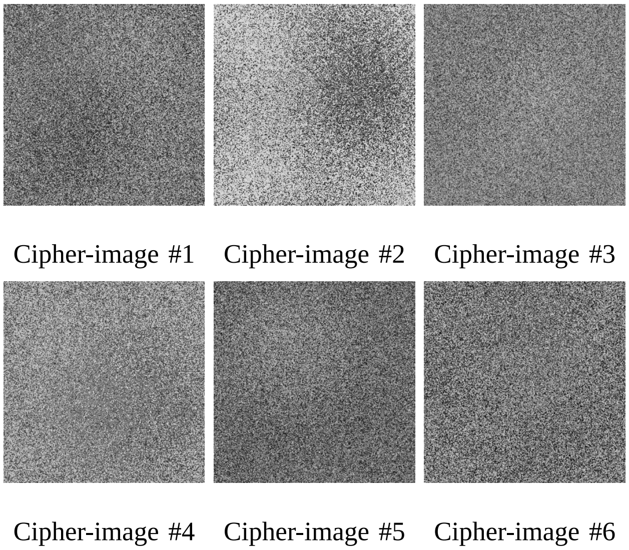
Fig. 2. The six 256×256 test images used in the experiments.

In the experiments, three different configurations of HCIE are used: $S_M = S_N = 256$, $S_M = S_N = 32$, $S_M = S_N = 16$. As mentioned above, the configuration of $S_M = S_N = 256$ corresponds to general permutation-only image ciphers working in spatial domain (without using hierarchical structures). It is shown that three known plain-images are always enough to achieve a good breaking performance, and that an almost perfect breaking performance can be achieved with four plain-images. Thus, the theoretical analysis given in the last section is verified. Also, it has been confirmed that the security of the two-level hierarchical encryption structure is weaker than the security of the non-hierarchical structure. As a result, the security of HCIE against known-plaintext attack is even weaker than the security of other common permutation-only image ciphers.

The chosen-plaintext attack is omitted in this section, since one can absolutely break the permutation matrix by choosing two plain-images f_1 and f_2 as shown in Eqs. (5) and (6). Of course, some experiments have been completed to verify this theoretical result and the correctness of the uniquely-determined permutation matrix.

A. The experimental results with $S_M = S_N = 256$

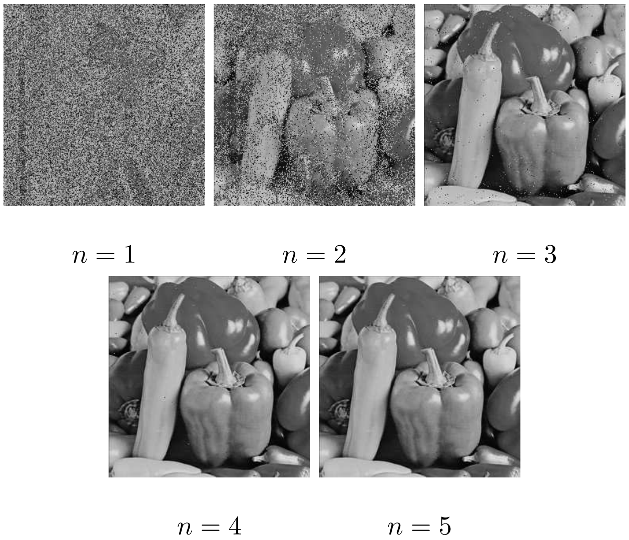
The public parameters are $\alpha = 6$, $\beta = 3$, $\gamma = 3$ and $no = 9$. The cipher-images of the six test images are shown in Fig. 3. When the first $n = 1 \sim 5$ test images and their cipher-images are known to the attacker, the five decrypted images of the sixth cipher-image are shown in Fig. 4. As can seen, one known plain-image cannot reveal any visual information, but two is capable to recover a rough view of the sixth test image, and three are enough to obtain a good recovery.



Cipher-image #1 Cipher-image #2 Cipher-image #3

Cipher-image #4 Cipher-image #5 Cipher-image #6

Fig. 3. The cipher-images of the six test images, when $S_M = S_N = 256$.



