Cryptanalysis of image ciphers with permutation-substitution network and chaos

Junxin Chen, Lei Chen, Yicong Zhou, *Senior Member, IEEE*

***Abstract*—In recent decades, the introduction of chaos to image encryption has drawn worldwide attention. The permutation-substitution architecture has been widely applied, and chaotic systems are generally employed to produce the required encryption elements. Although many security assessment tests have been conducted, some chaotic image ciphers are being cryptanalyzed. In this paper, we evaluate the security of a family of image ciphers whose encryption kernel consists of a bit-level or pixel-level permutation and a bit-wise exclusive OR substitution. After investigating the intrinsic linearity inside the outfitted structures and encryption techniques, we find that each ciphertext-plaintext pair can be represented as a combination of a set of ciphertext-plaintext bases. A chosen-ciphertext attack is proposed to construct the ciphertext-plaintext bases rather than the traditional solution to retrieve equivalent encryption elements. We further reveal that such weakness cannot be remedied by common enhancements such as more chaotic dynamics, complex permutation methods, and random pixel insertion during encryption. In addition, applications of the proposed attack to break 12 ciphers are theoretically presented and experimentally verified.**

***Index Terms*—Chosen-ciphertext attack, permutation substitution, bit-wise XOR, image encryption**

I. Introduction

In recent years, security and privacy have drawn increased attention by both individuals and governments. Together with the significant progress of multimedia applications over the Internet, the secure transmission of image content has become a hot topic. Encryption is a fundamental solution. Accordingly, traditional block ciphers, such as Advanced Encryption Standard (AES) and Data Encryption Standard (DES), are straightforwardly applicable to encrypting images in binary patterns. Such a technique is called 'naive' encryption and has been reported to lead to unsatisfactory implementation speed and security performance. These two issues further represented the primary motivations in developing specialized image ciphers [1]. Generally, specialized encryption schemes have been advocated to take advantage of the intrinsic properties of image content such as large data volumes and strong pixel correlations [2].

Starting from the encryption of a cat picture in 1967, chaos theory has shown great potential for image encryption [3]. By exploiting the intrinsic relationship between chaos and the famous permutation-substitution structure in cryptography, Fridrich first proposed an architecture for chaos-based image encryption [4]. It was then standardized by Chen [5], [6]. As illustrated in Fig. 1, permutation and substitution serve as the encryption kernel in this structure. Technically, the permutation phase relocates the plain pixels (or bits), while their values are subsequently modified in the substitution procedure. Various chaotic maps or other nonlinear dynamics are introduced to generate the permutation vector and substitution mask. They essentially act as the core secret elements in the encryption process. This structure has led an explosion of chaos-based image encryption schemes. Most such schemes focus on particular techniques for permutation/substitution or on complex chaotic maps for key stream generation. For example, Fridrich's scrambling methods were extended to three dimensions in [5], [6], while bit-level permutation was also developed [7]–[11]. A real-time chaotic video encryption system using a permutation-only technique was given in [12]. Regarding the substitution part, the improvements mainly consist of masking formulas [5] and novel substitution patterns [13]. In [14], [15], secret optical transforms were also introduced to enhance the confidentiality of the substitution phase. Under the assumption that complex chaotic maps give greater randomness to the key stream and thus more security to the encryption scheme, more complex chaotic systems have been employed for image encryption such as cascaded chaotic systems [16] and hyper chaotic maps [17]–[19].

On the other hand, arguments and cryptanalysis of chaos-based image encryption have also been proposed. As mentioned above, using AES or DES directly to encrypt images presented unsatisfactory efficiency and security, which represents the primary motivations of chaotic image encryption [1]. However, negative results were obtained recently [3]. By experimentally implementing popular chaos-based image ciphers and AES, chaos-based image cryptosystems were reported to be severely insufficient. This is because the encryption speed of AES has been significantly increased by resourceful cryptographic libraries, which are unavailable in chaotic cryptography and seem unlikely to be available in the near future. Regarding security concerns, it is widely known that AES with the proper mode of operation can achieve satisfactory security regardless of whether the plaintext is image data or other types

This work is funded by the National Natural Science Foundation of China (nos. 61802055 and 61771121), by The Science and Technology Development Fund, Macau SAR (File no. 189/2017/A3), and by University of Macau (File no. MYRG2018-00136-FST).

Junxin Chen is with the College of Medicine and Biological Information Engineering, Northeastern University, Shenyang 110004, China (E-mail: chenjx@bmie.neu.edu.cn). He is also with the Department of Computer and Information Science, University of Macau, Macau 999078, China.

Lei Chen is with the School of Sciences, Beijing University of Posts and Telecommunications, Beijing 100876, China (Email: clei@bupt.edu.cn).

Yicong Zhou is with Department of Computer and Information Science, University of Macau, Macau 999078, China (E-mail: yicongzhou@um.edu.mo). He is the corresponding author.

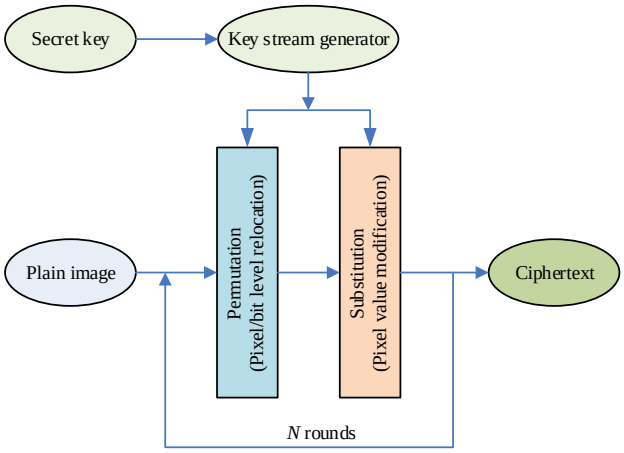


Fig. 1. Architecture of the studied image ciphers.

of data. The security strength has been proven mathematically and experimentally. In contrast, only statistical indicators, such as histograms, information entropy, and NIST (National Institute of Standards and Technology) randomness tests, have been used for security evaluation in chaotic cryptography. Unfortunately, such a statistical evaluation suite has been reported to be insufficient for security declaration [3], [20]. An obvious insecure cipher can pass these tests [3]; on the other hand, many of the so-called 'passed' image ciphers have been cryptanalyzed.

A. Related work

Many achievements focused on cryptanalyzing chaos-based image encryption schemes. There are approximately 195 results in the Web of Science [1].

The overwhelming majority of the image cryptanalysis publications were case specific. The cracking process in [21] benefited from the incomplete structure of the studied cipher, which was constructed without a permutation phase. By studying the intrinsic properties of the Chinese remainder theorem (CRT), Zhu's cipher [22], which used CRT for substitution, was cracked by a chosen-plaintext attack [23]. The bilateral substitution technique adopted in [13] was revealed to significantly decrease the key space, and the cipher was cryptanalyzed by a plaintext attack [24]. Summarizing, the specified cryptanalysis generally benefits from the weakness inside the adopted architecture or permutation/substitution particulars of the studied image cipher.

Some researchers have worked toward generalized cryptanalysis, yet the achievements were usually made under certain constraints. The normalization of permutation-only encryption is a typical approach [25]–[27], where a permutation cipher is generalized as an invertible key-dependent permutation vector. Similarly, the security evaluation of substitution-only image ciphers against a chosen-plaintext attack was conducted in [28]. By studying the cryptographic strength of the 'differential equation of modulo addition' [2], Zhang cracked a class of substitution techniques that evolved from Chen's work in [5]. However, this attack is not possible when the encryption kernel is repeated for many rounds. From these generalized achievements, we can see that some universal security drawbacks may exist in current image ciphers, and the evaluation and generalization of these loopholes may be more valuable than conventional case-by-case works.

Taking overall consideration of the aforementioned limitations in the previous cryptanalysis literature, the keywords describing this work include 'generalized cryptanalysis', 'permutation and substitution' and 'iteratively performed'. They are the primary motivations and innovations of this paper.

B. Our contribution

This paper presents a security evaluation of a family of chaos-based image ciphers. Specifically, the ciphers possessing the following properties may be vulnerable to the proposed attack.

1) The permutation-substitution structure is utilized.
2) Permutation is implemented at the pixel level or bit level, and bit-wise XOR is used for substitution.
3) The encryption kernel, i.e., permutation and substitution, can be iteratively performed.
4) The permutation vector and substitution mask solely depend on the secret key, in other words, the key is independent of the plaintext.

In the literature, 12 image ciphers of this type have been reported [8], [29]–[39]. In this paper, differential cryptanalysis is first conducted, and the common security loopholes inside these ciphers are found. Based on this, a universal chosen-ciphertext attack is proposed. The attack can crack this family of image ciphers without any modification. Our contributions are summarized as follows.

1) The security of a family of image ciphers using bit-level or pixel-level permutation and bit-wise XOR substitution is investigated.
2) Common vulnerabilities of these ciphers have been found, and a universal chosen-ciphertext attack is proposed.
3) The security bound of these ciphers is formalized, and some famous security enhancement techniques are revealed to be useless.
4) Applications of the proposed attack to break 12 image ciphers are theoretically described and experimentally verified.

This paper matches the conclusions in [3], [20], where widely adopted security metrics were reported to be insufficient for security declaration. The proposed cryptanalysis and attack can be regarded as an inheritance and extension of related works in [2], [40], whereas the attack particulars and applicable ciphers are very different from each other. It is noted that image ciphers using plaintext-related key generation mechanisms have been investigated in recent years [41]. The secret key is (partially) correlated with the plaintext, so different plaintexts operate under different keys and the cipher is thus more secure against various attacks. Depending on how the key is produced from the plaintext, case-specific

[1]Searching '(attack OR cryptanalysis OR breaking OR cracking OR (security analysis) OR cryptanalyzing OR comment) AND image AND (cipher OR cryptosystem OR encryption)' in the title domain (Sept 27, 2019).

cryptanalysis of these ciphers can be performed. But it is beyond the scope of this work.

Though this paper indicates that a family of permutation-substitution image ciphers are breakable, yet the permutation-substitution network itself has been proven to be a secure architecture for designing ciphers, as adopted in AES. It is the linearity of the bit-wise XOR substitution and invariance of the permutation vector that make the proposed attack feasible. In conjunction with the cryptanalysis in [2], nonlinear operations are strongly suggested integrating into the substitution kernel. Chen's 'mixed modulo addition and bit-wise XOR' may be a candidate [5]; however, it must be iterated to achieve a higher security level [2]. Similar to AES, substitution with a lookup table is selectable as well. In addition, the permutation vector is suggested correlating with the plaintext [41]. In this scenario, the permutation vector varies from the plaintext and further promote the security of the whole cryptosystem.

C. Organization of the paper

The remainder of this paper is organized as follows. The notations and the studied image ciphers are given in Section II. Starting with a basic encryption model, differential cryptanalysis and the proposed chosen-ciphertext attack are illustrated in Section III. Quantitative analysis and universality of this attack are discussed in Section IV, while its cryptographic applications are described in Section V. Finally, conclusions are drawn in the last section.

II. The Image Ciphers Under Study

This paper will theoretically and experimentally demonstrate that a family of chaotic image ciphers is breakable. In total, 12 image ciphers [8], [29]–[39] are vulnerable to the proposed chosen-ciphertext attack.

A. Notations

Unless otherwise indicated, bold uppercase $\boldsymbol{A}$ is used to denote an assembly. It may be an image (such as a plaintext/ciphertext) or the secret matrix used for encryption (such as the substitution mask). We use lowercase a to represent a variable or element of the corresponding assembly $\boldsymbol{A}$. Uppercase A always denotes a constant. The superscript bracket-within-number $\boldsymbol{A}^{(i)}$ denotes the involved factors in the i^{th} encryption rounds, while the subscript $\boldsymbol{A}_i$ refers to the image index. Some examples and other nomenclatures are illustrated as follows.

- Generally, $\boldsymbol{M}$ and $\boldsymbol{C}$ are used to denote the plaintext and ciphertext in a certain encryption round, respectively.
- The image size is assumed as $H \times W$; thus, the image can be denoted as $\boldsymbol{M} = \{m(1,1), m(1,2), \ldots, m(H,W)\}$, or $\boldsymbol{M} = \{m(1), m(2), \ldots, m(L)\}(L = H \times W)$. In this paper, the vectorial representation is preferable.
- As special cases, $\boldsymbol{M}^{(1)} = \boldsymbol{C}^{(0)}$ both refer to the input plaintext, while $\boldsymbol{C}^{(Out)} = \boldsymbol{C}^{(N)}$ both denote the output ciphertext, where N is the default iteration counts.
- The gray scale of the image is assumed as G, i.e., pixel values are within $[0, G-1]$ and represented by $B = \lceil log_2(G) \rceil$ bits.
- The bit-wise XOR of two images is defined as their differential, denoted as

$$\Delta \boldsymbol{M} = \boldsymbol{M}_1 \oplus \boldsymbol{M}_2. \tag{1}$$

In this paper, we use bit-level permutation to generalize the scrambling operations at both the pixel level and bit level. For images with L pixels and B-bit resolution, a total of $L \times B$ bits have to be relocated in the permutation phase. Following [25]–[27], all of the involved permutation techniques are finalized as a permutation vector in the following analysis. The permutation vector is denoted as $\boldsymbol{WB} = [wb(i) \in \mathbb{LB}], \mathbb{LB} = \{1, 2, \ldots, L \times B\}$, where the element $wb(i)$ represents the coordination of the plain bit that will be shuffled to the i^{th} position in the permutation ciphertext [2]. Obviously, $\boldsymbol{WB}$ is a $\mathbb{LB} \rightarrow \mathbb{LB}$ bijection.

