Optimal quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks

Chengqing Li[*,a,b], Kwok-Tung Lo[b]

[a] *College of Information Engineering, Xiangtan University, Xiangtan 411105, Hunan, China*
[b] *Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong*

Abstract

Recently, an image scrambling encryption algorithm of pixel bit based on chaos map was proposed. Considering the algorithm as a typical binary image scrambling/permutation algorithm exerting on plaintext of size $M \times (8N)$, this paper proposes a novel optimal method to break it with some known/chosen-plaintexts. The spatial complexity and computational complexity of the attack are only $O(32 \cdot MN)$ and $O(16 \cdot n_0 \cdot MN)$ respectively, where n_0 is the number of known/chosen-plaintexts used. The method can be easily extended to break any permutation-only encryption scheme exerting on plaintext of size $M \times N$ and with L different levels of values. The corresponding spatial complexity and computational complexity are only $O(MN)$ and $O(\lceil \log_L(MN) \rceil \cdot MN)$ respectively. In addition, some specific remarks on the performance of the image scrambling encryption algorithm are presented.

Key words: cryptanalysis, known-plaintext attack, chosen-plaintext attack, encryption, image

1. Introduction

With rapid development of digital information technology, image data is transmitted over all kinds of wired/wireless channels more and more frequently. Consequently, security of image data becomes more and more important. However, the traditional text encryption schemes fail to protect image data efficiently due to the big differences between image data and text, e.g. strong redundancy existing in uncompressed image data and its bulky size. In addition, image encryption schemes have some special requirements such as fast handling speed and easy concatenation of different components of the whole image processing system. Therefore, designing encryption schemes protecting image data specially becomes an urgent task. Due to the subtle similarities between cryptography and chaos, a great number of image encryption schemes based on chaos or other nonlinear theories have been proposed in the past decade [1–7]. Unfortunately, most of them have been found to be insecure to different extents from the viewpoint of modern cryptography [8–16]. For more discussion on chaos-based image encryption schemes, please refer to [17, 18].

In [19], an image permutation algorithm was proposed by scrambling/permuting binary bit of every pixel with pseudo-random number sequence generated by chaotic logistic map. Essentially, it is a permutation-only algorithm exerting on a binary image of size $M \times (8N)$. The

[*] Corresponding author.
Email address: chengqingg@gmail.com (Chengqing Li)

present paper focuses on security analysis of the algorithm and reports the following results: 1) an optimal method is proposed to break the image permutation algorithm under study with some known/chosen plaintexts; 2) the method is extended to break any permutation-only encryption scheme exerting on elements of any different levels of values; 3) some remarks on the performance of the image permutation algorithm under study are given.

The rest of this paper is organized as follows. Section 2 briefly introduces the image permutation algorithm under study. Section 3 proposes the optimal known/chosen-plaintext attack based on a binary tree to break the algorithm and points out some remarks on the performance on it. Extension of the optimal attack to break any permutation-only multimedia encryption schemes is discussed in Sec. 4. The last section concludes the paper.

2. The image permutation algorithm under study

The plaintext encrypted by the image permutation algorithm under study is a gray-scale image of size $M \times N$ (height×width), which can be denoted by an $M \times N$ matrix in domain $\mathbb{Z}_{256}$, $\boldsymbol{I} = [I(i,j)]_{i=0,j=0}^{M-1,N-1}$. The image $\boldsymbol{I}$ is further represented as an $M \times (8N)$ binary matrix $\boldsymbol{B} = [B(i,l)]_{i=0,l=0}^{M-1,8N-1}$, where $I(i,j) = \sum_{k=0}^{7} B(i,l) \cdot 2^k$, $l = 8 \cdot j + k$. The corresponding cipher-image is $\boldsymbol{I'} = [I'(i,j)]_{i=0,j=0}^{M-1,N-1}$, $I'(i,j) = \sum_{k=0}^{7} B'(i,l) \cdot 2^k$, $l = 8 \cdot j + k$. Then, the image permutation algorithm can be described as follows[1].

