A general quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks

Shujun Li [a,*], Chengqing Li [b], Guanrong Chen [b],
Nikolaos G. Bourbakis [c] and Kwok-Tung Lo [d]

[a] *FernUniversität in Hagen, Lehrgebiet Informationstechnik, Universitätsstraße 27, 58084 Hagen, Germany*

[b] *Department of Electronic Engineering, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon Tong, Hong Kong, China*

[c] *Information Technology Research Institute, College of Engineering and Computer Science, Wright State University, 3640 Glenn Hwy, Dayton, OH 45435, USA*

[d] *Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong SAR, China*

Abstract

In recent years secret permutations have been widely used for protecting different types of multimedia data, including speech files, digital images and videos. Based on a general model of permutation-only multimedia ciphers, this paper performs a quantitative cryptanalysis on the performance of these kind of ciphers against plaintext attacks. When the plaintext is of size $M \times N$ and with L different levels of values, the following quantitative cryptanalytic findings have been concluded under the assumption of a uniform distribution of each element in the plaintext: 1) all permutation-only multimedia ciphers are practically insecure against known/chosen-plaintext attacks in the sense that only $O\left(\log_L(MN)\right)$ known/chosen plaintexts are sufficient to recover not less than (in an average sense) half elements of the plaintext; 2) the computational complexity of the known/chosen-plaintext attack is only $O(n \cdot (MN)^2)$, where n is the number of known/chosen plaintexts used. When the plaintext has a non-uniform distribution, the number of required plaintexts and the computational complexity is also discussed. Experiments are given to demonstrate the real performance of the known-plaintext attack for a typical permutation-only image cipher.

Key words: permutation-only multimedia encryption, image, video, speech, cryptanalysis, known-plaintext attack, chosen-plaintext attack

1 Introduction

With the rapid progress of computer and communication network technologies, a great deal of concerns have been raised about the security of multimedia data transmitted over open networks. Also, secure storage of digital multimedia data is demanded in many real applications, such as confidential teleconferencing, pay-TV, medical and military imaging, and privacy-related multimedia services. Due to the prevalence of multimedia services in consumer electronic devices, users of handheld devices have started to require content protection of multimedia data including recorded speech segments, personal photos and private movie clips.

To meet all these needs in practice, encryption algorithms are required to offer a sufficient level of security for different multimedia applications. Apparently, the simplest way to encrypt multimedia data is to treat them as 1-D bit-streams, and then to encrypt them with any available cipher [1, 2]. In some multimedia applications, such a simple idea of *naive encryption* may be enough. However, in many other applications, especially when digital images and videos are involved, encryption schemes considering special features of the multimedia data, such as bulky sizes and large redundancy in uncompressed images/videos, are still required to achieve a better overall performance and to make the integration of the encryption scheme into the whole process easier. In the past several decades, different algorithms have been proposed to provide specific solutions to the encryption of images, videos and speech data. Meanwhile, many cryptanalytic results have been reported, leading to the conclusion that a number of multimedia encryption schemes are insecure from the cryptographical point of view. For recent surveys on image and video encryption algorithms, see [3–7], and for surveys on speech encryption, see [6, 8–10].

The use of secret permutations is very popular in analog pay-TV services as a main approach to protecting video signals in broadcast-TV [11–13]. Due to the specifici structure of analog video signals, there are only three major

[*] The corresponding author, contact him via his personal web site `http://www.hooklee.com`.

ways to perform secret permutations on lines: time reversal (transmitting randomly selected lines in reverse order), line cut and rotation (cut each line at a random point and swap the two halves of the line), and line shuffling. The condition becomes much better, however, when secret permutations are used to protect digital multimedia data. According to the format of the multimedia data to be encrypted, a lot of elements can be permuted in a secure way: pixels (samples), bitplanes, lines, rows, blocks, macroblocks, slices, transform coefficients, VLC (variable-length codewords) syntax elements, tree nodes, and so on [14–40]. While some multimedia encryption schemes combine secret permutations with other encryption techniques, there are many multimedia encryption algorithms that are entirely based on secret permutations [11–31]. These are called *permutation-only* multimedia ciphers in this paper. Note that some ciphers can also be classified as permutation-only ones, even though other encryption techniques are used together with secret permutations. For instance, the video ciphers proposed in [37–39] become permutation-only ciphers, if the sign bits of all encrypted data elements are neglected. The main advantages of using only secret permutations in a cipher include: i) they can be easily implemented; ii) when used properly, perceptual information about the plaintext can be efficiently concealed.

The security of *permutation-only* multimedia ciphers has been extensively studied. Almost all permutation-only analog pay-TV encryption schemes and some permutation-only ciphers had already been found insecure against ciphertext-only attacks, due to the high information redundancy in multimedia data and/or some specific weaknesses in the encryption algorithms [41–45]. In addition, it has been widely known that permutation-only multimedia ciphers are insecure against the known/chosen-plaintext attack [25, 30, 31, 44–53], which is quite understandable since the secret permutations can be recovered by comparing the plaintexts and the permuted ciphertexts. Though secret permutations suffer from the above security problems, many researchers still hope that it will be useful to design multimedia encryption schemes based on this technique, due to the following reasons:

(1) the insecurity against ciphertext-only attacks is not a problem for most digital permutation-only ciphers because of the use of more complicated permutations;
(2) the insecurity against plaintext attacks can be solved in practice, by using dynamically-updated and/or plaintext-dependent secret permutations;
(3) it is one of the simplest encryption techniques to maintain format compliance and size preservation simultaneously;
(4) by combining it with very simple substitution operations, multimedia encryption of high confidentiality can be achieved.