In addition, a function $\mathcal{WB}(\cdot)$ is introduced to normalize the permutation operation, as illustrated in Eq. (2), where $\boldsymbol{P}$ means the permutation ciphertext.

$$\boldsymbol{P} = \mathcal{WB}(\boldsymbol{M}). \tag{2}$$

B. The basic encryption model

The basic encryption model that we start with is constructed referring to Fig. 1, while bit-level permutation and bit-wise XOR substitution are used. The encryption kernel is repeated N rounds, with an individual permutation vector and substitution masks in each iteration. The encryption procedures are as follows.

1) *Key scheduling*. With the secret key *Seed*, produce a series of parameters to generate the permutation vector and substitution mask for the following encryption iterations. They are denoted as $PARA = Init(Seed) = \{Para^{(1)}, Para^{(2)}, \cdots, Para^{(i)}, \cdots, Para^{(N)}\}$.
2) *Encryption element generation*. With $Para^{(\cdot)}$ and the employed chaotic maps, produce the required permutation vector $\boldsymbol{WB}$ and substitution mask $\boldsymbol{K}$ in an encryption round.
3) *Permutation at the bit level*. Stretch the plaintext $\boldsymbol{M}$ into a binary image $\boldsymbol{MB}$, which is then shuffled with $\boldsymbol{WB}$ to produce the scrambled binary image $\boldsymbol{MS}$.
   Finally, convert $\boldsymbol{MS}$ into the pixel level to obtain the permutation ciphertext $\boldsymbol{P}$. In short, the whole bit-level permutation is generalized as Eq. (2).
4) *Substitution using XOR*. Substitute the pixels of $\boldsymbol{P}$ and produce the ciphertext according to

$$\boldsymbol{C} = \boldsymbol{P} \oplus \boldsymbol{K}. \tag{3}$$

5) *Iteration*. Repeat steps 2) – 4) N times with the parameter $Para^{(i)}$ in the i^{th} encryption round. The output ciphertext $\boldsymbol{C}^{(Out)}$ is mathematically obtained as

$$\begin{cases} \boldsymbol{C}^{(Out)} = \boldsymbol{C}^{(N)} \\ \boldsymbol{C}^{(i)} = \mathcal{WB}^{(i)}(\boldsymbol{M}^{(i)}) \oplus \boldsymbol{K}^{(i)} \\ \boldsymbol{M}^{(i)} = \boldsymbol{C}^{(i-1)} \\ \boldsymbol{C}^{(0)} = \boldsymbol{M}^{(1)} \end{cases}. \tag{4}$$

[2]Herein, $wb(i)$ is defined as the inverse of that in [25]–[27] for easing of the following analysis.

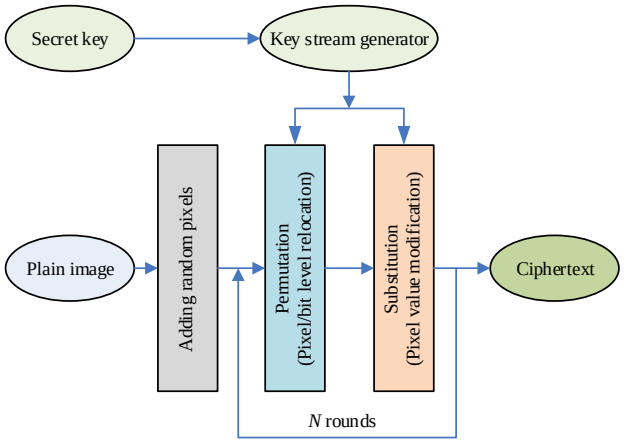


Fig. 2. Chaos-based image cipher with randomly pixel inserting.

Typical image ciphers similar to the basic encryption model can be found in [29]–[32].

C. Variants of the basic model

Based on this basic encryption model, some improved variants were subsequently proposed. Their primary innovations are identified into the following three categories. They are sketched in this subsection, while corresponding encryption details will be given in Section V, together with their cryptanalysis.

1) *Ciphers using substitution linking a previous ciphertext*. The avalanche effect, also called the diffusion performance, is an important security indicator for image ciphers. This means that a tiny difference (usually one bit) of the plaintext should scatter to a large scale into the ciphertext. Obviously, there is no avalanche effect in the basic model. A one-bit difference in the plaintext only affects one bit of the ciphertext, although it may be relocated by the permutation phase. For avalanche performance, researchers have proposed introducing a previous ciphertext into the current substitution operation, as shown in Eq. (5).

$$c(i) = p(i) \oplus k(i) \oplus c(i-1). \tag{5}$$

In this case, a one-bit difference will scatter into the whole ciphertext after several rounds of permutation and substitution. Typical ciphers can be found in [8], [33]–[35].

2) *Ciphers with random variables inserted in the encryption process*. Recently, researchers have proposed adding random pixels before the encryption kernel. Then, this enlarged image is encrypted by the permutation-substitution network. In this scenario, the same plaintext will be encrypted to distinct ciphertexts at different times, although an identical secret key is used. This is the so-called indistinguishability, which gives an encryption scheme satisfactory resistance against plaintext attacks. The encryption process is shown in Fig. 2, and typical ciphers can be found in [36], [37], in which Eq. (5) is adopted for the avalanche effect.

3) *Ciphers with a substitution-then-permutation structure*. In recent years, the permutation-then-substitution structure has shown vulnerability against a stylized cryptanalysis. A chosen-plaintext with identical pixels (such as an all-zero image) was always preferable to initially offset the pixel scrambling effect. Then, cryptanalysts worked toward the substitution phase, and conversely, the permutation vector was ultimately recovered. As a remedy, substitution was proposed to precede the permutation procedure. Typical ciphers can be found in [38], [39]. Furthermore, substitution linking the previous ciphertext (i.e., Eq. (5)), therein inserting a random pixel before core encryption, is also employed in [38] in the hope of improving the security level.

III. The Chosen-Ciphertext Attack

We start with the cryptanalysis of the basic encryption model described in Section II-B, and then, a chosen-ciphertext attack is proposed.

A. Attack assumption

According to Kerckhoffs's principle, a cipher should be secure even if everything about it is openly accessible except for the secret key [42]. By specifying the power of the opponent, four types of attacks are always adopted for the theoretical security analysis of a cipher. These attacks are the so-called chosen-ciphertext attack, chosen-plaintext attack, known-plaintext attack and ciphertext-only attack. By exploring the clues hiding inside the acquired knowledge, the common goal of these attacks is to extract the underlying plaintext of some other ciphertext generated by the same secret key [42].

We evaluate the security of the studied ciphers against the chosen-ciphertext attack. In a chosen-ciphertext attack, the adversary is able to access any ciphertext of his choice, and decrypt them to get the corresponding plaintext [3]. At first sight, this might seem strict. However, when the secret key is fixed by the manufacturer and the opponent can freely manipulate the device for a while, the chosen-ciphertext attack can be launched. Some potential scenarios facing chosen-ciphertext attacks, such as POS (point of sale) terminals and web application session token encryption, are discussed in [42], [43]. In summary, a chosen-ciphertext attack is also an important indicator for evaluating a cipher's practical security.

As mentioned above, the chosen-ciphertext attack aims to learn information about the underlying plaintext of some other ciphertext generated by the same key [42]. In other words, the secret key is assumed to be unchanged in the attack process. Straightforwardly, opponents can attempt to obtain the input secret key. As an alternative, they may also try to retrieve the key-dependent equivalent encryption elements. Regarding the aforementioned studied ciphers in this paper, the permutation vector $\boldsymbol{WB}$ and substitution mask $\boldsymbol{K}$ are key-dependent equivalent encryption elements. They are also assumed unchanged

[3]Of course, the received ciphertext cannot be directly decrypted, and the secret key cannot be obtained.

during the attack because they are completely dependent on the secret key.

It should be emphasized that the studied image ciphers may also be vulnerable to other attacks. However, we focus on their common weakness to a chosen-ciphertext attack.

B. Differential cryptanalysis

Suppose that there are two plaintexts $\boldsymbol{M}_1$, $\boldsymbol{M}_2$ and their ciphertexts $\boldsymbol{C}_1$, $\boldsymbol{C}_2$ in a certain encryption round. Referring to Eqs. (1) and (3), the differential of the ciphertexts is

$$\begin{aligned}\Delta\boldsymbol{C} = \boldsymbol{C}_1 \oplus \boldsymbol{C}_2 &= (\mathcal{WB}(\boldsymbol{M}_1) \oplus \boldsymbol{K}) \oplus (\mathcal{WB}(\boldsymbol{M}_2) \oplus \boldsymbol{K}) \\ &= \mathcal{WB}(\boldsymbol{M}_1) \oplus \mathcal{WB}(\boldsymbol{M}_2)\end{aligned}. \quad (6)$$

Bit-level permutation is adopted in this cipher, i.e., the original bits are scrambled without value modification. In addition, the bits in identical coordinates will be relocated to the same position in the ciphertexts when using identical permutation vectors. Therefore,

$$\mathcal{WB}(\boldsymbol{M}_1) \oplus \mathcal{WB}(\boldsymbol{M}_2) = \mathcal{WB}(\boldsymbol{M}_1 \oplus \boldsymbol{M}_2) = \mathcal{WB}(\Delta\boldsymbol{M}). \quad (7)$$

Combining Eqs. (6) and (7), we can obtain

$$\Delta\boldsymbol{C} = \boldsymbol{C}_1 \oplus \boldsymbol{C}_2 = \mathcal{WB}(\Delta\boldsymbol{M}). \quad (8)$$

Definition 1. *The differential transfer function (DTF) $\mathcal{F}_{(cipher)}(\cdot)$ is defined as the mapping from $\Delta\boldsymbol{M}$ to $\Delta\boldsymbol{C}$ in a certain encryption round, i.e.,*

$$\Delta\boldsymbol{C} = \mathcal{F}_{(cipher)}(\Delta\boldsymbol{M}),$$

where the subscript bracket-within-name denotes the name of the involved cipher.

Referring to Eq. (8), we can obtain the DTF of the basic encryption model as

$$\Delta\boldsymbol{C} = \mathcal{F}_{(basic)}(\Delta\boldsymbol{M}) = \mathcal{WB}(\Delta\boldsymbol{M}). \quad (9)$$

Property 1. *$\mathcal{F}_{(basic)}(\Delta\boldsymbol{M})$ is bijective, which means $\Delta\boldsymbol{M}_1 = \Delta\boldsymbol{M}_2$ if and only if $\mathcal{F}_{(basic)}(\Delta\boldsymbol{M}_1) = \mathcal{F}_{(basic)}(\Delta\boldsymbol{M}_2)$.*

Proof. It is well known that the permutation function $\mathcal{WB}(\cdot)$ is a one-to-one mapping; therefore, $\mathcal{F}_{(basic)}(\Delta\boldsymbol{M})$ has bijectivity. This completes the proof. □

Property 2. *$\mathcal{F}_{(basic)}(\Delta\boldsymbol{M})$ is binary multiplicative, which means*

$$\lambda \times \mathcal{F}_{(basic)}(\Delta\boldsymbol{M}) = \mathcal{F}_{(basic)}(\lambda \times \Delta\boldsymbol{M}), \lambda \in \{0, 1\}.$$

Proof. If $\lambda = 1$, $\lambda \times \mathcal{F}_{(basic)}(\Delta\boldsymbol{M}) = \mathcal{F}_{(basic)}(\lambda \times \Delta\boldsymbol{M})$ accordingly holds. In the case $\lambda = 0$, $\mathcal{F}_{(basic)}(\lambda \times \Delta\boldsymbol{M})$ is obviously an all-zero image because $\lambda \times \Delta\boldsymbol{M}$ is an all-zero image, and $\mathcal{F}_{(basic)}(\cdot)$ refers to a bit-level permutation function without value modification. On the other hand, $\lambda \times \mathcal{F}_{(basic)}(\Delta\boldsymbol{M})$ is also an all-zero image when $\lambda = 0$. Hence, the proof is completed. □

Property 3. *$\mathcal{F}_{(basic)}(\Delta\boldsymbol{M})$ can be bit-wise XORed, which means*

$$\mathcal{F}_{(basic)}(\Delta\boldsymbol{M}_1) \oplus \mathcal{F}_{(basic)}(\Delta\boldsymbol{M}_2) = \mathcal{F}_{(basic)}(\Delta\boldsymbol{M}_1 \oplus \Delta\boldsymbol{M}_2).$$

Proof. The proof is similar to the deduction of Eq. (7). □

Referring to the nomenclature given in Section II-A, we employ $\mathcal{F}^{(i)}_{(basic)}(\Delta\boldsymbol{M}) = \mathcal{WB}^{(i)}(\Delta\boldsymbol{M})$ to represent the DTF of the basic encryption model in the i^{th} iteration. One can observe that the DTFs in distinct iterations may be different from each other, yet Properties 1–3 always hold. In other words, the particulars of $\mathcal{F}^{(i)}_{(basic)}(\Delta\boldsymbol{M})$ are key dependent. However, they are all bijective and binary multiplicative and can be bit-wise XORed. These properties are key independent.

