Re-Evaluation of the Security of a Family of Image Diffusion Mechanisms

Junxin Chen, Leo Yu Zhang, *Member, IEEE*, and Yicong Zhou, *Senior Member, IEEE*

***Abstract*— In recent years, the use of permutation-diffusion architecture for digital image encryption has become increasingly popular. The permutation procedure scrambles the pixel locations, while the diffusion phase modifies the pixel values and gives rise to the avalanche effect. Various diffusion techniques have been developed, and their strength strongly impacts the security of the overall cryptosystem. In this paper, we re-evaluate the security of a family of image diffusion mechanisms that are based on mixing modulo addition with bitwise exclusive OR operations. The recovery of the encryption element of these diffusion mechanisms is comprehensively demonstrated, and the accuracy bounds under various conditions are proved mathematically. Compared to the state-of-the-art methods, our work improves the recovery accuracy of the encryption element while the required prior knowledge is decreased. The proposed analysis of the diffusion mechanisms is further used to cryptanalyze the whole cryptosystem theoretically and experimentally.**

***Index Terms*— Cryptanalysis, image encryption, modulo addition, bitwise exclusive OR.**

I. Introduction

IN RECENT years, secure transmission and storage of multimedia contents in public communication infrastructures have attracted intense attention [1], [2]. Traditional ciphers such as the data encryption standard (DES) and advanced encryption standard (AES) are applicable to encrypt the multimedia data in binary fashion. However, this straightforward encryption that does not consider the nature of multimedia data is inefficient, and in some cases is also insecure [3], [4]. Due to this concern, some researchers have advocated investigation of ad hoc encryption schemes by leveraging the intrinsic properties of multimedia data such as high pixel correlation and large volume [5]–[7].

The permutation-diffusion network is currently the most popular architecture in the literature on the design of image encryption schemes, [4], [8]–[13]. It was first proposed by Fridrich [14] based on the traditional substitution-permutation network, and then was formalized by Chen *et al*. [15], [16]. In this structure, the permutation procedure focuses on scrambling pixel locations while the diffusion phase modifies the pixel values and spreads the plaintext's information to the whole ciphertext. The permutation vector and diffusion masks are the secret elements required in this architecture, while chaotic systems and other nonlinear phenomenons are popular choice for their generation. Even though researchers have developed various permutation approaches, permutation itself is not very strong with respect to security and generalized cryptanalysis of the permutation techniques has been investigated in [17]–[22]. In this case, the security of the nonlinear diffusion phase either from the design or the analysis point of view, becomes critical.

Fridrich proposed to implement the image diffusion as

$$c(i) = p(i) \dot{+} F[c(i-1), k(i)],$$

where $p(i)$, $c(i)$ and $k(i)$ are the i-th plain pixel, cipher pixel and diffusion mask, respectively. Here, the operator $\dot{+}$ denotes the modulo addition while function $F(\cdot)$ is suggested to be nonlinear and computation-efficient. The introduction of $c(i-1)$ aims to spread a single pixel's information to the whole ciphertext, hence achieving the so-called avalanche effect. Following this work, Chen *et al*. [15], [16] developed a diffusion mechanism by mixing modulo addition with bitwise exclusive OR (XOR), according to

$$c(i) = c(i-1) \oplus [p(i) \dot{+} k(i)] \oplus k(i), \tag{1}$$

where $\oplus$ denotes bitwise XOR. Benefiting from the high implementation efficiency and nonlinear characteristic over GF(2), Eq. (1) has been frequently adopted as the diffusion part of an image encryption scheme [23]–[29]. Some of the cryptosystems directly employed Eq. (1) for diffusion [24], while some others slightly modified the equation. For example, the roles of $p(i)$ and $c(i-1)$ was swapped in the encryption scheme of [23] while three groups of diffusion masks were used in [26].

Together with its wide use for encryption, security evaluation of Eq. (1) has also attracted research attention of the community. If the diffusion mask $k(i)$ can be derived from some collected information, i.e., some known or chosen plain pixels and the corresponding ciphertexts, the security strength of this kind of permutation-diffusion cryptosystem is consequently decreased. Determination of $k(i)$ of Eq. (1) is the core problem of the related cryptanalysis. Previous works have been reported in [30]–[34]. They used differential analysis and reduced the problem to the derivation of k of the following

Manuscript received June 28, 2020; revised October 16, 2020; accepted January 21, 2021. Date of publication January 25, 2021; date of current version December 6, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61802055, Grant 61771121, Grant 61701103, and Grant 61702221; in part by the Fundamental Research Funds for the Central Universities under Grant N2019001 and Grant N181904002; in part by the Science and Technology Development Fund, Macau, under Grant 189/2017/A3; and in part by the University of Macau under Grant MYRG2018-00136-FST. This article was recommended by Associate Editor J. Xu. *(Corresponding author: Yicong Zhou.)*

Junxin Chen is with the College of Medicine and Biological Information Engineering, Northeastern University, Shenyang 110004, China, and also with the Department of Computer and Information Science, University of Macau, Macau 999078, China.

Leo Yu Zhang is with the School of Information Technology, Deakin University, Victoria 3216, Australia.

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

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2021.3054508.

Digital Object Identifier 10.1109/TCSVT.2021.3054508

differential equation of modulo addition (DEA),

$$y = (\alpha \dot{+} k) \oplus (\beta \dot{+} k). \tag{2}$$

With some known or chosen (y, α, β) tuples, k can be determined with certain probability. For attacking the image cryptosystems in [28], [29], Li *et al*. [30] first introduced DEA and reported that three chosen (α, β) queries and corresponding y are sufficient to derive k. On the other hand, Zhang *et al*. [32], [33] attempted to determine k from some known (y, α, β) tuples and extended the method to crack some image cryptosystems under a known-plaintext attack.

In this paper, we attempt to determine k from

$$y = (p \dot{+} k) \oplus k, \tag{3}$$

rather than by using the differential analysis in the previous works [30]–[34]. Then, the cryptographic strength of the Eq. (1) and similar diffusion mechanisms is directly indicated, and the generalized achievements can be extended in a straightforward manner to crack the permutation-diffusion kind cryptosystems. Determination of k of $y = (p \dot{+} k) \oplus k$ has been discussed in [35] where the authors proposed to narrow the possible values of k using some known (y, p) pairs. This is essentially an exhaustive search method. In this paper, we present a theoretical study of this problem. Generalized methods for determining k of $y = (p \dot{+} k) \oplus k$ are examined, and the proposed method can be straightforwardly extended for cryptanalysis applications. On the other hand, the related works have always focused on attacking a certain specified image cryptosystem [30], [31]. In addition, we summarize the theoretical bounds of determining k under various attack conditions, while the peer works generally solved this problem in a single scenario such as the known-plaintext attack assumption given in [33].

The contributions of this work are summarized as follows:

1) We comprehensively analyze the determination of k satisfying $y = (p \dot{+} k) \oplus k$ that is popularly used as a generic cryptographic component for image encryption.
2) The technical algorithm for deriving k is described in detail, and the accuracy bounds under various assumptions are mathematically proven.
3) The theoretical achievements are experimentally applied to cryptanalyze several image encryption schemes.
4) The source codes are open accessible for validation and extension.[1]

The remainder of this paper is organized as follows. Section II introduces the notations and related work. The theoretical achievements of this work are presented in Section III which also include the comparisons with the related methods. Cryptographic applications for attacking some image cryptosystems are given in Section IV, and conclusions are drawn in the last section.

II. Related Works

A. Notations and Assumptions

Some notations adopted in this paper are listed as follows.

- In this paper, the bold upper case is used to denote an assembly while a capital character always denotes a constant. For example, the image size is assumed as $H \times W$, and the pixels of image $\boldsymbol{M}$ are denoted as $\{m(0,0), m(0,1), \ldots, m(i,j), \ldots, m(H-1, W-1)\}$, or $\{m(0), m(1), \ldots, m(i), \ldots, m(L-1)\}$, $L = H \times W$ in vector formation.
- The pixels are assumed to have N-bits resolution, so that the pixel values range within $[0, 2^N - 1]$.
- For a n-bits resolution number x, its bits are denoted as $x_{n-1}, \cdots, x_1, x_0$, from the highest bit to the lowest one. The value of x is $x = \sum_{i=0}^{n-1} x_i \times 2^i$.
- The symbol $\wedge$ denotes bitwise AND operation, and we also use ab to represent $a \wedge b$ for simplicity [31], [33]

In a secret communication system, the ciphertext is assumed to be transmitted over public channels. Technically, everybody can eavesdrop and obtain the ciphertext, whereas its plaintext should be inaccessible without the key. In this scenario, cryptanalysis refers to the recovery of the plaintext without the key. The common four types of attack models are [36]

1) *Ciphertext-only attack:* the adversary only has a number of ciphertexts.
2) *Known-plaintext attack:* the adversary has a collection of plaintexts and their ciphertexts.
3) *Chosen-plaintext attack:* the adversary can construct any plaintexts on demand and obtain the corresponding ciphertexts.
4) *Chosen-ciphertext attack:* the adversary can construct any ciphertexts that he wants and obtain the corresponding plaintexts.

By exploiting the underlying clues inside the collected information, an attack is said to be successful if the receiving ciphertext can be recovered without the secret key.

B. Image Encryption Schemes Under Study

As mentioned above, mixing modulo addition with bitwise XOR has been frequently adopted for the diffusion part of image encryption. To some extent, most of them are variants of Chen's diffusion mechanism in Eq. (1). Three typical applications can be found in [23], [25], [26]. We briefly review these cryptosystems in this section.[2]

1) *Xie's cryptosystem in [25].* An image encryption scheme was developed in [25] based on the permutation-diffusion structure. Using the results on the security analysis of permutation ciphers [17], [19], [20], the 3D permutation in [25] is generalized as a vector for ease of the description.
   a) *Initialization.* With the help of the optics chaos and 3D cat map, a permutation vector $\boldsymbol{V} = \{v(1), v(2), \cdots, v(L)\}$ is produced. In addition, a vector of diffusion mask, i.e., $\boldsymbol{K} = \{k(0), k(2), \cdots, k(L-1)\}$, is generated from the Logistic map.
   b) *Permutation*. Scramble the plain image $\boldsymbol{M}$ according to

$$p(i) = m(v(i)). \tag{4}$$

The permutation ciphertext $\boldsymbol{P}$ is thus produced.