- *The secret key*: three positive integers m, n, and T, and the initial condition $x_0 \in (0,1)$ and control parameter $\mu \in (3.569945672, 4)$ of the following chaotic Logistic map:

$$f(x) = \mu \cdot x \cdot (1-x). \tag{1}$$

- *The initialization procedure*: 1) run the map Eq. (1) from x_0 to generate a chaotic sequence, $\{x_k\}_{k=1}^{\max\{(m+M),(n+8MN)\}}$; 2) generate an vector $\boldsymbol{T_M}$ of length M, where $\boldsymbol{S_M}(\boldsymbol{T_M}(i))$ is the $(i+1)$-th largest element of $\boldsymbol{S_M} = \{x_{m+k}\}_{k=1}^{M}$, $0 \leq i \leq (M-1)$; 3) generate a matrix $\boldsymbol{T_N}$ of size $M \times (8N)$, where, $\forall\ i \in \{0, \cdots, M-1\}$, $\boldsymbol{S_N}(\boldsymbol{T_N}(i,j))$ is the $(j+1)$-th largest element of $\boldsymbol{S_N} = \{x_{n+(8N)i+k}\}_{k=1}^{8N}$, $0 \leq j \leq N-1$.

- *The encryption procedure*:
  - *Step 1 – vertical permutation*: generate an intermediate matrix $\boldsymbol{B}^* = [B^*(i,l)]_{i=0,l=0}^{M-1,8N-1}$, where
$$B^*(i,:) = B(\boldsymbol{T_M}(i),:); \tag{2}$$
  - *Step 2 – horizontal permutation*: generate an intermediate matrix $\boldsymbol{B'} = [B'(i,l)]_{i=0,l=0}^{M-1,8N-1}$, where
$$B'(i,l) = B^*(i,\boldsymbol{T_N}(i,l)); \tag{3}$$
  - *Step 3 – repetition*: reset the value of x_0 to the current state of Eq. (1), and repeat the above operations from *the initialization procedure* for $(T-1)$ times.

[1] To make the presentation more concise and complete, some notations in the original paper are modified, and some details about the algorithm are also supplied or corrected.

- *The decryption procedure* is similar to the encryption one except the following simple modifications: 1) the different rounds of encryption are exerted in a reverse order; 2) the order of *Step 1* and *Step 2* in each round is reversed; 3) the left parts and right parts of Eq (2) and Eq. (3) are exchanged, respectively.

3. Cryptanalysis

3.1. Known-plaintext attack

The known-plaintext attack is an attack model for cryptanalysis where the attacker has some samples of both the plaintext and the corresponding ciphertext and make use of them to reveal secret information, such as secret keys and/or its equivalent ones.

Apparently, the combination of multiple rounds of the operations in *the encryption procedure* can be represented by an $M \times (8N)$ permutation matrix $\boldsymbol{W} = [w(i,l)]_{i=0,l=0}^{M,8N}$, where $w(i,l) = (i', l')$ denotes the secret position of the plain bit $B(i,j)$ in $\boldsymbol{B'}$. That is, the permutation matrix $\boldsymbol{W}$ defines a bijective map on set $\mathbb{M} \times \mathbb{N}^{+}$, where $\mathbb{M} = \{0, \cdots, M-1\}$ and $\mathbb{N}^{+} = \{0, \cdots, 8N-1\}$. Once the attacker recovers the permutation matrix $\boldsymbol{W}$ and then obtains its inverse $\boldsymbol{W}^{-1}$, he can use it as an equivalent key to decrypt any cipher-image encrypted with the same secret key.

Since that the secret permutation does not change the values of the permuted elements, a general algorithm was proposed in [20] for obtaining the permutation matrix by comparing the values of the elements of some plaintexts and the corresponding ciphertexts. Based on the same mechanism, a novel optimal method is proposed here to break the image permutation algorithm under study.

Actually, the proposed method is the construction process of a binary tree, where every node includes five components in order: a pointer holding address of the left child node, PT_L; two sets containing some entry positions of plain-image and cipher-image, respectively; cardinality of one of the two sets; a pointer holding address of the right child node, PT_R. Denote the two sets in the root node with $\mathbb{B}$ and $\mathbb{B}'$, respectively, where $\mathbb{B} = \mathbb{B}' = \mathbb{M} \times \mathbb{N}^{+}$. Obviously, cardinality of $\mathbb{B}$, $|\mathbb{B}| = 8MN$. Then, the binary tree can be constructed as follows.

