International Journal of Bifurcation and Chaos
© World Scientific Publishing Company

Cryptanalysis of some self-synchronous chaotic stream ciphers and their improved schemes

Baoju Chen[*], Simin Yu[†]
School of Automation, Guangdong University of Technology,
Guangzhou 510006, P. R. China
[*]*bogychan@foxmail.com*
[†]*siminyu@163.com*

David Day-Uei Li[‡]
Faculty of Science, University of Strathclyde,
Glasgow G4 0RE, U. K.
[‡]*David.Li@strath.ac.uk*

Jinhu Lü[§]
School of Automation Science and Electrical Engineering, Beihang University,
Beijing 100191, P. R. China
[§]*jhlu@iss.ac.cn*

Received (to be inserted by publisher)

In this paper, a cryptanalysis method that combines a chosen-ciphertext attack with a divide-and-conquer attack by traversing multiple non-zero component initial conditions (DCA-TMNCIC) is proposed. The method is used for security analysis of n-D (n=3,4,5,6,7,8) self-synchronous chaotic stream ciphers that employ a product of two chaotic variables and three chaotic variables (n-D SCSC-2 and n-D SCSC-3), taking 3-D SCSC-2 as a typical example for cryptanalysis. For resisting the combinational effect of the chosen-ciphertext attack and DCA-TMNCIC, several improved chaotic cipher schemes are designed, including 3-D SCSC based on a nonlinear nominal system (3-D SCSC-NNS) and n-D SCSC based on sinusoidal modulation (n-D SCSC-SM (n=3,4,5,6,7,8)). Theoretical security analysis validates the improved schemes.

Keywords: chosen-ciphertext attack; DCA-TMNCIC; chaotic stream cipher; n-D SCSC-2; 3-D SCSC-NNS; n-D SCSC-SM.

1. Introduction

From a theoretical perspective, there are many similarities between chaos theory and modern cryptography. Consequently, chaotic maps have been adopted to construct chaotic ciphers for information encryption. Since the logistic map was used in cryptography [Matthews, 1989], research on chaotic cryptosystems has been continuously and rapidly developed.

The working modes of chaotic cryptosystems can be roughly classified as non-feedback mode (NFM), plaintext association mode (PAM), and ciphertext feedback mode (CFM). For NFM, as plaintexts and ciphertexts are independent of the keystreams, the cryptanalyst can directly obtain equivalent keys using

Table 1. The deciphered NFM using basic cryptographic analysis methods

Analysis algorithm	Algorithm structure	Analysis method
Image encryption cryptosystem based on binary bit planes extraction and multiple chaotic maps [Shafique & Shahid, 2018]	Permutation-Diffusion	Chosen-plaintext attack [Wen & Yu, 2019]
Image encryption algorithm based on DNA encoding and spatiotemporal chaos [Song & Qiao, 2015]	Permutation-Diffusion	Chosen-plaintext attack and chosen-ciphertext attack [Wen *et al.*, 2019]
Image cipher based on 3D bit matrix and Latin cubes [Xu & Tian, 2019]	Permutation-Diffusion-Permutation	Chosen-plaintext attack and differential attack [Zhang & Yu, 2019]
Image encryption based on three-dimensional bit matrix permutation [Zhang *et al.*, 2016]	Permutation-Diffusion	Chosen-plaintext attack [Wu *et al.*, 2018]
Image encryption scheme using lookup table-based confusion and diffusion [Chen *et al.*, 2015]	Permutation-Diffusion	Chosen-plaintext attack and chosen-ciphertext attack [Hu *et al.*, 2017]

basic cryptanalysis methods (such as known-plaintext attack, chosen-plaintext attack, and chosen-ciphertext attack). Some examples are listed in Table 1. In addition, compressive sensing based methods have also been applied to the image encryption[Ye *et al.*, 2020b]. For encryption algorithms with compressive sensing, how to use the basic cryptanalysis methods to carry out the security analysis needs further research.

The main feature of PAM is that the keystream or the encryption process is related to plaintexts. It can be further classified into two subcategories: one is that the keystream depends on the original keys and the plaintext images [Ye & Huang, 2015; Zhao *et al.*, 2015; Mollaeefar *et al.*, 2017; Wu *et al.*, 2017; Diab, 2018]; another is that the encryption process is controlled by some specific characteristics of the plaintext images [Parvin *et al.*, 2016; Huang *et al.*, 2018; Niyat *et al.*, 2017; Zhou *et al.*, 2020]. Notably, many the cryptanalyst found that the equivalent keys in some PAM encryption algorithms can also be obtained using basic cryptographic methods. One example is the image encryption algorithm proposed in [Ye & Huang, 2015] based on autoblocking and electrocardiography, which was found [Li *et al.*, 2018a] to be weak because by the known-plaintext attack one can obtain the equivalent keys using only one pair of a known plain-image and its corresponding cipher-image. In [Li *et al.*, 2018b], the cryptanalysis of an encryption algorithm designed in [Niyat *et al.*, 2017] based on hybrid hyper-chaos and cellular automata proposed revealed three security drawbacks of the algorithm for which the equivalent keys can be obtained by using the chosen-plaintext attack. In PAM, the secret keys or features required for decryption are determined by plaintexts; however, the communication channels usually only transmit ciphertexts but not plaintexts. This makes it unsuitable for real-time transmission of multimedia data.

In CFM, ciphertexts are fed back into the encryption process or the underlying chaotic system, and the chaotic sequences generated by the chaotic system are related to the ciphertexts [Zhang *et al.*, 2019; Shahzadi *et al.*, 2019; Lin *et al.*, 2015; Chen *et al.*, 2018, 2020]. Therefore, unlike NFM and PAM, the cryptanalyst cannot directly obtain the equivalent keys but need to decipher the original keys. This is a clear advantage relative to the other two modes.

Note that the chaotic cryptosystems described above are all symmetric cipher. Besides, the asymmetric cipher is a hotspot in the research field of chaotic cryptosystems. For example, an asymmetric image encryption algorithm based on a fractional-order chaotic system and the RSA public-key cryptosystem was proposed in [Ye *et al.*, 2020a]. In this encryption algorithm, the public key is used for encryption and the private key for decryption, which can effectively address the issue of symmetric encryption key distribution and ensure the security of the encryption algorithm.

In [Lin *et al.*, 2015; Chen *et al.*, 2018, 2020], several self-synchronous chaotic stream ciphers in CFM, based on chaos anti-control principles, were designed and applied to multimedia chaotic secure communications. The ciphertext formats are shown in Table 2, where $m(k)$ denotes the plaintext, $p(k)$ denotes the ciphertext, and $x_i(k)(i = 1, 2, 3, \cdots)$ denotes the chaotic variable. According to Table 2, the n-D SCSC with the ciphertext $p(k) = \bmod\left(\lfloor x_i(k) \rfloor, 2^8\right) \oplus m(k)$ is called n-D SCSC-1; the n-D SCSC with the ciphertext $p(k) = \bmod \left(\left\lfloor x_i(k)x_j(k)/2^{N_1}\right\rfloor, 2^8\right) \oplus m(k)$ is called n-D SCSC-2; the n-D SCSC with the ciphertext $p(k) = \bmod \left(\left\lfloor x_i(k)x_j(k)x_l(k)/2^{N_2}\right\rfloor, 2^8\right) \oplus m(k)$ is called n-D SCSC-3, and so on, where $n = 3, 4, 5, \cdots$, $i, j, l = 1, 2, 3, \cdots$, $i \neq j \neq l$, N_1 and N_2 are the positive integers.

For n-D SCSC (n=3,7,8) listed in Table 2, any given initial conditions at the receiver can achieve asymptotic synchronization, so the cryptanalyst can succeed security analysis by arbitrarily selecting different initial conditions. When the plaintext is encrypted by the lower 8 bits derived from a signal state variable with a round-down operation and a modulo operation, the divide-and-conquer attack by traversing single non-zero component initial conditions (DCA-TSNCIC) can be used for the security analysis of n-D SCSC. For example, Lin et al. [Lin *et al.*, 2015] proposed an 8-D SCSC-1 to encrypt the RGB three primary colors, but Lin et al. [Lin *et al.*, 2018] proposed a method that combines known-plaintext attack, chosen-ciphertext attack, and DCA-TSNCIC to decipher the original keys. Notably, 8-D SCSC-1 was deciphered in [Lin *et al.*, 2018] using DCA-TSNCIC with eight single non-zero component initial conditions, where $c_i (i = 1, 2, \cdots, 8)$ was set as $2^{7+8i} (i = 0, 1, \cdots, 7)$, respectively. The cryptanalysis method used in [Lin *et al.*, 2018] has the following two main features:

Table 2. The ciphertext forms of n-D SCSC

Encryption algorithm	Ciphertext form
8-D SCSC-1 [Lin *et al.*, 2015]	Use $p(k) = \bmod \left(\lfloor x_i(k)\rfloor, 2^8\right) \oplus m(k)(i = 1, 2, 3)$ to encrypt the RGB three primary colors.
3-D SCSC-2 [Chen *et al.*, 2018]	Use $p(k) = \bmod \left(\left\lfloor x_1(k)x_2(k)/2^{27}\right\rfloor, 2^8\right) \oplus m(k)$ to encrypt the RGB three primary colors.
7-D SCSC-3 [Chen *et al.*, 2020]	Use $p(k) = \bmod \left(\left\lfloor x_i(k)x_j(k)x_l(k)/2^{24}\right\rfloor, 2^8\right) \oplus m(k)$ $(i = 1, 2, 3; j = 3, 4, 5; l = 5, 6, 7)$ to encrypt the RGB three primary colors.

(1) The encryption algorithm proposed in [Lin *et al.*, 2015] is relatively simple in that only 8-D SCSC-1 with $p(k) = \bmod\left(\lfloor x_i(k)\rfloor, 2^8\right) \oplus m(k)$ $(i = 1, 2, 3)$ was used. Eight initial conditions $\{(c_1, 0, \cdots, 0), \cdots, (0, 0, \cdots, c_8)\}$ were substituted into the iterative equation for the first iteration and for the first divide-and-conquer attack, respectively. Then, the secret keys $a_{ij} (1 \leq i \leq 3, 1 \leq j \leq 8)$ were deciphered directly, providing essential prerequisites to decipher the subsequent secret keys $a_{ij} (4 \leq i \leq 8, 4 \leq j \leq 8)$.

(2) The deciphered secret keys $a_{ij} (1 \leq i \leq 3, 1 \leq j \leq 8)$ were taken as the known conditions in a global substitution method, and the nonlinear equations obtained in each subsequent iteration were further simplified to be linear equations. Then, the unknown secret keys $a_{ij} (4 \leq i \leq 8, 4 \leq j \leq 8)$ were deciphered when the rank of the linear equations is equal to the number of $a_{ij} (4 \leq i \leq 8, 4 \leq j \leq 8)$.

However, the DCA-TSNCIC used in [Lin *et al.*, 2018] can only decipher the less-complicated n-D SCSC-1. For the more-sophisticated n-D SCSC-2 and n-D SCSC-3 $(n = 3, 4, 5, 6, 7, 8)$, only 3-D SCSC-2 and 3-D SCSC-3 can be deciphered after five iterations, and the others cannot be deciphered. To address this issue, a cryptanalysis method with a stronger attack intensity, the DCA-TMNCIC is presented in this paper, where all possible choices for multiple non-zero component initial conditions are traversed through the exhaustive method.

The main contributions of this paper are summarized as follows:

(1) A general method using DCA-TMNCIC is developed for security analyses of n-D SCSC-2 and n-D SCSC-3 (n=3,4,5,6,7,8).

(2) 3-D SCSC-2 proposed in [Chen *et al.*, 2018] is taken as a typical example for cryptanalysis by combining chosen-ciphertext attack and DCA -TMNCIC.

(3) Several new chaotic cipher schemes are designed, including 3-D SCSC-NNS and n-D SCSC-SM (n=3,4,5,6,7,8). Security analysis is performed, demonstrating that the improved schemes are secure against combined effect of chosen-ciphertext attack and divide-and-conquer attack.

The rest of the paper is organized as follows: Section 2 introduces the description and security analysis for n-D SCSC (n=3,4,5,6,7,8). Section 3 performs security analysis of 3-D SCSC-2 by combining chosen-ciphertext attack and DCA-TMNCIC. Section 4 gives the comparisons and discussions of DCA-TSNCIC and DCA-TMNCIC. Section 5 and Section 6 present several new improved chaotic cipher schemes, including 3-D SCSC-NNS and n-D SCSC-SM (n=3,4,5,6,7,8), along with their security analyses, respectively. Section 7 concludes the investigation.

2. Description and security analysis for n-D SCSC

2.1. *Description of n-D SCSC*

According to [Lin *et al.*, 2015; Chen *et al.*, 2018, 2020], n-D SCSC ($n = 3, 4, 5, 6, 7, 8$) is a class of self-synchronous chaotic stream ciphers based on chaos anti-control principles, the main features of n-D SCSC ($n = 3, 4, 5, 6, 7, 8$) are as follows:

(1) The ciphertexts containing the plaintext information are fed back into the underlying chaotic system to realize self-synchronization.

(2) With a round down operation and a modulo operation, only the lower 8 bits of a single chaotic variable or a product of multiple chaotic variables used for encryption-decryption, resulting in decreasing leakage of chaotic information.

(3) With the different dimension, n-D SCSC ($n = 3, 4, 5, 6, 7, 8$) can be divided into single-channel encryption scheme and multi-channel encryption scheme. With the difference of ciphertext forms, n-D SCSC ($n = 3, 4, 5, 6, 7, 8$)can be classified as n-D SCSC-1, n-D SCSC-2 and n-D SCSC-3, the typical examples are listed in Table 2.