To the best of our knowledge, all previous cryptanalytic results were performed for specific permutation-only image/video ciphers, and a general quantitative study about plaintext attacks has not been reported to clarify the number of required plaintexts and the computational complexity of such an attack[1]. As a result, some questions still remain to be answered, which include: i) Can the security of permutation-only multimedia encryption algorithms be effectively enhanced by designing new methods to generate "better" secret permutations? ii) How frequent should the secret permutations be updated to provide an acceptable security against plaintext attacks?

This paper reports a general cryptanalysis of permutation-only multimedia encryption algorithms against plaintext attacks, mainly focusing on the quantitative relation between the breaking performance and the number of required known/chosen plaintexts, and provides an estimation of the attack complexity. The cryptanalysis is performed on a general model of permutation-only multimedia ciphers by considering the plaintext (image, speech, frame of videos, etc.) as an $M \times N$ matrix in which each element has L possible distinct values. Under the assumption that each element in the matrix has an independent and uniform distribution, it will be shown that the number of plaintexts required to obtain an acceptable breaking performance in known plaintext attack is $\lceil \log_L(2(MN-1)) \rceil$. When the plaintext does not have a uniform distribution, this number will increase accordingly. This issue will also be studied on a special nonuniform distribution. For chosen-plaintext attack, it will be shown that only $\lceil \log_L(MN) \rceil$ plaintexts are enough to get a good breaking performance. In addition, an upper bound of the attack complexity will be obtained: $O(n \cdot (MN)^2)$, where n is the number of known/chosen plain-images.

The rest of this paper is organized as follows. In Sec. 2, a general model of permutation-only multimedia ciphers is described. Cryptanalysis on this normalized model is studied in detail in Sec. 3. Some experimental results are shown in Sec. 4 to support the theoretical cryptanalysis. The last section concludes the paper.

2 A General Model of Permutation-Only Multimedia Ciphers

Though different kinds of multimedia data require different kinds of secret permutations, it is possible to construct a general model by considering the plaintext as a 2-D $M \times N$ matrix. This is because the following reasons: i) 1-D speech data is just a special case of $M = 1$; ii) 3-D videos are generally encrypted frame by frame, and each frame is encrypted block by block, so permutation-

[1] Though there were some simple discussions on the quantitative aspects of known/chosen-plaintext attacks of bit-permutation ciphers in the cryptology community [54], this problem has not been systematically and quantitatively studied in a general way for any case, especially for permutation-only multimedia ciphers.

only video encryption is actually a generalized case of permutation-only image encryption; iii) the dimension remains unchanged when a multimedia signal is converted to transform domain. Thus, in the following of this section, we describe the general model based on a 2-D input plaintext. To facilitate the discussion below and to avoid potential confusion, we use a special term "particles" to denote elements in the 2-D plaintext that are permuted, such as pixels in a plain-image or transform coefficients in a block of a video frame.

As its name suggests, a *permutation-only* multimedia cipher encrypts a 2-D plaintext by permuting the positions of all particles 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. Although many different methods have been proposed to realize secret key-dependent pixel permutations, for a given plaintext of size $M\times N$, a permutation-only 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 plaintext $f = [f(i,j)]_{M\times N}$ and its corresponding ciphertext $f' = [f'(i,j)]_{M\times N}$, the encryption and decryption procedures of a permutation-only cipher can 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, we denote the encryption procedure by $f'(\boldsymbol{W}(\boldsymbol{I})) = f(\boldsymbol{I})$ and the decryption procedure by $f(\boldsymbol{W}^{-1}(\boldsymbol{I})) = f'(\boldsymbol{I})$, 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 property 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, $F_{\boldsymbol{W}} : \mathbb{M} \times \mathbb{N} \rightarrow \mathbb{M} \times \mathbb{N}$.

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

- *SCAN language* based methods [16–19, 32, 33]: define some different scan patterns of the 2-D plaintext and combine these patterns to obtain a permutation matrix by scanning the whole plaintext particle by particle;
- *quadtree* based methods [18,19]: divide the plaintext 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 [34–36]: iterate a discretized 2-D chaotic map over the $M \times N$ plaintext for many times to realize a permutation matrix;
- *Fractal curves* based methods [14, 15]: use a fractal(-like) curve to replace the normal scan order to realize a permutation matrix;
- *pseudo-random rotations* based methods [20, 23]: pseudo-randomly rotate particles along some straight lines for many times to realize a permutation matrix;
- *matrix transformation based methods* [21]: use (integer) transformations of matrix, such as n-dimensional Arnold transformation and Fibonacci-Q transformation, to define permutation matrices;
- *composite methods* [22]: combine different methods to realize more complicated permutation matrices.

Although different types of secret keys are used in different permutation-only multimedia 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 multimedia ciphers can be considered the same. This is the base for the security analysis to be carried out below in next section.

3 General Quantitative Cryptanalysis

In this section, we discuss the general quantitative cryptanalysis of plaintext attacks based on the above general model of permutation-only multimedia ciphers.