[1]The source codes are open accessible via https://github.com/lurenjia212.

[2]For simplicity, descriptions of the studied cryptosystems may be different from those in the original publications. However, the encryption kernels are identical.

c) *Diffusion*. The diffusion is implemented according to Eq. (1), that is

$$c(i) = c(i-1) \oplus [p(i) \dot{+} k(i)] \oplus k(i).$$

A similar secret communication system uses optics chaos and permutation-diffusion encryption can be found in [24], and the same diffusion equation is used.

2) *Parvin's cryptosystem in [23].* An encryption scheme was developed in [23] for encrypting a 256 gray-scale image.

a) *Initialization.* With the key *Seed* and two chaotic systems, three series of random numbers are generated. These are denoted as $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{K}$, respectively. The assembly $\boldsymbol{K}$ has $H \times W$ pseudorandom integers within $[0, 255]$, while $\boldsymbol{U}$, $\boldsymbol{V}$ contains H and W non-repetitive pseudorandom integers in the interval $[0, H-1]$ and $[0, W-1]$, respectively.

b) *Permutation.* Parvin's cryptosystem uses a two-stage permutation.[3] First, each row of the plaintext $\boldsymbol{M}$ is shuffled with the vector $\boldsymbol{U}$, that is

$$p'(u(i), j) = m(i, j). \tag{5}$$

The resultant image $\boldsymbol{P}'$ is then scrambled column by column by the vector $\boldsymbol{V}$, that is

$$p(i, v(j)) = p'(i, j). \tag{6}$$

The product $\boldsymbol{P}$ is the permutation ciphertext.

c) *Diffusion.* Compared to Eq. (1), Parvin's diffusion method swaps the roles of $c(i-1)$ and $p(i)$, yet the adopted arithmetic operations are identical. The permutation product is first stretched into a vector that is still denoted as $\boldsymbol{P}$ because this will not cause ambiguity. The diffusion is implemented according to

$$c(i) = p(i) \oplus [c(i-1) \dot{+} k(i)] \oplus k(i). \tag{7}$$

The resultant vector is rearranged into a $H \times W$ matrix, and then the ciphertext $\boldsymbol{C}$ is obtained.

3) *Sam's cryptosystem in [26].* This scheme is developed for encrypting color images. The secret key includes six odd integers ($\{r_u\}_{u=1}^{6}$), and three control parameters (k_1, k_2, k_3) and initial states (x_0, y_0, z_0) of the employed 3-D chaotic map. The plain image is denoted as $\boldsymbol{M}$, while its RGB channels are represented as $\boldsymbol{R}$, $\boldsymbol{G}$, $\boldsymbol{B}$, respectively.

a) *Initialization.* With $\{r_u\}_{u=1}^{6}$, three permutation matrices are first generated according to Eq. (8).

$$\begin{cases} ri(i) = \text{mod}\ (i \times r_1 \times 31, H)\ \ (i \in [0 \sim H-1]) \\ rj(j) = \text{mod}\ (j \times r_2 \times 31, W)\ \ (j \in [0 \sim W-1]) \\ gi(i) = \text{mod}\ (i \times r_3 \times 31, H)\ \ (i \in [0 \sim H-1]) \\ gj(j) = \text{mod}\ (j \times r_4 \times 31, W)\ \ (j \in [0 \sim W-1]) \\ bi(i) = \text{mod}\ (i \times r_5 \times 31, H)\ \ (i \in [0 \sim H-1]) \\ bj(j) = \text{mod}\ (j \times r_6 \times 31, W)\ \ (j \in [0 \sim W-1]). \end{cases} \tag{8}$$

Then, three series of diffusion masks are produced with $(k_1, k_2, k_3, x_0, y_0, z_0)$ and the adopted 3-D chaotic map, denoted as $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$. All of these have $L = H \times W$ elements range within [0, 255].

b) *Permutation.* The RGB channels are scrambled independently, with the permutation matrices produced by Eq. (8). The scrambling process is given by

$$\begin{cases} pr(i, j) = r(ri(i), rj(j)) \\ pg(i, j) = g(gi(i), gj(j)) \\ pb(i, j) = b(bi(i), bj(j)), \end{cases} \tag{9}$$

where $\boldsymbol{PR}$, $\boldsymbol{PG}$ and $\boldsymbol{PB}$ denote the permutation ciphertexts of $\boldsymbol{R}$, $\boldsymbol{G}$, $\boldsymbol{B}$, respectively.

c) *Diffusion.* Since the diffusion is operated on each channel independently, we only focus on the R channel in the following. The permutation ciphertext $\boldsymbol{PR}$ is first reshaped into a vector. A nonlinear diffusion procedure is then implemented according to

$$cr^{\dagger}(i) = [pr(i) \ggg 4 \dot{+} x(i)] \oplus y(i). \tag{10}$$

The product $\boldsymbol{CR}^{\dagger}$ is reshaped to a matrix and then rescanned in zigzag pattern to get $\boldsymbol{CR}^{\ddagger}$. A diffusion procedure is further performed according to

$$cr(i) = cr^{\ddagger}(i) \oplus cr(i-1) \oplus z(i), \tag{11}$$

where $cr(-1) = 0$. Without loss of the generality, we use $zig(i)$ to denote the correspondence between $cr^{\dagger}(i)$ and $cr^{\ddagger}(i)$, that is

$$cr^{\ddagger}(i) = cr^{\dagger}(zig(i)). \tag{12}$$

Combining Eqs. (10)-(12) together, we can integrate the whole diffusion as

$$\begin{aligned} cr(i) = {} & [pr(zig(i)) \ggg 4 \dot{+} x(zig(i))] \\ & \oplus y(zig(i)) \oplus cr(i-1) \oplus z(i), \end{aligned} \tag{13}$$

where $\ggg 4$ refers to the circular shift (towards right) operation by 4 bits. The G and B channels are encrypted in a same manner with identical diffusion masks. That is

$$\begin{aligned} cg(i) = {} & [pg(zig(i)) \ggg 4 \dot{+} x(zig(i))] \\ & \oplus y(zig(i)) \oplus cg(i-1) \oplus z(i). \end{aligned} \tag{14}$$

$$\begin{aligned} cb(i) = {} & [pb(zig(i)) \ggg 4 \dot{+} x(zig(i))] \\ & \oplus y(zig(i)) \oplus cb(i-1) \oplus z(i). \end{aligned} \tag{15}$$

Combining the encrypted RGB channels ($\boldsymbol{CR}$, $\boldsymbol{CG}$ and $\boldsymbol{CB}$) into a color image, the final ciphertext $\boldsymbol{C}$ is produced.

C. Existing Cryptanalysis on the Primitive

Finding the diffusion mask $\boldsymbol{K}$ is the core problem of the cryptanalysis of the studied diffusion mechanisms as well as the whole cryptosystems. Previous works have striven for solving this problem [30]–[34]. These works are based on a differential analysis. Taking Eq. (1) as an example, assuming that there are two pairs of plaintexts and corresponding

[3]Because Parvin's permutation technique can be attacked in a simpler manner than the universal cryptanalysis in [17], [19], [20], and we do not generalize Parvin's permutation as Eq. (4).

ciphertexts, i.e., ***M1***, ***M2***, ***C1***, ***C2***, we can obtain

$$\begin{cases} c1(i) = c1(i-1) \oplus [p1(i) \dot{+} k(i)] \oplus k(i) \\ c2(i) = c2(i-1) \oplus [p2(i) \dot{+} k(i)] \oplus k(i). \end{cases} \tag{16}$$

The differential of the ciphertexts is further calculated as

$$\begin{aligned} &c1(i) \oplus c2(i) \oplus c1(i-1) \oplus c2(i-1) \\ &\quad = [p1(i) + k(i)] \oplus [p2(i) + k(i)]. \end{aligned} \tag{17}$$

It is clear that the differential analysis of the Parvin's scheme by Eq. (7) can also be finalized into a similar form. For Sam's encryption scheme [26], we can directly obtain the differential result by XORing the ciphertexts of different channels. From Eqs. (13) and (14), we can obtain

$$\begin{aligned} &cr(i) \oplus cr(i-1) \oplus cg(i) \oplus cg(i-1) \\ &\quad = [pr(zig(i)) \ggg 4 \dot{+} x(zig(i))] \\ &\qquad \oplus [pg(zig(i)) \ggg 4 \dot{+} x(zig(i))] \end{aligned} \tag{18}$$

The DEA is the generalized expression of Eqs. (17) and (18), that is

$$y = (\alpha \dot{+} k) \oplus (\beta \dot{+} k).$$

For any value of i, finding the diffusion mask $k(i)$ finalizes the determination k of DEA. Previous works [30]–[34] have sought to determine k from some known or chosen (y, α, β) tuples. These can be classified into the following categories.