The general form of n-D SCSC ($n = 3, 4, 5, 6, 7, 8$) can be derived, as

$$x(k+1) = f(a_{ij},\ x(k),\ p(k)) + g(\sigma_l p(k),\ \varepsilon_l), \tag{1}$$

where $k = 0, 1, 2, 3, \cdots$, $p(k)$ denotes the ciphertext, $f(a_{ij},\ x(k),\ p(k))$ denotes a nominal system with ciphertext feedback, $g(\sigma_l p(k),\ \varepsilon_l)$ denotes the uniformly bounded controller with ciphertext feedback. $a_{ij}, \sigma_l, \varepsilon_l$ $(i, j, l = 1, 2, \cdots, 8)$ denote secret keys, $x(k+1) = (x_1(k+1), x_2(k+1), \cdots, x_n(k+1))^T$, $x(k) = (x_1(k), x_2(k), \cdots, x_n(k))^T$ $(n = 3, 4, 5, 6, 7, 8)$ denote chaotic variables.

In n-D SCSC-1, $p(k)$ is derived as

$$p(k) = \bmod\left(\lfloor x_i(k) \rfloor, 2^8\right) \oplus m(k) \rightarrow m(k) \oplus p(k) = \bmod\left(\lfloor x_i(k) \rfloor, 2^8\right). \tag{2}$$

In n-D SCSC-2, $p(k)$ is derived as

$$p(k) = \bmod\left(\left\lfloor x_i(k)x_j(k)/2^{N_1} \right\rfloor, 2^8\right) \oplus m(k) \rightarrow m(k) \oplus p(k) = \bmod\left(\left\lfloor x_i(k)x_j(k)/2^{N_1} \right\rfloor, 2^8\right). \tag{3}$$

In n-D SCSC-3, $p(k)$ is derived as

$$p(k) = \bmod\left(\left\lfloor x_i(k)x_j(k)x_l(k)/2^{N_2} \right\rfloor, 2^8\right) \oplus m(k) \rightarrow m(k) \oplus p(k) = \bmod\left(\left\lfloor x_i(k)x_j(k)x_l(k)/2^{N_2} \right\rfloor, 2^8\right), \tag{4}$$

where $i, j, l = 1, 2, \cdots, 8$, $i \neq j \neq l$, N_1 and N_2 are the positive integers.

2.2. *Loopholes of n-D SCSC*

According to the description in Section 2.1, the main problems existing in n-D SCSC ($n = 3, 4, 5, 6, 7, 8$) are as follows:

(1) In actual channel communications, such as LAN and WAN, any given initial conditions at the receiver can achieve asymptotic synchronization. It can be seen that the initial conditions of the chaotic system are weak secret keys. Therefore, in the process of security analysis, the cryptanalyst can select any initial conditions favorable for cryptanalysis.

(2) In n-D SCSC ($n = 3, 4, 5, 6, 7, 8$), ciphertexts are fed back into the underlying chaotic system, and the chaotic sequences generated by the chaotic system are related to the ciphertexts. However, with the

chosen-ciphertext attack, the chaotic iterative equation at the receiver will degenerate into a linear iterative equation, simplifying the calculation complexity of the chaotic iterative equation and facilitating the security analysis. For example, in Eq. (1), by setting $p(k) = 0$, Eq. (1) will degenerate into $x(k+1) = f(a_{ij},\ x(k))$. Compared with Eq. (1), the complexity of iterative operation of $x(k+1) = f(a_{ij},\ x(k))$ is greatly reduced.

(3) According to Eq. (2)-(4), the plaintext is encrypted by the lower 8 bits derived from chaotic variables with a round-down operation and a modulo operation. Thus, the cryptanalyst can only obtain the lower 8 bits of chaotic variables, but not all the information. However, DCA-TSNCIC proposed in [Lin *et al.*, 2018] can be used for the security analysis of n-D SCSC-1 ($n = 3, 4, 5, 6, 7, 8$). DCA-TMNCIC proposed in this paper can be used for the security analyses of n-D SCSC-2 and n-D SCSC-3 ($n = 3, 4, 5, 6, 7, 8$).

To sum up, for n-D SCSC ($n = 3, 4, 5, 6, 7, 8$), with the chosen-ciphertext attack, by setting $p(k) = 0$, Eq. (1) will degenerate into a linear iterative equation, one has

$$x(k+1) = f(a_{ij},\ x(k)). \tag{5}$$

Then Eq. (2)-(4) can also be simplified as

$$m(k) = \mathrm{mod}\left(\left\lfloor F^{(k)}(c, a_{ij})\right\rfloor, 2^8\right), \tag{6}$$

where $k = 0, 1, 2, 3, \cdots$, $F^{(k)}(c, a_{ij})$ denotes the relational expression between initial conditions and secret keys, a_{ij} denotes secret keys, c denotes the value of initial condition.

According to Eq. (6), the cryptanalyst can set the appropriate value of c and use the divide-and-conquer attack to obtain the relationship between secret keys and plaintexts. Then, the linear or nonlinear equations can be obtained for solving the secret keys.

2.3. *General methods for security analyses of n-D SCSC-2 and n-D SCSC-3*

A general method that combines a theoretical analysis with Matlab R2017a and Maple 2018 software is proposed for the security analyses of n-D SCSC-2, n-D SCSC-3, and other n-D SCSC ($n = 3, 4, 5, 6, 7, 8$). Note that n-D SCSC ($n = 3, 4, 5, 6, 7, 8$) is one of the ciphers of CFM, and the ciphertexts are fed back into the underlying chaotic system. Hence, the cryptanalyst needs to decipher the original keys but not the equivalent keys. The cryptanalysis method for obtaining the original keys is different from that for obtaining the equivalent keys. The general method proposed in this paper is the cryptanalysis method for n-D SCSC ($n = 3, 4, 5, 6, 7, 8$), which is not suitable for other CFM, NFM and PAM. The flowchart of the general method for security analysis is shown in Fig. 1.

Specifically, in Fig. 1, valid initial conditions are the initial conditions such that the secret keys appear in $p(k) = \mathrm{mod}\left(\left\lfloor x_i(k)x_j(k)/2^{N_1}\right\rfloor, 2^8\right) \oplus m(k)$ or $p(k) = \mathrm{mod}\left(\left\lfloor x_i(k)x_j(k)x_k(k)/2^{N_2}\right\rfloor, 2^8\right) \oplus m(k)$. In contrast, invalid initial conditions are the initial conditions such that the secret keys cannot appear therein.

Fig. 1 shows the flowchart using Matlab software to perform the divide-and-conquer attack and using Maple software to solve the nonlinear equations. When the $solve(equations, variables)$ function in the Maple software is used to solve the nonlinear equations, Maple software can find a numerical solution in a limited time, so the secret keys can be successfully deciphered. If the cryptanalysis equation is iterated for multiple times but the software is failed to converge, then one can confirm whether the secret keys have been deciphered.

When solving nonlinear equations by Maple 2018 software, the higher dimension of n-D SCSC ($n =$ 3,4,5,6,7,8), the longer time is needed to solve nonlinear equations. For n-D SCSC-2 and n-D SCSC-3 ($n =$ 3,4), the numerical solution can be solved in a few seconds. For n-D SCSC-2 ($n =$ 5,6,7,8), the numerical solution can be solved in a few minutes. For n-D SCSC-3 ($n =$ 5,6,7,8), it takes several hours.

As shown by Fig.1, the general method for security analyses of n-D SCSC-2 and n-D SCSC-3 (n=3,4,5,6,7,8) is as follows:

(1) Obtain all valid initial conditions, and substitute them into the iterative equation for performing the first iteration and divide-and-conquer attack. Then, the first nonlinear equation system about the secret keys is obtained and solved. If there is a solution, the secret keys are deciphered; otherwise, the iteration will continue to the second one.

(2) Substitute the first iteration result into the iterative equation for performing the second iteration

and divide-and-conquer attack. Then, the second nonlinear equation system about the secret keys is obtained. From the previous two iterations and divide-and-conquer attacks, two nonlinear equation systems are obtained for solving the secret keys. If there is a solution, the secret keys are deciphered; otherwise, the next iteration will continue.

(3) Obtain several (three or more) nonlinear equation systems about the secret keys by multiple (three or more) iterations and divide-and-conquer attacks. From the previous multiple iterations and divide-and-conquer attacks, several nonlinear equation systems are obtained for solving the secret keys. If there is a solution, the secret keys are deciphered; otherwise, the secret keys are considered unable to be deciphered.

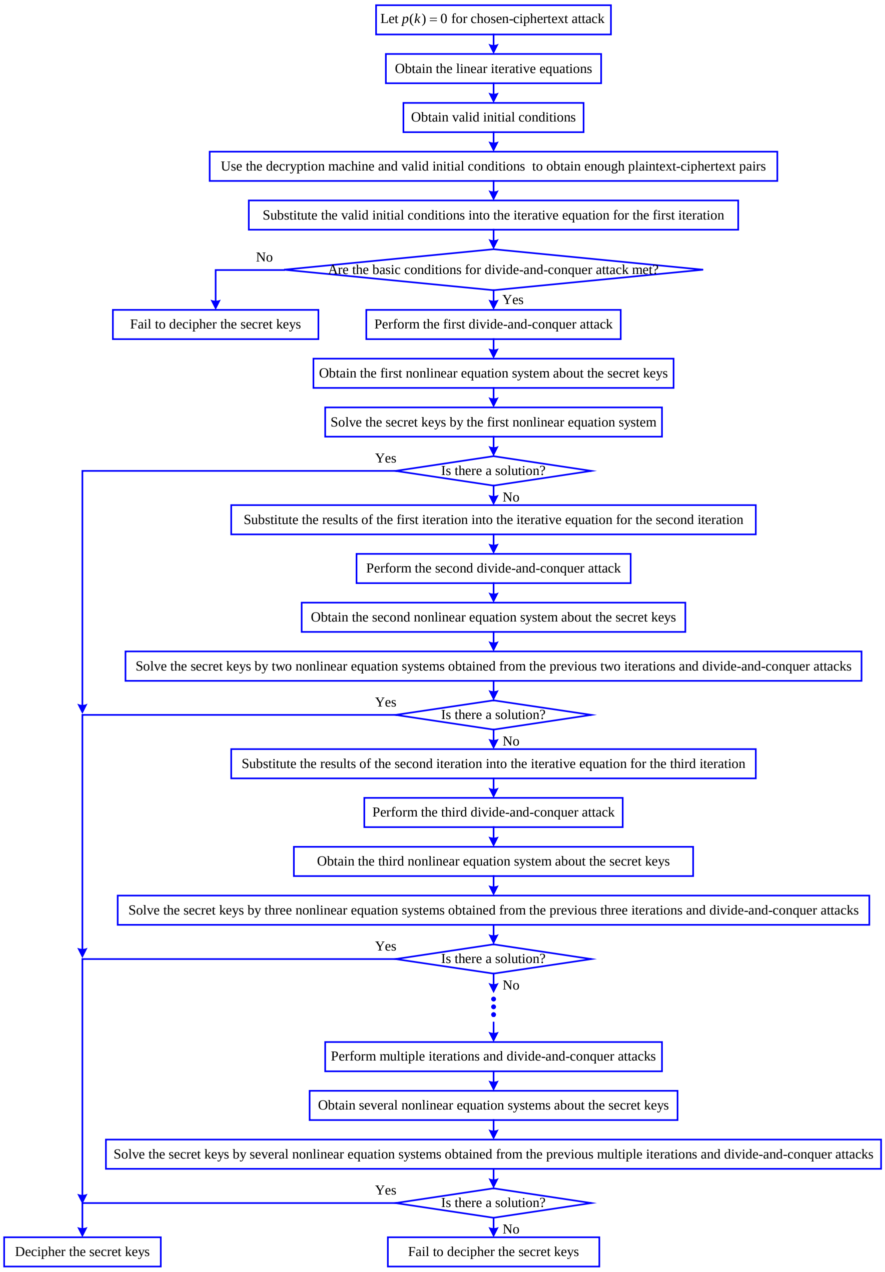

Let $p(k) = 0$ for chosen-ciphertext attack
Obtain the linear iterative equations
Obtain valid initial conditions
Use the decryption machine and valid initial conditions to obtain enough plaintext-ciphertext pairs
Substitute the valid initial conditions into the iterative equation for the first iteration
No
Are the basic conditions for divide-and-conquer attack met?
Yes
Fail to decipher the secret keys
Perform the first divide-and-conquer attack
Obtain the first nonlinear equation system about the secret keys
Solve the secret keys by the first nonlinear equation system
Yes
Is there a solution?
No
Substitute the results of the first iteration into the iterative equation for the second iteration
Perform the second divide-and-conquer attack
Obtain the second nonlinear equation system about the secret keys
Solve the secret keys by two nonlinear equation systems obtained from the previous two iterations and divide-and-conquer attacks
Yes
Is there a solution?
No
Substitute the results of the second iteration into the iterative equation for the third iteration
Perform the third divide-and-conquer attack
Obtain the third nonlinear equation system about the secret keys
Solve the secret keys by three nonlinear equation systems obtained from the previous three iterations and divide-and-conquer attacks
Yes
Is there a solution?
No
Perform multiple iterations and divide-and-conquer attacks
Obtain several nonlinear equation systems about the secret keys
Solve the secret keys by several nonlinear equation systems obtained from the previous multiple iterations and divide-and-conquer attacks
Yes
Is there a solution?
No
Decipher the secret keys
Fail to decipher the secret keys

Fig. 1. Flowchart of the general method for security analyses of n-D SCSC-2 and n-D SCSC-3

3. 3-D SCSC-2 and its security analysis

3.1. *Description of 3-D SCSC-2*

According to [Chen *et al.*, 2018], the iterative equation of 3-D SCSC-2 is given by

$$\begin{cases} x_1(k+1) = a_{11}x_1(k) + a_{12}x_2(k) + a_{13}x_3(k) \\ x_2(k+1) = a_{21}p(k) + a_{22}x_2(k) + a_{23}x_3(k) \\ x_3(k+1) = a_{31}p(k) + a_{32}x_2(k) + a_{33}x_3(k) + \varepsilon\sin(\sigma p(k)) \end{cases} . \tag{7}$$

The ciphertext $p(k)$ in Eq. (7) is given by

$$p(k) = \operatorname{mod}\left(\left\lfloor \frac{x_1(k)x_2(k)}{2^{27}} \right\rfloor, 2^8\right) \oplus m(k) \rightarrow m(k) \oplus p(k) = \operatorname{mod}\left(\left\lfloor \frac{x_1(k)x_2(k)}{2^{27}} \right\rfloor, 2^8\right), \tag{8}$$

where $k = 0, 1, 2, 3, 4\cdots$, $m(k)$ denotes the corresponding plaintext, $x_i(k)(i = 1, 2, 3)$ denotes the chaotic variable, and $a_{11} = 0.09$, $a_{12} = -0.37$, $a_{13} = 0.1$, $a_{21} = -0.1$, $a_{22} = -0.18$, $a_{23} = 0.37$, $a_{31} = 0.27$, $a_{32} = -0.27$, $a_{33} = 0.19$, $\varepsilon = 3.3 \times 10^8$, $\sigma = 2.5 \times 10^5$ denote the secret keys, 2^{27} denotes a constant.