- $\forall\ (i,l) \in \mathbb{B}$, do the following operations:

$$\begin{cases} \text{add } (i,l) \text{ into } \mathbb{B}_1 & \text{if } B(i,l) = 1; \\ \text{add } (i,l) \text{ into } \mathbb{B}_0 & \text{if } B(i,l) = 0. \end{cases} \tag{4}$$

- $\forall\ (i,l) \in \mathbb{B}'$, do the following operations:

$$\begin{cases} \text{add } (i,l) \text{ into } \mathbb{B}'_1 & \text{if } B'(i,l) = 1; \\ \text{add } (i,l) \text{ into } \mathbb{B}'_0 & \text{if } B'(i,l) = 0. \end{cases}$$

- Delete the elements in the two sets of root node and set the fourth item of the node as zero.

Since the secret permutation does not change the values of the permuted elements, the cardinalities of the two sets in the two child nodes are always the same. Hence, only the cardinality of one set is needed to record. The basic structure of the binary tree is shown in Fig. 1. With more pairs of known-images and the corresponding cipher-images, the binary tree will be updated and expanded iteratively with the following steps.

- Search for all nodes whose third item is greater than one, namely, both the corresponding two sets have more than one elements;
- Expand each found node with the similar operations shown above.

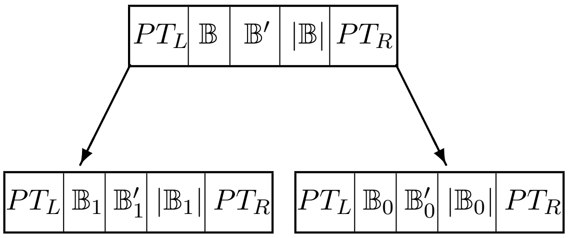

PT_L | $\mathbb{B}$ | $\mathbb{B}'$ | $|\mathbb{B}|$ | PT_R
PT_L | $\mathbb{B}_1$ | $\mathbb{B}_1'$ | $|\mathbb{B}_1|$ | PT_R
PT_L | $\mathbb{B}_0$ | $\mathbb{B}_0'$ | $|\mathbb{B}_0|$ | PT_R

Figure 1: Basic structure of the binary tree.

After the construction of the binary tree is completed, we now investigate how to obtain the estimated version of the permutation matrix $\boldsymbol{W}$ from the tree. To facilitate the following discussion, let $\mathbb{B}_i$, $\mathbb{B}_i'$ and $|\mathbb{B}_i|$ denote the middle three items of a leaf node, respectively. Apparently, $\boldsymbol{w}(i,l)$ can be uniquely determined if and only if $|\mathbb{B}_i| = 1$. Otherwise, one has to guess one from $|\mathbb{B}_i|!$ possible cases. For the whole permutation matrix, there are $\Pi_{i=1}^{P}(|\mathbb{B}_i|!)$ possible cases, where P is the number of leaf nodes in the binary tree. For simplicity, we derive the permutation matrix by mapping the elements in $\mathbb{B}_i$ and $\mathbb{B}_i'$ one by one in order.

Next, we investigate how many known-images are sufficient to achieve an acceptable breaking performance. Roughly speaking, $\Pi_{i=1}^{P}(|\mathbb{B}_i|!)$ rapidly decrease when the number of known-images, n_0, is increased. The less value of $\Pi_{i=1}^{P}(|\mathbb{B}_i|!)$ means more accurate estimation of the permutation matrix. To simplify the analyses, we assume that each element in $\boldsymbol{B}$ distributes uniformly over $\{0,1\}$, and any two elements are independent of each other. Then, the elements of $\mathbb{B}_i$ can be divided into the following two types:

- *the sole right position*, which occurs definitely;
- *other fake positions*, each of which occurs in $\mathbb{B}_i$ with a probability of $1/2^{n_0}$, since one condition in Eq. (4) is satisfied consecutively for n_0 times.