Definition 2. *The cascaded differential transfer function (CDTF) $\mathcal{F}^{(1)-(N)}_{(cipher)}(\cdot)$ is defined as the relationship between the differential matrix of the original input plaintexts, i.e., $\Delta\boldsymbol{M}^{(1)}$, with that of the output ciphertexts, i.e., $\Delta\boldsymbol{C}^{(N)}$, which means*

$$\Delta\boldsymbol{C}^{(N)} = \mathcal{F}^{(1)-(N)}_{(cipher)}(\Delta\boldsymbol{M}^{(1)}). \quad (10)$$

Property 4. *$\mathcal{F}^{(1)-(N)}_{(basic)}(\Delta\boldsymbol{M}^{(1)})$ is also bijective and binary multiplicative and can be bit-wise XORed.*

Proof. The proof is given in the Appendix A. □

Remark 1. *For the basic encryption model, the differential of the original plaintexts is correlated with that of the output ciphertexts as*

$$\Delta\boldsymbol{C}^{(N)} = \mathcal{F}^{(1)-(N)}_{(basic)}(\Delta\boldsymbol{M}^{(1)}).$$

This function is bijective and binary multiplicative and can be bit-wise XORed.

C. Chosen-ciphertext attack

Benefiting from Remark 1, a chosen-ciphertext attack is created as follows. Note that only the output ciphertext $\boldsymbol{C}^{(N)}$ and input plaintext $\boldsymbol{M}^{(1)}$ are available in the attack, whereas the temporary ciphertexts $\boldsymbol{C}^{(i)}(i < N)$ in the intermediate iterations are unobtainable. They are merely introduced for the aforementioned theoretical analysis.

1) Construct $1 + L \times B$ chosen-ciphertexts, where L is the pixel counts of the ciphertext and B refers to the bit counts of a pixel, as mentioned before. These ciphertexts are denoted as $\boldsymbol{C}^{(N)}_0$ and $\boldsymbol{C}^{(N)}_{l-b}, l \in [1, L],\ b \in [1, B]$, and their pixels are constructed by Eq. (11).

$$\begin{aligned}&{c_0}^{(N)}(i) = 0, \quad i \in [1, L] \\ &c^{(N)}_{l-b}(i) = \begin{cases} 2^{B-b}, i = l \\ 0 \quad\ \ , i \neq l, i \in [1, L] \end{cases}\end{aligned}. \quad (11)$$

   In short, $\boldsymbol{C}^{(N)}_{l-b}$ can be constructed one by one by flipping a single bit of an all-zero image ($\boldsymbol{C}^{(N)}_0$), from the first bit of the first pixel to the last bit of the last pixel. Such an arrangement ensures that any ciphertext $\boldsymbol{C}^{(N)}$ is representable by $\boldsymbol{C}^{(N)}_0$ and $\boldsymbol{C}^{(N)}_{l-b}$ with binary multiplication and bit-wise XOR operators.
2) Their input plaintexts are obtainable in the chosen-ciphertext attack, denoted as $\boldsymbol{M}^{(1)}_0$ and $\boldsymbol{M}^{(1)}_{l-b}$ analogously.
3) Obtain the differentials of the plaintexts according to

$$\Delta\boldsymbol{M}^{(1)}_{l-b} = \boldsymbol{M}^{(1)}_{l-b} \oplus \boldsymbol{M}^{(1)}_0. \quad (12)$$

4) For any ciphertext $\boldsymbol{C}^{(N)} = \{c^{(N)}(l), l \in [1, L]\}$, its plaintext is denoted as $\boldsymbol{M}^{(1)}$, whose differential between

$\boldsymbol{M}_0^{(1)}$ is further assumed as $\Delta\boldsymbol{M}^{(1)}$. We can calculate $\Delta\boldsymbol{M}^{(1)}$ from Eq. (13), where $bitC_{l-b}$ denotes the b^{th} bit of the ciphertext $c^{(N)}(l)$ and $\prod$ represents the continuous bit-wise XOR operation.

$$\Delta\boldsymbol{M}^{(1)} = \prod_{l=1}^{L}\prod_{b=1}^{B}[bitC_{l-b} \times \Delta\boldsymbol{M}_{l-b}^{(1)}]. \quad (13)$$

The deduction processes are as follows.

- Considering the involved images as binary matrices, it is easy to obtain $\boldsymbol{C}^{(N)} = \prod_{l=1}^{L}\prod_{b=1}^{B}[bitC_{l-b} \times \boldsymbol{C}_{l-b}^{(N)}]$.
- As $\boldsymbol{C}_0^{(N)}$ is an all-zero image, $\Delta\boldsymbol{C}_{l-b}^{(N)} = \boldsymbol{C}_{l-b}^{(N)} \oplus \boldsymbol{C}_0^{(N)} = \boldsymbol{C}_{l-b}^{(N)}$, and $\Delta\boldsymbol{C}^{(N)} = \boldsymbol{C}^{(N)} \oplus \boldsymbol{C}_0^{(N)} = \boldsymbol{C}^{(N)}$.
- Considering the above two items, we can obtain $\Delta\boldsymbol{C}^{(N)} = \prod_{l=1}^{L}\prod_{b=1}^{B}[bitC_{l-b} \times \Delta\boldsymbol{C}_{l-b}^{(N)}]$.
- Considering that $\Delta\boldsymbol{C}^{(N)} = \mathcal{F}_{(cipher)}^{(1)-(N)}(\Delta\boldsymbol{M}^{(1)})$ and that $\mathcal{F}_{(cipher)}^{(1)-(N)}(\Delta\boldsymbol{M}^{(1)})$ is bijective and binary multiplicative and can be bit-wise XORed, it is easy to deduce that $\Delta\boldsymbol{M}^{(1)} = \prod_{l=1}^{L}\prod_{b=1}^{B}[bitC_{l-b} \times \Delta\boldsymbol{M}_{l-b}^{(1)}]$.

5) Finally, the plaintext $\boldsymbol{M}^{(1)}$ is recovered as

$$\boldsymbol{M}^{(1)} = \Delta\boldsymbol{M}^{(1)} \oplus \boldsymbol{M}_0^{(1)}. \quad (14)$$

D. Pseudocode

Two primary modules can be identified from the aforementioned attack steps.

The first module consists of steps 1) – 3). It is used to produce $\boldsymbol{M}_0^{(1)}$ and the differentials between the plaintexts, i.e., $\Delta\boldsymbol{M}_{l-b}^{(1)}$. To some extent, $\boldsymbol{M}_0^{(1)}$ and $\Delta\boldsymbol{M}_{l-b}^{(1)}$ act as the atoms of the attack. Technically, Algorithm 1 can be utilized for practical implementation. In this and the following pseudocodes, the function $zeros(1, L)$ generates a vector with L elements, and each element has B bits.

Algorithm 1 Generate the atoms of the attack.

Input: Length L of the ciphertext, bit counts B of a pixel
Output: $1 + L \times B$ atoms

$\boldsymbol{C}_0^{(N)} = zeros(1, L)$;
$\boldsymbol{M}_0^{(1)} = decrypt(\boldsymbol{C}_0^{(N)})$;
for each $l \in [1, L]$ **do**
  for each $b \in [1, B]$ **do**
    $\boldsymbol{C}_{l-b}^{(N)} = zeros(1, L)$;
    $c_{l-b}^{(N)}(l) = 2^{B-b}$;
    $\boldsymbol{M}_{l-b}^{(1)} = decrypt(\boldsymbol{C}_{l-b}^{(N)})$;
    $\Delta\boldsymbol{M}_{l-b}^{(1)} = \boldsymbol{M}_{l-b}^{(1)} \oplus \boldsymbol{M}_0^{(1)}$;
  end for
end for
return $\boldsymbol{M}_0^{(1)}$ and $\Delta\boldsymbol{M}_{l-b}^{(1)}$

The second module refers to the image reconstruction part using the latter two steps. Algorithm 2 is the pseudocode for practical implementation. After obtaining the atoms by Algorithm 1, any received ciphertext can be straightforwardly recovered by Algorithm 2 without repeating Algorithm 1.

Algorithm 2 Recovery of the plaintext.

Input: A ciphertext $\boldsymbol{C}^{(N)}$, the atoms $\boldsymbol{M}_0^{(1)}$ and $\Delta\boldsymbol{M}_{l-b}^{(1)}$
Output: The recovered plaintext $\boldsymbol{M}^{(1)}$

$\Delta\boldsymbol{M}^{(1)} = zeros(1, L)$;
for each $l \in [1, L]$ **do**
  for each $b \in [1, B]$ **do**
    $bitC_{l-b} = [c^{(N)}(l)\&2^{B-b}]/2^{B-b}$;
    $\Delta\boldsymbol{M}^{(1)} = \Delta\boldsymbol{M}^{(1)} \oplus [bitC_{l-b} \times \Delta\boldsymbol{M}_{l-b}^{(1)}]$;
  end for
end for
$\boldsymbol{M}^{(1)} = \Delta\boldsymbol{M}^{(1)} \oplus \boldsymbol{M}_0^{(1)}$;
return $\boldsymbol{M}^{(1)}$

E. Example description

An illustrative experiment is given for better understanding. The basic encryption model in Section II-B is employed. The bit-level permutation is performed by chaotic-sorting technique [29], while Logistic map is used for generating the encryption elements, i.e., permutation vector and substitution mask. In addition, the encryption is iterated 10 times. The plaintext is assumed with 8-bit resolution, i.e., $B = 8$ and the pixel values are within the range $[0, 255]$. Interested readers can refer to the source code for more details [4].

Without loss of generality, we randomly construct a plaintext as $\boldsymbol{M}^{(1)} = \{45, 129; 199, 235\}$, which is encrypted to $\boldsymbol{C}^{(10)} = \{12, 35; 65, 230\}$ for confidential transmission. Then, opponents eavesdrop this ciphertext in public channels and attempt to recover the plaintext without the secret key. Following the procedures given in Section III-C, the attack is implemented step by step as follows.

1) Construct $1 + 4 \times 8 = 33$ chosen-ciphertexts according to Eq. (11), which are denoted by $\boldsymbol{C}_0^{(10)}$ and $\boldsymbol{C}_{l-b}^{(10)}, l \in [1, 4],\ b \in [1, 8]$, as listed in Table I.

Table I. The constructed chosen-ciphertexts.

$\boldsymbol{C}_0^{(10)}$	0,0;0,0						
$\boldsymbol{C}_{1-1}^{(10)}$	128,0;0,0	$\boldsymbol{C}_{2-1}^{(10)}$	0,128;0,0	$\boldsymbol{C}_{3-1}^{(10)}$	0,0;128,0	$\boldsymbol{C}_{4-1}^{(10)}$	0,0;0,128
$\boldsymbol{C}_{1-2}^{(10)}$	64,0;0,0	$\boldsymbol{C}_{2-2}^{(10)}$	0,64;0,0	$\boldsymbol{C}_{3-2}^{(10)}$	0,0;64,0	$\boldsymbol{C}_{4-2}^{(10)}$	0,0;0,64
$\boldsymbol{C}_{1-3}^{(10)}$	32,0;0,0	$\boldsymbol{C}_{2-3}^{(10)}$	0,32;0,0	$\boldsymbol{C}_{3-3}^{(10)}$	0,0;32,0	$\boldsymbol{C}_{4-3}^{(10)}$	0,0;0,32
$\boldsymbol{C}_{1-4}^{(10)}$	16,0;0,0	$\boldsymbol{C}_{2-4}^{(10)}$	0,16;0,0	$\boldsymbol{C}_{3-4}^{(10)}$	0,0;16,0	$\boldsymbol{C}_{4-4}^{(10)}$	0,0;0,16
$\boldsymbol{C}_{1-5}^{(10)}$	8,0;0,0	$\boldsymbol{C}_{2-5}^{(10)}$	0,8;0,0	$\boldsymbol{C}_{3-5}^{(10)}$	0,0;8,0	$\boldsymbol{C}_{4-5}^{(10)}$	0,0;0,8
$\boldsymbol{C}_{1-6}^{(10)}$	4,0;0,0	$\boldsymbol{C}_{2-6}^{(10)}$	0,4;0,0	$\boldsymbol{C}_{3-6}^{(10)}$	0,0;4,0	$\boldsymbol{C}_{4-6}^{(10)}$	0,0;0,4
$\boldsymbol{C}_{1-7}^{(10)}$	2,0;0,0	$\boldsymbol{C}_{2-7}^{(10)}$	0,2;0,0	$\boldsymbol{C}_{3-7}^{(10)}$	0,0;2,0	$\boldsymbol{C}_{4-7}^{(10)}$	0,0;0,2
$\boldsymbol{C}_{1-8}^{(10)}$	1,0;0,0	$\boldsymbol{C}_{2-8}^{(10)}$	0,1;0,0	$\boldsymbol{C}_{3-8}^{(10)}$	0,0;1,0	$\boldsymbol{C}_{4-8}^{(10)}$	0,0;0,1

2) Obtain their corresponding plaintexts, given in Table II.
3) Calculate the differentials of the plaintexts; the results are listed in Table III.
4) Decompose the eavesdropped ciphertext $\boldsymbol{C}^{(10)} = \{12, 35; 65, 230\}$ into binary sequence, as demonstrated

[4]The source codes are openly accessible via https://github.com/lurenjia212/Break_bitxor.