3.1 Known-plaintext attack

As discussed above, when a *permutation-only* multimedia cipher is used to encrypt a plaintext, the particle at the position (i, j) will be secretly permuted to another fixed position (i', j') while its value remains unchanged. Therefore, by comparing a number of known plaintexts and the corresponding ciphertexts, it is possible for an attacker to (partially or even totally) reconstruct the secret permutations of all particles, i.e., to derive the encryption/decryption keys – the permutation matrix $\boldsymbol{W}$ and its inverse $\boldsymbol{W}^{-1}$.

Given n known plaintexts $f_1 \sim f_n$ and their ciphertexts $f_1' \sim f_n'$, the deduction procedure of the two key-matrices $\boldsymbol{W}$ and $\boldsymbol{W}^{-1}$ can be described by a function `Get_Permutation_Matrix`. 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 particle ranges in $\{0, \cdots, L-1\}$, the function `Get_Permutation_Matrix` works as follows.

- *Step 1: compare pixel values within the n ciphertexts $f_1' \sim f_n'$ to get $(n \cdot L)$ sets of 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 particles 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 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 $\bigcup_{\substack{0 \leq i \leq M-1 \\ 0 \leq j \leq N-1}} \widehat{\boldsymbol{w}}(i, j) = \mathbb{M} \times \mathbb{N}$ and that $\widehat{\boldsymbol{w}}(i_1, j_1) = \widehat{\boldsymbol{w}}(i_2, j_2)$ may hold if $(i_1, j_1) \neq (i_2, j_2)$;
- *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 $\#\left(\widehat{\boldsymbol{w}}(0, 0)\right) = \cdots = \#\left(\widehat{\boldsymbol{w}}(M-1, N-1)\right) = 1$, i.e., each element of $\widehat{\boldsymbol{W}}$ contains only one 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 position, generally $\widetilde{\boldsymbol{W}}$ is not an exact estimation of $\boldsymbol{W}$. Assume that there are $(\widehat{N} \leq MN)$ distinct elements in $\widehat{\boldsymbol{W}}$, and that the $\widehat{N}$ 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 the genetic and simulated annealing algorithms. Our experiments have shown that even a simple algorithm may be enough to achieve a rather good estimation when $n \geq 3$ for 256×256 gray-scale images (see the next section for more details). 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 $\forall (i_1, j_1) \neq (i_2, j_2)$, $\widetilde{w}(i_1, j_1) \neq \widetilde{w}(i_2, j_2)$.

Next, we study 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 multimedia data, usually partially-recovered plaintext is 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 probability of success 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. This means that we will get a lower bound of the performance.

Now, the problem of counting correct elements in $\widetilde{\boldsymbol{W}}$ is simplified to be another one of counting singe-value elements in $\widehat{\boldsymbol{W}}$. From `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 any two particle values are independent of each other[2] and denote the occurrence probability of a particle value $l \in \{0, \cdots, L-1\}$ by P_l. Apparently, it is true that $\sum_{l=0}^{L-1} P_l = 1$ and $P_l = \frac{1}{L}$ for the uniform distribution of the particle value. 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 probability $P_{f_m(i,j)}$, 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 probability $\prod_{m=1}^{n} P_{f_m(i,j)}$.

[2] This is actually not true for most multimedia data, but we use this strong assumption to carry out a qualitative estimation.

Therefore, when the values of $f_1(i,j) \sim f_n(i,j)$ are fixed, the expected cardinality of $\widehat{\boldsymbol{w}}(i,j)$ is $1 + (MN-1)\prod_{m=1}^{n} P_{f_m(i,j)}$. Because it is very difficult to estimate a general result when the values of $P_1 \sim P_{L-1}$ are unknown, we only discuss two special distributions to demonstrate how to estimate a lower bound of the number of the required plaintexts.

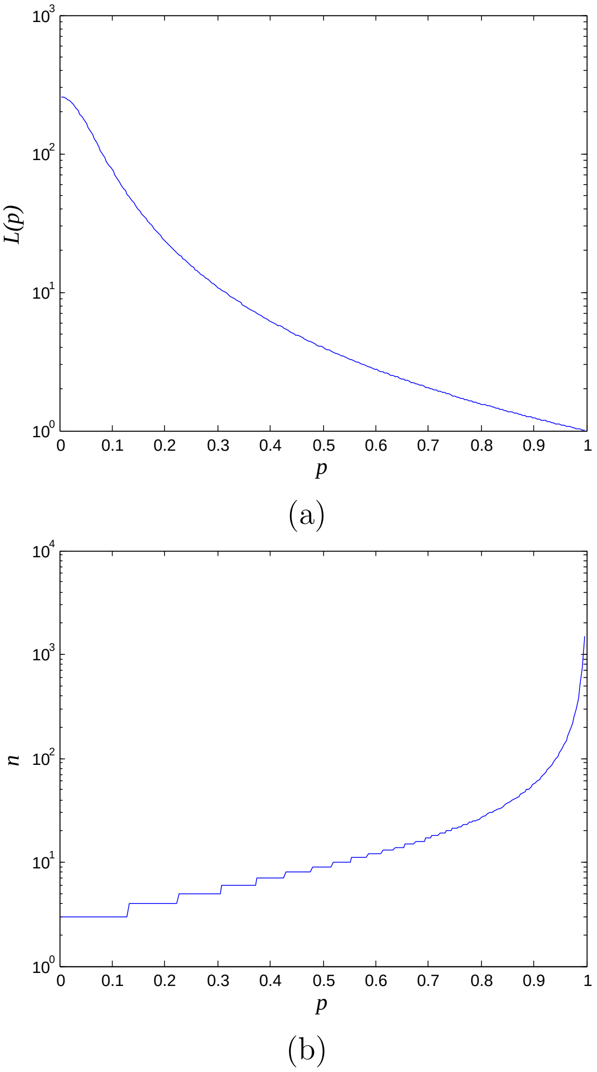
(1) *Uniform distribution*: In this case, $P_l = \frac{1}{L}$, $\forall l \in \{0, \cdots, L-1\}$. One can qualitatively deduce that the average cardinality of $\widehat{\boldsymbol{w}}(i,j)$ for any given position (i,j) is $1 + \frac{MN-1}{L^n}$, which approaches 1 exponentially as n increases. Generally speaking, when $1 + \frac{MN-1}{L^n} \leq 1.5$ or $\frac{MN-1}{L^n} \leq 0.5$, i.e., more than 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.