Let n_8 denote the number of error bits in a recovered pixel, it can be calculated that $Prob(n_8 = i) = \binom{8}{i} \cdot (1-p_b)^i \cdot (p_b)^{8-i}$, where $p_b = \frac{1}{1+(8MN-1)/2^{n_0}}$, $i = 0 \sim 8$. Generally speaking, when every bit is determined correctly with a probability larger than a half, namely $p_b > 0.5$, the decryption will be acceptable. Solve the inequality, one has

$$n_0 > \lceil \log_2(8MN-1) \rceil. \tag{5}$$

To verify the performance of the above known-plaintext attack, a number of experiments have been performed on a number of randomly selected natural images of size 256 × 256. In this case, Eq. (5) become $n_0 > \lceil \log_2(2^{19}-1) \rceil = 19$. With a randomly selected secret key $(x_0, \mu, m, n, T) = (0.2009, 3.98, 20, 51, 4)$, a plain-image "Peppers" and its encrypted version are shown in Figs. 2a) and b), respectively. Let $\boldsymbol{W}_{20}$ and $\boldsymbol{W}_{25}$ denote the estimated version of the permutation matrix $\boldsymbol{W}$ obtained from 20 and 25 known plain-images, respectively. The decrypted version of Fig. 2b) with $\boldsymbol{W}_{20}$ and $\boldsymbol{W}_{25}$ is shown in Figs. 2c) and d) respectively. It is found that most visual information contained in the original plain-image has been recovered in Fig. 2c), although only $23509/65536 \approx 35.8\%$ of plain pixels are correct in value. This is attributed to the following two main reasons:

- human eyes have a powerful capability for excluding image noises and recognizing significant information;
- due to strong redundancy in natural images, two pixel values are close to each other with a probability larger than the average one, hence many incorrectly recovered pixels are close to their true values with a probability larger than the average one also.

Distribution of difference between the recovered plain-image, shown in Fig. 2c), and the corresponding original plain-image is shown in Fig. 3a). The similar data on the recovered plain-image shown in Fig. 2d) is also illustrated in Fig. 3b) for comparison. With some noise reduction schemes, the recovered plain-images can be enhanced further. The results of two images shown in Figs. 2c) and d) with a 3×3 median filter are shown in Figs. 4a) and b), respectively.

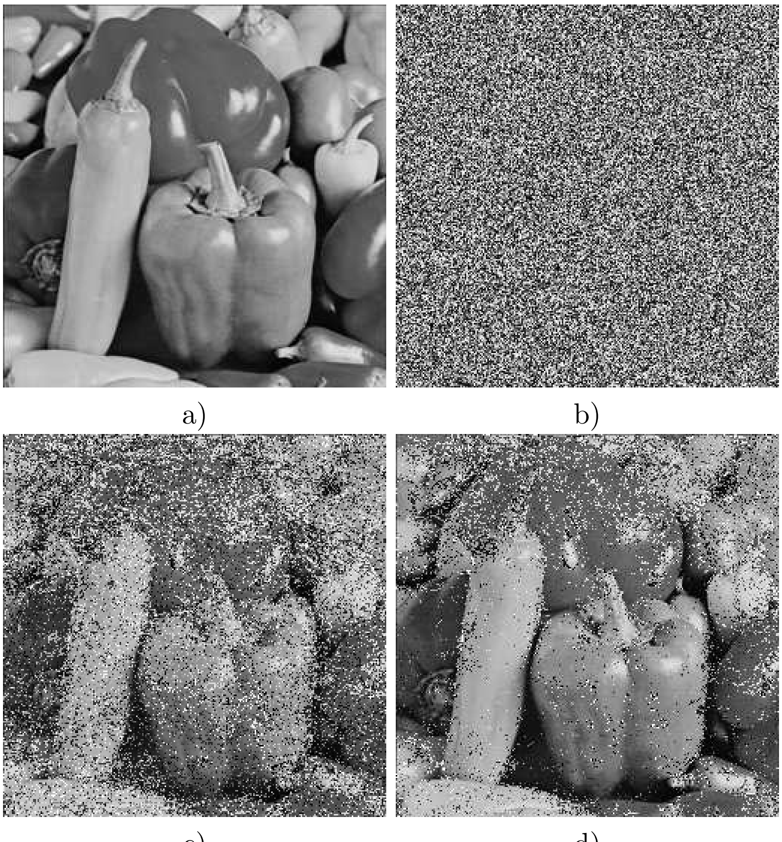


Figure 2: The image "Peppers" recovered by know-plaintext attack: a) "Peppers"; b) the encrypted "Peppers"; c) recovered "Peppers" via $\boldsymbol{W}_{20}$; d) recovered "Peppers" via $\boldsymbol{W}_{25}$.