Table II. The decryption results of the chosen-ciphertexts.

$\boldsymbol{M}_0^{(1)}$	6,171;223,218						
$\boldsymbol{M}_{1-1}^{(1)}$	6,171;95,218	$\boldsymbol{M}_{2-1}^{(1)}$	6,171;223,210	$\boldsymbol{M}_{3-1}^{(1)}$	134,171;223,218	$\boldsymbol{M}_{4-1}^{(1)}$	6,163;223,218
$\boldsymbol{M}_{1-2}^{(1)}$	6,171;221,218	$\boldsymbol{M}_{2-2}^{(1)}$	6,171;222,218	$\boldsymbol{M}_{3-2}^{(1)}$	4,171;223,218	$\boldsymbol{M}_{4-2}^{(1)}$	7,171;223,218
$\boldsymbol{M}_{1-3}^{(1)}$	6,171;219,218	$\boldsymbol{M}_{2-3}^{(1)}$	6,171;215,218	$\boldsymbol{M}_{3-3}^{(1)}$	2,171;223,218	$\boldsymbol{M}_{4-3}^{(1)}$	14,171;223,218
$\boldsymbol{M}_{1-4}^{(1)}$	6,171;223,154	$\boldsymbol{M}_{2-4}^{(1)}$	6,171;223,222	$\boldsymbol{M}_{3-4}^{(1)}$	6,235;223,218	$\boldsymbol{M}_{4-4}^{(1)}$	6,175;223,218
$\boldsymbol{M}_{1-5}^{(1)}$	6,171;223,219	$\boldsymbol{M}_{2-5}^{(1)}$	6,171;159,218	$\boldsymbol{M}_{3-5}^{(1)}$	6,170;223,218	$\boldsymbol{M}_{4-5}^{(1)}$	70,171;223,218
$\boldsymbol{M}_{1-6}^{(1)}$	6,171;223,202	$\boldsymbol{M}_{2-6}^{(1)}$	6,171;223,216	$\boldsymbol{M}_{3-6}^{(1)}$	6,187;223,218	$\boldsymbol{M}_{4-6}^{(1)}$	6,169;223,218
$\boldsymbol{M}_{1-7}^{(1)}$	6,171;223,90	$\boldsymbol{M}_{2-7}^{(1)}$	6,171;223,250	$\boldsymbol{M}_{3-7}^{(1)}$	6,43;223,218	$\boldsymbol{M}_{4-7}^{(1)}$	6,139,223,218
$\boldsymbol{M}_{1-8}^{(1)}$	6,171;255,218	$\boldsymbol{M}_{2-8}^{(1)}$	6,171;207,218	$\boldsymbol{M}_{3-8}^{(1)}$	38,171;223,218	$\boldsymbol{M}_{4-8}^{(1)}$	222,171;223,218

in Eq. (15).

$$\begin{cases} 12 \Rightarrow \{bitC_{1-1}, \cdots, bitC_{1-8}\} = \{0,0,0,0,1,1,0,0\} \\ 35 \Rightarrow \{bitC_{2-1}, \cdots, bitC_{2-8}\} = \{0,0,1,0,0,0,1,1\} \\ 65 \Rightarrow \{bitC_{3-1}, \cdots, bitC_{3-8}\} = \{0,1,0,0,0,0,0,1\} \\ 230 \Rightarrow \{bitC_{4-1}, \cdots, bitC_{4-8}\} = \{1,1,1,0,0,1,1,0\} \end{cases}. \tag{15}$$

Referring to Eqs. (13), (15) and Table III, we can obtain

$$\begin{aligned} \Delta\boldsymbol{M}^{(1)} &= \prod_{l=1}^{4}\prod_{b=1}^{8}[bitC_{l-b} \times \Delta\boldsymbol{M}_{l-b}^{(1)}] \\ &= \{43, 42; 24, 49\} \end{aligned}.$$

5) Finally, the plaintext $\boldsymbol{M}$ is calculated as

$$\begin{aligned} \boldsymbol{M}^{(1)} &= \Delta\boldsymbol{M}^{(1)} \oplus \boldsymbol{M}_0^{(1)} \\ &= \{43, 42; 24, 49\} \oplus \{6, 171; 223, 218\}. \\ &= \{45, 129; 199, 235\} \end{aligned}$$

The recovered pixels completely match the original plaintext. The proposed attack is therefore validated.

In addition, Fig. 3 demonstrates a set of experimental results using the 256 gray-scale Lena image with a size of 256×256. The image recovered by the proposed attack, shown in Fig. 3(f), has been numerically verified to be identical to the original plaintext.

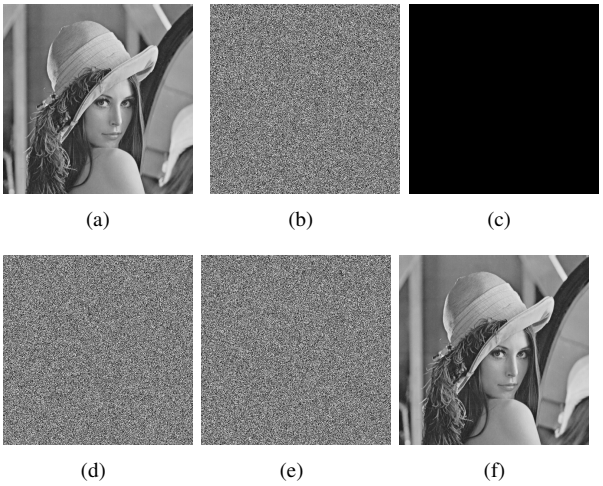


Fig. 3. Results of the proposed attack on cracking the basic encryption model: (a) plaintext Lena; (b) ciphertext of Lena; (c) the first chosen-ciphertext $\boldsymbol{C}_0^{(10)}$; (d) decryption result $\boldsymbol{M}_0^{(1)}$ of the first chosen-ciphertext $\boldsymbol{C}_0^{(10)}$; (e) the calculated differential image $\Delta\boldsymbol{M}^{(1)}$; (f) the recovered image.

IV. Quantitative analysis and universality

Hereinafter, the proposed chosen-ciphertext attack in Section III-C is abbreviated as PCCA. A cipher's DTF/CDTF has BMX properties, which means that it is bijective and binary multiplicative and can be bit-wise XORed.

A. Complexity

As mentioned above, $1 + L \times B$ chosen-ciphertexts and corresponding plaintexts are required in the attack; hence, both the computational and spatial complexity are $O((L \times B)^2)$. Once the atoms are produced by PCCA's first three steps, each received ciphertext can be recovered using a series of binary multiplication and bit-wise XOR operations.

Note that the complexity is independent of the iteration counts N. This is quite different from peer cryptanalysis, whose computation and storage requirements always explode with further encryption rounds such as in the security analysis of Fridrich's cipher given in [44], [45]. In addition, it should be emphasized that the atoms of the attack, i.e., $\boldsymbol{M}_0^{(1)}$ and $\Delta\boldsymbol{M}_{l-b}^{(1)}$, are required to be established only once (Algorithm 1). After that, all of the received ciphertexts are directly recoverable, referring to the attack steps 4) –5) (Algorithm 2).

B. Quantitative analysis

As aforementioned, a chosen-ciphertext attack means that the adversary can access any ciphertext of this choice and get the corresponding plaintext. Regarding PCCA, the adversary is able to decrypt the elaborately constructed $\boldsymbol{C}_0^{(N)}$ and $\boldsymbol{C}_{l-b}^{(N)}$ and obtain $\boldsymbol{M}_0^{(1)}$ and $\boldsymbol{M}_{l-b}^{(1)}$. With the clues underlying these ciphertext-plaintext pairs, the plaintext of any ciphertext can be exactly recovered with Eqs. (12)-(14). For the targeted ciphers, PCCA is capable of recovering the plaintext with a 100% success rate in a chosen-ciphertext attack.

In addition, PCCA is also beneficial for an adversary who doesn't have sufficient resources to implement a chosen-ciphertext attack. For example, when the adversary has collected most of the attack elements required in Eqs. (12)-(14) except certain $\Delta\boldsymbol{M}_{l-b}^{(1)}$, correct attack result is also achievable if corresponding $bitC_{l-b}$ equals to zero coincidently. A guessed $\Delta\boldsymbol{M}_{l-b}^{(1)}$ can be introduced as a replacement. Referring to Eq. (13), $\Delta\boldsymbol{M}_{l-b}^{(1)}$ cannot change the value of $\Delta\boldsymbol{M}^{(1)}$ when $bitC_{l-b} = 0$, furthermore, the attack result produced in Eq. (14) is also unaffected. It is important to note that this scenario is not a chosen-ciphertext attack which always means that any ciphertext can be constructed and decrypted on demand.

Table III. The differential matrices of the decrypted images.

$\Delta \boldsymbol{M}_{1-1}^{(1)}$	0,0;128,0	$\Delta \boldsymbol{M}_{2-1}^{(1)}$	0,0;0,8	$\Delta \boldsymbol{M}_{3-1}^{(1)}$	128,0;0,0	$\Delta \boldsymbol{M}_{4-1}^{(1)}$	0,8;0,0
$\Delta \boldsymbol{M}_{1-2}^{(1)}$	0,0;2,0	$\Delta \boldsymbol{M}_{2-2}^{(1)}$	0,0;8,0	$\Delta \boldsymbol{M}_{3-2}^{(1)}$	2,0;0,0	$\Delta \boldsymbol{M}_{4-2}^{(1)}$	1,0;0,0
$\Delta \boldsymbol{M}_{1-3}^{(1)}$	0,0;4,0	$\Delta \boldsymbol{M}_{2-3}^{(1)}$	0,0;8,0	$\Delta \boldsymbol{M}_{3-3}^{(1)}$	4,0;0,0	$\Delta \boldsymbol{M}_{4-3}^{(1)}$	8,0;0,0
$\Delta \boldsymbol{M}_{1-4}^{(1)}$	0,0;0,64	$\Delta \boldsymbol{M}_{2-4}^{(1)}$	0,0;0,4	$\Delta \boldsymbol{M}_{3-4}^{(1)}$	0,64;0,0	$\Delta \boldsymbol{M}_{4-4}^{(1)}$	0,4;0,0
$\Delta \boldsymbol{M}_{1-5}^{(1)}$	0,0;0,1	$\Delta \boldsymbol{M}_{2-5}^{(1)}$	0,0;64,0	$\Delta \boldsymbol{M}_{3-5}^{(1)}$	0,1;0,0	$\Delta \boldsymbol{M}_{4-5}^{(1)}$	64,0;0,0
$\Delta \boldsymbol{M}_{1-6}^{(1)}$	0,0;0,16	$\Delta \boldsymbol{M}_{2-6}^{(1)}$	0,0;0,2	$\Delta \boldsymbol{M}_{3-6}^{(1)}$	0,16;0,0	$\Delta \boldsymbol{M}_{4-6}^{(1)}$	0,2;0,0
$\Delta \boldsymbol{M}_{1-7}^{(1)}$	0,0;0,128	$\Delta \boldsymbol{M}_{2-7}^{(1)}$	0,0;0,32	$\Delta \boldsymbol{M}_{3-7}^{(1)}$	0,128;0,0	$\Delta \boldsymbol{M}_{4-7}^{(1)}$	0,32;0,0
$\Delta \boldsymbol{M}_{1-8}^{(1)}$	0,0;32,0	$\Delta \boldsymbol{M}_{2-8}^{(1)}$	0,0;16,0	$\Delta \boldsymbol{M}_{3-8}^{(1)}$	32,0;0,0	$\Delta \boldsymbol{M}_{4-8}$	16,0;0,0

C. Universality

The CDTF's BMX properties are the most critical issue for this attack. If a cipher's DTF has BMX, its CDTF also has BMX. Therefore, we can conclude that if a cipher's DTF or CDTF has BMX features, it is vulnerable to PCCA. We find that this security drawback exists in a family of image encryption schemes, which are essentially variants of the studied basic model. They have DTFs or CDTFs possessing BMX properties inside, whereas they usually share the following similar architecture in appearance.

1) They are based on the iterative permutation-substitution network.
2) Pixel-level or bit-level permutation is used, while bit-wise XOR is employed for substitution.
3) The permutation vector and substitution mask are key dependent.

We find that the image encryption schemes in [8], [29]–[39] fall within the scope of PCCA. In addition, it is indicated that the PCCA is independent of the permutation vector, substitution mask and encryption rounds. For a cipher whose DTF or CDTF has BMX drawbacks, we can therefore conclude that the following techniques are infeasible for security enhancement.