(2) Uniform distribution except for one particle value: Typical examples of this kind of distribution are images with large smooth background. Without loss of generality, assume $P_0 = p$ and $P_l = q = \frac{1-p}{L-1}$ for $l \in \{1, \cdots, L-1\}$. Then, if there are k values of $f_1(i,j) \sim f_n(i,j)$ equal to 0, which occurs with a probability of $\binom{n}{k} p^k (1-p)^{n-k}$, the expected cardinality of $\widehat{\boldsymbol{w}}(i,j)$ is $1 + (MN-1)p^k q^{n-k}$. As a result, one can get the average value of $\#\left(\widehat{\boldsymbol{w}}(i,j)\right) - 1$ as follows:

$$\begin{aligned}
\overline{\left(\#\left(\widehat{\boldsymbol{w}}(i,j)\right) - 1\right)} &= \sum_{k=0}^{n} \binom{n}{k} p^k (1-p)^{n-k} (MN-1) p^k q^{n-k} \\
&= (MN-1) \sum_{k=0}^{n} \binom{n}{k} (p^2)^k \left(\frac{(1-p)^2}{L-1} \right)^{n-k} \\
&= (MN-1) \left(p^2 + \frac{(1-p)^2}{L-1} \right)^n .
\end{aligned}$$

Let $(MN-1)\left(p^2 + \frac{(1-p)^2}{L-1}\right)^n \leq 0.5$. Then, one can get $n \geq \lceil \log_{L(p)}(2(MN-1)) \rceil$, where $L(p) = \frac{1}{p^2 + \frac{(1-p)^2}{L-1}}$. When $M = N = L = 256$, Figure 1 shows how the value of $L(p)$ and the lower bound of n change with respect to the value of p. It can be seen that the non-uniformity can cause an increase of the number of required plaintexts.

Though the distribution of most multimedia data is not uniform, our experiments on permutation-only image ciphers have shown that the above quantitative results obtained from the uniform distribution is basically correct for natural images: about $\log_L(2(MN-1))$ plain-images are sufficient to get a good breaking performance as will be shown in the next section. In fact, the



(a)

(b)

Fig. 1. The relationships between $L(p)$, $\lceil \log_{L(p)}(2(MN-1)) \rceil$ (the lower bound of n) and $p = 1/L, 2/L, \cdots, (L-1)/L$, when $M = N = L = 256$.

actual decryption performance is even better than the theoretical expectation because of 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, many wrongly-decrypted pixels are close to their true values with a probability larger than the average probability.

The above two points imply that the decryption performance of natural images will be better than that of noise-like images. For experimental verification and more explanations, see Sec. 4, Figs. 4 and 5. This perceptual phenomenon can also be generalized to audio and speech data.

Finally, 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. The time complexity of each step is as follows.

- *Step 1*: The L sets of each ciphertext f'_m are obtained by scanning f'_m once: for $i = 0 \sim (M-1)$ and $j = 0 \sim (N-1)$, add (i,j) into the set $\Lambda'_m(f'_m(i,j))$. Thus, the time complexity of this step is $O(nMN)$.
- *Step 2*: The average cardinality of $\Lambda'_m(l)$ is P_lMN and an upper bound of the time complexity of this step is $(MN)^n \sum_{(i,j)} \left(\prod_{m=1}^n P_{f_m(i,j)}\right)$. When the plaintext has a uniform distribution, $\sum_{(i,j)} \left(\prod_{m=1}^n P_{f_m(i,j)}\right) = \frac{MN}{L^n}$ and the upper bound becomes $MN \cdot \left(\frac{MN}{L}\right)^n$, which exponentially increases as n increases if $MN > L$. 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

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

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 can be calculated as follows for the uniform distribution:

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

As two typical examples, when $M = N = 256$ and $L = 2$, the complexity is about $(2^{29.2} \cdot n)$; when $M = N = 256$ and $L = 256$, the complexity is only $(2^{15} \cdot n)$. One can see that in both cases 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 smaller than $\frac{2^{\lceil \log_2 n \rceil}}{2L^2-1} \cdot (MN)^2 \leq \frac{2n}{2L^2-1} \cdot (MN)^2$.