$n = 1$ $n = 2$ $n = 3$

$n = 4$ $n = 5$

Fig. 4. The decrypted images of Cipher-Image #6 when the first n test images are known to the attacker, when $S_M = S_N = 256$.

To verify the fact that the breaking performance is better than the theoretical prediction based on the correctly-recovered elements in $\widetilde{W}$, let us see the decryption performance with $n = 2$ as an example. For this case, the number of the absolutely correct elements in $\widetilde{W}$ are only 10,600, and the number of all correct elements in $\widetilde{W}$ is 26,631. In comparison, the number of correctly-recovered pixels are 27,210. Although only about $\frac{27210}{65536} \approx 41.52\%$ of the pixels are recovered, most visual information in the plain-image #6 has been revealed successfully. Now, let us consider the correct pixels that are not recovered from the correct elements in $\widetilde{W}$, i.e, the $(27210 - 26631 = 579)$ more correct pixels. These pixels are correctly decrypted with a frequency $\frac{579}{65536-26631} \approx 0.0149$, which is larger than the average probability $L^{-1} \approx 0.0039$. If we also count those pixels whose values close to the right ones, this frequency will be even larger. In fact, excluding the pixels correctly determined by the 26,631 correct elements in $\widetilde{W}$, the histogram of the other $(65536 - 26631 = 38905)$ pixels of the difference image between the recovered image and the original plain-image #6 is a Gaussian-like function as

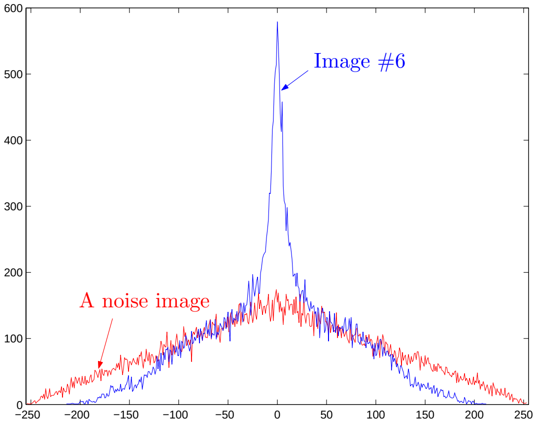


Fig. 5. The histogram of the difference image between the recovered image and the original plain-image, when the plain-image is Image #6 (the blue line) or a randomly-generated noise image (the red line).

shown in Fig. 5. In comparison, the histogram of the difference image corresponding to a randomly-generated noise image of the same size 256×256 is also shown. It is clear that the Gaussian-like histogram corresponding to Image #6 is caused by the correlation information existing in natural images. Note that the triangular histogram of the noise image can be easily deduced under the assumption that the two involved images (i.e., the noise image and the corresponding cipher-image) are independent of each other and have a uniform histogram: $\forall i = -255 \sim 255$, the occurrence probability of the difference value i in the histogram is: $\frac{256-|i|}{65536} = \frac{1}{256} - \frac{|i|}{65536}$.

B. *The experimental results with $S_M = S_N = 32$*

The public parameters are $\alpha = 4$, $\beta = 2$, $\gamma = 1$ and $no = 2$. The cipher-images of the six test images are all shown in Fig. 6. When the first $n = 1 \sim 5$ test images are known to the attacker, the five decrypted images of the sixth cipher-image are shown in Fig. 7. As can be seen, one known plain-image cannot reveal much useful visual information, but two is enough to obtain a good performance.

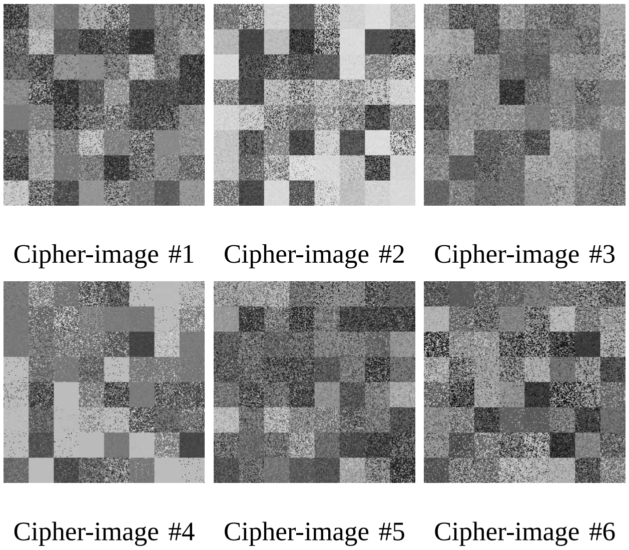


Fig. 6. The cipher-images of the six 256×256 test images, when $S_M = S_N = 32$.