1) *Determine k from some* (y, α, β) *tuples while* (α, β) *can be freely selected*. This scenario always corresponds to chosen-plaintext attack in the cryptanalyst-related image encryption schemes, because α and β refer to plain pixels in Eqs. (17) and (18). By constructing some special (α, β) and obtaining corresponding y, k had been proven to be recoverable. Li *et al*. first proposed in [30] that three special (α, β) pairs are required to derive k, and then demonstrated that two chosen queries are sufficient [31]. For cryptanalyzing a 256 gray-scale image encryption scheme, Liu *et al*. [34] specified that the two chosen queries are $(\hat{\alpha}, \hat{\beta}) = (0, 170)$ and $(\tilde{\alpha}, \tilde{\beta}) = (170, 85)$. In addition to Li's achievements, it was investigated in [33] that another two chosen queries in terms of (α, β) are also valid for determining k of DEA.
2) *Determine k when* (y, α, β) *tuples are known but unselectable*. This assumption appears to be similar to a known-plaintext attack. Unlike the precise recovery by some chosen (α, β) queries, determining k with some known (y, α, β) tuples is relatively difficult. Generally, researchers attempted to derive a probability of determining k_i in this scenario [31]–[33]. In [31], *Li* first sought to obtain k from known (y, α, β) tuples; however, the presented achievements are obtained by utilizing some special properties of the studies image encryption schemes [28], [29] and cannot be directly extended to other similar cryptosystems. In [35], it is proposed to continuously narrow the possible candidates of k and finally determine k using the known plaintexts and ciphertexts. Subsequently, Zhang *et al*. [32], [33] investigated a general method to determine k of DEA from g known (y, α, β) tuples. The recovery probability has been mathematically deduced and experimentally validated.
3) *Determine k from* (y, α, β) *tuples while y can be freely chosen*. This assumption generally corresponds to a chosen-ciphertext attack in cryptanalysis. In the previous works [30]–[34], there was no specific discussion about this issue, i.e., determining k when y can be freely chosen and the corresponding (α, β) are available. Based on the presented properties or propositions, a conclusion can also be drawn. Typical achievements can be found in [32], [33] who reported that the $k_0 \sim k_i$ can be determined if $y_0 \sim y_i$ are all ones. In other words, if consecutive ones were observed in y, then equal amounts of consecutive bits can be derived definitely. Accordingly, k can be fully determined with one (y, α, β) tuple if the bits of y are all ones.

It is noted that the required (y, α, β) counts for breaking DEA are not the numbers of the required chosen-plaintexts, known-plaintexts or chosen-ciphertexts when cryptanalyzing a real image cryptosystem. The (α, β) of the DEA refers to two plaintexts of an image cryptosystem, while y is the differential of the ciphertexts of α and β. Therefore, g tuples of (y, α, β) always require more than g plaintexts. For example, the two chosen queries in [31], [33] refer to three chosen-plaintexts and corresponding ciphertexts of the cryptosystem in [29], while the g known (y, α, β) tuples required in [32], [33] correspond to at least $g + 1$ couples of known plaintexts and ciphertexts.

III. Main Results

A. Problem Formulation

Unlike the previous works that seek to derive k by DEA, i.e., $y = (\alpha \dot{+} k) \oplus (\beta \dot{+} k)$, this work attempts to solve k directly from the diffusion equation itself. Without loss of the generality, Eq. (1) is first taken as an example. We can obviously obtain

$$c(n) \oplus c(n-1) = [p(n) \dot{+} k(n)] \oplus k(n).$$

A generalized form is thus derived as Eq. (3), that is

$$y = (p \dot{+} k) \oplus k.$$

The counterparts [31], [33] strive to solve this problem through a differential fashion as $y = (\alpha \dot{+} k) \oplus (\beta \dot{+} k)$. On the other hand, this paper seeks to determine k directly from its original fashion, i.e., $y = (p \dot{+} k) \oplus k$. Suppose $y, p, k \in \mathbb{Z}_2^N$, given a collection of (y, p) pairs, the following subsections focus on determining k satisfying $y = (p \dot{+} k) \oplus k$ in various assumptions.

B. Some Properties of $y = (p \dot{+} k) \oplus k$

We start with the bitwise representation of $y = (p \dot{+} k) \oplus k$. Assume that $t = p \dot{+} k$, its iteration form is [33]

$$\begin{cases} t_i = p_i \oplus k_i \oplus \gamma_i \\ \gamma_0 = 0; \\ \gamma_i = p_{i-1} k_{i-1} \oplus k_{i-1} \gamma_{i-1} \oplus \gamma_{i-1} p_{i-1}, \quad i \geq 1, \end{cases} \tag{19}$$

where γ_i is the carry bit of the i-th bit plane of $t = p \dot{+} k$. The i-th bit of y is further obtained as

$$\begin{aligned} y_i &= t_i \oplus k_i \\ &= p_i \oplus k_i \oplus \gamma_i \oplus k_i \\ &= p_i \oplus \gamma_i. \end{aligned} \tag{20}$$

Combining Eqs. (19) and (20), we can obtain the iteration pattern of $y = (p \dot{+} k) \oplus k$ as Eq. (21).

$$\begin{cases} y_i = p_i \oplus \gamma_i \\ \gamma_0 = 0; \\ \gamma_i = p_{i-1}k_{i-1} \oplus k_{i-1}\gamma_{i-1} \oplus \gamma_{i-1}p_{i-1}, \quad i \geq 1 \end{cases} \tag{21}$$

Proposition 1: The highest bit of k, i.e., k_{N-1}, has no effect on the result of $y = (p \dot{+} k) \oplus k$. That means that if k satisfies $y = (p \dot{+} k) \oplus k$, then $k \oplus 2^{N-1}$ is an equivalent solution.

Proof: As revealed from Eq. (21), $y_{N-1} = p_{N-1} \oplus \gamma_{N-1}$ while γ_{N-1} is produced by p_{N-2}, k_{N-2} and γ_{N-2}. The result of y_{N-1} is unrelated to k_{N-1}. Proof over. □

This proposition indicates that the adversary only needs to recover $k_0 \sim k_{N-2}$. The highest bit k_{N-1} is not required in the cryptanalysis. Even though described in different fashions and for different motivations, this property had been proved in peer works, such as Proposition 1 of [34].

Proposition 2: The bit $y_i = 1$ is the sole necessary condition for recovering k_i ($i \in [0, N-2]$).

Proof: When $i \in [0, N-2]$, Eq. (21) can be further calculated as

$$\begin{aligned} y_{i+1} &= p_{i+1} \oplus \gamma_{i+1} \\ &= p_{i+1} \oplus p_i k_i \oplus k_i \gamma_i \oplus \gamma_i p_i \\ &= p_{i+1} \oplus k_i (p_i \oplus \gamma_i) \oplus \gamma_i p_i \\ &= p_{i+1} \oplus k_i y_i \oplus \gamma_i p_i \end{aligned}. \tag{22}$$

As indicated, the information of k_i is only preserved in the form of $k_i y_i$. When $y_i = 1$, Eq. (22) will forward the information of k_i to y_{i+1}, otherwise, the information of k_i will be lost. Thus, $y_i = 1$ is necessary for recovering k_i.

Furthermore, when $y_i = 1$, k_i can be recovered by

$$k_i = y_{i+1} \oplus p_{i+1} \oplus \gamma_i p_i.$$

Referring to Eq. (21), $y_i = \gamma_i \oplus p_i = 1$ indicates $\gamma_i p_i \equiv 0$, k_i is consequently finalized as

$$k_i = y_{i+1} \oplus p_{i+1}. \tag{23}$$

Because y_{i+1} and p_{i+1} are known under the assumption that some (y, p) pairs have been collected, the value of k_i can be determined once $y_i = 1$.

To conclude, $y_i = 1$ is the sole necessary condition for recovering k_i. Hence the proof is completed. □

It should be emphasized that Proposition 2 is a significant advance similar to that in [32], where $y_i = 1$ is described as a necessary but not the sole necessary condition for recovering k_i. Proposition 2 also indicates that the highest bit of k, i.e., k_{N-1} cannot be recovered, because y_N of Eq. (23) was discarded by the modulo addition. Fortunately, k_{N-1} has been revealed to be unnecessary by Proposition 1.

Proposition 3: The recovery of k_i is independent of the recovery of $k_j (j \neq i)$.

Proof: This proposition can be regarded as a derivative of Proposition 2. Clearly, we can conclude from Eq. (22) that whether k_i is recoverable depends only on the value of y_i. In addition, it is straightforward from Eq. (23) that the value of k_i is completely determined by the values of y_{i+1} and p_{i+1} that are known in the assumption. Whether $k_j (j \neq i)$ is being recovered cannot change the recoverability as well as the derived value of k_i. Thus, the recoveries of k_i are independent of each other. Proof over. □

Proposition 3 has remarkable advantages over the cryptanalysis in [33], where recovering k_i relies on the value of k_{i-1}. This proposition also promotes the recovery accuracy of k_i when some (y, p) pairs are known but unselectable. Numerical comparisons will be given in Section III-D.

C. Determine k Under Various Assumptions

Suppose that g pairs of (y, p) satisfying $y = (p \dot{+} k) \oplus k$ have been collected, and they are denoted as $\mathbb{S} = \{[y(1), p(1)], [y(2), p(2)], \cdots, [y(g), p(g)]\}$.[4] Based on the aforementioned propositions, Algorithm 1 is developed to determine k from $\mathbb{S}$.

Algorithm 1 The Retrieval of k

Input: A set $\mathbb{S}$ including g pairs of (y, p)
Output: k satisfying $y = (p \dot{+} k) \oplus k$

```
Set k to a random number in [0, 2^N − 1]
for each i ∈ [0, N − 2] do
    for each j ∈ [1, g] do
        if y(j)_i == 1 then
            Update k_i as
            k_i = y(j)_{i+1} ⊕ p(j)_{i+1};
            break;
        end if
    end for
end for
return k;
```

Then, we discuss the accuracy of k under three conditions, i.e., when y is selectable, when p is selectable and when both y and p are unselectable. Mathematical proofs are given.

First, let us discuss the recovery of k when y is selectable. In specific, this scenario assumes that y can be freely chosen while the corresponding p is also known. We can conclude from Proposition 2 that when $y = 2^N - 1$ (all the bits of y are 1) or $y = 2^{N-1} - 1$ (all the bits of y are 1, except the highest bit), it is able to recover $k_i (i \in [0, N-2])$ from Eq. (23) or Algorithm 1 exactly. The values derived from $y = 2^N - 1$ and $y = 2^{N-1} - 1$ are equivalent. As discussed in Proposition 2, it is unable to recover the highest bit k_{N-1}. However, the recovered value is an equivalent of the original value, because Proposition 1 has proven that the highest bit k_{N-1} is not necessary for the cryptanalysis.