1) Introducing more random nonlinear dynamics for producing the permutation vector and substitution mask.
2) Developing complex permutation techniques, regardless of whether they are performed at the pixel level or bit level.
3) Increasing the encryption rounds, where even round keys, i.e., different encryption elements ($\boldsymbol{WB}$ and $\boldsymbol{K}$), are used.
4) It will be further demonstrated in Section V that substitution chaining with a previous ciphertext, inserting random pixels in the encryption, and repositioning the permutation and substitution procedures are also ineffective.

V. Applications to the Cipher Family

In this section, we will theoretically and experimentally demonstrate that PCCA is feasible for a family of chaos-based image ciphers. These ciphers were proposed in [8], [29]–[39].

A. Applications to ciphers similar to the basic model

The ciphers proposed in [29]–[32] are very similar to the basic encryption model described in Section II-B [5]. Specifically, they usually adopt a certain bit-level or pixel-level permutation method to shuffle the plain image and then modify the pixel values via $p(n) \oplus k(n)$. Their primary innovations lie in the novel permutation techniques or complex chaotic maps, which have been found to be ineffective for resisting PCCA. Since they are very similar to the basic encryption model, their description and cryptanalysis are sketched as follows [6].

1) Li's cipher in [29].
   - *Brief review*. With the key $Seed$, a sequence $\boldsymbol{X}$ is first generated by the Logistic map and then mapped to $\boldsymbol{Y}$ and $\boldsymbol{Z}$ using a 'projective transform' technique. The permutation vector $\boldsymbol{WB}$ is produced by $\boldsymbol{Y}$, while the pixel mask $\boldsymbol{K}$ is quantized from $\boldsymbol{Z}$. The plain image $\boldsymbol{M}$ is shuffled at the pixel level to obtain the permutation ciphertext $\boldsymbol{P}$, and the ciphertext is subsequently produced according to $\boldsymbol{C} = \boldsymbol{P} \oplus \boldsymbol{K}$.
   - *Cryptanalysis*. This scheme possesses an identical architecture to the basic encryption model; thus, it is easy to conclude that its DTF has BMX properties. The generation mechanism of the chaotic variable is Li's primary contribution [29]. However, it has been found to be ineffective for security enhancement because PCCA is independent of the encryption elements.
2) Tang's cipher in [30].
   - *Brief review*. There are three stages of encryption. First, the plaintext is randomly divided into blocks, which are then shuffled. Second, individual pixel-level permutation is performed for each block. Third, bit-wise XOR is implemented between the shuffled image $\boldsymbol{P}$ and pixel masks $\boldsymbol{K}$ for substitution. The encryption elements in these steps are all derived from the secret key $Seed$.
   - *Cryptanalysis*. Apparently, we can combine the first two shuffling procedures into a single permutation process. Thus, this cipher is finalized as a standard permutation-substitution encryption scheme. Similar

[5] In [46], [47], Diaconu proposed an image cipher consisting of one round of plaintext-related (not our focused key dependent) permutation and bit-wise XOR substitution. A slightly modified version of the proposed attack is feasible; interested readers can refer to the source codes for more details.

[6] The notations here may be different from those in the original publications, but the encryption cores are identical.

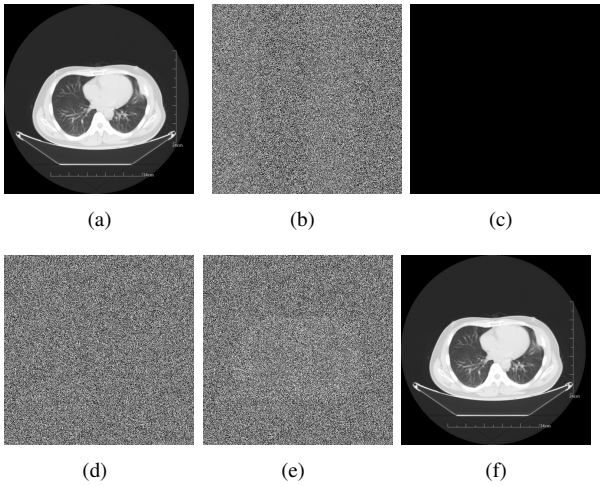


Fig. 4. Results of the proposed attack on cracking Dai's cipher [32]: (a) plaintext CT_Abdomen; (b) ciphertext of CT_Abdomen; (c) the first chosen-ciphertext $\boldsymbol{C}_0^{(1)}$; (d) decryption result $\boldsymbol{M}_0^{(1)}$ of the first chosen-ciphertext $\boldsymbol{C}_0^{(1)}$; (e) calculated differential image $\Delta\boldsymbol{M}^{(1)}$; (f) the recovered image.

to the basic encryption model, it is breakable by PCCA. The primary contribution is the permutation technique using random dividing and shuffling operations; however, it cannot increase the security level.

3) Tang's cipher in [31].
   - *Brief review*. Four plaintexts are combined into one binary matrix, which is then shuffled using the random division and shuffling method developed in [30]. Four images are decomposed from the permutation ciphertext, and then, bit-wise XOR substitution is employed for pixel value modification.
   - *Cryptanalysis*. The permutation approach developed in [30] is implemented at the bit level in this cipher; they [30], [31] suffer from the same vulnerability against PCCA accordingly.
4) Dai's cipher in [32].
   - *Brief review*. The plain image $\boldsymbol{M}$ is first decomposed into binary planes. The Cat map is then employed to shuffle the highest two bit planes 100 times and the following two bit planes 50 times, whereas the other bit planes remain unchanged. Subsequently, $\boldsymbol{P}\oplus\boldsymbol{K}$ is used for substitution. The Logistic map and Henon map are employed for key stream generation.
   - *Cryptanalysis*. Although this cipher uses selective permutation for different bit planes, the entire scrambling process can also be summarized as a single permutation vector $\boldsymbol{WB}$. The PCCA is thus applicable.

The experimental results of cryptanalyzing Dai's cipher [32] are demonstrated in Fig. 4. Numerical comparison verifies that the original input image has been accurately recovered.

Remark 2. *Based on the basic model, improvements in terms of more complex dynamics, novel permutation techniques, and more iterations with round keys are ineffective at enhancing security against PCCA.*

B. Applications to ciphers with substitution linking a previous ciphertext

Ciphers in this category [8], [33]–[35] perform substitution linking a previous ciphertext, i.e., $c(i) = p(i) \oplus k(i) \oplus c(i-1)$, to obtain the avalanche effect. Taking Fu's cipher [8] as an example, we will show that such an enhancement is also insecure against PCCA.

Fu's cipher [8] is reviewed as follows.

1) *Key scheduling*. The control parameters of the Cat map for the permutation, control parameter and initial value of the Logistic map for substitution jointly constitute the secret key.
2) *Encryption element generation*. With the secret key and Cat map, B distinct permutation vectors are generated, i.e., $\boldsymbol{WB}_1, \boldsymbol{WB}_2, \cdots, \boldsymbol{WB}_B$, and a pixel masking sequence $\boldsymbol{K}$ is produced using the Logistic map.
3) *Permutation*. The plain image $\boldsymbol{M}$ is decomposed into B binary planes, which are shuffled independently using the produced permutation vectors. Then, the scrambled bit planes are re-combined as the permutation ciphertext $\boldsymbol{P}$.
4) *Substitution*. The pixels are encrypted one by one according to Eq. (5), where $c(0)$ is set as a constant for masking the first pixel. The ciphertext $\boldsymbol{C}$ is thus generated.
5) *Iteration*. The above processes are iteratively performed based on the security and efficiency requirements.

The cryptanalysis starts with the substitution of Fu's cipher. It is easy to rewrite Eq. (5) as

$$c(i) = c(0) \oplus \prod_{j=1}^{i} p(j) \oplus \prod_{j=1}^{i} k(j), i \in [1, L].$$

Suppose that there are two plaintexts $\boldsymbol{M}_1, \boldsymbol{M}_2$ and their ciphertexts $\boldsymbol{C}_1$, $\boldsymbol{C}_2$ in a certain encryption round. The differential of these two ciphertexts is calculated as

$$\begin{aligned}\Delta c(i) &= c_1(i) \oplus c_2(i)\\ &= c(0) \oplus \prod_{j=1}^{i} p_1(j) \oplus \prod_{j=1}^{i} k(j)\\ &\oplus c(0) \oplus \prod_{j=1}^{i} p_2(j) \oplus \prod_{j=1}^{i} k(j).\\ &= \prod_{j=1}^{i} p_1(j) \oplus \prod_{j=1}^{i} p_2(j)\\ &= \prod_{j=1}^{i} \Delta p(j)\end{aligned} \tag{16}$$

Without loss of generality, we use $\Delta\boldsymbol{C} = \prod(\Delta\boldsymbol{P})$ to finalize the mapping from $\Delta\boldsymbol{P}$ to $\Delta\boldsymbol{C}$ in matrix form. In addition, we can combine the permutation effects of $\boldsymbol{WB}_1, \boldsymbol{WB}_2, \cdots, \boldsymbol{WB}_B$ as one bit-level permutation vector $\boldsymbol{WB}$, i.e., $\Delta\boldsymbol{P} = \mathcal{WB}(\Delta\boldsymbol{M})$. Therefore, we obtain the DTF of Fu's cipher as

$$\Delta\boldsymbol{C} = \mathcal{F}_{(Fu)}(\Delta\boldsymbol{M}) = \prod[\mathcal{WB}(\Delta\boldsymbol{M})]. \tag{17}$$

Appendix B summarizes the BMX properties of $\mathcal{F}_{(Fu)}(\Delta\boldsymbol{M})$. The PCCA is able to break this cipher without any modification.

Cryptanalysis of peer ciphers [33]–[35] of this type is briefly given as follows.

1) Zhou's cipher in [33].

- *Brief review*. First, the plaintext $\boldsymbol{M}$ is shuffled with the permutation vector $\boldsymbol{WB}$ to obtain the permutation ciphertext $\boldsymbol{P}$. Subsequently, a row-by-row and column-by-column substitution is successively performed. For each row/column substitution, only the first pixel is encrypted, whereas other pixels are modified by the previous ciphertext. Taking the i^{th} row as an example, $c(i,1) = p(i,1) \oplus k_r(i)$ while $c(i,j) = p(i,j) \oplus c(i,j-1), j \neq 1$. The permutation vector $\boldsymbol{WB}$ and row/column substitution mask $\boldsymbol{K}_r$/$\boldsymbol{K}_c$ are produced by the Henon map and Line map.
- *Cryptanalysis*. The substitution is indeed a particular case of $c(i,j) = p(i,j) \oplus k(i,j) \oplus c(i,j-1)$. This is the same as in the scenario where $k(i,j) = 0$ ($j \neq 1$) in the row-by-row substitution process and $k(i,j) = 0$ ($i \neq 1$) in the column-by-column substitution. Since the PCCA is independent of the encryption elements, it is consequently able to break this cipher.

2) Mirzaei's cipher in [34].
   - *Brief review*. First, divide the plaintext $\boldsymbol{M}$ into four identical blocks, which are then shuffled with a random sequence $\boldsymbol{WA}$. Second, scramble the image again with shuffling vectors $\boldsymbol{WR}$ and $\boldsymbol{WC}$. Third, modify the plain pixels according to $c_j(i) = p_j(i) \oplus k_j(i) \oplus c_{mod(j-1,4)}(i-1)$, where $c_j(i)$ denotes the i^{th} pixel in the j^{th} block. The encryption elements $\boldsymbol{WA}$, $\boldsymbol{WR}$, $\boldsymbol{WC}$ and $\boldsymbol{K}$ all depend on the secret key.
   - *Cryptanalysis*. In this cipher, the substitution process for $p(i)$ is not linked to $c(i-1)$ but rather to a ciphertext in another block. Its position is denoted as $ws(i)$ without loss of generality. The most important observation is that $ws(i)$ is fixed and never visited again in the entire substitution process. This is the same as in the scenario where another permutation vector $\boldsymbol{WS}$ is implemented to the ciphertext that is produced by $c(i) = p(i) \oplus k(i) \oplus c(i-1)$. Padding another permutation phase after the substitution process cannot significantly enhance the security; even multiple rounds of encryption are vulnerable to PCCA.

3) Ye's cipher in [35].
   - *Brief review*. First, two images are individually derived from the plaintext's higher four and lower four bit planes. The discretized Cat map is then employed for shuffling these images to produce the permutated image $\boldsymbol{P}$. Second, pixel substitution is performed row by row and then column by column by Eq. (5), where the substitution masks $\boldsymbol{K}$ are generated from a continuous Cat map.
   - *Cryptanalysis*. This cipher can be regarded as a permutation-substitution-permutation-substitution cipher when nothing is done in the second permutation. Since PCCA is effective for an iterative permutation-substitution network, it is consequently able to crack this cipher.

Fu's cipher [8] is employed as an example, the experimental results are shown in Fig. 5, therein assuming that the encryption kernel is repeated 3 times. The recovered image (Fig. 5(f)) has been numerically verified identical to the plaintext.