When the distribution is not uniform, the reduction of each intersection becomes not easy to calculate. To simplify the deduction, we use the average size for all sets and assume that the reduction is proportional to a factor $\frac{1}{L^*}$ (as an analogue of $\frac{1}{L}$). The value of $\frac{1}{L^*}$ can be calculated as a weighted sum of all the possible sizes divided by MN: $\frac{1}{L^*} = \sum_{l=0}^{L-1} P_l \cdot (P_l MN)/MN = \sum_{l=0}^{L-1} P_l^2$. Then, by replacing L in Eq. (2) with $L^* = \frac{1}{\sum_{l=0}^{L-1} P_l^2}$, the overall complexity becomes $\frac{n(MN)^2}{2(L^*)^2-1}$. Taking the special nonuniform distribution studied before, $P_0 = p$ and $P_l = \frac{1-p}{L-1}$ for $l \in \{1, \cdots, L-1\}$, the value of L^* can be obtained as $L(p) = \frac{1}{p^2 + \frac{(1-p)^2}{L-1}}$, whose relation with p has been shown in Figure 1(a). As can be seen from the formula and the figure, $L(p)$ goes to 1 decreasingly with respect to the value of p, so the complexity will goes to $n(MN)^2$ as p approaches 1. Fortunately, this does not change the level of the complexity.

- *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 $O(MN)$.

Combining the above discussions, the final time complexity of the function `Get_Permutation_Matrix` 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 function `Get_Permutation_Matrix`, 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 $O\left(n \cdot \frac{(MN)^2}{L^*}\right)$.

3.2 Chosen-plaintext attack

The chosen-plaintext attack works in the same way as the known-plaintext attack, but the plaintext 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 plaintexts $f_1 \sim f_n$:

- the histogram of each chosen plaintext should be as uniform as possible;
- the i-dimensional ($2 \leq i \leq n$) histogram of any i chosen plaintexts 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 $M = N = L = 256$. In this case, the following two chosen plaintexts are enough to ensure a perfect estimation of the permutation matrix $\boldsymbol{W}$: $f_1 = [f_1(i,j) = i]_{256\times256}$ and $f_2 = [f_2(i,j) = j]_{256\times256}$, i.e.,

$$f_1 = f_2^T = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ i & \cdots & i \\ \vdots & \ddots & \vdots \\ 255 & \cdots & 255 \end{bmatrix}_{256\times256} \tag{3}$$

and

$$f_2 = f_1^T = \begin{bmatrix} 0 & \cdots & j & \cdots & 255 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & j & \cdots & 255 \end{bmatrix}_{256\times256} . \tag{4}$$

For the two chosen plaintexts, $(f_1(i_1,j_1), f_1(i_2,j_2)) \neq (f_2(i_1,j_1), f_2(i_2,j_2))$, $\forall (i_1,j_1) \neq (i_2,j_2)$. This ensures that $\#\left(\Lambda_1'(l_1) \cap \Lambda_2'(l_2)\right) = 1, \forall l_1, l_2 \in \{0, \cdots, L-1\}$.

In general cases, it can be easily deduced that $n = \lceil \log_L(MN) \rceil$ orthogonal plaintexts 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 plaintexts in the known-plaintext attack with a good breaking performance. This means the chosen-plaintext attack is a little (but not so much) stronger than the ~~chosen~~-plaintext attack.

4 Experiments

To verify the decryption performance of the above-discussed known-plaintext attack[3], some experiments have been performed on a typical permutation-only image cipher called CIE [20], in which the secret permutations are pseudo-randomly generated by iterations of a chaotic map. Figure 2 shows six 256×256 test images used in the experiments, both of which are in 256 gray scales. In the experiments, the "taking-the-first" algorithm was used to generate $\widetilde{\boldsymbol{W}}$ from $\widehat{\boldsymbol{W}}$ in the `Get_Permutation_Matrix` function. It turned out that such a simple algorithm was enough to achieve a considerable performance in real attacks.