[4]Note that, $p(i)$ does not denote an image's pixel at coordinate i, it represents the i-th element of the collected g known plaintexts in an attack.

TABLE I

REQUIRED EQUIVALENT (y, p) PAIRS FOR PRECIOUSLY DETERMINING k

	Zhang's [33]	Li's [31]	Proposed
y is selectable	2	2	1
p is selectable	3	3	1

Remark 1: Given a pair of (y, p) of Eq. (3), $k_i (i \in [0, N-2])$ can be solely determined in the case that $y = 2^N - 1$ or $y = 2^{N-1} - 1$.

Second, we discuss the recovery of k when p is selectable. In this scenario, p can be freely constructed on demand and the corresponding y is known at the same time. Appendix A proves that only two chosen queries in terms of p and their corresponding y are sufficient to determine k of $y = (p \dot{+} k) \oplus k$ exactly. The chosen queries are $(\hat{p} = \sum_{j=0}^{\lceil N/2 \rceil - 1} 4^j,\ \tilde{p} = \sum_{j=0}^{\lfloor N/2 \rfloor - 1} 2 \cdot 4^j)$ or $(\hat{p} = \sum_{j=0}^{\lceil N/2 \rceil - 1} 4^j,\ \tilde{p} = \sum_{j=0}^{\lfloor N/2 \rfloor - 1} 2 \cdot 4^j + 1)$.[5] Taking $N = 8$ as an example, each pixel has 8-bit resolution and the gray scale is $2^8 - 1 = 255$. The aforementioned two chosen queries of p are $(\hat{p} = 85,\ \tilde{p} = 170)$ or $(\hat{p} = 85,\ \tilde{p} = 171)$. In binary representation, $\hat{p} = 85 = (01010101)_2$, $\tilde{p} = 170 = (10101010)_2$ or $\tilde{p} = 171 = (10101011)_2$.

Remark 2: Given two pairs of (y, p) of Eq. (3), i.e., $(\hat{y}, \hat{p})$ and $(\tilde{y}, \tilde{p})$, k_i $(i \in [0, N-2])$ can be solely determined in the case that $\hat{p} = \sum_{j=0}^{\lceil N/2 \rceil - 1} 4^j$ while $\tilde{p} = \sum_{j=0}^{\lfloor N/2 \rfloor - 1} 2 \cdot 4^j$ or $\tilde{p} = \sum_{j=0}^{\lfloor N/2 \rfloor - 1} 2 \cdot 4^j + 1$.

Finally, we try to recover k when y and p are known but both unselectable. Suppose that g pairs of (y, p) that satisfy $y = (p \dot{+} k) \oplus k$ have been collected. With the help of Properties 2 and 3, $k_i (i \in [0, N-2])$ can be recovered independently by Eqs. (22) and (23). Assuming that p and y are uniformly distributed, the probability that $y_i = 1$ of a known y is $1/2$. For g pairs of (y, p), the probability that there exists at least one y satisfies $y_i = 1$ is $1 - (1/2)^g$. Benefiting from Properties 2 and 3, Algorithm 1 is capable of determining k_i $(i \in [0, N-2])$ with probability $1 - (1/2)^g$.

Remark 3: Given g known pairs of (y, p) of Eq. (3), k_i $(i \in [0, N-2])$ can be independently determined with probability $1 - (1/2)^g$.

D. Discussion and Comparison

Our primary goal is to determine k of $y = (p \dot{+} k) \oplus k$ which is identical as the goals of the peer works [31], [33]. This paper innovatively solves this problem in its original form rather than using differential analysis [31], [33], and we found that each bit of k can be determined independently. Furthermore, the recovery accuracy of the proposed approach has advantages over the counterpart methods, as listed in Tables I and II.

First, when y is selectable, a single (y, p) pair is sufficient to recover k in the case that y is $2^N - 1$ or $2^{N-1} - 1$. On the other hand, even though a (y, α, β) tuple is also sufficient to recover k of $y = (\alpha \dot{+} k) \oplus (\beta \dot{+} k)$, it corresponds to two (y, p) pairs that discussed here. The proposed approach decreases the number of the required (y, p) pairs from two to one in the scenario that y is selectable.

[5]Other types of representation of these two numbers are also possible, as adopted in [30], [31], [33]

TABLE II

PROBABILITY OF DETERMINING k_i WITH g UNSELECTABLE (y, p) PAIRS

	Zhang's [33]	Li's [31]	Proposed
y and p are in-selectable	$(1 - (\frac{1}{2})^{g-1})^{i+1}$	$1 - (1 - \frac{1}{2^{i+1}})^{g-1}$	$1 - (\frac{1}{2})^g$

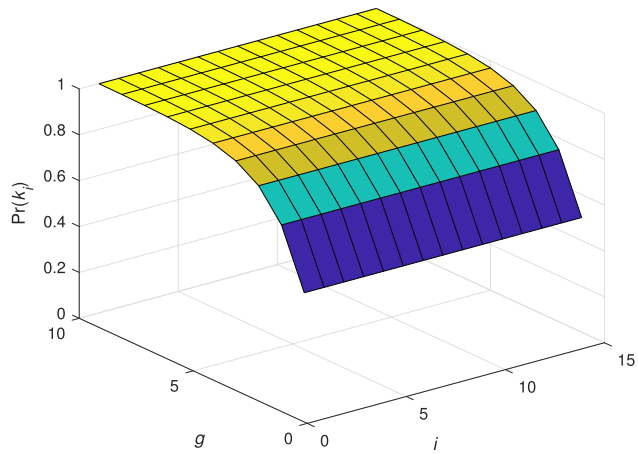


Fig. 1. Probability of determining k_i of Eq. (3) with respect to g known (y, p) pairs.

Second, as given in Remark 2, two (y, p) pairs can solely determine k given that the values of p are selectable. Comparatively, two (y, α, β) tuples that are equivalent to three (y, p) pairs are required by peer works [31], [33] to recover k precisely. The proposed method also has advantages to recover k when p is selectable.

Third, we discuss the probability of deriving k_i when g pairs of (y, p) are known yet unselectable. As concluded in Remark 3, k_i is recovered independently in this work. The probability for recovering k_i from g known (y, p) pairs is

$$\Pr(k_i) \equiv 1 - (\frac{1}{2})^g. \tag{24}$$

As indicated, each bit has identical probability to be recovered, and the probability increases exponentially with g. Even if only one (y, p) pair is known to the adversary, k_i can be recovered with the probability of 50%, and this value will be as large as 87.5% when 3 known pairs are available.

Our algorithm displays remarkable advances in comparison with peer works [31], [33]. The derivation of k_i relies on the recovery of k_{i-1} in [33], so that $\Pr(k_i)$ is equal to the probability of recovering all of the bits $k_0 \sim k_i$. In addition, Zhang *et al.* recovered k from the DEA, g pairs of (y, p) in this paper is equal to at most $g - 1$ known tuples (y, α, β) in [33]. The probability of recovering $k_i (i \in [0, N-2])$ by Zhang's algorithm [33], from g known pairs of (y, p) is [6]

$$\Pr(k_i) = (1 - (\frac{1}{2})^{g-1})^{i+1}.$$

On the other hand, there was no specific discussion regarding the derivation of k_i from known (y, p) pairs in [31]. The

[6]Interested readers can refer to the original paper for more details, and the probability is given in Section V-A on Page 7 of [33].

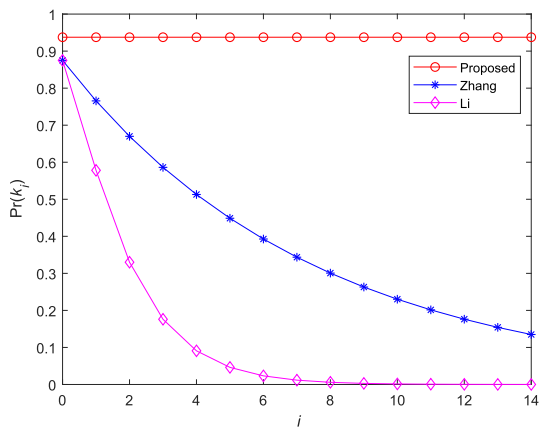


Fig. 2. Probability of determining k_i of Eq. (3) when 4 (y, p) pairs are available.

derivation of k_i in [31] also relies on the successful recovery of the previous bits, and can be presumed to have a probability of observing $i+1$ consecutive ones in a single y sample. Suppose that y is uniformly distributed, the probability of observing $i+1$ consecutive ones in a single y is $1/2^{i+1}$. Here, recovering k_i by Li's work [31] from g known-only (y, p) pairs is presumed with probability

$$\Pr(k_i) = 1 - (1 - \frac{1}{2^{i+1}})^{g-1}.$$

The calculated probabilities of the proposed algorithm and its counterparts [31], [33] are plotted in Fig. 2 where $N = 16$ and $g = 4$ are taken as an example. Even though the compared algorithms can obtain relatively similar probabilities to recover k_0, the probabilities of the counterparts [31], [33] when recovering higher bits decrease exponentially. On the other hand, k_i is recovered independently in our work with a fixed probability as described by Eq. (24). In addition, it is widely-known that higher-plane bit carries more information. As indicated from Fig. 2, our algorithm is more advantageous for recovering the bits at higher bit planes, and consequently superior for deriving the final values of k. It is also easy to observe that our algorithm become more advantageous relative to its with decreasing g. In practical applications, it is reasonable to collect a small number rather than a large number of plaintext-ciphertext pairs. Therefore, the proposed algorithm is more practical in real applications.