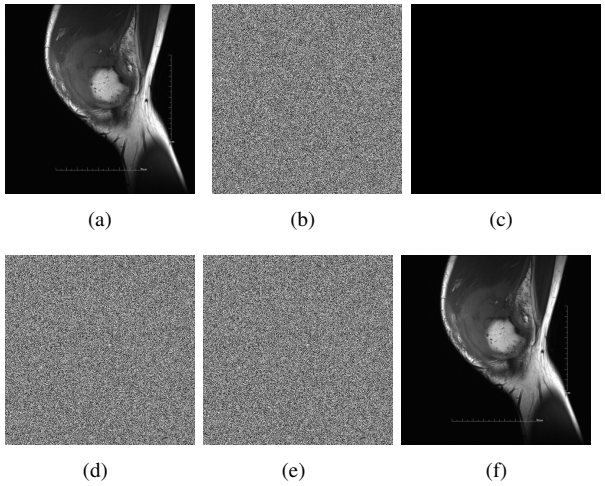


Fig. 5. Results of the proposed attack on cracking Fu's cipher [8]: (a) plaintext MRI_Knee; (b) ciphertext of MRI_Knee; (c) the first chosen-ciphertext $\boldsymbol{C}_0^{(3)}$; (d) decryption result $\boldsymbol{M}_0^{(1)}$ of the first chosen-ciphertext $\boldsymbol{C}_0^{(3)}$; (e) calculated differential image $\Delta\boldsymbol{M}^{(1)}$; (f) the recovered image.

Remark 3. *Substitution linking a previous ciphertext can achieve avalanche performance; however, it is also vulnerable to the proposed cryptanalysis. The BMX properties exist regardless, and the PCCA is applicable in a straightforward manner.*

C. Applications to ciphers with random variables inserted in the encryption process

As mentioned above, adding random pixels before the core encryption process can produce indistinguishability and further obtain satisfactory resistance against plaintext attacks [36], [37]. However, such an improved method is also found to be insecure against PCCA.

Hua's cipher in [36] is first employed as an example, and its encryption processes are described as follows.

1) *Key scheduling*. The secret key $Seed$ is a 232-bit binary sequence, which is used to generate the parameters of the employed 2D logistic-adjusted-sine map (2D-LASM).
2) *Inserting random pixels*. Add $2 \times M + 2 \times N + 4$ random pixels around the input plaintext $\boldsymbol{M}$ to obtain an enlarged image $\boldsymbol{MI}$ of size $(H+2) \times (W+2)$.
3) *Encryption element generation*. With $Seed$ and the 2D-LASM, produce the permutation vector $\boldsymbol{WB}$ and pixel masking sequence $\boldsymbol{K}$. They are essentially dependent on the key and uncorrelated with the plaintext [7]
4) *Permutation*. Using $\boldsymbol{WB}$ to shuffle $\boldsymbol{MI}$ at the pixel level, the permutation ciphertext is denoted as $\boldsymbol{P}$.

[7]In [36], $\boldsymbol{WB}$ is obtained by sorting a sequence whose particle is one-by-one produced by combining the bits of a chaotic variable BS and a plain pixel BP as one variable. Since BP is placed in the lower positions, it cannot influence the sorting results and thus essentially has nothing to do with the permutation vector.

5) *Substitution*. The substitution process is performed by Eq. (18). Note that $L = (H+2) \times (W+2)$ here, referring to the pixel counts of the ciphertext.

$$\begin{cases} c(i) = p(i) \oplus k(i) \oplus p(L), i = 1 \\ c(i) = p(i) \oplus k(i) \oplus c(i-1), i \neq 1 \end{cases}. \tag{18}$$

6) *Iteration*. The permutation and substitution processes are repeated twice, using different permutation vectors and substitution masks in each round.

Without loss of generality, the randomly inserted variables are denoted as $\boldsymbol{R}$, and the symbol $||$ is introduced to denote the pixel insertion operation. Therefore, we can obtain

$$\boldsymbol{MI}^{(1)} = \boldsymbol{M}^{(1)} || \boldsymbol{R}.$$

Note that the pixel insertion operation is implemented only once, which means that $\boldsymbol{MI}^{(1)}$ is the actual input of the permutation-substitution network. On the other hand, it also denotes the products of the permutation-substitution network in the decryption phase before removing the edge pixels. Similar to Fu's cipher, the differential of the enlarged plaintexts $\boldsymbol{MI}$ is correlated with that of the ciphertexts $\boldsymbol{C}$ in a certain encryption round as Eq. (19), where $\Delta p(L)$ denotes the last pixel of $\mathcal{WB}(\Delta \boldsymbol{MI})$.

$$\begin{cases} \Delta \boldsymbol{C} = \mathcal{FI}_{(Hua)}(\Delta \boldsymbol{MI}) \\ \mathcal{FI}_{(Hua)}(\Delta \boldsymbol{MI}) = \Delta p(L) \oplus \prod [\mathcal{WB}(\Delta \boldsymbol{MI})] \end{cases}. \tag{19}$$

Referring to the BMX properties of $\mathcal{F}_{(Fu)}(\Delta \boldsymbol{M})$ given in Appendix B, we can easily conclude that $\mathcal{FI}_{(Hua)}(\Delta \boldsymbol{MI})$ also possesses BMX properties.

It should be emphasized that $\boldsymbol{R}$ is random and unobtainable in the encryption process; however, it is well-defined in the decryption procedure, although it is also unobtainable. This is also the case for $\boldsymbol{MI}^{(1)}$.

Following the attack procedures given in Section III-C, PCCA is found to be feasible for Hua's cipher.

1) Obviously, the required $1+L \times B$ chosen-ciphertexts $\boldsymbol{C}_0^{(2)}$ and $\boldsymbol{C}_{l-b}^{(2)}, l \in [1, L],\ b \in [1, B]$ can be constructed. All of them have a size of $(H+2) \times (W+2)$.
2) Subsequently, $1 + L \times B$ decryption results, $\boldsymbol{M}_0^{(1)}$ and $\boldsymbol{M}_{l-b}^{(1)}$, are obtainable. However, their sizes are all $H \times W$.
3) We can obtain the differentials of the plaintexts as

$$\Delta \boldsymbol{M}_{l-b}^{(1)} = \boldsymbol{M}_{l-b}^{(1)} \oplus \boldsymbol{M}_0^{(1)}. \tag{20}$$

4) For any ciphertext $\boldsymbol{C}^{(2)} = \{c^{(2)}(l), l \in [1, L]\}$, its plaintext is denoted as $\boldsymbol{M}^{(1)}$, and the differential between $\boldsymbol{M}^{(1)}$ and $\boldsymbol{M}_0^{(1)}$ is assumed as $\Delta \boldsymbol{M}^{(1)}$. We can obtain $\Delta \boldsymbol{M}^{(1)}$ by Eq. (21), where $bitC_{l-b}$ denotes the b^{th} bit of pixel $c^{(2)}(l)$, as mentioned above.

$$\Delta \boldsymbol{M}^{(1)} = \prod_{l=1}^{L} \prod_{b=1}^{B} [bitC_{l-b} \times \Delta \boldsymbol{M}_{l-b}^{(1)}]. \tag{21}$$

The proof can be found in Appendix C.

5) Thus, the plaintext is recovered as $\boldsymbol{M}^{(1)} = \Delta \boldsymbol{M}^{(1)} \oplus \boldsymbol{M}_0^{(1)}$.

In conclusion, PCCA can be directly launched to crack Hua's cipher [36] without any modification. This is because no pixel-chaining operation exists in PCCA, and the unknown inserted random pixels cannot affect the recovery of the plain pixels. In addition, the raw materials ($bitC_{l-b}$, $\boldsymbol{M}_0^{(1)}$ and $\Delta \boldsymbol{M}_{l-b}^{(1)}$) are sufficient. Hua's source codes are employed for verification [36] [8], and the results are presented in Fig. 6. With PCCA, the plaintext is recovered as Fig. 6(f), which exactly equals the input plaintext shown in Fig. 6(a).

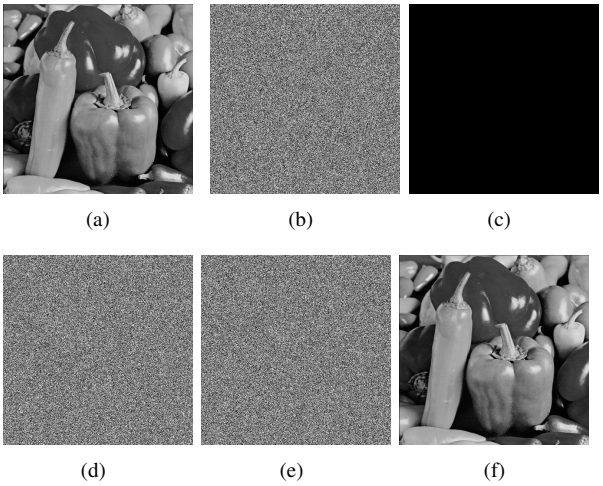


Fig. 6. Results of the proposed attack on cracking Hua's cipher [36]: (a) plaintext peppers; (b) ciphertext of peppers (a circle larger than the plaintext); (c) the first chosen-ciphertext $\boldsymbol{C}_0^{(2)}$ (a circle larger than the plaintext); (d) decryption result of the first chosen-ciphertext $\boldsymbol{M}_0^{(1)}$; (e) calculated differential image $\Delta \boldsymbol{M}^{(1)}$; (f) the recovered image.

In [37], a cipher named MIE-BX is proposed for medical image encryption [9]. It is similar to the cipher discussed above and is also vulnerable to PCCA.

1) The MIE-BX in [37].
   - *Brief review*. First, deriving from a 256-bit secret key $Seed$ and the employed logistic-sine system, two suites of permutation vectors $\boldsymbol{WB}$ and substitution masks $\boldsymbol{K}$ are generated. Second, add random pixels along the four sides of the original plain image $\boldsymbol{M}$ such that an enlarged image $\boldsymbol{MI}$ is generated. Third, shuffle $\boldsymbol{MI}$ at the pixel level to obtain the permutation ciphertext $\boldsymbol{P}$. Fourth, perform pixel substitution according to Eq. (18). The encryption core is iterated twice, with the individual permutation vectors and substitution masks produced in the first step.
   - *Cryptanalysis*. This cipher [37] is very similar to the cipher in [36], except for the adopted permutation techniques and chaotic maps. As mentioned in Section IV-C, the enhancements in terms of permutation techniques and nonlinear dynamics cannot enhance the cipher's security against PCCA. In other words, PCCA is applicable without any modification.

Remark 4. *Inserting random pixels in the encryption process provides indistinguishability to a cipher for resisting plaintext*

[8] Hua's source codes are open accessible at https://sites.google.com/site/huazyum/home/2DLASMEncryption.rar.

[9] Two ciphers were proposed in [37], and we focus on MIE-BX only. The other cipher (MIE-MA), which uses modulo addition for pixel modification, is beyond the scope of this paper.

attacks. However, such randomness vanishes in the decryption phase. The intrinsic BMX properties of the encryption kernel facilitate the applicability of PCCA.

D. Applications to ciphers with substitution-then-permutation structure

Slightly different from the widely adopted structure, the ciphers in [38], [39] perform pixel substitution ahead of the permutation phase. They are also vulnerable to PCCA.

1) Zhou's cipher in [38].
   - *Brief review*. There are three primary procedures in this encryption scheme. First, add random pixels along the left side of the plaintext $\boldsymbol{M}$, and an enlarged image $\boldsymbol{MI}$ is produced. Second, perform pixel substitution row by row according to $s(i) = mi(i) \oplus k(i) \oplus s(i-1)$, whereas $s(1) = mi(1)$ is specifically defined. This means that the first pixel is not encrypted in the substitution stage. Third, rotate the substitution ciphertext 90° to produce the ciphertext $\boldsymbol{C}$ in this round. The random pixel insertion, substitution and rotation processes are implemented four times to produce the final ciphertext.
   - *Cryptanalysis*. The primary feature of this cipher is that the random pixel insertion process is performed in each round. However, the 90° rotation and fixed insertion manner (left side) help us to specialize this cipher to a special case of Hua's algorithm [36]. Specifically, this is equivalent to randomly inserting pixels around all four sides and then performing substitution in row-by-row (left to right), column-by-column (bottom to top), row-by-row (right to left), column-by-column (top to bottom) manners. Similar to the cryptanalysis of Hua's cipher, PCCA also can break Zhou's encryption algorithm [38].
2) Tong's cipher in [39].
   - *Brief review*. A new one-dimensional chaotic system was developed in [39], and it was further used to generate the substitution mask $\boldsymbol{K}$ and two permutation vectors $\boldsymbol{WR}$ and $\boldsymbol{WC}$. In this cipher, substitution is first implemented according to $\boldsymbol{S} = \boldsymbol{M} \oplus \boldsymbol{K}$. Then, permutation is performed row by row with $\boldsymbol{WR}$ and then column by column using $\boldsymbol{WC}$.
   - *Cryptanalysis*. As indicated before, we can combine the two scrambling procedures into a single permutation. This cipher is thus relaxed to a standard substitution-then-permutation cipher. This cipher can also be viewed as a two-round permutation-then-substitution cipher. Accordingly, Tong's cipher is vulnerable to PCCA.