Figure 5 shows the percentage of correctly-recovered elements, including plain-bit, plain-pixel and the elements of permutation matrix, with respect to the number of known plain-images. One can see that the breaking performance is good when $n_0 \geq 20$. It can also be observed that the slopes of the three lines shown in Fig. 5 are very flat when $n_0 \geq 25$. This is due to the negative impact incurred by the strong redundancy in natural images such as the MSBs of neighboring pixels are the same with a high probability. This point has been proved quantitatively in [20]. Now, one can see that the non-uniform distribution of most natural images has two opposite influences on

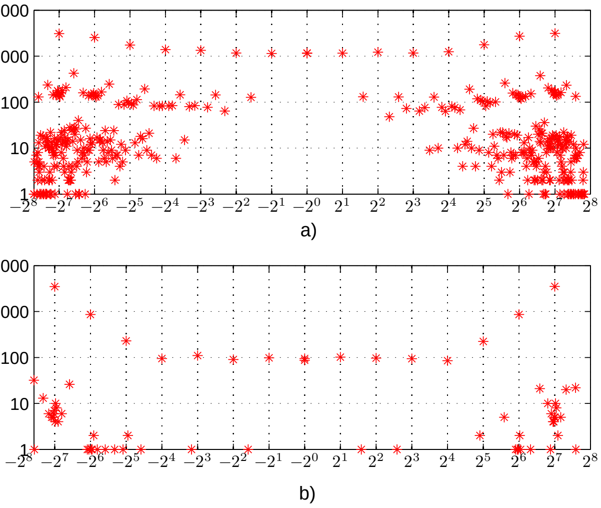
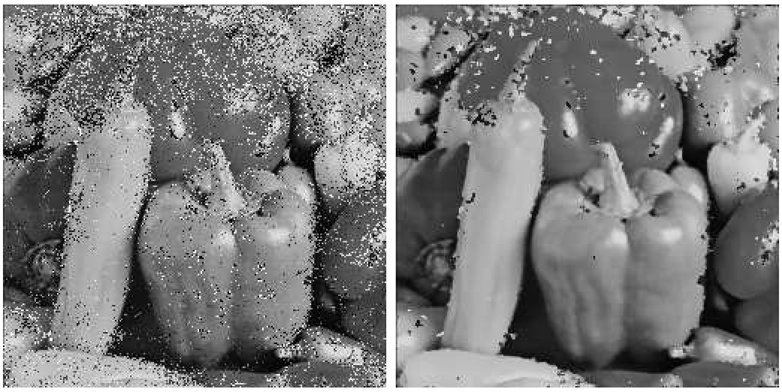


Figure 3: Distribution of non-zero difference between two recovered plain-images and the original plain-image: a) the one shown in Fig. 2c); b) the one shown in Fig. 2d), where x-axis denotes specific difference and y-axis denotes the corresponding counts.

the final breaking. The experiments have shown that the above analysis result obtained under the assumption of uniform distribution also holds for the case of ordinary natural images.



a) b)

Figure 4: The enhanced results of applying a 3 × 3 median filter to the two recovered plain-images: a) the image shown in Fig. 2c); b) the image shown in Fig. 2b).

Obviously, the spatial complexity and the computational complexity of the proposed known-plaintext attack are only $O(32 \cdot MN)$ and $O(16 \cdot n_0 \cdot MN)$, respectively.

3.2. Chosen-plaintext attack

The chosen-plaintext attack is an attack model for cryptanalysis which assumes that the attacker has the capability to arbitrarily choose some plaintexts to be encrypted and observe the corresponding ciphertexts. The goal of the attack is to optimize the attack performance considering the special properties of the encryption scheme when the plaintexts are with a specific structure.

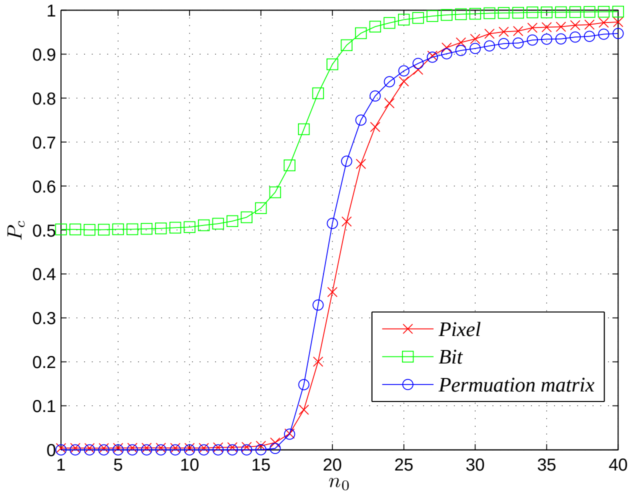


Figure 5: The percentage of correctly-recovered elements with respect to the number of known plain-images.