In addition, the proposed algorithm is clearly advantageous when solving problems that are described as $y = (\alpha \dot{+} k) \oplus (\beta \dot{+} k)$. By fixing β as zero, deriving k of $y = (\alpha \dot{+} k) \oplus (\beta \dot{+} k)$ turns into the studied problem (Eq. (3)). The most important achievement is that k_i is found to be recoverable independently in this work. Compared with the counterparts [31], [33], Algorithm 1 significantly promotes the accuracy since determining k_i in [31], [33] relies on the values of previous bits of k.

This section indicates that the diffusion primitive $y = (p \dot{+} k) \oplus k$ is insecure and the encryption element k can be retrieved under various conditions. In addition, the following sections will further demonstrate that some permutation-diffusion encryption schemes using $y = (p \dot{+} k) \oplus k$ as encryption kernel are also insecure. However, the permutation-diffusion (substitution) network itself has been proven to be a secure architecture, as employed in AES. It is the vulnerability of $y = (p \dot{+} k) \oplus k$ that makes the whole cryptosystem insecure. In this direction, secure diffusion kernel is suggested to cooperate with a permutation module to build a complete cryptosystem. A diffusion equation originating from the Helix cipher, i.e.,

$$(k_1 \dot{+} k_2) \oplus (k_1 \dot{+} (k_2 \oplus p)) = y \tag{25}$$

can be used as an alternative [33], [37]. Paul *et al.* [38] has reported that the required queries of finding the unknown (k_1, k_2) of Eq. (25) is 2^{N-2} that approximates the theoretical value 2^N. Referring to the approach adopted in AES, diffusion with a lookup table is also suggested, yet the permutation-diffusion network must be iterated many times to promote the security.

IV. Applications for Cryptanalysis

In this section, cryptanalysis applications of the theoretical achievements in Section III will be demonstrated. Without loss of the generality, the test images are assumed to have a size of 512×512 and the gray scale is set as 256 (i.e., $N = 8$).

A. Cryptanalysis of Xie's Cryptosystem

Observing the diffusion formula of Xie's cryptosystem [25], we can obtain

$$c(i) \oplus c(i-1) = [p(i) \dot{+} k(i)] \oplus k(i),$$

where $c(i)$ and $c(i-1)$ represent the ciphered pixels and $p(i)$ refers to a pixel of the permutation ciphertext. Because the permutation procedure will change pixel locations, we cannot easily obtain the $p(i)$ on demand. Fortunately, the permutation cannot change pixel values, so that a plain image with identical pixels will remain the same after the permutation procedure. Benefiting from Remark 2, a chosen-plaintext attack is derived for cracking this cryptosystem [25].

First, two chosen-plaintexts and corresponding ciphertexts are sufficient to solely determine the diffusion masks $\boldsymbol{K}$. Referring to Remark 2, the pixels of the first chosen-plaintext are all 85 while those of the second chosen-plaintext are 170, as shown in Figs. 3(a) and 3(b), respectively. The recovered diffusion masks are shown in 3(c) where their highest bits are set to zero, because the highest bits are not necessary referring to Proposition 1. After recovering $\boldsymbol{K}$, the whole cryptosystem is relaxed as a permutation-only cipher for which the permutation vectors can be recovered by the some generalized methods in [17], [19], [20]. With the retrieved permutation vectors and diffusion masks, any receiving ciphertext can be recovered, as demonstrated in Figs. 3(d)-3(f).

B. Cryptanalysis of Parvin's Cryptosystem

As described in Section II-B, Parvin's cryptosystem [23] consists of a row/column circular permutation and a diffusion procedure based on Eq. (7). A $divide - and - conquer$ strategy is employed to independently recover the permutation element and diffusion mask under a chosen-plaintext attack.

The permutation vectors $\boldsymbol{U}$ and $\boldsymbol{V}$ are retrieved first. Compared with the generalized methods proposed in [17], [19], [20], by exploiting the security defects of the adopted

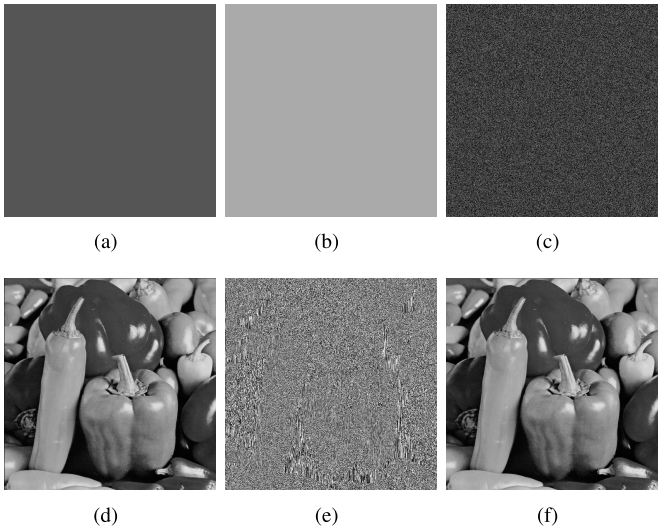


Fig. 3. Attacking results of [25]: (a) chosen-plaintext with pixel values of 85; (b) chosen-plaintext with pixel values of 170; (c) recovered diffusion masks; (d) a plaintext; (e) ciphertext of (d); (f) recovered image.

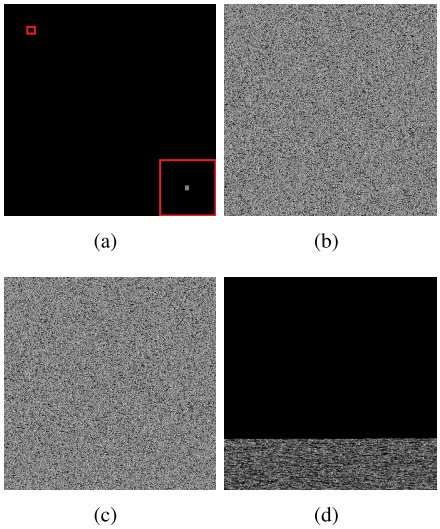


Fig. 4. Recovering the permutation vector of [23]: (a) the differential image of the plaintexts $\boldsymbol{M}$ and $\boldsymbol{M}'$; (b) the ciphertext $\boldsymbol{C}$; (c) the ciphertext $\boldsymbol{C}'$; (d) differential image of $\boldsymbol{C}$ and $\boldsymbol{C}'$.

row/column circular permutation, $\boldsymbol{U}$ and $\boldsymbol{V}$ can be recovered in a more efficient manner. Suppose that we have a chosen-plaintext $\boldsymbol{M} = \{m(i, j) \equiv 0\}$, and the corresponding ciphertext is $\boldsymbol{C}$. Then, we construct another plaintext $\boldsymbol{M}'$ as

$$\begin{cases} m'(l, l) = 127 \\ m'(i, j) = 0 \quad i \neq l, \ \ j \neq l,, \end{cases}$$

and denote its ciphertext as $\boldsymbol{C}'$. Of course, $m'(l, l)$ can be set to other values. According to Eqs. (5) and (6), this different pixel will be swapped to $(u(l), v(l))$ of the permutation ciphertext. Furthermore, this different pixel will cause large scale different pixels after the diffusion module. Based on Eq. (7), the different pixels between $\boldsymbol{C}$ and $\boldsymbol{C}'$ are sequentially distributed, starting from $(u(l), v(l))$ to (H, W). Therefore, $(u(l), v(l))$ is retrieved. An illustrative example is shown in Fig. 4, assuming $l = 66$ without loss of the generality. Figure 4(a) is the differential image of $\boldsymbol{M}$ and $\boldsymbol{M}'$, there is only one different

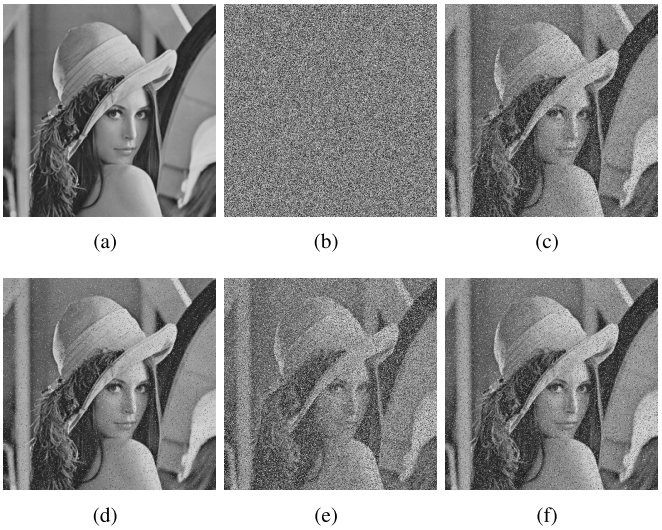


Fig. 5. Attacking results of [23]: (a) plaintext with size 512×512; (b) the ciphertext of (a); (c) recovered image with $\boldsymbol{K}$ derived from 2 plaintext-ciphertext pair by the proposed algorithm; (d) recovered image with $\boldsymbol{K}$ derived from 4 plaintexts-ciphertexts pairs by the proposed algorithm; (e) recovered image with $\boldsymbol{K}$ derived from 2 plaintexts-ciphertexts pairs by Zhang's algorithm [33]; (f) recovered image with $\boldsymbol{K}$ derived from 4 plaintexts-ciphertexts pairs by Zhang's algorithm [33].

pixel at $(66, 66)$. The ciphertexts are given in Figs. 4(b) and 4(c), and they are noise-like in appearance. However, their differential image clearly shows the different pixel between $\boldsymbol{C}$ and $\boldsymbol{C}'$. From numerical comparison, the first non-zero pixel of the differential image is found at $(389, 338)$. Therefore, we can conclude that $u(66) = 389$ and $v(66) = 338$. By traversing all of the diagonal pixels of $\boldsymbol{M}$, all of the elements of $\boldsymbol{U}$ and $\boldsymbol{V}$ can be recovered.