Experiments on cracking Zhou's cipher [38] are performed, and the results are shown in Fig. 7. A fingerprint image in the original paper is employed [10]. Complying with the theoretical analysis, the plaintext has been accurately recovered.

[10]The source code of Zhou's cipher is available via https://www.fst.um.edu.mo/en/staff/documents/fstycz/Bao2014SP.rar.

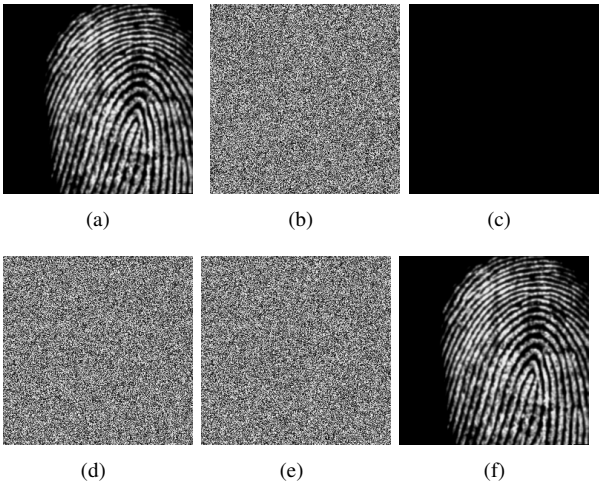


Fig. 7. Results of the proposed attack on cracking Zhou's cipher [38]: (a) plaintext fingerprint; (b) ciphertext of the fingerprint (a circle larger than the plaintext); (c) the first chosen-ciphertext $\boldsymbol{C}_0^{(4)}$ (a circle larger than the plaintext); (d) decryption result of the first chosen-ciphertext $\boldsymbol{M}_0^{(1)}$; (e) calculated differential image $\Delta\boldsymbol{M}^{(1)}$; (f) the recovered image.

Remark 5. *It is shown that repositioning the permutation and substitution procedures cannot increase resistance against PCCA. This is easy to understand, as substitution-then-permutation is essentially a typical case of two rounds of permutation-then-substitution encryption.*

VI. Conclusions

In this paper, we have investigated the security of a family of image encryption schemes. The studied ciphers adopt an iterative permutation-substitution network. The ciphers employ a bit-level or pixel-level permutation approach for image shuffling and use bit-wise XOR for pixel substitution. By differential cryptanalysis, we have revealed the BMX properties inside the mapping from the differential of the ciphertexts to that of the plaintexts. On this basis, a universal chosen-ciphertext attack has been theoretically proposed and experimentally verified. A total of 12 image ciphers have been found vulnerable to the proposed attack. We have further indicated that various types of enhancements, such as complex dynamics, improved permutation methods, substitution linking previous ciphertexts, inserting randomness before the encryption kernel, and repositioning the permutation and substitution, are incapable of enhancing security. Plaintext-related permutation and nonlinear substitution are highly suggested integrating into the future permutation-substitution type image ciphers.

Appendix A
Proof of Property 4

As indicated from Eq. (4), the output of the $(n-1)^{th}$ encryption round will be the input of the subsequent iteration.

Therefore, we can obtain

$$\begin{aligned}\Delta\boldsymbol{C}^{(n)} &= \mathcal{F}^{(i)}_{(basic)}(\Delta\boldsymbol{M}^{(i)})\\ &= \mathcal{F}^{(i)}_{(basic)}(\Delta\boldsymbol{C}^{(i-1)})\\ &= \mathcal{F}^{(i)}_{(basic)}[\mathcal{F}^{(i-1)}_{(basic)}(\Delta\boldsymbol{C}^{(i-2)})]\\ &= \dots\\ &= \mathcal{F}^{(i)}_{(basic)}\{\mathcal{F}^{(i-1)}_{(basic)}[\dots\mathcal{F}^{(1)}_{(basic)}(\Delta\boldsymbol{C}^{(0)})]\}\\ &= \mathcal{F}^{(i)}_{(basic)}\{\mathcal{F}^{(i-1)}_{(basic)}[\dots\mathcal{F}^{(1)}_{(basic)}(\Delta\boldsymbol{M}^{(1)})]\}\end{aligned}.$$

In other words,

$$\mathcal{F}^{(1)-(N)}_{(basic)}(\Delta\boldsymbol{M}^{(1)}) = \mathcal{F}^{(N)}_{(basic)}\{\mathcal{F}^{(N-1)}_{(basic)}[\dots\mathcal{F}^{(1)}_{(basic)}(\Delta\boldsymbol{M}^{(1)})]\}.$$

As $\mathcal{F}^{(i)}_{(basic)}(\Delta\boldsymbol{M}), i \in [1, N]$ has Properties 1–3, $\Delta\boldsymbol{C}^{(N)} = \mathcal{F}^{(1)-(N)}_{(cipher)}(\Delta\boldsymbol{M}^{(1)})$ is consequently bijective and binary multiplicative and can be bit-wise XORed. Hence, the proof is completed.

APPENDIX B
THE BMX PROPERTIES FU'S CIPHER'S DTF

For easy reference, we rewrite $\mathcal{F}_{(Fu)}(\Delta\boldsymbol{M})$ here, that is,

$$\mathcal{F}_{(Fu)}(\Delta\boldsymbol{M}) = \prod[\mathcal{WB}(\Delta\boldsymbol{M})]. \tag{22}$$

The bijectivity is easy to obtain. As shown in Eq. (22), $\mathcal{F}_{(Fu)}(\Delta\boldsymbol{M})$ consists of bit-wise XOR and bit-level permutation operations; they are both one-to-one mappings. Thus, $\mathcal{F}_{(Fu)}$ is bijective.

Then, we come to the binary multiplicability of $\mathcal{F}_{(Fu)}(\Delta\boldsymbol{M})$, that is,

$$\lambda \times \mathcal{F}_{(Fu)}(\Delta\boldsymbol{M}) = \mathcal{F}_{(Fu)}(\lambda \times \Delta\boldsymbol{M}), \lambda \in \{0, 1\}.$$

When $\lambda = 1$, $\lambda \times \mathcal{F}_{(Fu)}(\Delta\boldsymbol{M}) = \mathcal{F}_{(Fu)}(\lambda \times \Delta\boldsymbol{M})$ accordingly holds. In the case $\lambda = 0$, $\lambda \times \mathcal{F}_{(Fu)}(\Delta\boldsymbol{M})$ are zeros. Then, we refer to Eq. (22) and move to the right part. It is obvious that $\lambda \times \Delta\boldsymbol{M}$ are zeros, as is $\mathcal{WB}(\lambda \times \Delta\boldsymbol{M})$. Since $\prod$ denotes consecutive bit-wise XOR operation, $\mathcal{F}_{(Fu)}(0 \times \Delta\boldsymbol{M}) = \prod[\mathcal{WB}(0 \times \Delta\boldsymbol{M})]$ equals zero as well. In other words, $\lambda \times \mathcal{F}_{(Fu)}(\Delta\boldsymbol{M}) = \mathcal{F}_{(Fu)}(\lambda \times \Delta\boldsymbol{M})$ also holds when $\lambda = 0$. Therefore, we can conclude the binary multiplicability of $\mathcal{F}_{(Fu)}(\Delta\boldsymbol{M})$.

Finally, we prove that $\mathcal{F}_{(Fu)}(\Delta\boldsymbol{M})$ can be bit-wise XORed, that is,

$$\mathcal{F}_{(Fu)}(\Delta\boldsymbol{M}_1) \oplus \mathcal{F}_{(Fu)}(\Delta\boldsymbol{M}_2) = \mathcal{F}_{(Fu)}(\Delta\boldsymbol{M}_1 \oplus \Delta\boldsymbol{M}_2).$$

Referring to Eq. (16), we can obtain

$$\prod\nolimits_{j=1}^{i}\Delta p_1(j) \oplus \prod\nolimits_{j=1}^{i}\Delta p_2(j) = \prod\nolimits_{j=1}^{i}[\Delta p_1(j) \oplus \Delta p_2(j)],$$

which is valid for each $i \in [1, L]$. In matrix form, it can be represented via $\prod(\Delta\boldsymbol{P}_1) \oplus \prod(\Delta\boldsymbol{P}_2) = \prod(\Delta\boldsymbol{P}_1 \oplus \Delta\boldsymbol{P}_2)$. Considering that $\Delta\boldsymbol{P} = \mathcal{WB}(\Delta\boldsymbol{M})$ and referring to the deduction of Eq. (7), we can obtain

$$\begin{aligned}&\prod[\mathcal{WB}(\Delta\boldsymbol{M}_1)] \oplus \prod[\mathcal{WB}(\Delta\boldsymbol{M}_2)]\\ &= \prod[\mathcal{WB}(\Delta\boldsymbol{M}_1) \oplus \mathcal{WB}(\Delta\boldsymbol{M}_2)]\\ &= \prod[\mathcal{WB}(\Delta\boldsymbol{M}_1 \oplus \Delta\boldsymbol{M}_2)]\end{aligned}.$$

Therefore, $\mathcal{F}_{(Fu)}(\Delta\boldsymbol{M})$ can be bit-wise XORed.

Summarizing, $\mathcal{F}_{(Fu)}(\Delta\boldsymbol{M})$ has BMX properties.

APPENDIX C
THE PROOF OF EQ. (21)

As temporary variables, $\boldsymbol{MI}^{(1)}_0$, $\boldsymbol{MI}^{(1)}_{l-b}$, $\boldsymbol{R}_0$ and $\boldsymbol{R}_{l-b}$ exist in the decryption process. They are unknown but not random; in addition

$$\begin{cases}\boldsymbol{MI}^{(1)}_0 = \boldsymbol{M}^{(1)}_0||\boldsymbol{R}_0\\ \boldsymbol{MI}^{(1)}_{l-b} = \boldsymbol{M}^{(1)}_{l-b}||\boldsymbol{R}_{l-b}\end{cases}. \tag{23}$$

Next, we can convert Eq. (20) as

$$\Delta\boldsymbol{MI}^{(1)}_{l-b} = \boldsymbol{MI}^{(1)}_{l-b} \oplus \boldsymbol{MI}^{(1)}_0 = (\boldsymbol{M}^{(1)}_{l-b}||\boldsymbol{R}_{l-b}) \oplus (\boldsymbol{M}^{(1)}_0||\boldsymbol{R}_0).$$

Since $||$ denotes the pixel insertion operation and since pixels in different positions of a matrix cannot affect each other in bit-wise XOR operation,

$$\begin{aligned}\Delta\boldsymbol{MI}^{(1)}_{l-b} &= (\boldsymbol{M}^{(1)}_{l-b}||\boldsymbol{R}_{l-b}) \oplus (\boldsymbol{M}^{(1)}_0||\boldsymbol{R}_0)\\ &= (\boldsymbol{M}^{(1)}_{l-b} \oplus \boldsymbol{M}^{(1)}_0)||(\boldsymbol{R}_{l-b} \oplus \boldsymbol{R}_0).\\ &= \Delta\boldsymbol{M}^{(1)}_{l-b}||\Delta\boldsymbol{R}_{l-b}\end{aligned} \tag{24}$$

Similarly, Eq. (25) holds for $\boldsymbol{M}^{(1)}$, which is the plaintext of $\boldsymbol{C}^{(2)}$ awaiting recovery. The enlarged image $\boldsymbol{MI}^{(1)}$ and the randomly inserted pixels $\boldsymbol{R}$ exist, although they are unobtainable.

$$\begin{aligned}\Delta\boldsymbol{MI}^{(1)} &= (\boldsymbol{M}^{(1)}||\boldsymbol{R}) \oplus (\boldsymbol{M}^{(1)}_0||\boldsymbol{R}_0)\\ &= (\boldsymbol{M}^{(1)} \oplus \boldsymbol{M}^{(1)}_0)||(\boldsymbol{R} \oplus \boldsymbol{R}_0).\\ &= \Delta\boldsymbol{M}^{(1)}||\Delta\boldsymbol{R}\end{aligned} \tag{25}$$

Referring to the BMX properties of $\mathcal{FI}_{(Hua)}(\Delta\boldsymbol{MI})$ and taking Eq. (24) into consideration, we can also obtain $\Delta\boldsymbol{MI}^{(1)}$ by Eq. (26), where $bitC_{l-b}$ denotes the b^{th} bit of pixel $c^{(2)}(l)$.

$$\begin{aligned}\Delta\boldsymbol{MI}^{(1)} &= \prod\nolimits_{l=1}^{L}\prod\nolimits_{b=1}^{B}[bitC_{l-b} \times \Delta\boldsymbol{MI}^{(1)}_{l-b}]\\ &= \prod\nolimits_{l=1}^{L}\prod\nolimits_{b=1}^{B}[bitC_{l-b} \times (\Delta\boldsymbol{M}^{(2)}_{l-b}||\Delta\boldsymbol{R}_{l-b})]\end{aligned}. \tag{26}$$

Since $bitC_{l-b}$ is binary,

$$\begin{aligned}&bitC_{l-b} \times (\Delta\boldsymbol{M}^{(1)}_{l-b}||\Delta\boldsymbol{R}_{l-b})\\ &= [bitC_{l-b} \times \Delta\boldsymbol{M}^{(1)}_{l-b}]||[bitC_{l-b} \times \Delta\boldsymbol{R}_{l-b}]\end{aligned}. \tag{27}$$

Once again, since $||$ denotes the pixel insertion operation and since pixels in different positions cannot affect each other in bit-wise XOR of two matrices, we can combine Eqs. (26) and (27) as

$$\begin{aligned}\Delta\boldsymbol{MI}^{(1)} &= \prod\nolimits_{l=1}^{L}\prod\nolimits_{b=1}^{B}[(bitC_{l-b} \times \Delta\boldsymbol{M}^{(1)}_{l-b})||(bitC_{l-b} \times \Delta\boldsymbol{R}_{l-b})]\\ &= \prod\nolimits_{l=1}^{L}\prod\nolimits_{b=1}^{B}[(bitC_{l-b} \times \Delta\boldsymbol{M}^{(1)}_{l-b})] \;||\\ &\quad \prod\nolimits_{l=1}^{L}\prod\nolimits_{b=1}^{B}[(bitC_{l-b} \times \Delta\boldsymbol{R}_{l-b})]\end{aligned}. \tag{28}$$

Comparing Eq. (25) with (28), we can obtain

$$\Delta\boldsymbol{M}^{(1)} = \prod\nolimits_{l=1}^{L}\prod\nolimits_{b=1}^{B}[bitC_{l-b} \times \Delta\boldsymbol{M}^{(1)}_{l-b}].$$

This concludes the proof.