As for the image permutation algorithm under study, at least $n_c = \lceil \log_2(8MN) \rceil = 3 + \lceil \log_2(MN) \rceil$ chosen plain-images are needed to make cardinality of the set in every leaf node of the binary tree for breaking is equal to one, namely, every element of the permutation matrix can be recovered exactly. First, construct an $M \times (8N)$ matrix in domain $\mathbb{Z}_{2^{n_c}}$, $\boldsymbol{B}^+ = [B^+(i,l)]_{i=0,l=0}^{M-1,8N-1}$, whose elements are different from each other. Then, the chosen plain-images $\{\boldsymbol{I}_t\}_{i=0}^{n_c-1}$ can be chosen/constructed as follows:

$$I_t(i,j) = \sum\nolimits_{k=0}^{7} B_t^+(i,l) \cdot 2^k,$$

where $\sum_{t=0}^{n_c-1} B_t^+(i,l) \cdot 2^t = B^+(i,l)$ and $l = 8 \cdot j + k$.

3.3. Some remarks on the performance on the image permutation algorithm under study

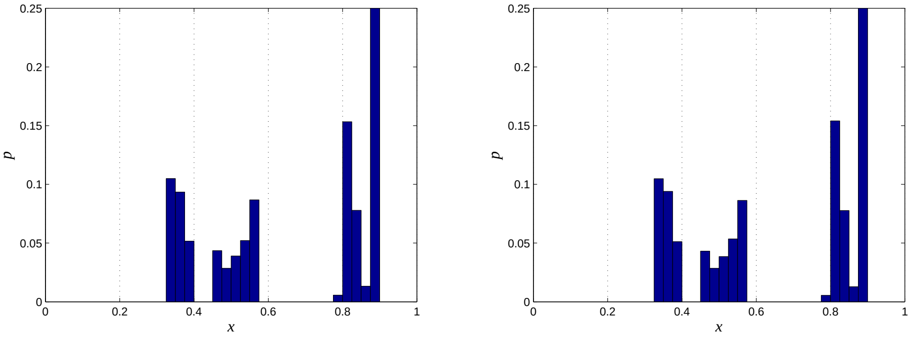
- *Weak randomness of vectors $\boldsymbol{T_M}$ and $\boldsymbol{T_N}$*

  Obviously, randomness of vectors $\boldsymbol{T_M}$ and $\boldsymbol{T_N}$ depends on randomness of the sequences generated by iterating Logistic map. Note that convincing argument for good randomness of a sequence is not the so-called complex theories based but whether it can pass a series of objective tests [21]. As shown in [22, Sec. 3.1], Logistic maps cannot serve as a good random number generator. This point can also be guessed by observing distribution of the trajectory of Logistic map, which is mainly determined by the control parameter [23]. For illustration, distribution of two trajectories of Logistic map with the same initial condition and control parameter used in [19, Sec. 2] is shown in Fig. 6.

- *Fail to encrypt images of fixed value zero or 255*

  For both the two images, all elements of the corresponding intermediate matrix $\boldsymbol{B}^*$ are the same, which make all the encryption procedures do nothing.

- *Insensitivity with respect to changes of plaintext*



a) b)

Figure 6: Distribution of two trajectories of the map Eq. (1) with control parameter $\mu = 3.5786$: a) $x_0 = 0.3333$; b) $x_0 = 0.5656$.

Sensitivity with respect to changes of plaintext is very important for any image encryption scheme since an image and its watermarked version may be encrypted at the same time. In cryptography, the most ideal state about the sensitivity is that the change of any single bit of plaintext will make every bit of the corresponding ciphertext change with a probability of one half. Unfortunately, the image permutation algorithm under study does not consider this property at all.

- *Equivalent sub-key*

  Since $f(x) = f(1-x)$, x_0 and $(1-x_0)$ are equivalent sub-keys for decryption.

- *Unchanged histogram on plain-bit*

  Although the histogram on plain-byte is changed by the image permutation algorithm under study, the one on plain-bit keep unchanged, which make the cipher-image still reveal some information of the plain-image.