3.2. *Security analysis flowchart of 3-D SCSC-2*

In 3-D SCSC-2, the lower 8 bits derived from the product of two chaotic variables are used to encrypt the plaintext. When the method of combining the chosen-ciphertext attack and DCA-TMNCIC is adopted, all the other secret keys can be deciphered except the secret keys multiplied by ciphertexts and depending on nonlinear functions. The security analysis flowchart of 3-D SCSC-2 is shown in Fig. 2.

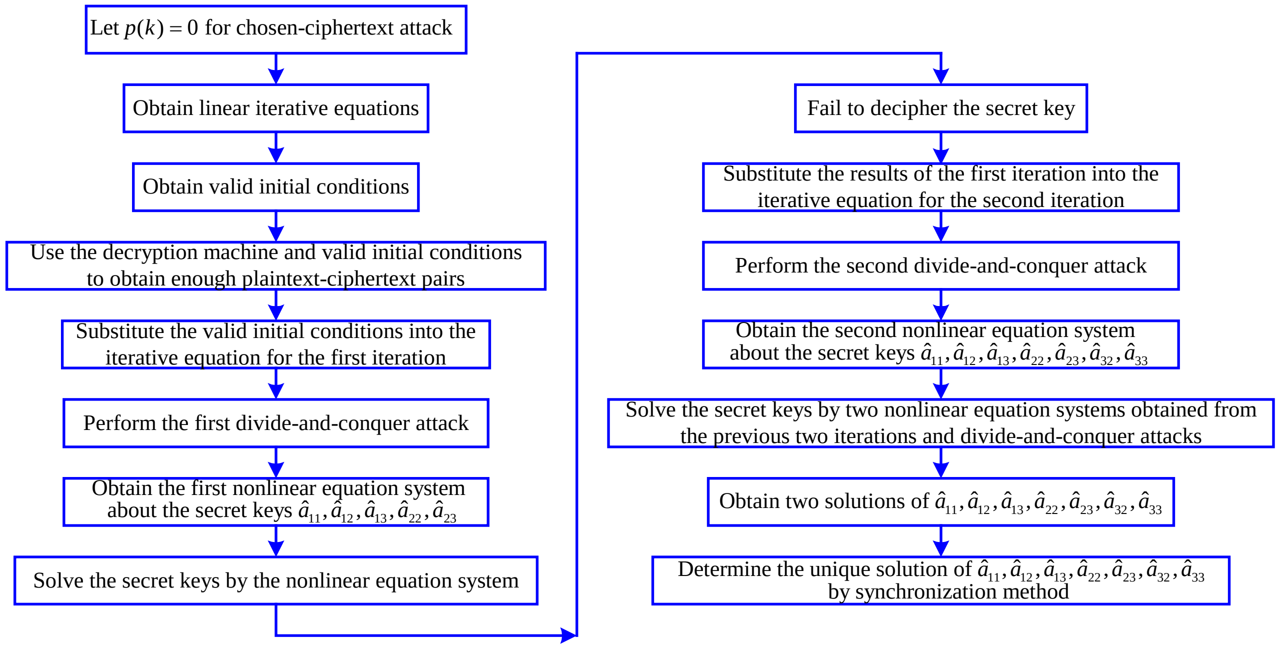

Let $p(k) = 0$ for chosen-ciphertext attack
Obtain linear iterative equations
Obtain valid initial conditions
Use the decryption machine and valid initial conditions to obtain enough plaintext-ciphertext pairs
Substitute the valid initial conditions into the iterative equation for the first iteration
Perform the first divide-and-conquer attack
Obtain the first nonlinear equation system about the secret keys $\hat{a}_{11}, \hat{a}_{12}, \hat{a}_{13}, \hat{a}_{22}, \hat{a}_{23}$
Solve the secret keys by the nonlinear equation system
Fail to decipher the secret key
Substitute the results of the first iteration into the iterative equation for the second iteration
Perform the second divide-and-conquer attack
Obtain the second nonlinear equation system about the secret keys $\hat{a}_{11}, \hat{a}_{12}, \hat{a}_{13}, \hat{a}_{22}, \hat{a}_{23}, \hat{a}_{32}, \hat{a}_{33}$
Solve the secret keys by two nonlinear equation systems obtained from the previous two iterations and divide-and-conquer attacks
Obtain two solutions of $\hat{a}_{11}, \hat{a}_{12}, \hat{a}_{13}, \hat{a}_{22}, \hat{a}_{23}, \hat{a}_{32}, \hat{a}_{33}$
Determine the unique solution of $\hat{a}_{11}, \hat{a}_{12}, \hat{a}_{13}, \hat{a}_{22}, \hat{a}_{23}, \hat{a}_{32}, \hat{a}_{33}$ by synchronization method

Fig. 2. Security analysis flowchart of 3-D SCSC-2

3.3. *Chosen-ciphertext attack*

According to Fig. 2, with the chosen-ciphertext attack, the cryptanalyst can select ciphertexts favorable for deciphering the secret keys of 3-D SCSC-2 so as to obtain the corresponding plaintexts. According to Eq.

(7), by setting $p(k) = 0$, a linear iterative equation is derived as

$$\begin{cases} x_1(k+1) = a_{11}x_1(k) + a_{12}x_2(k) + a_{13}x_3(k) \\ x_2(k+1) = a_{22}x_2(k) + a_{23}x_3(k) \\ x_3(k+1) = a_{32}x_2(k) + a_{33}x_3(k) \end{cases}, \tag{9}$$

where $k = 0, 1, 2, 3, 4, \cdots$. Then, $p(k) = 0$ is substituted into Eq. (8), which yields

$$m(k) = \mathrm{mod}\left(\left\lfloor \frac{x_1(k)x_2(k)}{2^{27}} \right\rfloor, 2^8\right). \tag{10}$$

Since a multiplication, a round-down operation, and a modulo operation are involved in Eq. (10), the cryptanalyst can only reveal the information of the lower 8 bits of $x_1(k)x_2(k)$, but cannot obtain all the information of $x_1(k)x_2(k)$. However, combined with the chosen-ciphertext attack, DCA-TMNCIC can be used to decipher the secret keys.

Under DCA-TMNCIC, there are seven options to select initial conditions. The set of all options is derived as

$$(x_1(0), x_2(0), x_3(0)) \in \left\{ \begin{matrix} (c_1, 0, 0), (0, c_2, 0), (0, 0, c_3), \\ (c_1, c_2, 0), (c_1, 0, c_3), (0, c_2, c_3), (c_1, c_2, c_3) \end{matrix} \right\}, \tag{11}$$

where $x_i(0) (i = 1, 2, 3)$ denotes the initinal condition, $c_i \neq 0$ $(i = 1, 2, 3)$. Then, according Eq. (9)-(10), one gets

$$\begin{aligned} m(1) &= \mathrm{mod}\left(\left\lfloor \frac{x_1(1)x_2(1)}{2^{27}} \right\rfloor, 2^8\right) \\ &= \mathrm{mod}\left(\left\lfloor \frac{(a_{11}x_1(0) + a_{12}x_2(0) + a_{13}x_3(0)) \times (a_{22}x_2(0) + a_{23}x_3(0))}{2^{27}} \right\rfloor, 2^8\right). \end{aligned} \tag{12}$$

Seven initial conditions in Eq. (11) are substituted into Eq. (12) one by one. If $m(1) \neq 0$ is satisfied, it is called a valid initial condition; if $m(1) = 0$ is satisfied, it is called an invalid initial condition. Note that the attack intensity grows with the number of valid initial conditions.

3.4. *DCA-TMNCIC*

For 3-D SCSC-2 in Eqs. (7)-(8), any given initial conditions at the receiver can achieve asymptotic synchronization, so the cryptanalyst can select any initial conditions favorable for deciphering 3-D SCSC-2. According to Eq. (11), by setting appropriate values of c_i $(i = 1, 2, 3)$ with $p(k) = 0$, a prerequisite is provided to decipher 3-D SCSC-2 by using DCA-TMNCIC. The procedure of DCA-TMNCIC is discussed below.

3.4.1. *Divide 64-bit binary numbers*

The divide-and-conquer attack is a cryptanalysis method that divides the secret key into several independent sub-blocks and then solves them block by block, reducing computational loads, to achieve cryptographic deciphering. Thus, in 3-D SCSC-2, firstly the secret keys a_{ij} $(i = 1, 2, 3; j = 1, 2, 3)$ are represented by the 64-bit binary numbers $(a_{ij})_2$ $(i = 1, 2, 3; j = 1, 2, 3)$, where the 1st bit denotes the sign, and the remaining 63 bits denote the data. Then, $(a_{ij})_2$ $(i = 1, 2, 3; j = 1, 2, 3)$ are divided into 8 sub-blocks, respectively. Each sub-block is of 8 bits, denoted by $(a_{ij}^{(k)})_2$ $(i = 1, 2, 3; j = 1, 2, 3; k = 1, 2, \cdots, 8)$. Finally, the relationship between $(a_{ij})_2$ $(i = 1, 2, 3; j = 1, 2, 3)$ and the secret key sub-block $(a_{ij}^{(k)})_2$ $(i = 1, 2, 3; j = 1, 2, 3; k = 1, 2, \cdots, 8)$ is derived, as

$$(a_{ij})_2 = (a_{ij}^{(1)})_2(a_{ij}^{(2)})_2(a_{ij}^{(3)})_2(a_{ij}^{(4)})_2(a_{ij}^{(5)})_2(a_{ij}^{(6)})_2(a_{ij}^{(7)})_2(a_{ij}^{(8)})_2 \ (i = 1, 2, 3; j = 1, 2, 3). \tag{13}$$

Similarly, the estimated value sub-block $(\hat{a}_{ij}^{(k)})_2$ $(i = 1,2,3; j = 1,2,3; k = 1,2,\cdots,8)$ of $(a_{ij}{}^{(k)})_2$ $(i = 1,2,3; j = 1,2,3; k = 1,2,\cdots,8)$ can be obtained, and the estimated value $(\hat{a}_{ij})_2$ of $(a_{ij})_2$ $(i = 1,2,3; j = 1,2,3)$ can be derived, as

$$(\hat{a}_{ij})_2 = (\hat{a}_{ij}^{(1)})_2(\hat{a}_{ij}^{(2)})_2(\hat{a}_{ij}^{(3)})_2(\hat{a}_{ij}^{(4)})_2(\hat{a}_{ij}^{(5)})_2(\hat{a}_{ij}^{(6)})_2(\hat{a}_{ij}^{(7)})_2(\hat{a}_{ij}^{(8)})_2 \ (i = 1,2,3; j = 1,2,3). \tag{14}$$

For more complicated secret key expressions, they can also be represented by 64-bit binary numbers. For example, when the secret key expression is $a_{ij}a_{ij}$ $(i = 1,2,3; j = 1,2,3)$, according to the above-described method, the relationship between $(a_{ij}a_{ij})_2$ $(i = 1,2,3; j = 1,2,3)$ and the secret key expression sub-block $((a_{ij}a_{ij})^{(k)})_2$ $(i = 1,2,3; j = 1,2,3; k = 1,2,\cdots,8)$ can be derived, as

$$((a_{ij}a_{ij}))_2 = ((a_{ij}a_{ij})^{(1)})_2((a_{ij}a_{ij})^{(2)})_2\cdots((a_{ij}a_{ij})^{(7)})_2((a_{ij}a_{ij})^{(8)})_2 \ (i = 1,2,3; j = 1,2,3). \tag{15}$$

Similarly, the estimated values $((\hat{a}_{ij}\hat{a}_{ij}))_2$ $(i = 1,2,3; j = 1,2,3)$ of $(a_{ij}a_{ij})_2$ $(i = 1,2,3; j = 1,2,3)$ can be derived, as

$$((\hat{a}_{ij}\hat{a}_{ij}))_2 = ((\hat{a}_{ij}\hat{a}_{ij})^{(1)})_2((\hat{a}_{ij}\hat{a}_{ij})^{(2)})_2\cdots((\hat{a}_{ij}\hat{a}_{ij})^{(7)})_2((\hat{a}_{ij}\hat{a}_{ij})^{(8)})_2 \ (i = 1,2,3; j = 1,2,3). \tag{16}$$

3.4.2. *Divide 64-bit binary numbers $\hat{a}_{11}, \hat{a}_{12}, \hat{a}_{13}, \hat{a}_{22}, \hat{a}_{23}$ through the first iteration*

Firstly, by substituting $k = 0$ into Eq. (9), the first iteration result is derived as

$$\begin{cases} x_1(1) = a_{11}x_1(0) + a_{12}x_2(0) + a_{13}x_3(0) \\ x_2(1) = a_{22}x_2(0) + a_{23}x_3(0) \\ x_3(1) = a_{32}x_2(0) + a_{33}x_3(0) \end{cases}. \tag{17}$$

Then, by substituting $x_1(0) = c_1, x_2(0) = c_2, x_3(0) = c_3$ into Eq. (17), one has

$$\begin{cases} x_1(1) = a_{11}c_1 + a_{12}c_2 + a_{13}c_3 \\ x_2(1) = a_{22}c_2 + a_{23}c_3 \\ x_3(1) = a_{32}c_3 + a_{33}c_3 \end{cases}. \tag{18}$$

According to the seven initial conditions given in Eq. (11), with the chosen-ciphertext attack, the corresponding plaintext $m_i(1)$ $(i = 1,2,\cdots,7)$ can be obtained by substituting $p_i(1) = 0$ $(i = 1,2,\cdots,7)$ into Eq. (10). Note that both $p_i(1) = 0$ $(i = 1,2,\cdots,7)$ and $m_i(1)$ $(i = 1,2,\cdots,7)$ are known, $m_i(1)$ $(i = 1,2,\cdots,7)$ are changed with different c_i $(i = 1,2,3)$, although $p_i(1) = 0$ $(i = 1,2,\cdots,7)$ remains the same. According to Eq. (18), consider the first iteration as follows:

(1) By substituting the first initial condition $(c_1, 0, 0)$ into Eq. (18), one obtains $x_1(1) = a_{11}c_1 \neq 0$, $x_2(1) = x_3(1) = 0$, satisfying $x_1(1)x_2(1) = 0$. According to Eq. (10), one can see that $(c_1, 0, 0)$ is an invalid initial condition.