After obtaining the permutation vectors, the cryptosystem is relaxed as a diffusion-only system. The diffusion mask $\boldsymbol{K}$ is the remainder encryption elements to be recovered. Observing Parvin's diffusion by Eq. (7), we can obtain

$$c(i) \oplus p(i) = [c(i-1) \dot{+} k(i)] \oplus k(i). \tag{26}$$

Because $c(i)$ and $c(i-1)$ are un-controllable in plaintext attacks, recovering $k(i)$ of Eq. (26) consequently corresponds to determining k of $y = (p \dot{+} k) \oplus k$ in the scenario where y and p are both unselectable. We can refer to Algorithm 1 and determine $k(i)$ one by one with the probability given in Remark 3. With the permutation vectors $\boldsymbol{U}$, $\boldsymbol{V}$, and the diffusion mask $\boldsymbol{K}$, every ciphertext can be recovered. The results are demonstrated in Fig. 5, where Fig. 5(a) is the 512×512 plaintext and its ciphertext is shown in Fig. 5(b). When $\boldsymbol{K}$ is derived from 2 and 4 plaintext-ciphertext pairs, the corresponding deciphered images are shown in Figs. 5(c) and 5(d), respectively. The recovered image's quality increases significantly with the counts of the collected plaintext-ciphertext pairs, and it matches the probability given in Remark 3.

In addition, Parvin's encryption scheme was also cryptanalyzed in [33]. Figures 5(e) and 5(f) demonstrate the decrypted images using Zhang's algorithm to recover $\boldsymbol{K}$ from 2 and 4 plaintext-ciphertext pairs, respectively. We can visually observe that Fig. 5(e) is noisier than Fig. 5(c) even though equal counts of plaintext-ciphertext pairs are employed to

recover the diffusion element $\boldsymbol{K}$. Specifically, the total count of error bits of Fig. 5(e), compared with the plaintext Fig. 5(a), is 358257, whereas that of Fig. 5(c) is only 181070. Similarly, when 4 plaintext-ciphertext pairs are available for determining $\boldsymbol{K}$, the decrypted image obtained by the proposed algorithm (Fig. 5(d)) has only 48487 error bits while that decrypted by Zhang's algorithm has 110287 incorrect bits (Fig. 5(f)). The advantage of the proposed algorithm has been well demonstrated.

C. Cryptanalysis of Sam's Cryptosystem

The cryptosystem in [26] is developed for encrypting color images, and the diffusion masks of the RGB channels are identical. In other words, we can directly obtain three chosen (y, p) pairs from one color plaintext and its ciphertext.

Only one chosen plaintext and the corresponding ciphertext is sufficient to derive the equivalent diffusion masks. Specifically, we can derive the equivalent diffusion masks with a chosen-plaintext for which the pixels in the R, G, B channels are identical and are 0, 85, 170, respectively. Observing that $0 = 0 \ggg 4$, $85 = 85 \ggg 4$, $170 = 170 \ggg 4$ and both Sam's permutation and zigzag rescanning are useless for shuffling an image with identical pixels, the cipher pixels in the RGB channels are obtained according to Eqs. (9)-(15) as

$$\begin{cases} cr(i) = x(zig(i)) \oplus y(zig(i)) \oplus cr(i-1) \oplus z(i) \\ cg(i) = [85 + x(zig(i))] \oplus y(zig(i)) \oplus cg(i-1) \oplus z(i) \\ cb(i) = [170 + x(zig(i))] \oplus y(zig(i)) \oplus cr(i-1) \oplus z(i). \end{cases} \tag{27}$$

Furthermore, we can obtain

$$\begin{cases} cr(i) \oplus cr(i-1) \oplus cg(i) \oplus cg(i-1) \\ \quad = [85 \dot{+} x(zig(i))] \oplus x(zig(i)) \\ cr(i) \oplus cr(i-1) \oplus cb(i) \oplus cb(i-1) \\ \quad = [170 \dot{+} x(zig(i))] \oplus x(zig(i)). \end{cases} \tag{28}$$

Benefiting from Remark 2, $x(zig(i))$ can be first determined. Then, referring to Eq. (27), $y(zig(i)) \oplus z(i)$ is obtained via

$$y(zig(i)) \oplus z(i) = cr(i) \oplus cr(i-1) \oplus x(zig(i)).$$

We can straightforwardly use $x(zig(i))$ and $y(zig(i)) \oplus z(i)$ as the equivalent diffusion masks for decrypting the ciphertexts of Sam's cryptosystem, rather than deriving the specific values of $x(i)$, $y(i)$ and $z(i)$.

With the retrieved $x(zig(i))$ and $y(zig(i)) \oplus z(i)$, Sam's cryptosystem degrades into a permutation-only encryption scheme. Due to the intrinsic loophole of Eq. (8), it was revealed in [35] that only one more chosen-plaintext is feasible to accurately derive the permutation vector. In summary, two chosen plaintexts and their ciphertexts are sufficient to recover the permutation vector and equivalent diffusion masks of Sam's cryptosystem.

Experimental results on cracking Sam's cryptosystem [26] are shown in Fig. 6. The chosen-plaintext for deriving the (equivalent) diffusion masks is given in Fig. 6(a), while Figs. 6(b) and 6(c) show the retrieved matrices of $x(zig(i))$ and $y(zig(i)) \oplus z(i)$, respectively. A color image *peppers* is employed for validation, and Fig. 6(e) shows its ciphertext

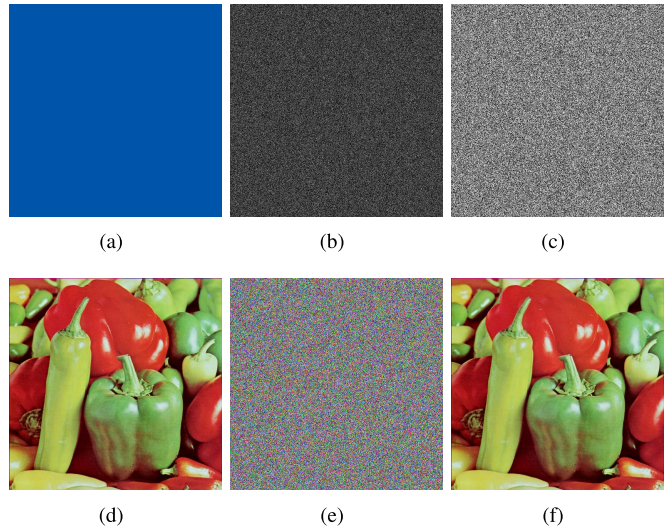


Fig. 6. Attacking results of [26]: (a) chosen-plaintext with size 512×512; (b) retrieved matrix of $x(zig(i))$; (c) retrieved matrix of $y(zig(i)) \oplus z(i)$; (d) a plaintext *peppers*; (e) ciphertext of (d); (f) recovered image obtained with the derived equivalent encryption elements.

while Fig. 6(f) shows the attack result using the derived equivalent encryption elements. Numerical comparison verifies that the original image has been accurately recovered.

V. Conclusion

By studying the recovery of k satisfying $y = (p \dot{+} k) \oplus k$, this paper re-evaluates the security of a family of image diffusion mechanisms. The determination of k in various conditions have been understood, and the accuracy bounds are mathematically deduced. Compared to the counterpart methods, our algorithm can determine k with higher probability and less prior knowledge. The proposed method is further applied to break three cryptosystems using variants of $y = (p \dot{+} k) \oplus k$ for image diffusion. Experimental results are given for validation. The security strength of $y = (p \dot{+} k) \oplus k$ indicated by this paper is expected to benefit both the design and cryptanalysis of a family of image encryption schemes.

Appendix A
Deduction of Remark 2

Given some pairs of $(y\ \ p)$ of Eq. (3), this appendix will demonstrate the required patterns of p for solely determining k_i $(i \in [0, N-2])$.

1) *Recovering* k_0. Benefiting from Proposition 2, if there is a $(\hat{p}, \hat{y})$ pair of Eq. (3) that ensures $\hat{y}_0 = 1$, then k_0 can be recovered from Eq. (23). Referring to Eq. (21), $\hat{y}_0 = 1$ definitely holds in the case that $\hat{p}_0 = 1$.
2) *Recovering* k_1. Referring to Eqs. and (21) and (22), we can obtain

$$\begin{aligned} \hat{y}_1 &= \hat{p}_1 \oplus k_0 \hat{y}_0 \oplus \hat{\gamma}_0 \hat{p}_0 \\ &= \hat{p}_1 \oplus k_0 \wedge 1 \oplus 0 \wedge \hat{p}_0 \\ &= \hat{p}_1 \oplus k_0 \end{aligned} \tag{29}$$

As mentioned above, k_0 is guaranteed to be recoverable when $\hat{p}_0 = 1$, yet its value may be 0 or 1. A fixed value of $\hat{p}_1$ cannot ensure $\hat{y}_1 = 1$ which is the sole necessary condition for recovering k_1. Therefore, another pair of

TABLE III

THE VALUES OF $\hat{y}_1$ AND $\tilde{y}_1$ IN VARIOUS CONDITIONS

Col	$\hat{p}_1$	$\tilde{p}_1$	$\tilde{p}_0$	k_0	$\hat{y}_1$	$\tilde{y}_1$
1	0	0	0	0	0	0
2	0	0	0	1	1	0
3	0	0	1	0	0	0
4	0	0	1	1	1	1
5	0	1	0	0	0	1
6	0	1	0	1	1	1
7	0	1	1	0	0	1
8	0	1	1	1	1	0
9	1	0	0	0	1	0
10	1	0	0	1	0	0
11	1	0	1	0	1	0
12	1	0	1	1	0	1
13	1	1	0	0	1	1
14	1	1	0	1	0	1
15	1	1	1	0	1	1
16	1	1	1	1	0	0

Eq. (3), denoted as $(\tilde{p}, \tilde{y})$, is required. With the help of Eqs. (21) and (22), we can obtain $\tilde{y}_1$ as

$$\begin{aligned}\tilde{y}_1 &= \tilde{p}_1 \oplus k_0\tilde{y}_0 \oplus \tilde{\gamma}_0\tilde{p}_0 \\ &= \tilde{p}_1 \oplus k_0\tilde{y}_0 \oplus 0 \wedge \tilde{p}_0 \\ &= \tilde{p}_1 \oplus k_0\tilde{y}_0 \end{aligned} \tag{30}$$

To recover k_1, the values of $\hat{y}_1$ and $\tilde{y}_1$ must include at least a 1. By exhaustive search in term of $\hat{y}_1$, $\tilde{y}_1$, $\tilde{y}_0$ and k_0, the values of $\hat{y}_1$ and $\tilde{y}_1$ are listed in Table III. Observing columns 5∼8 of Table III, $\hat{y}_1$ and $\tilde{y}_1$ have at least one positive value when $(\hat{p}_1, \tilde{p}_1) = (0, 1)$, independent of $\tilde{p}_0$ and k_0.