- *Low efficiency*

  The generation method of permutation matrixes $\boldsymbol{T_M}$ and $\boldsymbol{T_N}$ make the computational complexity of the image permutation algorithm under study is $O(M\cdot(8N)^2)$, which is even much larger than that of DES.

4. Known/chosen-plaintext attack on any permutation-only multimedia encryption schemes

As shown in [20], no matter what the permuted elements of multimedia data are, all permutation-only multimedia encryption schemes on a plaintext of size $M \times N$ can be represented as

$$I^*(w(i,j)) = I(i,j),$$

where the permutation matrix $\boldsymbol{W} = [w(i,j) = (i',j') \in \mathbb{M} \times \mathbb{N}]_{M\times N}$, $\mathbb{M} = \{0,\cdots,M-1\}$ and $\mathbb{N} = \{0,\cdots,N-1\}$. Then it was analyzed that permutation-only multimedia encryption can be broken with only $O(\lceil \log_L(MN)\rceil)$ known/chosen plaintexts, where L is the number of different elements in the plaintext. The breaking method proposed in [20, Sec. 3.1] includes the following three key steps:

- *Step 1*: obtain a set containing all possible secret positions for each entry of the plaintext by comparing every pair of plaintext and the corresponding ciphertext;
- *Step 2*: solve the intersection of the different sets corresponding to each entry of plaintext;
- *Step 3*: get an estimated version of the permutation matrix by choosing a secret position of an entry plaintext from the final set corresponding to it.

The operations in *Step 2* make the computational complexity of the above attack become $O(n_0 \cdot MN^2)$. It is found that the complex intersection operations can be avoided by extending the idea proposed in Sec. 3.1, namely constructing a multi-branch tree, where every node includes 2^L+3 components: 2^L pointers holding addresses of 2^L possible child nodes; two sets containing some entry positions in plain-image and cipher-image, respectively; and cardinality of one of the two sets. Let $\mathbb{B}$, $\mathbb{B}'$, and $|\mathbb{B}|$ denote the last three items in the root node. Initially, set $\mathbb{B} = \mathbb{B}' = \mathbb{M} \times \mathbb{N}$, and $|\mathbb{B}| = MN$. Then, the multi-branch tree can be constructed as follows.

- $\forall$ $(i, j) \in \mathbb{B}$, add (i, j) to the first set of the child nodes to which the $I(i, j)$-th item of the current node points;
- $\forall$ $(i, j) \in \mathbb{B}'$, add (i, j) to the second set of the child nodes to which the $I'(i, j)$-th item of the current node points.
- Delete the elements of the sets in the root node and set the last item of the node as zero.

With more pairs of known/chosen plain-images and the corresponding cipher-images, the multi branch can be updated and expanded iteratively with the following steps:

- Search for all nodes whose last item is greater than one;
- Expand and update each found node with the similar operations shown above.

Once construction of the multi-branch tree is completed, the permutation matrix $\boldsymbol{W}$ can be estimated by simply mapping the elements in the two sets of every leaf node in order.

Finally, combine the performance analysis of the known/chosen-plaintext attack presented in [20], we can conclude that any permutation-only multimedia encryption schemes exerting on plaintext of size $M \times N$ can be efficiently broken with $O(\lceil \log_L(MN) \rceil)$ known/chosen-plaintexts, and spatial complexity and computational complexity of the attack are $O(MN)$ and $O(n_0 \cdot MN)$, respectively. Substitute n_0 with $\lceil \log_L(MN) \rceil$, the complexity becomes $O(\lceil \log_L(MN) \rceil \cdot MN)$, which is lower much than $O(\lceil \log_L(MN) \rceil \cdot MN^2)$, the whole attack complexity estimated in [20].

5. Conclusion

In this paper, the security of an image permutation encryption algorithm is analyzed. An optimal method, in term of spatial complexity and computational complexity, is proposed to break the binary image permutation algorithm. Furthermore, the method can be extended to break any permutation-only multimedia encryption schemes with an optimal performance also. In addition, some specific remarks on the performance of the image scrambling encryption algorithm under study are provided. Again, this cryptanalysis paper proves that the security of a good image encryption scheme should rely on a combination of traditional cryptography and special properties of image data, like the schemes proposed in [24, 25].