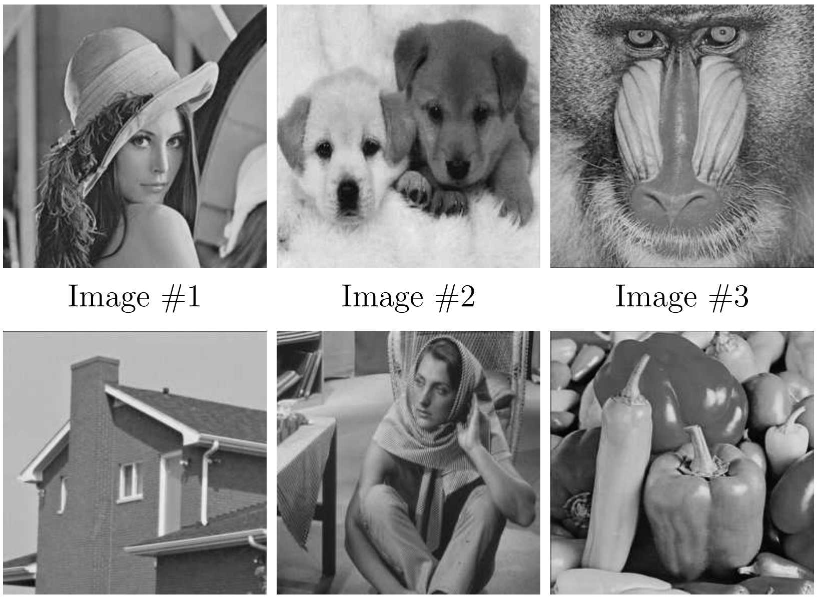


Image #1 Image #2 Image #3

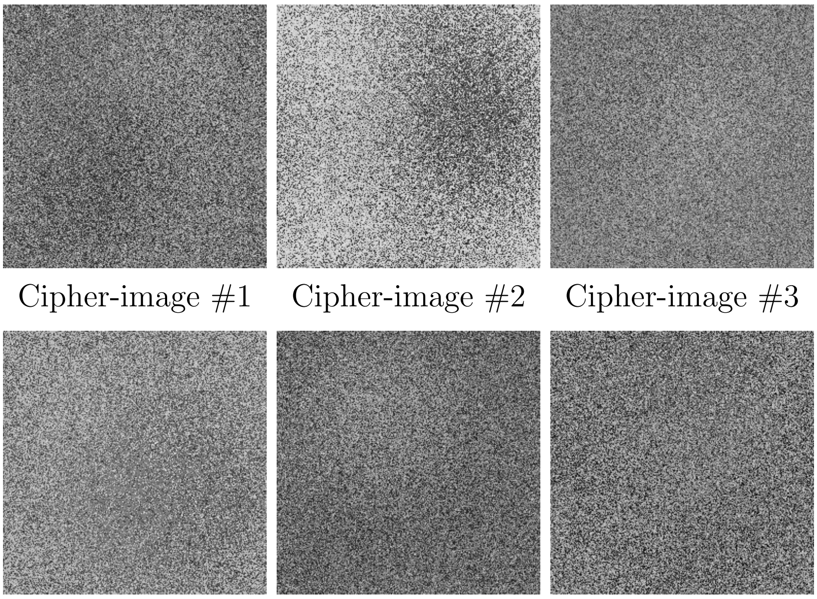
Image #4 Image #5 Image #6

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

The cipher-images of the six test images are shown in Fig. 3. When the first n $(= 1 \sim 5)$ test image(s) and the corresponding cipher-image(s) are known to the attacker, the breaking results of Cipher-Image #6 are demonstrated in Fig. 4. It can be seen that one known plain-image is not enough to reveal any visual information about the 6th test image, but two are capable to recover a rough view, and three or more are quite enough to achieve a very good performance.

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

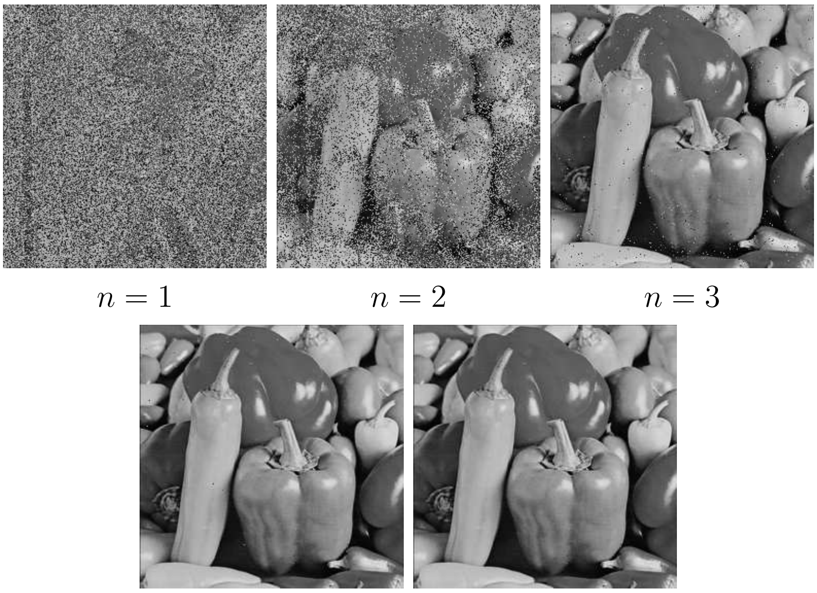
[3] The chosen-plaintext attack is omitted in this section, since one can absolutely break the permutation matrix by choosing two plaintexts f_1 and f_2 as shown in Eqs. (3) and (4).



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.



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

$n = 4$ $n = 5$

Fig. 4. The decrypted images of Cipher-Image #6 when the first n test images are known to the attacker.

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 $\frac{579}{65536-26631} \approx 0.0149$,

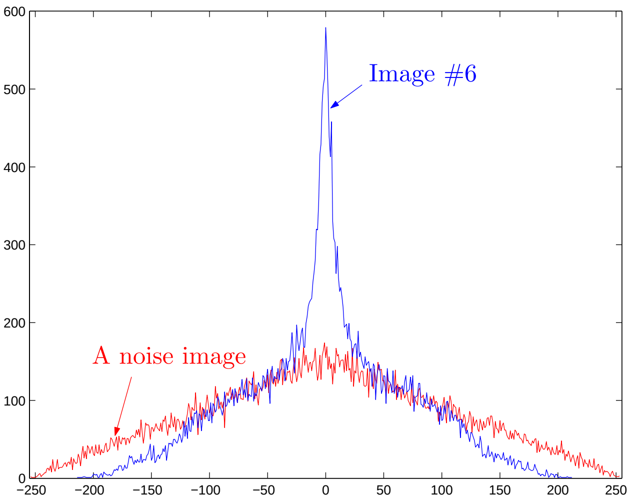


Fig. 5. The histogram of the difference image between the recovered image and the original plain-image, when the plain-image is Image #6 (the blue line) or a randomly-generated noise image (the red line).