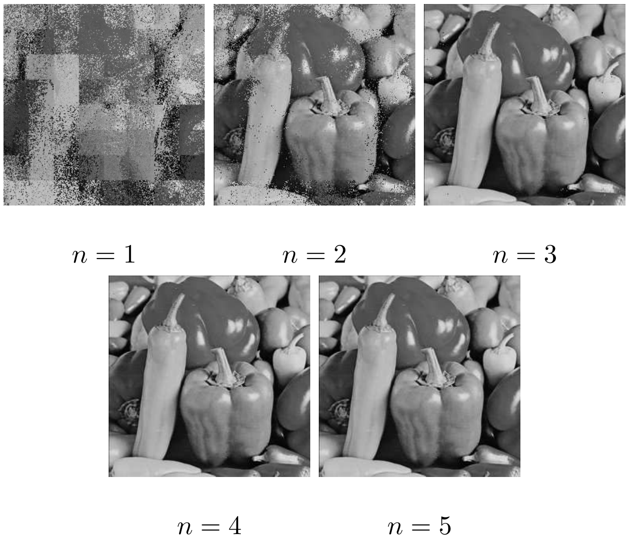
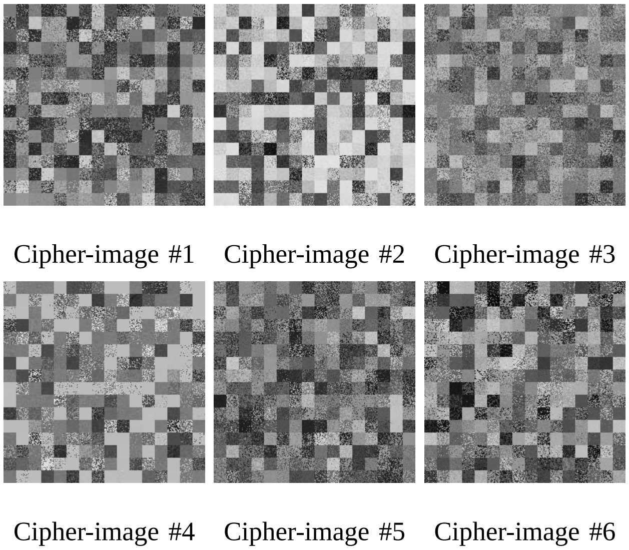


Fig. 7. The decrypted image of Cipher-Image #6 when the first n test images are known to the attacker, when $S_M = S_N = 32$.

C. *The experimental results with $S_M = S_N = 16$*

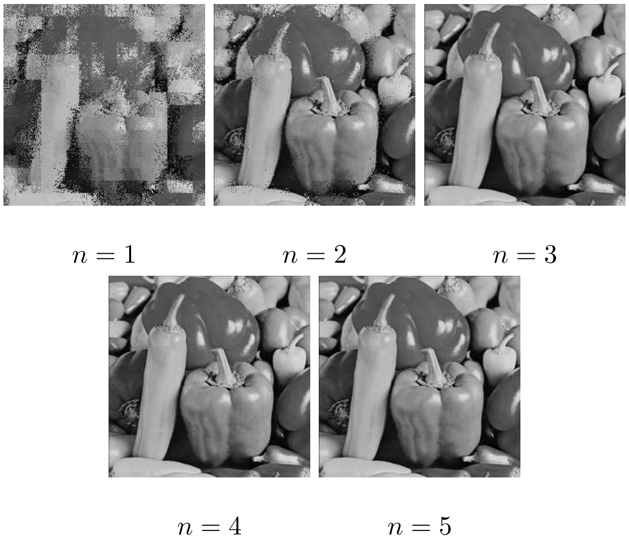
The public parameters are $\alpha = 4$, $\beta = 2$, $\gamma = 1$ and $no = 2$. The cipher-images of the six test images are all shown in Fig. 8. When the first $n = 1 \sim 5$ test images are known to the attacker, the five decrypt images of the sixth cipher-image are shown in Fig. 9. As can be seen, even one known plain-image can reveal a rough view of the plain-image, and two is enough to obtain a nearly-perfect recovery.