Acknowledgement

The work of Chengqing Li was partially supported by The Hong Kong Polytechnic University's Postdoctoral Fellowships Scheme under grant no. G-YX2L.

References

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

[2] H.-C. Chen, J.-C. Yen, A new cryptography system and its VLSI realization, Journal of Systems Architecture 49 (7–9) (2003) 355–367.

[3] G. Chen, Y. Mao, C. K. Chui, A symmetric image encryption scheme based on 3D chaotic cat maps, Chaos, Solitons & Fractals 21 (3) (2004) 749–761.

[4] N. J. Flores-Carmona, M. Carpio-Valadez, Encryption and decryption of images with chaotic map lattices, Chaos 16 (3) (2006) art. no. 033118.

[5] A. P. Kurian, S. Puthusserypady, Self-synchronizing chaotic stream ciphers, Signal Processing 88 (10) (2008) 2442–2452.

[6] X. Tong, M. Cui, Image encryption scheme based on 3d baker with dynamical compound chaotic sequence cipher generator, Signal Processing 89 (4) (2009) 480–491.

[7] X. Liao, S. Lai, Q. Zhou, A novel image encryption algorithm based on self-adaptive wave transmission, Signal Processing 90 (9) (2010) 2714–2722.

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

[9] C. Li, S. Li, D.-C. Lou, On the security of the Yen-Guo's domino signal encryption algorithm (DSEA), Journal of Systems and Software 79 (2) (2006) 253–258.

[10] K. Wang, W. Pei, L. Zou, A. Song, Z. He, On the security of 3D cat map based symmetric image encryption scheme, Physics Letters A 343 (6) (2005) 432–439.

[11] D. Arroyo, R. Rhouma, G. Alvarez, S. Li, V. Fernandez, On the security of a new image encryption scheme based on chaotic map lattices, Chaos 18 (3) (2008) art. no. 033112.

[12] C. Li, G. Chen, On the security of a class of image encryption schemes, in: Proceedings of 2008 IEEE Int. Symposium on Circuits and Systems, 2008, pp. 3290–3293.

[13] S. Li, C. Li, G. Chen, K.-T. Lo, Cryptanalysis of the RCES/RSES image encryption scheme, Journal of Systems and Software 81 (7) (2008) 1130–1143.

[14] R. Rhouma, S. Belghith, Cryptanalysis of a spatiotemporal chaotic image/video cryptosystem, Physics Letters A 372 (36) (2008) 5790–5794.

[15] C. Li, S. Li, M. Asim, J. Nunez, G. Alvarez, G. Chen, On the security defects of an image encryption scheme, Image and Vision Computing 27 (9) (2009) 1371–1381.

[16] G. Alvarez, S. Li, Cryptanalyzing a nonlinear chaotic algorithm (NCA) for image encryption, Communications in Nonlinear Science and Numerical Simulation 14 (11) (2009) 3743–3749.

[17] G. Álvarez, S. Li, Some basic cryptographic requirements for chaos-based cryptosystems, International Journal of Bifurcation and Chaos 16 (8) (2006) 2129–2151.

[18] 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, 2004, Ch. 4, pp. 133–167.

[19] G. Ye, Image scrambling encryption algorithm of pixel bit based on chaos map, Pattern Recognition Letters 31 (5) (2010) 347–354.

[20] S. Li, C. Li, G. Chen, N. G. Bourbakis, K.-T. Lo, A general quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks, Signal Processing: Image Communication 23 (3) (2008) 212–223.

[21] A. Rukhin, et al., A statistical test suite for random and pseudorandom number generators for cryptographic applications, NIST Special Publication 800-22, available online at `http://csrc.nist.gov/rng/rng2.html` (2001).

[22] C. Li, S. Li, G. Álvarez, G. Chen, K.-T. Lo, Cryptanalysis of two chaotic encryption schemes based on circular bit shift and xor operations, Physics Letters A 369 (1-2) (2007) 23–30.

[23] K. Wang, W. Pei, X. Hou, Y. Shen, Z. He, Symbolic dynamics approach to parameter estimation without initial value, Physics Letters A 374 (1) (2009) 44–49.

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

[25] C.-C. Chang, M.-S. Hwang, T.-S. Chen, A new encryption algorithm for image cryptosystems, Joural of Systems and Software 58 (2) (2001) 83–91.