(2) By substituting the second initial condition $(0, c_2, 0)$ into Eq. (18), one obtains $x_1(1) = a_{12}c_2 \neq 0$, $x_2(1) = a_{22}c_2 \neq 0$, $x_3(1) = a_{32}c_2 \neq 0$, satisfying $x_1(1)x_2(1) \neq 0$. According to Eq. (10), one can see that $(0, c_2, 0)$ is a valid initial condition.

(3) By substituting the third initial condition $(0, 0, c_3)$ into Eq. (18), one obtains $x_1(1) = a_{13}c_3 \neq 0$, $x_2(1) = a_{23}c_3 \neq 0$, $x_3(1) = a_{33}c_3 \neq 0$, satisfying $x_1(1)x_2(1) \neq 0$. According to Eq. (10), one can see that $(0, 0, c_3)$ is a valid initial condition.

(4) By substituting the fourth initial condition $(c_1, c_2, 0)$ into Eq. (18), one obtains $x_1(1) = a_{11}c_1 + a_{12}c_2 \neq 0$, $x_2(1) = a_{22}c_2 \neq 0$, $x_3(1) = a_{32}c_2 \neq 0$, satisfying $x_1(1)x_2(1) \neq 0$. According to Eq. (10), one can see that $(c_1, c_2, 0)$ is a valid initial condition.

(5) By substituting the fifth initial condition $(c_1, 0, c_3)$ into Eq. (18), one obtains $x_1(1) = a_{11}c_1 + a_{13}c_3 \neq 0$, $x_2(1) = a_{23}c_3 \neq 0$, $x_3(1) = a_{33}c_3 \neq 0$, satisfying $x_1(1)x_2(1) \neq 0$. According to Eq. (10), one can see that $(c_1, 0, c_3)$ is a valid initial condition.

(6) By substituting the sixth initial condition $(0, c_2, c_3)$ into Eq. (18), one obtains $x_1(1) = a_{12}c_2 + a_{13}c_3 \neq 0$, $x_2(1) = a_{22}c_2 + a_{23}c_3 \neq 0$, $x_3(1) = a_{32}c_2 + a_{33}c_3 \neq 0$, satisfying $x_1(1)x_2(1) \neq 0$. According to Eq. (10), one can see that $(0, c_2, c_3)$ is a valid initial condition.

(7) By substituting the seventh initial condition (c_1, c_2, c_3) into Eq. (18), one obtains $x_1(1) = a_{11}c_1 + a_{12}c_2 + a_{13}c_3 \neq 0$, $x_2(1) = a_{22}c_2 + a_{23}c_3 \neq 0$, $x_3(1) = a_{32}c_2 + a_{33}c_3 \neq 0$, satisfying $x_1(1)x_2(1) \neq 0$. According to Eq. (10), one can see that (c_1, c_2, c_3) is a valid initial condition.

In summary, since only the valid initial conditions are considered, by substituting the calculation results of $x_1(1)x_2(1)$ into Eq. (10), and setting $c_i = c$ $(i = 1, 2, 3)$, one obtains $m_i(1)$ $(i = 2, 3, \cdots, 7)$ as follows:

$$\begin{cases} m_2(1) = \left(\mathrm{mod}\left(\left\lfloor \frac{a_{12}a_{22}c_2^2}{2^{27}} \right\rfloor, 2^8\right)\right) = \left(\mathrm{mod}\left(\left\lfloor \frac{a_{12}a_{22}c^2}{2^{27}} \right\rfloor, 2^8\right)\right) \\ m_3(1) = \left(\mathrm{mod}\left(\left\lfloor \frac{a_{13}a_{23}c_3^2}{2^{27}} \right\rfloor, 2^8\right)\right) = \left(\mathrm{mod}\left(\left\lfloor \frac{a_{13}a_{23}c^2}{2^{27}} \right\rfloor, 2^8\right)\right) \\ m_4(1) = \left(\mathrm{mod}\left(\left\lfloor \frac{a_{22}c_2(a_{11}c_1 \ + \ a_{12}c_2)}{2^{27}} \right\rfloor, 2^8\right)\right) = \left(\mathrm{mod}\left(\left\lfloor \frac{(a_{11}a_{22} + a_{12}a_{22})c^2}{2^{27}} \right\rfloor, 2^8\right)\right) \\ m_5(1) = \left(\mathrm{mod}\left(\left\lfloor \frac{a_{23}c_3(a_{11}c_1 \ + \ a_{13}c_3)}{2^{27}} \right\rfloor, 2^8\right)\right) = \left(\mathrm{mod}\left(\left\lfloor \frac{(a_{11}a_{23} + a_{13}a_{23})c^2}{2^{27}} \right\rfloor, 2^8\right)\right) \\ m_6(1) = \left(\mathrm{mod}\left(\left\lfloor \frac{(a_{12}c_2 \ + \ a_{13}c_3) \times (a_{22}c_2 \ + \ a_{23}c_3)}{2^{27}} \right\rfloor, 2^8\right)\right) \\ \qquad\ \ = \left(\mathrm{mod}\left(\left\lfloor \frac{(a_{12} + a_{13}) \times (a_{22} + a_{23})c^2}{2^{27}} \right\rfloor, 2^8\right)\right) \\ m_7(1) = \left(\mathrm{mod}\left(\left\lfloor \frac{(a_{22}c_2 \ + \ a_{23}c_3) \times (a_{11}c_1 \ + \ a_{12}c_2 \ + \ a_{13}c_3)}{2^{27}} \right\rfloor, 2^8\right)\right) \\ \qquad\ \ = \left(\mathrm{mod}\left(\left\lfloor \frac{(a_{22} + a_{23}) \times (a_{11} + a_{12} + a_{13})c^2}{2^{27}} \right\rfloor, 2^8\right)\right) \end{cases} . \tag{19}$$

Refer $F(\cdot)$ in $\lfloor F(\cdot) \rfloor$ to as a secret key expression. Note that in Eq. (19), when $c_1 \neq c_2 \neq c_3$, $m_i(1)$ $(i = 2, 3, \cdots, 7)$ cannot be further simplified, so the estimated values of secret key expressions cannot be obtained by the divide-and-conquer attack. However, when $c_1 = c_2 = c_3 = c$, the common factor c^2 in $\mathrm{mod}(\cdot)$ can be extracted, so an appropriate value of c can be set to obtain the estimated values of secret key expressions by the divide-and-conquer attack.

The following is an example of $m_4(1)$ in Eq. (19) to illustrate this situation. Similarly to Eqs. (15)-(16), the relationship between $(a_{11}a_{22} + a_{12}a_{22})_2$ and the sub-block $((a_{11}a_{22} + a_{12}a_{22})^{(k)})_2$ $(k = 1, 2, \cdots, 8)$ is derived as

$$(a_{11}a_{22} + a_{12}a_{22})_2 = ((a_{11}a_{22} + a_{12}a_{22})^{(1)})_2((a_{11}a_{22} + a_{12}a_{22})^{(2)})_2 \cdots ((a_{11}a_{22} + a_{12}a_{22})^{(8)})_2.$$

A similar way, the relationship between $(\hat{a}_{11}\hat{a}_{22} + \hat{a}_{12}\hat{a}_{22})_2$ and $((\hat{a}_{11}\hat{a}_{22} + \hat{a}_{12}\hat{a}_{22})^{(k)})_2$ $(k = 1, 2, \cdots, 8)$ is derived as

$$(\hat{a}_{11}\hat{a}_{22} + \hat{a}_{12}\hat{a}_{22})_2 = ((\hat{a}_{11}\hat{a}_{22} + \hat{a}_{12}\hat{a}_{22})^{(1)})_2((\hat{a}_{11}\hat{a}_{22} + \hat{a}_{12}\hat{a}_{22})^{(2)})_2 \cdots ((\hat{a}_{11}\hat{a}_{22} + \hat{a}_{12}\hat{a}_{22})^{(8)})_2.$$

From the above description, the value of $(\hat{a}_{11}\hat{a}_{22} + \hat{a}_{12}\hat{a}_{22})$ can be obtained by the divide-and-conquer attack. The corresponding flowchart is shown in Fig. 3.

According to Fig. 3, one has the divide-and-conquer attack algorithm for obtaining $(\hat{a}_{11}\hat{a}_{22} + \hat{a}_{12}\hat{a}_{22})$, as summarized in Algorithm 1.

In Algorithm 1, nthroot($\cdot$,2) performs a square-root operation on a decimal number, dec2bin(mod($\cdot$,2^8),8) converts a decimal integer to an 8-bit binary number, bin2dec($\cdot$,64) converts a 64-bit binary number to a decimal.

According to Fig. 3 and Algorithm 1, one can obtain the estimated value of the secret key expression $(\hat{a}_{11}\hat{a}_{22} + \hat{a}_{12}\hat{a}_{22})$. In the same way, one can further obtain the nonlinear equation system about $\hat{a}_{11}, \hat{a}_{12}, \hat{a}_{13}, \hat{a}_{22}, \hat{a}_{23}$. Detailed steps are as follows:

Step 1. Obtain the first 8-bit of $(a_{11}a_{22} + a_{12}a_{22})_2$.

(1) Set $c^{(1)} = \sqrt{2^{7+27}}$.

According to $m_4(1)$ in Eq. (19), for eliminating 2^{27} in $\mathrm{mod}\left(\left\lfloor (a_{11}a_{22} + a_{12}a_{22}) \times (c^{(1)})^2/2^{27} \right\rfloor, 2^8\right)$, $c^{(1)} = \sqrt{2^{7+27}}$ is set. Then, the corresponding plaintext pixel value $m_4^{(1)}(1)$ can be obtained (by using the