Cipher-image #1 Cipher-image #2 Cipher-image #3

Cipher-image #4 Cipher-image #5 Cipher-image #6

Fig. 8. The cipher-images of the six 256×256 test images, when $S_M = S_N = 16$.



$n = 1$ $n = 2$ $n = 3$

$n = 4$ $n = 5$

Fig. 9. The decrypted images of Cipher-Image #6 when the first n test images are known to an attacker, when $S_M = S_N = 16$.

D. A comparison of the performances

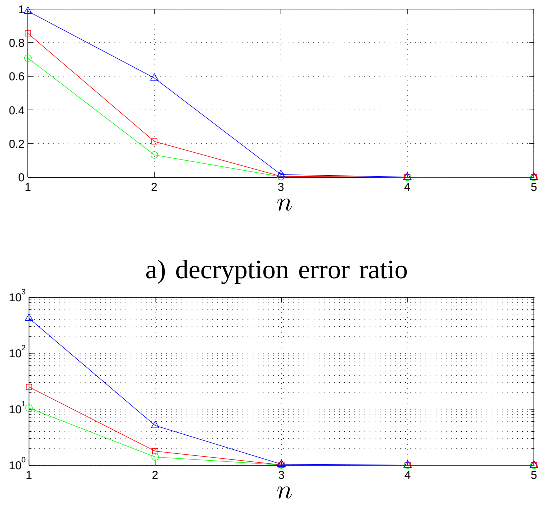
This subsection gives a performance comparison of the known-plaintext attack to HCIE with the above three different configurations. Figure 10a shows the quantitative relation between the number of known plain-images and the decryption quality (represented by the decryption error ratio). It can be seen that three known plain-images are enough for all three configurations to achieve an acceptable breaking performance, and two can reveal quite a lot of pixels (which means that most significant visual information is revealed). Also, it is shown that the breaking performance is dependent on the configuration: when $S_M = S_N = 16$, the best performance is achieved, which coincides with the expectation from Eq. (9): n is minimized when $S_M = S_N = \sqrt{256} = 16$.

Figure 10b shows the average cardinality of the elements in $\widehat{\boldsymbol{W}}$, which is an indicator of the probability of getting correct permutation elements in $\widetilde{\boldsymbol{W}}$ and an indicator of the time complexity as analyzed above. Comparing Figures 10a and 10b, one can see that the occurrence probability of decryption errors has a good correspondence with the average cardinality.

From the above comparison, it is true that the security of HCIE with a hierarchical structure is even weaker than the security of general permutation-only image ciphers without hierarchical structures: when $S_M = S_N = 32$ and $S_M = S_N = 16$, two known plain-images are enough to achieve an acceptable breaking performance; while when $S_M = S_N = 256$, the breaking performance with two known plain-images is not satisfactory, and three plain-images are needed to achieve an acceptable performance. Therefore, from the viewpoint of security against known/chosen-plaintext attacks, the hierarchical idea proposed in HCIE has no technical merits. This verifies the theoretical analysis given in the last section.

VI. Generalization of the Cryptanalysis

In the previous sections, it has been shown that permutation-only image cipher working in spatial domain are not secure against known/chosen-plaintext attacks. In this section, the above cryptanalysis results is generalized to permutation-only image ciphers working in frequency domain and also to permutation-only video ciphers. Since the cryptanalysis procedure is almost identical except for the format of plaintexts and ciphertexts, the following discussions only focus on a



a) decryption error ratio

b) the average cardinality $\overline{\#(\widehat{w}(i,j))}$

(Legend: △ – $S_M = S_N = 256$, □ – $S_M = S_N = 32$, ○ – $S_M = S_N = 16$)

Fig. 10. A performance comparison of the known-plaintext attack to HCIE

rough comparison of the breaking performances in different situations.