REFERENCES

[1] B. Furht and D. Kirovski, *Multimedia security handbook*. CRC Press, 2004, ch. Chaos-based encryption for digital images and videos.

[2] L. Y. Zhang, Y. Liu, K.-W. Wong, F. Pareschi, Y. Zhang, R. Rovatti, and G. Setti, "On the security of a class of diffusion mechanisms for image encryption," *IEEE Transactions on Cybernetics*, no. 99, pp. 1–13, 2017.

[3] M. Preishuber, T. Hütter, S. Katzenbeisser, and A. Uhl, "Depreciating motivation and empirical security analysis of chaos-based image and video encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2137–2150, 2018.

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

[5] G. Chen, Y. Mao, 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.

[6] Y. Mao, G. Chen, and S. Lian, "A novel fast image encryption scheme based on 3D chaotic baker maps," *International Journal of Bifurcation and Chaos*, vol. 14, no. 10, pp. 3613–3624, 2004.

[7] Z. Zhu, W. Zhang, K. Wong, and H. Yu, "A chaos-based symmetric image encryption scheme using a bit-level permutation," *Information Sciences*, vol. 181, no. 6, pp. 1171–1186, 2011.

[8] C. Fu, W.-H. Meng, Y.-F. Zhan, Z.-L. Zhu, F. C. Lau, K. T. Chi, and H.-F. Ma, "An efficient and secure medical image protection scheme based on chaotic maps," *Computers in Biology and Medicine*, vol. 43, no. 8, pp. 1000–1010, 2013.

[9] S. Sun, "A novel hyperchaotic image encryption scheme based on DNA encoding, pixel-level scrambling and bit-level scrambling," *IEEE Photonics Journal*, vol. 10, no. 2, pp. 1–14, 2018.

[10] X. Liu, D. Xiao, and Y. Xiang, "Quantum image encryption using intra and inter bit permutation based on logistic map," *IEEE Access*, vol. 7, pp. 6937–6946, 2019.

[11] J. Chen, Z. Zhu, L. Zhang, Y. Zhang, and B. Yang, "Exploiting self-adaptive permutation-diffusion and DNA random encoding for secure and efficient image encryption," *Signal Processing*, vol. 142, pp. 340 – 353, 2018.

[12] Z. Lin, S. Yu, J. Lü, S. Cai, and G. Chen, "Design and ARM-embedded implementation of a chaotic map-based real-time secure video communication system," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 7, pp. 1203–1216, 2015.

[13] X. Tong, "The novel bilateral-diffusion image encryption algorithm with dynamical compound chaos," *Journal of Systems and Software*, vol. 85, no. 4, pp. 850 – 858, 2012.

[14] X. Kang and R. Tao, "Color image encryption using pixel scrambling operator and reality-preserving MPFRHT," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 7, pp. 1919–1932, 2019.

[15] X. Kang, A. Ming, and R. Tao, "Reality-preserving multiple parameter discrete fractional angular transform and its application to color image encryption," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 6, pp. 1595–1607, 2019.

[16] Y. Zhou, Z. Hua, C.-M. Pun, and C. L. P. Chen, "Cascade chaotic system with applications," *IEEE Transactions on Cybernetics*, vol. 45, no. 9, pp. 2001–2012, 2015.

[17] S. Zhang and T. Gao, "A coding and substitution frame based on hyper-chaotic systems for secure communication," *Nonlinear Dynamics*, vol. 84, no. 2, pp. 833–849, 2016.

[18] X. Wang and H.-l. Zhang, "A novel image encryption algorithm based on genetic recombination and hyper-chaotic systems," *Nonlinear Dynamics*, vol. 83, no. 1-2, pp. 333–346, 2016.

[19] S. Chen, S. Yu, J. Lü, G. Chen, and J. He, "Design and FPGA-based realization of a chaotic secure video communication system," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2359–2371, 2018.

[20] F. Özkaynak, "Brief review on application of nonlinear dynamics in image encryption," *Nonlinear Dynamics*, vol. 92, no. 2, pp. 305–313, 2018.

[21] C. Li, D. Lin, J. Lü, and F. Hao, "Cryptanalyzing an image encryption algorithm based on autoblocking and electrocardiography," *IEEE Multimedia*, vol. 25, no. 4, pp. 46–56, 2018.

[22] H. Zhu, C. Zhao, and X. Zhang, "A novel image encryption–compression scheme using hyper-chaos and Chinese remainder theorem," *Signal Processing: Image Communication*, vol. 28, no. 6, pp. 670–680, 2013.

[23] C. Li, Y. Liu, L. Y. Zhang, and K.-W. Wong, "Cryptanalyzing a class of image encryption schemes based on Chinese remainder theorem," *Signal Processing: Image Communication*, vol. 29, no. 8, pp. 914–920, 2014.

[24] M. Su, W. Wen, and Y. Zhang, "Security evaluation of bilateral-diffusion based image encryption algorithm," *Nonlinear Dynamics*, vol. 77, no. 1-2, pp. 243–246, 2014.

[25] S. Li, C. Li, G. Chen, N. G. Bourbakis, and K. Lo, "A general quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks," *Signal Processing: Image Communication*, vol. 23, no. 3, pp. 212–223, 2008.

[26] C. Li and K.-T. Lo, "Optimal quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks," *Signal Processing*, vol. 91, no. 4, pp. 949–954, 2011.

[27] A. Jolfaei, X. Wu, and V. Muthukkumarasamy, "On the security of permutation-only image encryption schemes," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 2, pp. 235–246, 2016.

[28] Y. Zhang and D. Xiao, "Cryptanalysis of s-box-only chaotic image ciphers against chosen plaintext attack," *Nonlinear Dynamics*, vol. 72, no. 4, pp. 751–756, 2013.

[29] B. Li, X. Liao, and Y. Jiang, "A novel image encryption scheme based on logistic map and dynatomic modular curve," *Multimedia Tools and Applications*, vol. 77, no. 7, pp. 8911–8938, 2018.

[30] Z. Tang, X. Zhang, and W. Lan, "Efficient image encryption with block shuffling and chaotic map," *Multimedia Tools and Applications*, vol. 74, no. 15, pp. 5429–5448, 2015.

[31] Z. Tang, J. Song, X. Zhang, and R. Sun, "Multiple-image encryption with bit-plane decomposition and chaotic maps," *Optics and Lasers in Engineering*, vol. 80, pp. 1–11, 2016.

[32] Y. Dai, H. Wang, and Y. Wang, "Chaotic medical image encryption algorithm based on bit-plane decomposition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 30, no. 4, p. 1657001, 2016.

[33] G. Zhou, D. Zhang, Y. Liu, Y. Yuan, and Q. Liu, "A novel image encryption algorithm based on chaos and Line map," *Neurocomputing*, vol. 169, pp. 150–157, 2015.

[34] O. Mirzaei, M. Yaghoobi, and H. Irani, "A new image encryption method: parallel sub-image encryption with hyper chaos," *Nonlinear Dynamics*, vol. 67, no. 1, pp. 557–566, 2012.

[35] R. Ye, M. Ye, Y. Li, X. Shi, and W. Ye, "A permutation-substitution based image encryption scheme with bit-plane exchanging strategy," in *2015 3rd International Conference on Machinery, Materials and Information Technology Applications*. Atlantis Press, 2015.

[36] Z. Hua and Y. Zhou, "Image encryption using 2D Logistic-adjusted-Sine map," *Information Sciences*, vol. 339, pp. 237–253, 2016.

[37] Z. Hua, S. Yi, and Y. Zhou, "Medical image encryption using high-speed scrambling and pixel adaptive diffusion," *Signal Processing*, vol. 144, pp. 134–144, 2018.

[38] Y. Zhou, L. Bao, and C. P. Chen, "A new 1D chaotic system for image encryption," *Signal Processing*, vol. 97, pp. 172–182, 2014.

[39] X. Tong and M. Cui, "Image encryption with compound chaotic sequence cipher shifting dynamically," *Image and Vision Computing*, vol. 26, no. 6, pp. 843–850, 2008.

[40] J. Chen, L. Chen, and Y. Zhou, "Universal chosen-ciphertext attack for a family of image encryption schemes," *IEEE Transactions on Multimedia*, pp. 1–1, 2020.

[41] Y. Song, Z. Zhu, W. Zhang, H. Yu, and Y. Zhao, "Efficient and secure image encryption algorithm using a novel key-substitution architecture," *IEEE Access*, vol. 7, pp. 84 386–84 400, 2019.

[42] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. Chapman and Hall/CRC, 2014.

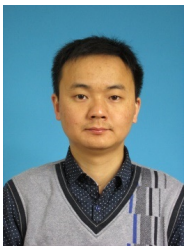
[43] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *International Journal of Bifurcation and Chaos*, vol. 16, no. 8, pp. 2129–2151, 2006.

[44] E. Solak, C. Çokal, O. T. Yildiz, and T. Biyikoğlu, "Cryptanalysis of Fridrich's chaotic image encryption," *International Journal of Bifurcation and Chaos*, vol. 20, no. 5, pp. 1405–1413, 2010.

[45] E. Y. Xie, C. Li, S. Yu, and J. Lü, "On the cryptanalysis of fridrich's chaotic image encryption scheme," *Signal Processing*, vol. 132, pp. 150–154, 2017.

[46] A.-V. Diaconu, "Circular inter–intra pixels bit-level permutation and chaos-based image encryption," *Information Sciences*, vol. 355, pp. 314–327, 2016.

[47] A.-V. Diaconu, V. Ionescu, G. Iana, and J. M. Lopez-Guede, "A new bit-level permutation image encryption algorithm," in *2016 International Conference on Communications (COMM)*. IEEE, 2016, pp. 411–416.



Junxin Chen received the B. Sc., M.Sc. and Ph.D degrees all in communications engineering from Northeastern University, Shenyang, China, in 2007, 2009, and 2016 respectively. He is currently an associate professor at College of Medicine and Biological Information Engineering, Northeastern University, Shenyang, China; he is also with the department of computer and information science, University of Macau, Macau SAR, China. He has authored/co-authored over 50 scientific paper in peer-reviewed journals and conferences, including IEEE Transactions of Industrial Informatics, IEEE Internet of Things Journal, IEEE Photonics Journal, Information Sciences, etc. His research interests include biosignal processing, compressive sensing, security and privacy.

Lei Chen received the Ph.D. degree in electronic science and technology from the Beijing University of Posts and Telecommunications, Beijing, China, in 2018. He is currently a Postdoctoral Fellow with the Research Institute of Information Technology (RIIT), Tsinghua University, Beijing, China. He is also with Nsfocus Information Technology Co., Ltd., Beijing, China. His research interests include multimedia security, cryptanalysis, data security, and machine learning for cyberspace security.

Yicong Zhou (M'07-SM'14) received the B.S. degree from Hunan University, Changsha, China, and the M.S. and Ph.D. degrees from Tufts University, Massachusetts, USA, all in electrical engineering. He is currently an Associate Professor and Director of the Vision and Image Processing Laboratory in the Department of Computer and Information Science at University of Macau. His research interests include image processing, computer vision, machine learning, and multimedia security.

Dr. Zhou is a senior member of the International Society for Optical Engineering (SPIE). He was a recipient of the Third Price of Macao Natural Science Award in 2014 and 2020. He is a Co-Chair of Technical Committee on Cognitive Computing in the IEEE Systems, Man, and Cybernetics Society. He serves as an Associate Editor for IEEE Transactions on Neutral Networks and Learning Systems, IEEE Transactions on Circuits and Systems for Video Technology, IEEE Transactions on Geoscience and Remote Sensing, and four other journals.