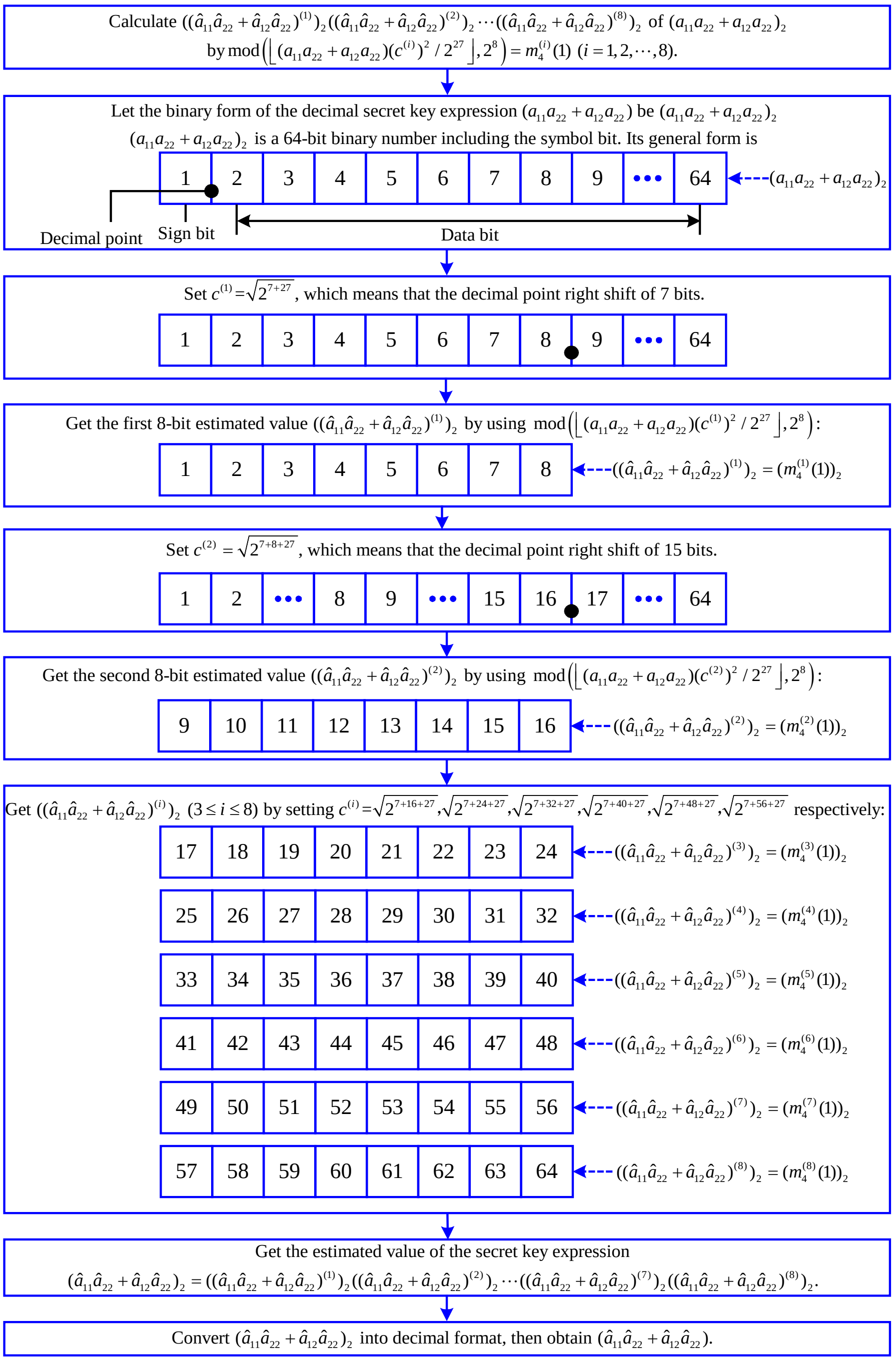

Calculate $((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(1)})_2((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(2)})_2\cdots((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(8)})_2$ of $(a_{11}a_{22}+a_{12}a_{22})_2$
by $\mathrm{mod}\left(\left\lfloor(a_{11}a_{22}+a_{12}a_{22})(c^{(i)})^2/2^{27}\right\rfloor,2^8\right)=m_4^{(i)}(1)$ $(i=1,2,\cdots,8)$.
Let the binary form of the decimal secret key expression $(a_{11}a_{22}+a_{12}a_{22})$ be $(a_{11}a_{22}+a_{12}a_{22})_2$
$(a_{11}a_{22}+a_{12}a_{22})_2$ is a 64-bit binary number including the symbol bit. Its general form is
1 2 3 4 5 6 7 8 9 ••• 64
$(a_{11}a_{22}+a_{12}a_{22})_2$
Decimal point
Sign bit
Data bit
Set $c^{(1)}=\sqrt{2^{7+27}}$, which means that the decimal point right shift of 7 bits.
1 2 3 4 5 6 7 8 9 ••• 64
Get the first 8-bit estimated value $((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(1)})_2$ by using $\mathrm{mod}\left(\left\lfloor(a_{11}a_{22}+a_{12}a_{22})(c^{(1)})^2/2^{27}\right\rfloor,2^8\right)$:
1 2 3 4 5 6 7 8
$((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(1)})_2=(m_4^{(1)}(1))_2$
Set $c^{(2)}=\sqrt{2^{7+8+27}}$, which means that the decimal point right shift of 15 bits.
1 2 ••• 8 9 ••• 15 16 17 ••• 64
Get the second 8-bit estimated value $((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(2)})_2$ by using $\mathrm{mod}\left(\left\lfloor(a_{11}a_{22}+a_{12}a_{22})(c^{(2)})^2/2^{27}\right\rfloor,2^8\right)$:
9 10 11 12 13 14 15 16
$((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(2)})_2=(m_4^{(2)}(1))_2$
Get $((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(i)})_2$ $(3\le i\le 8)$ by setting $c^{(i)}=\sqrt{2^{7+16+27}},\sqrt{2^{7+24+27}},\sqrt{2^{7+32+27}},\sqrt{2^{7+40+27}},\sqrt{2^{7+48+27}},\sqrt{2^{7+56+27}}$ respectively:
17 18 19 20 21 22 23 24
$((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(3)})_2=(m_4^{(3)}(1))_2$
25 26 27 28 29 30 31 32
$((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(4)})_2=(m_4^{(4)}(1))_2$
33 34 35 36 37 38 39 40
$((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(5)})_2=(m_4^{(5)}(1))_2$
41 42 43 44 45 46 47 48
$((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(6)})_2=(m_4^{(6)}(1))_2$
49 50 51 52 53 54 55 56
$((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(7)})_2=(m_4^{(7)}(1))_2$
57 58 59 60 61 62 63 64
$((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(8)})_2=(m_4^{(8)}(1))_2$
Get the estimated value of the secret key expression
$(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})_2=((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(1)})_2((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(2)})_2\cdots((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(7)})_2((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(8)})_2$.
Convert $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})_2$ into decimal format, then obtain $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})$.

Fig. 3. Flowchart of the divide-and-conquer attack for obtaining $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})$

decrypting machine).

(2) Obtain the relational expression between $m_4^{(1)}(1)$ and $c^{(1)}=\sqrt{2^{7+27}}$.

By substituting $c^{(1)}=\sqrt{2^{7+27}}$ into $m_4(1)$ in Eq. (19), the relational expression between them is derived as

$$\begin{aligned} m_4^{(1)}(1) &= \mathrm{mod}\left(\left\lfloor(a_{11}a_{22}+a_{12}a_{22})\times\left(c^{(1)}\right)^2/2^{27}\right\rfloor,2^8\right) \\ &= \mathrm{mod}\left(\left\lfloor(a_{11}a_{22}+a_{12}a_{22})\times 2^7\right\rfloor,2^8\right). \end{aligned} \tag{20}$$

Algorithm 1 The divide-and-conquer attack algorithm for obtaining $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})$

```
Input: $x_1(0),\ x_2(0),\ x_3(0)$
Output: $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})$
  for $round \leftarrow 0$ to 7 do
    $x_1(0) \leftarrow$ nthroot($(2^7)\times(2^{27})\times(2^8)^{round}$, 2);
    $x_2(0) \leftarrow$ nthroot($(2^7)\times(2^{27})\times(2^8)^{round}$, 2);
    $x_3(0) \leftarrow 0$;
    for $k \leftarrow 0$ do
      $x_1(k+1) \leftarrow a_{11}x_1(k)+a_{12}x_2(k)+a_{13}x_3(k)$;
      $x_2(k+1) \leftarrow a_{22}x_2(k)+a_{23}x_3(k)$;
      $x_3(k+1) \leftarrow a_{32}x_2(k)+a_{33}x_3(k)$;
    end for
    $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})_2 \leftarrow$ [$(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})_2$,dec2bin(mod(floor($x_1(1)\times x_2(1)/2^{27}$),$2^8$),8)];
    $round \leftarrow round+1$;
  end for
  $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22}) \leftarrow$ bin2dec($(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})_2$,64);
  return $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})$
```

(3) Obtain the first 8-bit of $(a_{11}a_{22}+a_{12}a_{22})_2$.

According to Eq. (20), the first 8-bit of $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})_2$ is derived as

$$((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(1)})_2=\left(\text{mod}\left(\lfloor(a_{11}a_{22}+a_{12}a_{22})\rfloor\times 2^7,2^8\right)\right)_2=(m_4^{(1)}(1))_2, \tag{21}$$

where $((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(1)})_2$ denotes the first 8-bit of $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})_2$, including a sign bit and seven data bits. Here, $\left(\text{mod}\left(\left\lfloor(a_{11}a_{22}+a_{12}a_{22})\times 2^7\right\rfloor,2^8\right)\right)_2$ denotes the binary form of mod $\left(\left\lfloor(a_{11}a_{22}+a_{12}a_{22})\times 2^7\right\rfloor,2^8\right)$ and $(m_4^{(1)}(1))_2$ denotes the binary form of $m_4^{(1)}(1)$.

Step 2. Obtain the second 8-bit of $(a_{11}a_{22}+a_{12}a_{22})_2$.

(1) Set $c^{(2)}=\sqrt{2^{7+8+27}}$.

According to $m_4(1)$ in Eq. (19), for eliminating 2^{27} in mod$\left(\left\lfloor(a_{11}a_{22}+a_{12}a_{22})\times(c^{(2)})^2/2^{27}\right\rfloor,2^8\right)$, $c^{(2)}=\sqrt{2^{7+8+27}}$ is set. Then, the corresponding plaintext pixel value $m_4^{(2)}(1)$ can be obtained (by using the decrypting machine).

(2) Obtain the relational expression between $m_4^{(2)}(1)$ and $c^{(2)}=\sqrt{2^{7+8+27}}$.

By substituting $c^{(2)}=\sqrt{2^{7+8+27}}$ into $m_4(1)$ in Eq. (19), the relational expression between them is derived as

$$\begin{aligned} m_4^{(2)}(1)&=\text{mod}\left(\left\lfloor(a_{11}a_{22}+a_{12}a_{22})\times(c^{(2)})^2/2^{27}\right\rfloor,2^8\right)\\ &=\text{mod}\left(\left\lfloor(a_{11}a_{22}+a_{12}a_{22})\times 2^{7+8}\right\rfloor,2^8\right). \end{aligned} \tag{22}$$

(3) Obtain the second 8-bit of $(a_{11}a_{22}+a_{12}a_{22})_2$.

According to Eq. (22), the second 8-bit of $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})_2$ is derived as

$$((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(2)})_2=\left(\text{mod}\left(\lfloor(a_{11}a_{22}+a_{12}a_{22})\rfloor\times 2^{7+8},2^8\right)\right)_2=(m_4^{(2)}(1))_2. \tag{23}$$

Step 3. Obtain the third to eighth 8-bit of $(a_{11}a_{22}+a_{12}a_{22})_2$.

Similarly, according to $m_4(1)$ in Eq. (19), $c^{(3)}=\sqrt{2^{7+16+27}}$, $c^{(4)}=\sqrt{2^{7+24+27}}$, $c^{(5)}=\sqrt{2^{7+32+27}}$, $c^{(6)}=\sqrt{2^{7+40+27}}$, $c^{(7)}=\sqrt{2^{7+48+27}}$, $c^{(8)}=\sqrt{2^{7+56+27}}$ are set. Then, the corresponding plaintext pixel values $(m_4^{(3)}(1))_2,(m_4^{(4)}(1))_2,(m_4^{(5)}(1))_2,(m_4^{(6)}(1))_2,(m_4^{(7)}(1))_2,(m_4^{(8)}(1))_2$ can be obtained (by using the decrypting machine). On this basis, the third to eight 8-bits can be further derived as

$$((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(3)})_2,((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(4)})_2,\cdots,((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(7)})_2,((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(8)})_2. \tag{24}$$

Step 4. Obtain $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})$ by $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})_2$.

(1) Use $((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(i)})_2(i=1,2,\cdots,8)$ to splice a 64-bit binary number $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})_2$.

Splice $((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(i)})_2 (i=1,2,\cdots,8)$ from small to large according to the label i. Then, the binary form of $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})_2$ is derived as

$$(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})_2=((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(1)})_2((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(2)})_2\cdots((\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})^{(8)})_2. \tag{25}$$

(2) Convert $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})_2$ into a decimal format and then obtain $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})$.

Convert the binary number $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})_2$ to the decimal number $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})$. The conversion method is shown in Algorithm 2, where $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})_2(i)$ denotes the i^{th} of binary number $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})_2$.

Algorithm 2 The conversion algorithm for obtaining $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})$

```
Input: $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})_2$
Output: $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})$
  $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22}) \leftarrow 0$;
  for $i \leftarrow 2$ to 64 do
    if $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})_2(i) == 1$ then
      $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22}) \leftarrow (\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22}) + 2^{-(i-1)}$;
    end if
  end for
  if $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})_2(1) == 1$ then
    $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22}) \leftarrow (\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22}) - 1$;
  end if
  return $(\hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22})$
```

Similarly, calculate $m_i(1)$ $(i=2,3,5,6,7)$ in Eq. (19) by using similar methods of the divide-and-conquer attack presented in Step 1 to Step 4. Then, the corresponding estimated values $\hat{a}_{12}\hat{a}_{22}, \hat{a}_{13}\hat{a}_{23}, \hat{a}_{11}\hat{a}_{23}+\hat{a}_{13}\hat{a}_{23}, (\hat{a}_{12}+\hat{a}_{13})\times(\hat{a}_{22}+\hat{a}_{23})$ and $(\hat{a}_{22}+\hat{a}_{23})\times(\hat{a}_{11}+\hat{a}_{12}+\hat{a}_{13})$ of the secret key expressions $a_{12}a_{22}, a_{13}a_{23}, a_{11}a_{23}+a_{13}a_{23}, (a_{12}+a_{13})\times(a_{22}+a_{23})$ and $(a_{22}+a_{23})\times(a_{11}+a_{12}+a_{13})$ can be obtained.

Finally, the nonlinear equation system about $\hat{a}_{11}, \hat{a}_{12}, \hat{a}_{13}, \hat{a}_{22}, \hat{a}_{23}$ is derived as

$$\begin{cases} \hat{a}_{12}\hat{a}_{22} = 0.0666 \\ \hat{a}_{13}\hat{a}_{23} = 0.037 \\ \hat{a}_{11}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22} = 0.0504 \\ \hat{a}_{11}\hat{a}_{23}+\hat{a}_{13}\hat{a}_{23} = 0.0703 \\ (\hat{a}_{12}+\hat{a}_{13})\times(\hat{a}_{22}+\hat{a}_{23}) = -0.0513 \\ (\hat{a}_{22}+\hat{a}_{23})\times(\hat{a}_{11}+\hat{a}_{12}+\hat{a}_{13}) = -0.0342 \end{cases}. \tag{26}$$

3.4.3. *Establish a nonlinear equation system about* $\hat{a}_{11}, \hat{a}_{12}, \hat{a}_{13}, \hat{a}_{22}, \hat{a}_{23}, \hat{a}_{32}, \hat{a}_{33}$ *through the second iteration*

Firstly, by substituting $k=1$ into Eq. (9), the second iteration result is derived as

$$\begin{cases} x_1(2) = a_{11}x_1(1) + a_{12}x_2(1) + a_{13}x_3(1) \\ x_2(2) = a_{22}x_2(1) + a_{23}x_3(1) \\ x_3(2) = a_{32}x_2(1) + a_{33}x_3(1) \end{cases}. \tag{27}$$

Then, by substituting Eq. (18) into Eq. (27), one has

$$\begin{cases} x_1(2) = a_{11}(a_{11}c_1 + a_{12}c_2 + a_{13}c_3) \\ \qquad + a_{12}(a_{22}c_2 + a_{23}c_3) + a_{13}(a_{22}c_2 + a_{23}c_3) \\ x_2(2) = a_{22}(a_{22}c_2 + a_{23}c_3) + a_{23}(a_{22}c_2 + a_{23}c_3) \\ x_3(2) = a_{32}(a_{22}c_2 + a_{23}c_3) + a_{33}(a_{22}c_2 + a_{23}c_3) \end{cases}. \tag{28}$$