A. Cryptanalysis of Permutation-Only Image Ciphers Working in Frequency Domain

Many digital images are stored by lossy compression techniques, which generally work in frequency domain, especially in DCT or wavelet domain. Accordingly, when permutation-only image ciphers are used to encrypt such images, the secret permutations are exerted on the transformation coefficients in frequency domain, not on the pixels in the spatial domain. In most transformation-based compression formats, the image is divided into many blocks of smaller size to reduce the time complexity of compression. For example, in DCT-based formats, the image is generally divided into 8×8 blocks; and in wavelet-based formats, the image is generally divided into a quadtree. In this case, the secret permutations can also be exerted on the blocks or the nodes of the tree, i.e., there may exist a hierarchical encryption structure.

Generally speaking, it is easy to directly generalize the above known/chosen-plaintext cryptanalysis, by considering the transformed image $\mathfrak{T}(f)$ as the plain-image f, i.e., considering the transform coefficients as the pixels in spatial domain. The only difference between the two cases are that the there exists energy concentration in $\mathfrak{T}(f)$ – generally most significant transform coefficients distribute within low-frequency band. What does this mean for cryptanalysis? Apparently, to achieve an acceptable breaking performance, one can only reconstruct the elements in $\boldsymbol{W}$ and $\boldsymbol{W}^{-1}$ that correspond to low-frequency coefficients. This implies that the reduction of the image size, which immediately leads to a smaller number of required known/chosen plain-images and to the decline of the security against known/chosen-plaintext attacks. In fact, in [42, Sec. 3.4.2], it has been pointed out that the non-uniform distribution of DCT coefficients in MPEG videos (also for JPEG images) can even be used to partially break the secret permutations in *ciphertext-only attacks*. For example, one can correctly locate the DC coefficient of each 8×8 block with a large probability since the DC coefficient generally has the largest amplitude among all 64 DCT coefficients.

As shown in previous sections, the existence of hierarchical structures in compression techniques further reduces the security. What's more, some elements in $\boldsymbol{W}$ and $\boldsymbol{W}^{-1}$ that correspond to high-frequency coefficients can also be determined in the attacks, which can further help refine the visual quality of the recovered plain-image.

As a result, generally permutation-only image ciphers working in frequency domain are less secure against known/chosen-plaintext attack than those working in spatial domain. If it is possible to avoid the energy-concentration property and the hierarchical structure, the security at best will be equivalent to that in the spatial-domain case.

B. Cryptanalysis of Permutation-Only Video Ciphers

A video stream is composed of a series of 2-D consecutive images, which are called *frames* of the video. Essentially, permutation-only video ciphers work in the same way on each frame as permutation-only image ciphers. Due to the bulky size of most videos, transform-based lossy compression techniques are widely used for storage and transmission of videos. Also, the block-based or quadtree-based hierarchical structure is widely used in various video formats. So, despite the details of different video formats, the security of most permutation-only video ciphers against known/chosen-plaintext attacks is in the same order as that of permutation-only image ciphers working in frequency domain.

As a result, the security of a video cipher can be evaluated by considering it as an image cipher encrypting the following two types of plain-images: 1) independent frames, such as I-frames in MPEG videos; 2) frames dependent on others, such as B/P-frames in MPEG videos. In such a way, the security analysis of the video cipher becomes simpler and clearer. The major extra consideration in the design of a video cipher is how to make the cipher faster and easier for implementation in the whole video processing system.

Here, note the following fact: if the permutation matrix is fixed for all frames, then only one partially-known/chosen plain-video is enough to reveal the secret permutation matrix. From such a point of view, the security of a permutation-only video cipher may be even weaker than its image counterpart. However, if the permutation matrix has to be changed from frame to frame, it will be more difficult to maintain the fast speed of the video cipher. This is another consideration in the design of a good video cipher.