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 Laplacian-like function as 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 Laplacian-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: $\frac{256-|i|}{65536} = \frac{1}{256} - \frac{|i|}{65536}$.

5 Conclusions

Based on a general model of permutation-only multimedia ciphers and from a general perspective, the present paper analyzes the security this type of ciphers against plaintext attacks. When the plaintext is of size $M \times N$ and distributed uniformly with L possible values, it is found that only $O\left(\log_L(MN)\right)$ plaintexts are enough to achieve a good breaking performance. It has also been found that the attack complexity is practically small – only $O(n \cdot (MN)^2)$, where n denotes the number of known/chosen plaintexts. Some experiments on a permutation-only image cipher have been shown to demonstrate the performance of the proposed known-plaintext attack. From the results of this paper, we draw the following conclusions: for permutation-only ciphers, 1) no better secret permutations can be achieved to offer a higher security level against plaintext attacks (compared with the general model discussed in this paper); 2) the secret permutations should be updated in a frequency smaller than $\log_L(MN)$ to offer an acceptable level of security against plaintext attacks, or they have to be combined with other encryption techniques to achieve this goal.

Acknowledgements

Shujun Li was supported by the Alexander von Humboldt Foundation and by The Hong Kong Polytechnic University's Postdoctoral Fellowship Scheme (under Grant No. G-YX63). The work of Nikolaos G. Bourbakis was supported by the AIIS Inc., NY, USA. The work of K.-T. Lo was supported by the Research Grants Council of the Hong Kong SAR Government under Project No. 523206 (PolyU 5232/06E).

References

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

[2] A. Menezes, P. van Oorschot, S. Vanstone, Handbook of Applied Cryptography, CRC Press, Inc., 1996.

[3] B. Furht, D. Socek, A. M. Eskicioglu, Fundamentals of multimedia encryption techniques, in: B. Furht, D. Kirovski (Eds.), Multimedia Security Handbook, CRC Press, LLC, 2004, Ch. 3, pp. 93–131.

[4] S. Li, G. Chen, X. Zheng, Chaos-based encryption for digital images and videos, in: B. Furht, D. Kirovski (Eds.), Multimedia Security Handbook, CRC Press, LLC, 2004, Ch. 4, pp. 133–167, preprint available at `http://www.hooklee.com/pub.html`.

[5] A. Uhl, A. Pommer, Image and Video Encryption: From Digital Rights Management to Secured Personal Communication, Springer, 2005.

[6] B. Furht, E. Muharemagic, D. Socek, Multimedia Encryption and Watermarking, Springer, 2005.

[7] W. Zeng, H. Yu, C.-Y. Lin (Eds.), Multimedia Security Technologies for Digital Rights Management, Academic Press, 2006.

[8] H. J. Beker, F. C. Piper, Secure Speech Communications, Academic, 1985.

[9] I. J. Kumar, Cryptology of speech signal, in: Cryptology: System Identification and Key-Clustering, Aegean Park Press, 1997, Ch. 6, pp. 309–380.

[10] R. K. Nichols, P. C. Lekkas, Speech cryptology, in: Wireless Security: Models, Threats, and Solutions, McGraw-Hill, 2002, Ch. 6, pp. 253–327.

[11] L. Brown, Comparing the security of pay-TV systems for use in Australia, Australian Telecommunication Research 24 (2) (1990) 1–8.

[12] A. Kudelski, Method for scrambling and unscrambling a video signal, U.S. Patent 5375168 (1994).

[13] Wikipedia, Television encryption, online document (2007).
URL `http://en.wikipedia.org/wiki/Television_encryption`

[14] Y. Matias, A. Shamir, A video scramble technique based on space filling curve (extended abstract), in: Advances in Cryptology - Crypto'87, Vol. 293 of Lecture Notes in Computer Science, 1987, pp. 398–417.

[15] R. Zunino, Fractal circuit layout for spatial decorrelation of images, Electronics Letters 34 (20) (1998) 1929–1930.

[16] N. G. Bourbakis, C. Alexopoulos, Picture data encryption using SCAN patterns, Pattern Recognition 25 (6) (1992) 567–581.

[17] C. Alexopoulos, N. G. Bourbakis, N. Ioannou, Image encryption method using a class of fractals, J. Electronic Imaging 4 (3) (1995) 251–259.

[18] H. K.-C. Chang, J.-L. Liu, A linear quadtree compression scheme for image encryption, Signal Processing: Image Communication 10 (4) (1997) 279–290.

[19] K.-L. Chung, L.-C. Chang, Large encryption binary images with higher security, Pattern Recognition Letters 19 (5-6) (1998) 461–468.

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

[21] D. Qi, J. Zou, X. Han, A new class of scrambling transformation and its application in the image information covering, Science in China - Series E (English Edition) 43 (3) (2000) 304–312.

[22] X.-Y. Zhao, G. Chen, Ergodic matrix in image encryption, in: Proc. Second International Conference on Image and Graphics, Vol. 4875 of Proc. SPIE, 2002, pp. 394–401.

[23] J.-C. Yen, J.-I. Guo, Efficient hierarchical chaotic image encryption algorithm and its VLSI realisation, IEE Proc. - Vision, Image and Signal Processing 147 (2) (2000) 167–175.