If the initial condition is determined to be valid in the first iteration, it is still valid after several iterations. Similarly, if the initial condition is determined to be invalid in the first iteration, it is still invalid after several iterations. Only the valid initial conditions are considered. By substituting the calculation results of $x_1(2)x_2(2)$ into Eq. (10), and setting $c_i = c$ $(i = 1, 2, 3)$, the expression of $m_i(2)$ $(i = 2, 3, \cdots, 7)$ is obtained as

$$\left\{\begin{aligned}
m_2(2) &= \left(\bmod\left(\left\lfloor \frac{(a_{22}^2 + a_{23}a_{32}c_2) \times (a_{11}a_{12}c_2 + a_{12}a_{22}c_2 + a_{13}a_{32}c_2)}{2^{27}} \right\rfloor, 2^8\right)\right) \\
&= \left(\bmod\left(\left\lfloor \frac{(a_{22}^2 + a_{23}a_{32}) \times (a_{11}a_{12} + a_{12}a_{22} + a_{13}a_{32})c^2}{2^{27}} \right\rfloor, 2^8\right)\right) \\
m_3(2) &= \left(\bmod\left(\left\lfloor \frac{(a_{22}a_{23}c_3 + a_{23}a_{33}c_3) \times (a_{11}a_{13}c_3 + a_{12}a_{23}c_3 + a_{13}a_{33}c_3)}{2^{27}} \right\rfloor, 2^8\right)\right) \\
&= \left(\bmod\left(\left\lfloor \frac{(a_{22}a_{23} + a_{23}a_{33}) \times (a_{11}a_{13} + a_{12}a_{23} + a_{13}a_{33})c^2}{2^{27}} \right\rfloor, 2^8\right)\right) \\
m_4(2) &= \left(\bmod\left(\left\lfloor \frac{(a_{22}^2 + a_{23}a_{32}c_2) \times (a_{11}^2c_1 + a_{11}a_{12}c_2 + a_{12}a_{22}c_2 + a_{13}a_{32}c_2)}{2^{27}} \right\rfloor, 2^8\right)\right) \\
&= \left(\bmod\left(\left\lfloor \frac{(a_{22}^2 + a_{23}a_{32}) \times (a_{11}^2 + a_{11}a_{12} + a_{12}a_{22} + a_{13}a_{32})c^2}{2^{27}} \right\rfloor, 2^8\right)\right) \\
m_5(2) &= \left(\bmod\left(\left\lfloor \frac{(a_{22}a_{23}c_3 + a_{23}a_{33}c_3) \times (a_{11}^2c_1 + a_{11}a_{13}c_3 + a_{12}a_{23}c_3 + a_{13}a_{33}c_3)}{2^{27}} \right\rfloor, 2^8\right)\right) \\
&= \left(\bmod\left(\left\lfloor \frac{(a_{22}a_{23} + a_{23}a_{33}) \times (a_{11}^2 + a_{11}a_{13} + a_{12}a_{23} + a_{13}a_{33})c^2}{2^{27}} \right\rfloor, 2^8\right)\right) \\
m_6(2) &= \left(\bmod\left(\left\lfloor \frac{(a_{22}^2c_2 + a_{22}a_{23}c_3 + a_{23}a_{32}c_2 + a_{23}a_{33}c_3) \times (a_{11}a_{12}c_2 + a_{11}a_{13}c_3 + a_{12}a_{22}c_2 + a_{12}a_{23}c_3 + a_{13}a_{32}c_2 + a_{13}a_{33}c_3)}{2^{27}} \right\rfloor, 2^8\right)\right) \\
&= \left(\bmod\left(\left\lfloor \frac{(a_{22}^2 + a_{22}a_{23} + a_{23}a_{32} + a_{23}a_{33}) \times (a_{11}a_{12} + a_{11}a_{13} + a_{12}a_{22} + a_{12}a_{23} + a_{13}a_{32} + a_{13}a_{33})c^2}{2^{27}} \right\rfloor, 2^8\right)\right) \\
m_7(2) &= \left(\bmod\left(\left\lfloor \frac{(a_{22}^2c_2+a_{22}a_{23}c_3+a_{23}a_{32}c_2+a_{23}a_{33}c_3)\times(a_{12}a_{22}c_2+a_{12}a_{23}c_3+a_{13}a_{32}c_2+a_{13}a_{33}c_3+a_{11}^2c_1+a_{11}a_{12}c_2+a_{11}a_{13}c_3)}{2^{27}} \right\rfloor, 2^8\right)\right) \\
&= \left(\bmod\left(\left\lfloor \frac{(a_{22}^2+a_{22}a_{23}+a_{23}a_{32}+a_{23}a_{33})\times(a_{12}a_{22}+a_{12}a_{23}+a_{13}a_{32}+a_{13}a_{33}+a_{11}^2+a_{11}a_{12}+a_{11}a_{13})c^2}{2^{27}} \right\rfloor, 2^8\right)\right)
\end{aligned}\right. . \tag{29}$$

Note that, in Eq. (29), when $c_1 \neq c_2 \neq c_3$, the expression of $m_i(2)(i = 2, 3, \cdots, 7)$ cannot be further simplified, so the estimated values of secret key expressions cannot be obtained by the divide-and-conquer attack. However, when $c_1 = c_2 = c_3 = c$ the common factor c^2 in $\bmod(\cdot)$ can be extracted, so an appropriate value of c can be set to obtain the estimated values of secret key expressions by the divide-and-conquer attack.

Similarly, calculate $m_i(2)(i = 2, 3, 5, 6, 7)$ in Eq. (29) by using similar methods of divide-and-conquer attack proposed in Section 3.4.2. Then, the corresponding estimated values of the secret key expressions can be obtained.

Finally, the nonlinear equation system about $\hat{a}_{11}, \hat{a}_{12}, \hat{a}_{13}, \hat{a}_{22}, \hat{a}_{23}, \hat{a}_{32}, \hat{a}_{33}$ is derived as

$$\begin{cases}
(\hat{a}_{22}^2 + \hat{a}_{23}\hat{a}_{32}) \times (\hat{a}_{11}\hat{a}_{12} + \hat{a}_{12}\hat{a}_{22} + \hat{a}_{13}\hat{a}_{32}) = -0.00042525 \\
(\hat{a}_{22}\hat{a}_{23} + \hat{a}_{23}\hat{a}_{33}) \times (\hat{a}_{11}\hat{a}_{13} + \hat{a}_{12}\hat{a}_{23} + \hat{a}_{13}\hat{a}_{33}) = -0.00040293 \\
(\hat{a}_{22}^2 + \hat{a}_{23}\hat{a}_{32}) \times (\hat{a}_{11}^2 + \hat{a}_{11}\hat{a}_{12} + \hat{a}_{12}\hat{a}_{22} + \hat{a}_{13}\hat{a}_{32}) = -0.00097200 \\
(\hat{a}_{22}\hat{a}_{23} + \hat{a}_{23}\hat{a}_{33}) \times (\hat{a}_{11}^2 + \hat{a}_{11}\hat{a}_{13} + \hat{a}_{12}\hat{a}_{23} + \hat{a}_{13}\hat{a}_{33}) = -0.00037296 \\
(\hat{a}_{22}^2 + \hat{a}_{22}\hat{a}_{23} + \hat{a}_{23}\hat{a}_{32} + \hat{a}_{23}\hat{a}_{33}) \times (\hat{a}_{11}\hat{a}_{12} + \hat{a}_{11}\hat{a}_{13} + \hat{a}_{12}\hat{a}_{22} + \hat{a}_{12}\hat{a}_{23} + \hat{a}_{13}\hat{a}_{32} + \hat{a}_{13}\hat{a}_{33}) = 0.00654588 \\
(\hat{a}_{22}^2 + \hat{a}_{22}\hat{a}_{23} + \hat{a}_{23}a_{32} + \hat{a}_{23}\hat{a}_{33}) \times (\hat{a}_{12}\hat{a}_{22} + \hat{a}_{12}\hat{a}_{23} + \hat{a}_{13}\hat{a}_{32} + \hat{a}_{13}\hat{a}_{33} + \hat{a}_{11}^2 + \hat{a}_{11}\hat{a}_{12} + \hat{a}_{11}\hat{a}_{13}) = 0.00602910
\end{cases} . \tag{30}$$

3.4.4. *Solve the estimated values of secret keys* $\hat{a}_{11}, \hat{a}_{12}, \hat{a}_{13}, \hat{a}_{22}, \hat{a}_{23}, \hat{a}_{32}, \hat{a}_{33}$

Firstly, take the nonlinear equation system established by the first iteration as the known condition, and observe the internal relation between the known condition and the nonlinear equation system established by the second iteration. Then, the nonlinear equation system established by the second iteration is simplified by using the global substitution method. Finally, the solution of several sets of estimated secret keys is solved.

According to Eq. (26), considering the expression of multiplying two secret keys together, one has

$$\begin{cases}
\hat{a}_{12}\hat{a}_{22} = 0.0666, \hat{a}_{13}\hat{a}_{23} = 0.037, \hat{a}_{11}\hat{a}_{22} = -0.0162, \\
\hat{a}_{11}\hat{a}_{23} = 0.0333, \hat{a}_{12}\hat{a}_{23} + \hat{a}_{13}\hat{a}_{22} = -0.1549
\end{cases} . \tag{31}$$

Expand the first to the fourth equations of Eq. (30), and recombine the expressions of secret key multiplications, one gets

$$\begin{cases} \hat{a}_{11}\hat{a}_{22}\hat{a}_{12}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22}\hat{a}_{22}\hat{a}_{22} + \hat{a}_{13}\hat{a}_{22}\hat{a}_{22}\hat{a}_{32}+\hat{a}_{11}\hat{a}_{23}\hat{a}_{12}\hat{a}_{32} \\ +\hat{a}_{12}\hat{a}_{22}\hat{a}_{23}\hat{a}_{32}+\hat{a}_{13}\hat{a}_{23}\hat{a}_{32}\hat{a}_{32}= -\ 0.00042525 \\ \hat{a}_{11}\hat{a}_{22}\hat{a}_{13}\hat{a}_{23}+\hat{a}_{12}\hat{a}_{22}\hat{a}_{23}\hat{a}_{23}+\hat{a}_{13}\hat{a}_{23}\hat{a}_{22}\hat{a}_{33} + \hat{a}_{11}\hat{a}_{23}\hat{a}_{13}\hat{a}_{33} \\ +\hat{a}_{12}\hat{a}_{23}\hat{a}_{23}\hat{a}_{33}+\hat{a}_{13}\hat{a}_{23}\hat{a}_{33}\hat{a}_{33}= -\ 0.00040293 \\ \hat{a}_{11}\hat{a}_{22}\hat{a}_{11}\hat{a}_{22}+\hat{a}_{11}\hat{a}_{22}\hat{a}_{12}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{22}\hat{a}_{22}\hat{a}_{22}+\hat{a}_{13}\hat{a}_{22}\hat{a}_{22}\hat{a}_{32} \\ +\hat{a}_{11}\hat{a}_{23}\hat{a}_{11}\hat{a}_{32}+\hat{a}_{11}\hat{a}_{23}\hat{a}_{12}\hat{a}_{32}+\hat{a}_{12}\hat{a}_{22}\hat{a}_{23}\hat{a}_{32}+\hat{a}_{13}\hat{a}_{23}\hat{a}_{32}\hat{a}_{32}= -\ 0.00097200 \\ \hat{a}_{11}\hat{a}_{22}\hat{a}_{11}\hat{a}_{23}+\hat{a}_{11}\hat{a}_{22}\hat{a}_{13}\hat{a}_{23}+\hat{a}_{12}\hat{a}_{22}\hat{a}_{23}\hat{a}_{23}+\hat{a}_{11}\hat{a}_{23}\hat{a}_{11}\hat{a}_{33} \\ +\hat{a}_{11}\hat{a}_{23}\hat{a}_{13}\hat{a}_{33}+(\hat{a}_{13}\hat{a}_{22}+\hat{a}_{12}\hat{a}_{23})\hat{a}_{23}\hat{a}_{33}+\hat{a}_{13}\hat{a}_{23}\hat{a}_{33}\hat{a}_{33}= -\ 0.00037296 \end{cases} . \tag{32}$$

By substituting Eq. (31) into Eq. (32), one gets

$$\begin{cases} 0.0666\hat{a}_{22}^2 + 0.0333\hat{a}_{12}\hat{a}_{32} + 0.0666\hat{a}_{23}\hat{a}_{32} + 0.037\hat{a}_{32}^2 + \hat{a}_{13}\hat{a}_{22}^2\hat{a}_{32} = 0.00065367 \\ 0.0666\hat{a}_{23}^2 + 0.037\hat{a}_{22}\hat{a}_{33} + 0.0333\hat{a}_{13}\hat{a}_{33} + 0.037\hat{a}_{33}^2 + \hat{a}_{12}\hat{a}_{23}^2\hat{a}_{33} = 0.00019647 \\ 0.0666\hat{a}_{22}^2 + 0.0333\hat{a}_{11}\hat{a}_{32} + 0.0333\hat{a}_{12}\hat{a}_{32} + 0.0666\hat{a}_{23}\hat{a}_{32} + 0.037\hat{a}_{32}^2 + \hat{a}_{13}\hat{a}_{22}^2\hat{a}_{32} = -0.00015552 \\ 0.0666\hat{a}_{23}^2 + 0.0333\hat{a}_{11}\hat{a}_{33} + 0.0333\hat{a}_{13}\hat{a}_{33} - 0.1549\hat{a}_{23}\hat{a}_{33} + 0.037\hat{a}_{33}^2 = 0.0007659 \end{cases} . \tag{33}$$

Then, Eq. (31) is further simplified, and $\hat{a}_{12}$ is used to represent $\hat{a}_{11}, \hat{a}_{13}, \hat{a}_{22}, \hat{a}_{23}$, to get

$$\begin{cases} \hat{a}_{11}= -\frac{9}{37}\hat{a}_{12}, \hat{a}_{13} = -\frac{10}{37}\hat{a}_{12} \\ \hat{a}_{22} = \frac{0.0666}{\hat{a}_{12}}, \hat{a}_{23} = -\frac{0.1369}{\hat{a}_{12}} \end{cases} . \tag{34}$$