VII. CONCLUSIONS

By normalizing the encryption and decryption procedures of permutation-only image ciphers working in spatial domain, the present paper analyzes the security of such image ciphers against known/chosen-plaintext attacks from a general perspective, and then generalizes the basic results to permutation-only image ciphers working in frequency domain and also to permutation-only video ciphers. A recently-proposed permutation-only image cipher, named HCIE, has been studied as a typical example for illustrating the cryptanalysis. When the plain-images have size $M \times N$ with L possible pixel values, it is found that only $O\left(\log_L(MN)\right)$ known/chosen plain-images are enough for an attacker to achieve a rather good breaking performance, leading to the conclusion that all permutation-only ciphers are not secure enough against known/chosen-plaintext attacks. Also, it has been found that the attack complexity is practically small – only $O(n \cdot (MN)^2)$, when n plain-images are known or chosen to use. Some experiments have been shown to support the cryptanalysis of general permutation-only image ciphers as well as the specific HCIE. As a natural result, it is also found that hierarchical permutation-only image ciphers such as HCIE are less secure than normal permutation-only image ciphers without using hierarchical encryption structures.

In summary, secret permutations are incapable of providing a sufficiently high level of security against known/chosen-plaintext attacks, so they must be used together with other encryption techniques in the design of highly secure multimedia encryption algorithms. To the best of our knowledge, it is the first time to clarify this security principle on multimedia encryption algorithms, from both theoretical and experimental points of view.

References

[1] B. Schneier, *Applied Cryptography – Protocols, Algorithms, and Souce Code in C*, 2nd ed. New York: John Wiley & Sons, Inc., 1996.

[2] N. G. Bourbakis and C. Alexopoulos, "Picture data encryption using SCAN patterns," *Pattern Recognition*, vol. 25, no. 6, pp. 567–581, 1992.

[3] C. Alexopoulos, N. G. Bourbakis, and N. Ioannou, "Image encryption method using a class of fratcals," *J. Electronic Imaging*, vol. 4, no. 3, pp. 251–259, 1995.

[4] H. K.-C. Chang and J.-L. Liu, "A linear quadtree compression scheme for image encryption," *Signal Processing: Image Communication*, vol. 10, no. 4, pp. 279–290, 1997.

[5] R. Zunino, "Fractal circuit layout for spatial decorrelation of images," *Electronics Letters*, vol. 34, no. 20, pp. 1929–1930, 1998.

[6] K.-L. Chung and L.-C. Chang, "Large encryption binary images with higher security," *Pattern Recognition Letters*, vol. 19, no. 5–6, pp. 461–468, 1998.

[7] J. Scharinger, "Fast encryption of image data using chaotic Kolmogorov flows," *J. Electronic Imaging*, vol. 7, no. 2, pp. 318–325, 1998.

[8] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," *Int. J. Bifurcation and Chaos*, vol. 8, no. 6, pp. 1259–1284, 1998.

[9] J.-C. Yen and J.-I. Guo, "A new chaotic image encryption algorithm," in *Proc. (Taiwan) National Symposium on Telecommunications*, 1998, pp. 358–362.

[10] ——, "A new hierarchical chaotic image encryption algorithm and its hardware architecture," in *Proc. 1998 Ninth VLSI DESIGN/CAD Symposium*, 1998.

[11] J.-I. Guo, J.-C. Yen, and J.-C. Yeh, "The design and realization of a new chaotic image encryption algorithm," in *Proc. 1999 International Symposium on Communications*, 1999, pp. 210–214.

[12] J.-C. Yen and J.-I. Guo, "Efficient hierarchical chaotic image encryption algorithm and its VLSI realisation," *IEE Proc. – Vis. Image Signal Process.*, vol. 147, no. 2, pp. 167–175, 2000.

[13] D. Qi, J. Zou, and X. Han, "A new class of scrambling transformation and its application in the image information covering," *Science in China - Series E (English Edition)*, vol. 43, no. 3, pp. 304–312, 2000.

[14] X.-Y. Zhao and G. Chen, "Ergodic matrix in image encryption," in *Proc. Second International Conference on Image and Graphics*, ser. Proc. SPIE, vol. 4875, 2002, pp. 394–401.

[15] A. Pommer and A. Uhl, "Selective encryption of wavelet-packet encoded image data: Efficiency and security," *Multimedia Systems*, vol. 9, no. 3, pp. 279–287, 2003.