To conclude, $(\hat{p}, \hat{y})$ and $(\tilde{p}, \tilde{y})$ are required to derive k_0 and k_1 of Eq. (3), while $\hat{p}_0 = 1$, $\hat{p}_1 = 0$ and $\tilde{p}_1 = 1$ yet $\tilde{p}_0$ can be 0 or 1.

3) *Recovering* $k_i (i \in [2, N-2])$. A generalized deduction is employed for deriving $k_i (i \in [2, N-2])$, under the assumption that k_{i-1} has been successfully recovered with two chosen queries $(\widehat{p}, \widehat{y})$, $(\breve{p}, \breve{y})$ with $\widehat{p}_{i-1} = 0$ and $\breve{p}_{i-1} = 1$. In this circumstance, Appendix B demonstrates that when $\widehat{p}_i = 1$ and $\breve{p}_i = 0$, the values of $\widehat{y}_i$, $\breve{y}_i$ have at least one 1. Therefore, k_i can be determined according to Proposition 2.

Clearly, recovering k_2 corresponds to $i = 2$, $\hat{p} = \widehat{p}$, $\tilde{p} = \breve{p}$. In other words, $\hat{p}_2 = 1$ while $\tilde{p}_2 = 0$ can guarantee the recovery of k_2. Similarly, deriving k_3 corresponds to $i = 3$, $\hat{p} = \breve{p}$, $\tilde{p} = \widehat{p}$. Therefore, $\hat{p}_3 = 0$ while $\tilde{p}_3 = 1$ helps to determine k_3. Such a deduction can be repeated for the recovery of $k_4, k_5, \cdots, k_{N-2}$ analogously. To summarize, we obtain the rule of $\hat{p}$ and $\tilde{p}$ as Eq. (31).

$$\begin{cases} \hat{p}_0 = 1 \\ \hat{p}_i = 1 \oplus \hat{p}_{i-1} \ (i \geq 1) \\ \tilde{p}_0 = 1 \text{ or } 0 \\ \tilde{p}_i = 1 \oplus \hat{p}_i \ (i \geq 1) \end{cases} \tag{31}$$

Numerically,[7] $\hat{p} = \sum_{j=0}^{\lceil N/2 \rceil - 1} 4^j$ while $\tilde{p} = \sum_{j=0}^{\lfloor N/2 \rfloor - 1} 2 \cdot 4^j$ or $\sum_{j=0}^{\lfloor N/2 \rfloor - 1} 2 \cdot 4^j + 1$.

[7]Clearly, there are other representations of these numbers [30], [31], [33].

TABLE IV

VALUES OF $\widehat{y}_{i-1}$ AND $\breve{y}_{i-1}$ IN VARIOUS CONDITIONS

Col	$\widehat{p}_{i-1}$	$\breve{p}_{i-1}$	$\widehat{\gamma}_{i-1}$	$\breve{\gamma}_{i-1}$	$\widehat{y}_{i-1}$	$\breve{y}_{i-1}$
1	0	1	0	0	0	1
2	0	1	0	1	0	0
3	0	1	1	0	1	1
4	0	1	1	1	1	0

TABLE V

POSSIBLE VALUES OF $\widehat{y}_i$ AND $\breve{y}_i$ WHEN $\widehat{p}_i = 0$ AND $\breve{p}_i = 0$

Col	$\widehat{p}_{i-1}$	$\breve{p}_{i-1}$	$\widehat{\gamma}_{i-1}$	$\breve{\gamma}_{i-1}$	k_i	$\widehat{y}_{i-1}$	$\breve{y}_{i-1}$
1	0	1	0	0	0	0	0
2	0	1	1	0	0	0	0
3	0	1	1	1	0	0	1
4	0	1	0	0	1	0	1
5	0	1	1	0	1	1	1
6	0	1	1	1	1	1	1

TABLE VI

POSSIBLE VALUES OF $\widehat{y}_i$ AND $\breve{y}_i$ WHEN $\widehat{p}_i = 0$ AND $\breve{p}_i = 1$

Col	$\widehat{p}_{i-1}$	$\breve{p}_{i-1}$	$\widehat{\gamma}_{i-1}$	$\breve{\gamma}_{i-1}$	k_i	$\widehat{y}_{i-1}$	$\breve{y}_{i-1}$
1	0	1	0	0	0	0	1
2	0	1	1	0	0	0	1
3	0	1	1	1	0	0	0
4	0	1	0	0	1	0	0
5	0	1	1	0	1	1	0
6	0	1	1	1	1	1	0

Summarizing the aforementioned three items, two pairs of chosen queries in terms of p are sufficient to derive k of Eq. (3). They are denoted as $(\hat{y}, \hat{p})$ and $(\tilde{y}, \tilde{p})$, requiring that $\hat{p} = \sum_{j=0}^{\lceil N/2 \rceil - 1} 4^j$ while $\tilde{p} = \sum_{j=0}^{\lfloor N/2 \rfloor - 1} 2 \cdot 4^j$ or $\sum_{j=0}^{\lfloor N/2 \rfloor - 1} 2 \cdot 4^j + 1$.

APPENDIX B
GENERALIZED REQUIREMENTS FOR RECOVERING k_i $(i \geq 2)$

Assuming that k_{i-1} of Eq. (3) has been successfully recovered with two chosen queries $(\widehat{p}, \widehat{y})$, $(\breve{p}, \breve{y})$ with $\widehat{p}_{i-1} = 0$ and $\breve{p}_{i-1} = 1$, here we demonstrate the required patterns of $\widehat{p}_i$ and $\breve{p}_i$ for recovering k_i. By Proposition 2, it is equivalent to ensure that $\widehat{y}_i$ and $\breve{y}_i$ have at least one 1.

According to Proposition 2 and Eq. (21), the aforementioned assumption further indicates that when $\widehat{p}_{i-1} = 0$ and $\breve{p}_{i-1} = 1$, $\widehat{y}_{i-1} = \widehat{p}_{i-1} \oplus \widehat{\gamma}_{i-1}$ and $\breve{y}_{i-1} = \breve{p}_{i-1} \oplus \breve{\gamma}_{i-1}$ have at least one 1. The possible values of $\widehat{y}_{i-1}$ and $\breve{y}_{i-1}$ when $\widehat{p}_{i-1} = 0$ and $\breve{p}_{i-1} = 1$ have been listed in Table IV. To ensure $\widehat{y}_i$ and $\breve{y}_i$ have at least one 1, the valid combinations of $(\widehat{\gamma}_{i-1}, \breve{\gamma}_{i-1})$ are $(0, 0)$, $(1, 0)$ and $(1, 1)$.

Referring to Eq. (21), the values of $\widehat{y}_i$, $\breve{y}_i$ is determined by $\widehat{p}_i$, $\breve{p}_i$ $\widehat{p}_{i-1}$, $\breve{p}_{i-1}$, $\widehat{\gamma}_{i-1}$, $\breve{\gamma}_{i-1}$, and k_{i-1}. Because $(\widehat{p}_{i-1}, \breve{p}_{i-1})$ are definite as $(0, 1)$, $(\widehat{\gamma}_{i-1}, \breve{\gamma}_{i-1})$ have three possible combinations as mentioned above and k_i has two possible values, together with four candidates of $(\widehat{p}_i, \breve{p}_i)$, there are a total of 24 combinations for calculating $\widehat{y}_i$ and $\breve{y}_i$. Aiming to obtain the suggested $(\widehat{p}_i, \breve{p}_i)$ candidate, we categorize the $(\widehat{y}_i, \breve{y}_i)$ results in terms of the four $(\widehat{p}_i, \breve{p}_i)$ candidates. Consequently, there are four suites of results and each suite has six items, as listed in Tables V ∼ VIII.

TABLE VII

POSSIBLE VALUES OF $\widehat{y}_i$ AND $\breve{y}_i$ WHEN $\widehat{p}_i = 1$ AND $\breve{p}_i = 0$

Col	$\widehat{p}_{i-1}$	$\breve{p}_{i-1}$	$\widehat{\gamma}_{i-1}$	$\breve{\gamma}_{i-1}$	k_i	$\widehat{y}_{i-1}$	$\breve{y}_{i-1}$
1	0	1	0	0	0	1	0
2	0	1	1	0	0	1	0
3	0	1	1	1	0	1	1
4	0	1	0	0	1	1	1
5	0	1	1	0	1	0	1
6	0	1	1	1	1	0	1

TABLE VIII

POSSIBLE VALUES OF $\widehat{y}_i$ AND $\breve{y}_i$ WHEN $\widehat{p}_i = 1$ AND $\breve{p}_i = 1$

Col	$\widehat{p}_{i-1}$	$\breve{p}_{i-1}$	$\widehat{\gamma}_{i-1}$	$\breve{\gamma}_{i-1}$	k_i	$\widehat{y}_{i-1}$	$\breve{y}_{i-1}$
1	0	1	0	0	0	1	1
2	0	1	1	0	0	1	1
3	0	1	1	1	0	1	0
4	0	1	0	0	1	0	0
5	0	1	1	0	1	0	0
6	0	1	1	1	1	0	0

It is observed that only in the case that $(\widehat{p}_i, \breve{p}_i) = (1, 0)$, the resultant values of $\widehat{y}_i$ and $\breve{y}_i$ include at least one 1. The recovery of k_i can be further ensured.