By substituting Eq. (34) into Eq. (33), the equations of $\hat{a}_{12}, \hat{a}_{32}, \hat{a}_{33}$ can be derived, as

$$\begin{cases} \frac{0.0333\hat{a}_{12}^3\hat{a}_{32}-0.01031634\hat{a}_{12}\hat{a}_{32}+0.037\hat{a}_{12}^2\hat{a}_{32}^2+0.000295408296}{\hat{a}_{12}^2} = 0.00065367 \\ \frac{-0.009\hat{a}_{12}^3\hat{a}_{33}+0.02120581\hat{a}_{12}\hat{a}_{33}+0.037\hat{a}_{33}^2\hat{a}_{12}^2+0.001248191226}{\hat{a}_{12}^2} = 0.00019647 \\ \frac{0.0252\hat{a}_{12}^3\hat{a}_{32}-0.01031634\hat{a}_{12}\hat{a}_{32}+0.0370\hat{a}_{12}^2\hat{a}_{32}^2+0.000295408296}{\hat{a}_{12}^2} = -0.00015552 \\ \frac{-0.0171\hat{a}_{12}^3\hat{a}_{33}+0.02120581\hat{a}_{12}\hat{a}_{33}+0.037\hat{a}_{33}^2\hat{a}_{12}^2+0.001248191226}{\hat{a}_{12}^2} = 0.0007659 \end{cases} . \tag{35}$$

Combine the first equation and the third equation in Eq. (35), one gets

$$\begin{cases} \frac{0.0333\hat{a}_{12}^3\hat{a}_{32}-0.01031634\hat{a}_{12}\hat{a}_{32}+0.037\hat{a}_{12}^2\hat{a}_{32}^2+0.000295408296}{\hat{a}_{12}^2} = 0.00065367 \\ \frac{0.0252\hat{a}_{12}^3\hat{a}_{32}-0.01031634\hat{a}_{12}\hat{a}_{32}+0.0370\hat{a}_{12}^2\hat{a}_{32}^2+0.000295408296}{\hat{a}_{12}^2} = -0.00015552 \end{cases} . \tag{36}$$

According to Eq. (36), two sets of the solution of $\hat{a}_{12}, \hat{a}_{32}$ are solved as

$$\begin{cases} \hat{a}_{12} = -0.37, \hat{a}_{32} = -0.27 \\ \hat{a}_{12} = 0.37, \hat{a}_{32} = 0.27 \end{cases} . \tag{37}$$

Combining the second equation and the fourth equation in Eq. (35), one gets

$$\begin{cases} \frac{-0.009\hat{a}_{12}^3\hat{a}_{33}+0.02120581\hat{a}_{12}\hat{a}_{33}+0.037\hat{a}_{33}^2\hat{a}_{12}^2+0.001248191226}{\hat{a}_{12}^2} = 0.00019647 \\ \frac{-0.0171\hat{a}_{12}^3\hat{a}_{33}+0.02120581\hat{a}_{12}\hat{a}_{33}+0.037\hat{a}_{33}^2\hat{a}_{12}^2+0.001248191226}{\hat{a}_{12}^2} = 0.0007659 \end{cases} . \tag{38}$$

According to Eq. (38), two sets of the solution of $\hat{a}_{12}, \hat{a}_{33}$ are solved, yielding

$$\begin{cases} \hat{a}_{12} = -0.37, \hat{a}_{33} = 0.19 \\ \hat{a}_{12} = 0.37, \hat{a}_{33} = -0.19 \end{cases} . \tag{39}$$

According to Eq. (37) and Eq. (39), two sets of the solution of $\hat{a}_{12}, \hat{a}_{32}, \hat{a}_{33}$ are obtained as

$$\begin{cases} \hat{a}_{12} = -0.37, \hat{a}_{32} = -0.27, , \hat{a}_{33} = 0.19 \\ \hat{a}_{12} = 0.37, \hat{a}_{32} = 0.27, , \hat{a}_{33} = -0.19 \end{cases} . \tag{40}$$

By substituting Eq. (40) into Eq. (33), two sets of the solution of $\hat{a}_{11}, \hat{a}_{12}, \hat{a}_{13}, \hat{a}_{22}, \hat{a}_{23}, \hat{a}_{32}, \hat{a}_{33}$ are solved as

$$\begin{cases} \hat{a}_{11} = 0.09, \hat{a}_{12} = -0.37, \hat{a}_{13} = 0.1, \hat{a}_{22} = -0.18, \hat{a}_{23} = 0.37, \hat{a}_{32} = -0.27, \hat{a}_{33} = 0.19 \\ \hat{a}_{11} = -0.09, \hat{a}_{12} = 0.37, \hat{a}_{13} = -0.1, \hat{a}_{22} = 0.18, \hat{a}_{23} = -0.37, \hat{a}_{32} = 0.27, \hat{a}_{33} = -0.19 \end{cases}. \tag{41}$$

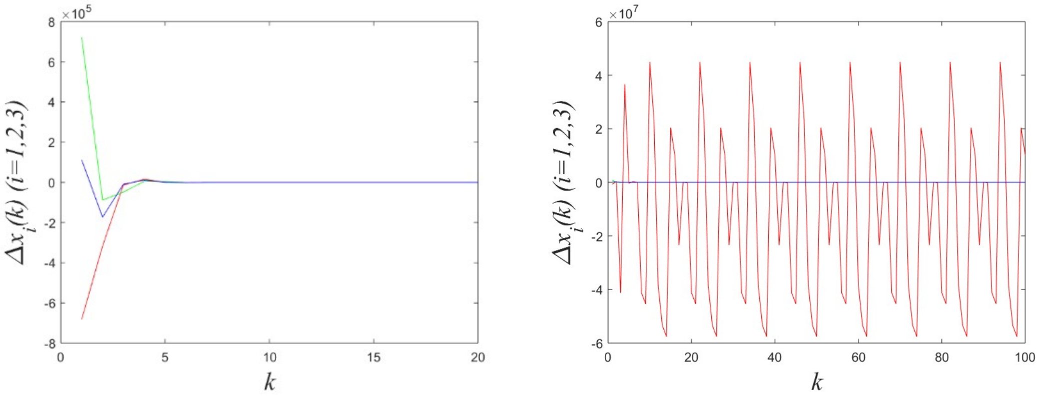
Noted that Eq.(31) - Eq.(41) is the solving process of the nonlinear equations using the method of gobal substitution and simplified evaluation. When using Maple 2018 software to solve the nonlinear equations in Eq.(26) and Eq.(30), the numerical solution can be obtained in about 1.3 seconds.

3.4.5. *Confirm the unique solution of* $\hat{a}_{11}, \hat{a}_{12}, \hat{a}_{13}, \hat{a}_{22}, \hat{a}_{23}, \hat{a}_{32}, \hat{a}_{33}$

To confirm the correctness and uniqueness of the solutions of $\hat{a}_{11}, \hat{a}_{12}, \hat{a}_{13}, \hat{a}_{22}, \hat{a}_{23}, \hat{a}_{32}, \hat{a}_{33}$ in Eq. (41), they are substituted into Eq. (7), yielding

$$\begin{cases} x_1(k+1) = \hat{a}_{11}x_1(k) + \hat{a}_{12}x_2(k) + \hat{a}_{13}x_3(k) \\ x_2(k+1) = a_{21}p(k) + \hat{a}_{22}x_2(k) + \hat{a}_{23}x_3(k) \\ x_3(k+1) = a_{31}p(k) + \hat{a}_{32}x_2(k) + \hat{a}_{33}x_3(k) + \varepsilon\sin(\sigma p(k)) \end{cases}. \tag{42}$$

When the secret keys of the sender and the receiver match, Eq. (7) and Eq. (42) can realize self-synchronization. Therefore, Eq. (41) is substituted into Eq. (42) to investigate the synchronization between Eq. (7) and Eq. (42). When Eq. (7) and Eq. (42) are synchronized, the corresponding synchronization error tends to 0; otherwise it does not tend to 0. The simulation results are shown in Figs. 4 (a) and (b), respectively.



(a)Simulation result of synchronization error when $\hat{a}_{11} = 0.09$, $\hat{a}_{12} = -0.37$, $\hat{a}_{13} = 0.1$, $\hat{a}_{22} = -0.18$, $\hat{a}_{23} = 0.37$, $\hat{a}_{32} = -0.27$, $\hat{a}_{33} = 0.19$.

(b)Simulation result of synchronization error when $\hat{a}_{11} = -0.09$, $\hat{a}_{12} = 0.37$, $\hat{a}_{13} = -0.1$, $\hat{a}_{22} = 0.18$, $\hat{a}_{23} = -0.37$, $\hat{a}_{32} = 0.27$, $\hat{a}_{33} = -0.19$.

Fig. 4. Simulation results of synchronization errors

In Fig. 4, the simulation results of synchronization errors show that when $\hat{a}_{11} = 0.09, \hat{a}_{12} = -0.37, \hat{a}_{13} = 0.1, \hat{a}_{22} = -0.18, \hat{a}_{23} = 0.37, \hat{a}_{32} = -0.27, \hat{a}_{33} = 0.19$, Eq. (7) and Eq. (42) can realize self-synchronization. Thus, the unique solution of $\hat{a}_{11}, \hat{a}_{12}, \hat{a}_{13}, \hat{a}_{22}, \hat{a}_{23}, \hat{a}_{32}, \hat{a}_{33}$ can be confirmed. Experimental results show that the deciphered secret keys $\hat{a}_{11}, \hat{a}_{12}, \hat{a}_{13}, \hat{a}_{22}, \hat{a}_{23}, \hat{a}_{32}, \hat{a}_{33}$ are exactly equal to the given values of the secret keys $a_{11}, a_{12}, a_{13}, a_{22}, a_{23}, a_{32}, a_{33}$, which verifies the effectiveness of the proposed DCA-TMNCIC.

4. Comparisons and discussions of DCA-TSNCIC and DCA-TMNCIC

For n-D SCSC ($n = 3, 4, 5, \cdots$), under DCA-TSNCIC, there are n options to select initial conditions. The set of all options is derived as

$$\mathbb{S}_1 = \{(c_1, 0, \cdots, 0), (0, c_2, 0, \cdots, 0), \cdots, (0, \cdots, 0, c_{n-1}, 0), (0, 0, \cdots, c_n)\}, \tag{43}$$

where c_i $(i = 1, 2, \cdots, n)$ denotes the non-zero constant.

Moreover, under DCA-TMNCIC, there are $2^n - 1$ options to select initial conditions. The set of all options is derived as

$$\mathbb{S}_2 = \{(c_1, 0, \cdots, 0), (0, c_2, 0, \cdots, 0), \cdots, (0, 0, \cdots, c_n), (c_1, c_2, 0, \cdots, 0), \cdots, (c_1, c_2, \cdots, c_n)\}. \tag{44}$$

Obviously, one can know that Eq. (43) is just a subset of Eq. (44). When n=3,4,5,6,7,8, comparison results of the numbers of initial condition options under the same dimension are shown in Table 3. With the same dimension, one can see that the number of all options under DCA-TMNCIC is more than the number of all options under DCA-TSNCIC.

Table 3. Comparison results of the numbers of initial condition options under the same dimension

n	Number of all options under DCA-TSNCIC	Number of all options under DCA-TMNCIC
3	3	7
4	4	15
5	5	31
6	6	63
7	7	127
8	8	255

For example, in 3-D SCSC-2, there are three options to select initial conditions under DAC-TSNCIC. The set of all options is given by $(x_1(0), x_2(0), x_3(0)) \in \{(c_1, 0, 0), (0, c_2, 0), (0, 0, c_3)\}$. Obviously, $(c_1, 0, 0)$ is an invalid initial condition, and others are valid initial conditions. When DCA-TSNCIC is used for the security analysis of 3-D SCSC-2, the number of valid initial conditions is only two. In the first iteration, only two nonlinear equations can be established under the two valid initial conditions. At this time, to combine more nonlinear equations, more iterations are needed. However, more iterations cause a higher complexity of nonlinear equations, thereby increasing the workload of deciphering the secret keys. Especially in n-D SCSC-2 and n-D SCSC-3 $(n = 4, 5, 6, 7, 8)$, more secret keys are introduced. With more iterations, the relationship between secret keys will become more complex, increasing the difficulty of solving nonlinear equations.

However, when DCA-TMNCIC is used for the security analysis of 3-D SCSC-2, the number of valid initial conditions obtained by DAC-TMNCIC is six. In the first iteration, six nonlinear equations can be established under the six valid initial conditions. At this time, only two iterations are needed to solve the nonlinear equations. For n-D SCSC-2 and n-D SCSC-3 $(n = 3, 4, 5, 6, 7, 8)$, more valid initial conditions mean that, in each iteration, more nonlinear equations are obtained, consequently less iterations are needed to decipher the secret keys. Less iterations will make a lower degree of the product terms in the nonlinear equations and decrease the workload of deciphering the secret keys.

Most numerical analysis results show that, when the proposed DCA-TMNCIC is used for security analyses of n-D SCSC-2 and n-D SCSC-3 (n=4,5,6,7,8), the secret keys can be deciphered, usually only after two or three iterations. In contrast, the DCA-TSNCIC proposed in [Lin *et al.*, 2018] cannot decipher the secret keys even with a much larger number of iterations. For example, when DCA-TMNCIC is used for security analysis of 4-D SCSC-2, the secret keys can be deciphered after three iterations. However, when DCA-TSNCIC is used for security analysis of 4-D SCSC-2, the secret keys cannot be deciphered even after six iterations.

According to the above comparisons and discussions, one can summarize the comparisons of DCA-TSNCIC and DCA-TMNCIC for n-D SCSC (n=3,4,5,6,7,8), as shown in Table 4.

From Table 4, one can see that DCA-TMNCIC is not only suitable for n-D SCSC-1 (n=3,4,5,6,7,8), 3-D SCSC-2 and 3-D SCSC-3, but also for n-D SCSC-2 and n-D SCSC-3 (n=4,5,6,7,8). Note that the attack intensity grows with the numbers of valid initial conditions and the nonlinear equations obtained in the same iteration. Therefore, the attack intensity of DCA-TMNCIC is higher than that of DCA-TSNCIC.