[16] Y. Matias and A. Shamir, "A video scramblng technique based on space filling curve (extended abstract)," in *Advances in Cryptology – Crypto'87*, ser. Lecture Notes in Computer Science, volume 293, 1987, pp. 398–417.

[17] Y. Mao, G. Chen, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons & Fractals*, vol. 21, no. 3, pp. 749–761, 2004.

[18] Y. Mao, G. Chen, and S. Lian, "A novel fast image encryption scheme based on 3D chaotic Baker maps," Accepted by *Int. J. Bifurcation and Chaos* in June 2003.

[19] X. Wu and P. W. Moo, "Joint image/video compression and encryption via high-order conditional entropy coding of wavelet coefficients," in *Porc. IEEE Conference on Multimedia Computing and Systems (CMS'99)*, 1999, pp. 908–912.

[20] H. C. H. Cheng, "Partial encryption for image and video communication." Master Thesis, Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada, Fall 1998.

[21] H. Cheng and X. Li, "Partial encryption of compressed images and videos," *IEEE Trans. Signal Processing*, vol. 48, no. 8, pp. 2439–2451, 2000.

[22] T. Uehara, R. Safavi-Naini, and P. Ogunbona, "Securing wavelet compression with random permutations," in *Proc. IEEE Pacific-Rim Conference on Multimedia (IEEE-PCM'2000)*, 2000, pp. 332–335.

[23] N. G. Bourbakis and A. Dollas, "SCAN-based compression-encryption-hiding for video on demand," *IEEE Multimedia*, vol. 10, no. 3, pp. 79–87, 2003.

[24] S. S. Maniccam and N. G. Bourbakis, "Image and video encryption using SCAN patterns," *Pattern Recognition*, vol. 37, no. 4, pp. 725–737, 2004.

[25] L. Tang, "Methods for encrypting and decrypting MPEG video data efficiently," in *Proc. 4th ACM Int. Conference on Multimedia*, 1996, pp. 219–230.

[26] L. Qiao and K. Nahrsted, "Comparison of MPEG encryption algorithms," *Computers & Graphics*, vol. 22, no. 4, pp. 437–448, 1998.

[27] S. U. Shin, K. S. Sim, and K. H. Rhee, "A secrecy scheme for MPEG video data using the joint of compression and encryption," in *Information Security: Second Int. Workshop (ISW'99) Proc.*, ser. Lecture Notes in Computer Science, volume 1729, 1999, pp. 191–201.

[28] C.-P. Wu and C.-C. J. Kuo, "Fast encryption methods for audiovisual data confidentiality," in *Multimedia Systems and Applications III*, ser. Proc. SPIE, volume 4209, 2001, pp. 284–295.

[29] ——, "Efficient multimedia encryption via entropy codec design," in *Security and Watermarking of Multimedia Contents III*, ser. Proc. SPIE, volume 4314, 2001, pp. 128–138.

[30] M. Pazarci and V. Dipçin, “A MPEG2-transparent scrambling technique,” *IEEE Trans. Consuming Eletronics*, vol. 48, no. 2, pp. 345–355, 2002.

[31] F. Chiaraluce, L. Ciccarelli, E. Gambi, P. Pierleoni, and M. Reginelli, “A new chaotic algorithm for video encryption,” *IEEE Trans. Consuming Eletronics*, vol. 48, no. 4, pp. 838–844, 2002.

[32] J. Wen, M. Severa, W. Zeng, M. H. Luttrell, and W. Jin, “A format-compliant configurable encryption framework for access control of video,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 545–557, 2002.

[33] W. Zeng and S. Lei, “Efficient frequency domain selective scrambling of digital video,” *IEEE Trans. Multimedia*, vol. 5, no. 1, pp. 118–129, 2003.

[34] D. Xie and C.-C. J. Kuo, “An enhanced MHT encryption scheme for chosen plaintext attack,” in *Internet Multimedia Management Systems IV*, ser. Proc. SPIE, volume 5242, 2003, pp. 175–183.