[24] H.-C. Chen, J.-I. Guo, L.-C. Huang, J.-C. Yen, Design and realization of a new signal security system for multimedia data transmission, EURASIP J. Applied Signal Processing 2003 (13) (2003) 1291–1305.

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

[26] L. Tang, Methods for encrypting and decrypting MPEG video data efficiently, in: Proc. 4th ACM Int. Conference on Multimedia, 1996, pp. 219–229.

[27] S. U. Shin, K. S. Sim, 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., Vol. 1729 of Lecture Notes in Computer Science, 1999, pp. 191–201.

[28] W. Zeng, S. Lei, Efficient frequency domain selective scrambling of digital video, IEEE Trans. Multimedia 5 (1) (2003) 118–129.

[29] S. Sridharan, E. Dawson, B. Goldburg, Speech encryption in the transform domain, Electronics Letters 26 (10) (1990) 655–657.

[30] S. Sridharan, E. Dawson, B. Goldburg, Fast Fourier transform based speech encryption system, IEE Proc. I - Communications, Speech and Vision 138 (3) (1991) 215–223.

[31] B. Goldburg, S. Sridharan, E. Dawson, Design and cryptanalysis of transform-based analog speech scramblers, IEEE J. Select. Areas Commun. 11 (5) (1993) 735–744.

[32] N. G. Bourbakis, A. Dollas, SCAN-based compression-encryption-hiding for video on demand, IEEE Multimedia 10 (3) (2003) 79–87.

[33] S. S. Maniccam, N. G. Bourbakis, Image and video encryption using SCAN patterns, Pattern Recognition 37 (4) (2004) 725–737.

[34] J. Scharinger, Fast encryption of image data using chaotic Kolmogorov flows, J. Electronic Imaging 7 (2) (1998) 318–325.

[35] J. Fridrich, Symmetric ciphers based on two-dimensional chaotic maps, Int. J. Bifurcation and Chaos 8 (6) (1998) 1259–1284.

[36] Y. Mao, G. Chen, S. Lian, A novel fast image encryption scheme based on 3D chaotic Baker maps, Int. J. Bifurcation and Chaos 14 (10) (2004) 3613–3624.

[37] S. Lian, X. Wang, J. Sun, Z. Wang, Perceptual cryptography on wavelet-transform encoded videos, in: Proc. IEEE Int. Symp. on Intelligent Multimedia, Video and Speech Processing (ISIMP'2004), 2004, pp. 57–60.

[38] S. Lian, J. Sun, Z. Wang, Perceptual cryptography on SPIHT compressed images or videos, in: Proc. IEEE Int. Conf. Multimedia & Expo (ICME'2004), 2004.

[39] S. Lian, J. Sun, Z. Wang, Perceptual cryptography on JPEG2000 compressed images or videos, in: Proc. Int. Conf. Computer and Information Technology (CIT'2004), IEEE Computer Society, 2004, pp. 78–83.

[40] Y. Mao, M. Wu, A joint signal processing and cryptographic approach to multimedia encryption, IEEE Trans. Image Processing 15 (7) (2006) 2061–2075.

[41] M. Bertilsson, E. F. Brickell, I. Ingemarson, Cryptanalysis of video encryption based on space-filling curves, in: Advances in Cryptology - EuroCrypt'88, Vol. 434 of Lecture Notes in Computer Science, 1989, pp. 403–411.

[42] M. Kuhn, AntiSky - an image processing attack on VideoCrypt, Online document, available at `http://www.cl.cam.ac.uk/~mgk25/tv-crypt/image-processing/antisky.html` (1994).

[43] M. Kuhn, Analysis for the nagravision video scrambling method, Online document, available at `http://www.cl.cam.ac.uk/~mgk25/nagra.pdf` (1998).

[44] J. H. Dolske, Secure MPEG video: Techniques and pitfalls, available online at `http://www.dolske.net/old/gradwork/cis788r08/` (June 1997).

[45] L. Qiao, Multimedia security and copyright protection, Ph.D. thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA (1998).

[46] J.-K. Jan, Y.-M. Tseng, On the security of image encryption method, Information Processing Letters 60 (5) (1996) 261–265.

[47] L. Qiao, 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. 226–229.

[48] L. Qiao, K. Nahrsted, Comparison of MPEG encryption algorithms, Computers & Graphics 22 (4) (1998) 437–448.

[49] H. C. H. Cheng, Partial encryption for image and video communication., Master thesis, Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada (1998).

[50] T. Uehara, 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.

[51] H. Cheng, X. Li, Partial encryption of compressed images and videos, IEEE Trans. Signal Processing 48 (8) (2000) 2439–2451.

[52] C.-C. Chang, T.-X. Yu, Cryptanalysis of an encryption scheme for binary images, Pattern Recognition Letters 23 (14) (2002) 1847–1852.

[53] X.-Y. Zhao, G. Chen, D. Zhang, X.-H. Wang, G.-C. Dong, Decryption of pure-position permutation algorithms, Journal of Zhejiang University SCIENCE 5 (7) (2004) 803–809.

[54] D. Wagner, G. G. Rose, T. Ritter, T. Jakobsen, N. Ferguson, D. R. Stinson, Transposition ciphers, Online Discussions in news group sci.crypt.research at google.com, available online at `http://groups-beta.google.com/group/sci.crypt.research/browse_thread/thread/3cd88407a3485cb1/58ff17304187ce74#58ff17304187ce74` (2001).