REFERENCES

[1] Z. Qian, H. Zhou, X. Zhang, and W. Zhang, "Separable reversible data hiding in encrypted JPEG bitstreams," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 6, pp. 1055–1067, Nov. 2018.

[2] D. Megias, "Improved privacy-preserving P2P multimedia distribution based on recombined fingerprints," *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 2, pp. 179–189, Mar. 2015.

[3] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystem," *Int. J. Bifurcation Chaos*, vol. 16, no. 8, pp. 2129–2151, 2006.

[4] M. Preishuber, T. Hutter, S. Katzenbeisser, and A. Uhl, "Depreciating motivation and empirical security analysis of chaos-based image and video encryption," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 9, pp. 2137–2150, Sep. 2018.

[5] S. Chen, S. Yu, J. Lu, G. Chen, and J. He, "Design and FPGA-based realization of a chaotic secure video communication system," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 9, pp. 2359–2371, Sep. 2018.

[6] X. Kang and R. Tao, "Color image encryption using pixel scrambling operator and reality-preserving MPFRHT," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 7, pp. 1919–1932, Jul. 2019.

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

[8] C. Li, Y. Zhang, and E. Y. Xie, "When an attacker meets a cipher-image in 2018: A year in review," *J. Inf. Secur. Appl.*, vol. 48, Oct. 2019, Art. no. 102361.

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

[10] M. Li, H. Fan, Y. Xiang, Y. Li, and Y. Zhang, "Cryptanalysis and improvement of a chaotic image encryption by first-order time-delay system," *IEEE Multimedia Mag.*, vol. 25, no. 3, pp. 92–101, Jul. 2018.

[11] M. Zhou and C. Wang, "A novel image encryption scheme based on conservative hyperchaotic system and closed-loop diffusion between blocks," *Signal Process.*, vol. 171, Jun. 2020, Art. no. 107484.

[12] C. Xu, J. Sun, and C. Wang, "An image encryption algorithm based on random walk and hyperchaotic systems," *Int. J. Bifurcation Chaos*, vol. 30, no. 4, Mar. 2020, Art. no. 2050060.

[13] S. Wang, C. Wang, and C. Xu, "An image encryption algorithm based on a hidden attractor chaos system and the Knuth–Durstenfeld algorithm," *Opt. Lasers Eng.*, vol. 128, May 2020, Art. no. 105995.

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

[15] 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, Jul. 2004.

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

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

[18] J. Chen, L. Chen, and Y. Zhou, "Universal chosen-ciphertext attack for a family of image encryption schemes," *IEEE Trans. Multimedia*, early access, Jul. 24, 2020, doi: 10.1109/TMM.2020.3011315.

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

[20] A. Jolfaei, X.-W. Wu, and V. Muthukkumarasamy, "On the security of permutation-only image encryption schemes," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 2, pp. 235–246, Feb. 2016.

[21] L. Y. Zhang, Y. Liu, C. Wang, J. Zhou, Y. Zhang, and G. Chen, "Improved known-plaintext attack to permutation-only multimedia ciphers," *Inf. Sci.*, vols. 430–431, pp. 228–239, Mar. 2018.

[22] J. Chen, L. Chen, and Y. Zhou, "Cryptanalysis of image ciphers with permutation-substitution network and chaos," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Sep. 4, 2020, doi: 10.1109/TCSVT.2020.3021908.

[23] Z. Parvin, H. Seyedarabi, and M. Shamsi, "A new secure and sensitive image encryption scheme based on new substitution with chaotic function," *Multimedia Tools Appl.*, vol. 75, no. 17, pp. 10631–10648, Sep. 2016.

[24] L. Li *et al.*, "Exploiting optical chaos for color image encryption and secure resource sharing in cloud," *IEEE Photon. J.*, vol. 11, no. 3, pp. 1–12, Jun. 2019.

[25] Y. Xie, J. Li, Z. Kong, Y. Zhang, X. Liao, and Y. Liu, "Exploiting optics chaos for image encryption-then-transmission," *J. Lightw. Technol.*, vol. 34, no. 22, pp. 5101–5109, Nov. 15, 2016.

[26] I. S. Sam, P. Devaraj, and R. S. Bhuvaneswaran, "A novel image cipher based on mixed transformed logistic maps," *Multimedia Tools Appl.*, vol. 56, no. 2, pp. 315–330, Jan. 2012.

[27] B. Mondal, P. Kumar, and S. Singh, "A chaotic permutation and diffusion based image encryption algorithm for secure communications," *Multimedia Tools Appl.*, vol. 77, no. 23, pp. 31177–31198, Dec. 2018.

[28] K. D. Rao and C. Gangadhar, "Modified chaotic key-based algorithm for image encryption and its VLSI realization," in *Proc. 15th Int. Conf. Digit. Signal Process.*, Jul. 2007, pp. 439–442.

[29] C. Gangadhar and K. Deergha Rao, "Hyperchaos based image encryption," *Int. J. Bifurcation Chaos*, vol. 19, no. 11, pp. 3833–3839, Nov. 2009.

[30] C. Li, M. Z. Q. Chen, and K.-T. Lo, "Breaking an image encryption algorithm based on chaos," *Int. J. Bifurcation Chaos*, vol. 21, no. 7, pp. 2067–2076, Jul. 2011.

[31] C. Li, Y. Liu, L. Y. Zhang, and M. Z. Q. Chen, "Breaking a chaotic image encryption algorithm based on modulo addition and XOR operation," *Int. J. Bifurcation Chaos*, vol. 23, no. 4, Apr. 2013, Art. no. 1350075.

[32] L. Y. Zhang, Y. Zhang, Y. Liu, A. Yang, and G. Chen, "Security analysis of some diffusion mechanisms used in chaotic ciphers," *Int. J. Bifurcation Chaos*, vol. 27, no. 10, Sep. 2017, Art. no. 1750155.

[33] L. Y. Zhang *et al.*, "On the security of a class of diffusion mechanisms for image encryption," *IEEE Trans. Cybern.*, vol. 48, no. 4, pp. 1163–1175, Apr. 2018.

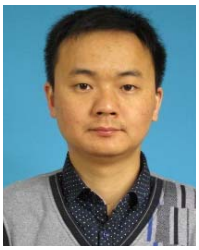
[34] Y. Liu, L. Y. Zhang, J. Wang, Y. Zhang, and K.-W. Wong, "Chosen-plaintext attack of an image encryption scheme based on modified permutation–diffusion structure," *Nonlinear Dyn.*, vol. 84, no. 4, pp. 2241–2250, Jun. 2016.

[35] Y. Zhang, D. Xiao, W. Wen, and M. Li, "Cryptanalyzing a novel image cipher based on mixed transformed logistic maps," *Multimedia Tools Appl.*, vol. 73, no. 3, pp. 1885–1896, Dec. 2014.

[36] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. London, U.K.: Chapman & Hall/CRC, 2014.

[37] J. Chen, L. Chen, L. Y. Zhang, and Z.-L. Zhu, "Medical image cipher using hierarchical diffusion and non-sequential encryption," *Nonlinear Dyn.*, vol. 96, no. 1, pp. 301–322, Apr. 2019.

[38] S. Paul and B. Preneel, "Near optimal algorithms for solving differential equations of addition with batch queries," in *Proc. Int. Conf. Cryptol. India*. New York, NY, USA: Springer, 2005, pp. 90–103.



Junxin Chen received the B.Sc., M.Sc., and Ph.D. degrees in communications engineering from Northeastern University, Shenyang, China, in 2007, 2009, and 2016, respectively. He is currently an Associate Professor with the College of Medicine and Biological Information Engineering, Northeastern University. He has authored/coauthored over 50 scientific paper in peer-reviewed journals and conferences, including IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE INTERNET OF THINGS JOURNAL, IEEE PHOTONICS JOURNAL, and *Information Sciences*. His research interests include the Internet of Medical Things, bio-signal processing, and compressive sensing, security, and privacy.

Leo Yu Zhang (Member, IEEE) received the bachelor's and master's degrees in computational mathematics from Xiangtan University, Xiangtan, China, in 2009 and 2012, respectively, and the Ph.D. degree from the City University of Hong Kong, Hong Kong, in 2016. He is currently a Lecturer with the School of Information Technology, Deakin University, Burwood, VIC, Australia. He held various research positions with the City University of Hong Kong, the University of Macau, Macau, the University of Ferrara, Ferrara, Italy, and the University of Bologna, Bologna, Italy. His current research interests include applied cryptography and AI-related security.

Yicong Zhou (Senior Member, IEEE) received the B.S. degree in electrical engineering from Hunan University, Changsha, China, and the M.S. and Ph.D. degrees in electrical engineering from Tufts University, Medford, MA, USA.

In 2011, he joined the University of Macau as an Assistant Professor with the Department of Computer and Information Science, where he is currently an Associate Professor and the Director of the Vision and Image Processing Laboratory. His research interests include image processing, computer vision, machine learning, and multimedia security. He is a fellow of the Society of Photo-Optical Instrumentation Engineers (SPIE). He received the Third Price of the Macao Natural Science Award as a sole winner in 2020 and a co-winner in 2014. Since 2015, he has been a leading Co-Chair of Technical Committee on Cognitive Computing in the IEEE Systems, Man, and Cybernetics (SMC) Society. He serves as an Associate Editor for IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, and four other journals. He received the Best Editor Award in 2020 to recognize his contribution to *Journal of Visual Communication and Image Representation*. He was recognized as the "Highly Cited Researcher" in Web of Science in 2020.