[35] M. S. Kankanhalli and T. T. Guan, “Compressed-domain scrambler/descrambler for digital video,” *IEEE Trans. Consuming Eletronics*, vol. 48, no. 2, pp. 356–365, 2002.

[36] B. Bhargava, C. Shi, and S.-Y. Wang, “MPEG video encryption algorithms,” *Multimedia Tools and Applications*, vol. 24, no. 1, pp. 57–79, 2004.

[37] M. Bertilsson, E. F. Brickell, and I. Ingemarson, “Cryptanalysis of video encryption based on space-filling curves,” in *Advances in Cryptology – EuroCrypt’88*, ser. Lecture Notes in Computer Science, volume 434, 1989, pp. 403–411.

[38] I. Agi and L. Gong, “An empirical study of secure MPEG video transmission,” in *Proc. ISOC Symposium on Network and Distributed Systems Security (SNDSS’96)*, 1996, pp. 137–144.

[39] J.-K. Jan and Y.-M. Tseng, “On the security of image encryption method,” *Information Processing Letters*, vol. 60, no. 5, pp. 261–265, 1996.

[40] L. Qiao and K. Nahrstedt, “Is MPEG encryption by using random list instead of ZigZag order secure?” in *Proc. IEEE Int. Symposium on Consumer Electronics (ISCE’97)*, 1997, pp. 332–335.

[41] J. H. Dolske, “Secure MPEG video: Techniques and pitfalls,” available online at http://www.dolske.net/old/gradwork/cis788r08/, June 1997.

[42] L. Qiao, “Multimedia security and copyright protection,” Ph.D. dissertation, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, 1998.

[43] T. Uehara and R. Safavi-Naini, “Chosen DCT coefficients attack on MPEG encryption schemes,” in *Proc. IEEE Pacific-Rim Conference on Multimedia (IEEE-PCM’2000)*, 2000, pp. 316–319.

[44] C.-C. Chang and T.-X. Yu, “Cryptanalysis of an encryption scheme for binary images,” *Pattern Recognition Letters*, vol. 23, no. 14, pp. 1847–1852, 2002.

[45] X.-Y. Zhao, G. Chen, D. Zhang, X.-H. Wang, and G.-C. Dong, “Decryption of pure-position permutation algorithms,” *Journal of Zhejiang University SCIENCE*, vol. 5, no. 7, pp. 803–809, 2004.

[46] X. Liu and A. M. Eskicioglu, “Selective encryption of multimedia content in distribution networks: Challenges and new directions,” in *Proc. IASTED Int. Conference on Communications, Internet and Information Technology (CIIT’2003)*, 2003.

[47] T. D. Lookabaugh, D. C. Sicker, D. M. Keaton, W. Y. Guo, and I. Vedula, “Security analysis of selectively encrypted MPEG-2 streams,” in *Multimedia Systems and Applications VI*, ser. Proc. SPIE, volume 5241, 2003, pp. 10–21.

[48] S. Li and X. Zheng, “Cryptanalysis of a chaotic image encryption method,” in *Proc. IEEE Int. Symposium on Circuits and Systems*, vol. II, 2002, pp. 708–711.

[49] ——, “On the security of an image encryption method,” in *Proc. IEEE Int. Conference on Image Processing*, vol. 2, 2002, pp. 925–928.

[50] S. Li, C. Li, G. Chen, and X. Mou, “Cryptanalysis of the RCES/RSES image encryption scheme,” Submitted to *IEEE Trans. Image Processing*, 2004.

[51] C. Li, S. Li, D. Zhang, and G. Chen, “Breaking a chaotic neural network based multimedia encryption scheme,” submitted to the 5th Pacific-Rim Conference on Multimedia (PCM’2004), May 2004.

[52] S. Li, G. Chen, and X. Zheng, “Chaos-based encryption for digital images and videos,” in *Multimedia Security Handbook*, B. Furht and D. Kirovski, Eds. CRC Press, 2004, ch. 4, the preprint is available at http://www.hooklee.com/pub.html (in press).

[53] Hao Bai-Lin, *Starting with Parabolas: An Introduction to Chaotic Dynamics* (in Chinese). Shanghai, China: Shanghai Scientific and Technological Education Publishing House, 1993.