Table 4. Comparisons of DCA-TSNCIC and DCA-TMNCIC for n-D SCSC

	DCA-TSNCIC proposed in [Lin *et al.*, 2018]	DCA-TMNCIC proposed in this paper
Number of initial conditions	n	$2^n - 1$
Number of valid initial conditions	Fewer	More
Number of equation iterations required to decipher a given cipher	More	Fewer
Number of nonlinear equations obtained in the same iteration	Fewer	More
Attack intensity	Weaker	Stronger
Applicable ciphers	n-D SCSC-1, 3-D SCSC-2, 3-D SCSC-3	n-D SCSC-1, n-D SCSC-2, n-D SCSC-3

5. 3-D SCSC-NNS and its security analysis

According to the security analysis results in Section 3, 3-D SCSC-2 cannot resist DCA-TMNCIC. To further resolve this problem, an improved scheme named 3-D SCSC-NNS is proposed, to satisfy as many invalid initial conditions as possible. Under the invalid initial conditions, the secret keys will never appear in the encryption-decryption equations, so the invalid initial conditions need not be considered in the analysis. Under valid initial conditions, the secret keys will appear in the encryption-decryption equations, so that the valid initial conditions can be considered in the analysis. Less valid initial conditions mean that, in each iteration, less nonlinear equations are obtained; therefore, more iterations are needed. However, more iterations cause a higher degree of product terms thereby increasing the workload of deciphering the secret keys. This design method can make the cryptanalysis equation fail to meet the basic conditions of the divide-and-conquer attack, so the secret keys cannot be deciphered further by the divide-and-conquer attack.

5.1. *Design of 3-D SCSC-NNS*

A nominal matrix of nonlinear functions is designed to satisfy as many invalid initial conditions as possible. The expression of the nominal matrix of nonlinear function F can be designed as

$$F = \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix} = \begin{pmatrix} \frac{a_{11}x_2(k)}{\mu} & \frac{a_{12}x_3(k)}{\mu} & \frac{a_{13}x_2(k)}{\mu} \\ a_{21} & \frac{a_{22}x_3(k)}{\mu} & a_{23} \\ a_{31} & a_{32} & \frac{a_{33}x_2(k)}{\mu} \end{pmatrix}, \tag{45}$$

where $\mu = 10^{10}$, $a_{11} = 0.09$, $a_{12} = -0.37$, $a_{13} = 0.1$, $a_{21} = -0.1$, $a_{22} = -0.18$, $a_{23} = 0.37$, $a_{31} = 0.27$, $a_{32} = -0.27$, $a_{33} = 0.19$, $||x_i(k)||/\mu < 1 (i = 2, 3)$. According to Eq. (45), the controlled chaotic system is designed as

$$\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{pmatrix} = \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix} \begin{pmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ g(\sigma x(k), \varepsilon) \end{pmatrix}, \tag{46}$$

where $g(\sigma x(k), \varepsilon) = \varepsilon \sin(\sigma x_1(k))$, $\varepsilon = 3.3 \times 10^8$, $\sigma = 2.5 \times 10^5$.

According to Eq. (46), the chaotic cipher is design as

$$\begin{cases} x_1(k+1) = F_{11}x_1(k) + F_{12}x_2(k) + F_{13}x_3(k) \\ x_2(k+1) = F_{21}p(k) + F_{22}x_2(k) + F_{23}x_3(k) \\ x_3(k+1) = F_{31}p(k) + F_{32}x_2(k) + F_{33}x_3(k) + \varepsilon \sin(\sigma p(k)) \end{cases}. \tag{47}$$

The ciphertext $p(k)$ in Eq. (47) is derived as

$$p(k) = \mathrm{mod}\left(\left\lfloor \frac{x_1(k)x_2(k)x_3(k)}{2^{27}} \right\rfloor, 2^8\right) \oplus m(k) \to m(k) \oplus p(k) = \mathrm{mod}\left(\left\lfloor \frac{x_1(k)x_2(k)x_3(k)}{2^{27}} \right\rfloor, 2^8\right), \tag{48}$$

where $k = 1, 2, \cdots$, $m(k)$ denotes the corresponding plaintext.

5.2. *Security analysis of 3-D SCSC-NNS*

Consider the chosen-ciphertext attack. According to Eq. (47), by setting $p(k) = 0$, one gets

$$\begin{cases} x_1(k+1) = F_{11}x_1(k) + F_{12}x_2(k) + F_{13}x_3(k) \\ x_2(k+1) = F_{22}x_2(k) + F_{23}x_3(k) \\ x_3(k+1) = F_{32}x_2(k) + F_{33}x_3(k) \end{cases} . \tag{49}$$

According to Eq. (45) and Eq. (49), the nonlinear iterative equation is derived as

$$\begin{cases} x_1(k+1) = \frac{a_{11}}{\mu}x_1(k)x_2(k) + \left(\frac{a_{12}}{\mu} + \frac{a_{13}}{\mu}\right)x_2(k)x_3(k) \\ x_2(k+1) = \frac{a_{22}}{\mu}x_2(k)x_3(k) + a_{23}x_3(k) \\ x_3(k+1) = a_{32}x_2(k) + \frac{a_{33}}{\mu}x_2(k)x_3(k) \end{cases} , \tag{50}$$

where $k = 1, 2, \cdots$.

5.2.1. *Situations of the first iteration*

By substituting $k = 0$ into Eq. (50), the first iteration result is derived as

$$\begin{cases} x_1(1) = \frac{a_{11}}{\mu}x_1(0)x_2(0) + \left(\frac{a_{12}}{\mu} + \frac{a_{13}}{\mu}\right)x_2(0)x_3(0) \\ x_2(1) = \frac{a_{22}}{\mu}x_2(0)x_3(0) + a_{23}x_3(0) \\ x_3(1) = a_{32}x_2(0) + \frac{a_{33}}{\mu}x_2(0)x_3(0) \end{cases} . \tag{51}$$

By substituting $x_1(0) = c_1, x_2(0) = c_2, x_3(0) = c_3$ into Eq. (51), one has

$$\begin{cases} x_1(1) = (\frac{a_{11}}{\mu})c_1c_2 + \left(\frac{a_{12}}{\mu} + \frac{a_{13}}{\mu}\right)c_2c_3 \\ x_2(1) = \frac{a_{22}}{\mu}c_2c_3 + a_{23}c_3 \\ x_3(1) = a_{32}c_2 + \frac{a_{33}}{\mu}c_2c_3 \end{cases} . \tag{52}$$

According to the seven initial conditions given in Eq. (11), with the chosen-ciphertext attack, the corresponding plaintext $m_i(1)(i = 1, 2, \cdots, 7)$ can be obtained by substituting $p_i(1) = 0(i = 1, 2, \cdots, 7)$ into Eq. (48). According to Eq. (52), consider the first iteration as follows:

(1) By substituting the first initial condition $(c_1, 0, 0)$ into Eq. (52), one obtains $x_1(1) = x_2(1) = x_3(1) = 0$, satisfying $x_1(1)x_2(1)x_3(1) = 0$. According to Eq. (48), one can see that $(c_1, 0, 0)$ is an invalid initial condition.

(2) By substituting the second initial condition $(0, c_2, 0)$ into Eq. (52), one obtains $x_1(1) = x_2(1) = 0$, $x_3(1) = a_{32}c_2 \neq 0$, satisfying $x_1(1)x_2(1)x_3(1) = 0$. According to Eq. (48), one can see that $(0, c_2, 0)$ is an invalid initial condition.

(3) By substituting the third initial condition $(0, 0, c_3)$ into Eq. (52), one obtains $x_1(1) = x_3(1) = 0$, $x_2(1) = a_{23}c_3 \neq 0$, satisfying $x_1(1)x_2(1)x_3(1) = 0$. According to Eq. (48), one can see that $(0, 0, c_3)$ is an invalid initial condition.

(4) By substituting the fourth initial condition $(c_1, c_2, 0)$ into Eq. (52), one obtains $x_1(1) = (a_{11}/\mu)c_1c_2 \neq 0$, $x_2(1) = 0$, $x_3(1) = a_{32}c_2 \neq 0$, satisfying $x_1(1)x_2(1)x_3(1) = 0$. According to Eq. (48), one can see that $(c_1, c_2, 0)$ is an invalid initial condition.

(5) By substituting the fifth initial condition $(c_1, 0, c_3)$ into Eq. (52), one obtains $x_1(1) = x_3(1) = 0$, $x_2(1) = a_{23}c_3 \neq 0$, satisfying $x_1(1)x_2(1)x_3(1) = 0$. According to Eq. (48), one can see that $(c_1, 0, c_3)$ is an invalid initial condition.

(6) By substituting the sixth initial condition $(0, c_2, c_3)$ into Eq. (52), one gets

$$\begin{cases} x_1(1) = \left(\frac{a_{12}}{\mu} + \frac{a_{13}}{\mu}\right)c_2c_3 \neq 0 \\ x_2(1) = \frac{a_{22}}{\mu}c_2c_3 + a_{23}c_3 \neq 0 \\ x_3(1) = a_{32}c_2 + \frac{a_{33}}{\mu}c_2c_3 \neq 0 \end{cases} , \tag{53}$$

which satisfies

$$x_1(1)x_2(1)x_3(1) = \left(\left(\frac{a_{12}}{\mu} + \frac{a_{13}}{\mu}\right) c_2c_3\right) \times \left(\frac{a_{22}}{\mu}c_2c_3 + a_{23}c_3\right) \times \left(a_{32}c_2 + \frac{a_{33}}{\mu}c_2c_3\right) \neq 0. \tag{54}$$

According to Eq. (48), one can see that $(0, c_2, c_3)$ is a valid initial condition.

(7) By substituting the seventh initial condition (c_1, c_2, c_3) into Eq. (52), one gets

$$\begin{cases} x_1(1) = \frac{a_{11}}{\mu}c_1c_2 + \left(\frac{a_{12}}{\mu} + \frac{a_{13}}{\mu}\right) c_2c_3 \neq 0 \\ x_2(1) = \frac{a_{22}}{\mu}c_2c_3 + a_{23}c_3 \neq 0 \\ x_3(1) = a_{32}c_2 + \frac{a_{33}}{\mu}c_2c_3 \neq 0 \end{cases}, \tag{55}$$

which satisfies

$$x_1(1)x_2(1)x_3(1) = \left(\frac{a_{11}}{\mu}c_1c_2 + \left(\frac{a_{12}}{\mu} + \frac{a_{13}}{\mu}\right) c_2c_3\right) \times \left(\frac{a_{22}}{\mu}c_2c_3 + a_{23}c_3\right) \times \left(a_{32}c_2 + \frac{a_{33}}{\mu}c_2c_3\right) \neq 0. \tag{56}$$

According to Eq. (48), one can see that (c_1, c_2, c_3) is a valid initial condition.

Since only valid initial conditions are considered, by substituting Eq. (53) and Eq. (55) into Eq. (48), one obtains

$$\begin{cases} m_6(1) = \bmod\left(\left\lfloor\left(\left(\frac{a_{12}}{\mu} + \frac{a_{13}}{\mu}\right) c_2c_3\right) \times \left(\frac{a_{22}}{\mu}c_2c_3 + a_{23}c_3\right) \times \left(a_{32}c_2 + \frac{a_{33}}{\mu}c_2c_3\right)\Big/2^{27}\right\rfloor, 2^8\right) \\ m_7(1) = \bmod\left(\left\lfloor\left(\frac{a_{11}}{\mu}c_1c_2 + \left(\frac{a_{12}}{\mu} + \frac{a_{13}}{\mu}\right) c_2c_3\right) \times \left(\frac{a_{22}}{\mu}c_2c_3 + a_{23}c_3\right) \times \left(a_{32}c_2 + \frac{a_{33}}{\mu}c_2c_3\right)\Big/2^{27}\right\rfloor, 2^8\right) \end{cases}. \tag{57}$$

5.2.2. *Situations of the second iteration*

By substituting $k = 1$ into Eq. (50), the second iteration result is derived as

$$\begin{cases} x_1(2) = \frac{a_{11}}{\mu}x_1(1)x_2(1) + \left(\frac{a_{12}}{\mu} + \frac{a_{13}}{\mu}\right) x_2(1)x_3(1) \\ x_2(2) = \frac{a_{22}}{\mu}x_2(1)x_3(1) + a_{23}x_3(1) \\ x_3(2) = a_{32}x_2(1) + \frac{a_{33}}{\mu}x_2(1)x_3(1) \end{cases}. \tag{58}$$

When the initial condition is determined to be invalid in the first iteration, it is still invalid after several iterations. Thus, one can see that $(0, c_2, c_3)$ and (c_1, c_2, c_3) are valid initial conditions in the second iteration. Consider the valid initial condition $(0, c_2, c_3)$. By substituting Eq. (53) into Eq. (58), the second iteration result is obtained, as

$$\begin{cases} x_1(2) = \frac{a_{11}}{\mu}\left(\frac{a_{12}}{\mu} + \frac{a_{13}}{\mu}\right) c_2c_3 \left(\frac{a_{22}}{\mu}c_2c_3 + a_{23}c_3\right) \\ \qquad + \left(\frac{a_{12}}{\mu} + \frac{a_{13}}{\mu}\right) \times \left(\frac{a_{22}}{\mu}c_2c_3 + a_{23}c_3\right) \times \left(a_{32}c_2 + \frac{a_{33}}{\mu}c_2c_3\right) \\ x_2(2) = \frac{a_{22}}{\mu}\left(\frac{a_{22}}{\mu}c_2c_3 + a_{23}c_3\right) \times \left(a_{32}c_2 + \frac{a_{33}}{\mu}c_2c_3\right) + a_{23}\left(a_{32}c_2 + \frac{a_{33}}{\mu}c_2c_3\right) \\ x_3(2) = a_{32}\left(\frac{a_{22}}{\mu}c_2c_3 + a_{23}c_3\right) + \frac{a_{33}}{\mu}\left(\frac{a_{22}}{\mu}c_2c_3 + a_{23}c_3\right) \times \left(a_{32}c_2 + \frac{a_{33}}{\mu}c_2c_3\right) \end{cases}. \tag{59}$$

Consider the valid initial condition (c_1, c_2, c_3). By substituting Eq. (55) into Eq. (58), the second iteration result is obtained as

$$\begin{cases} x_1(2) = \frac{a_{11}}{\mu}\left(\frac{a_{11}}{\mu}c_1c_2 + \left(\frac{a_{12}}{\mu} + \frac{a_{13}}{\mu}\right) c_2c_3\right) \times \left(\frac{a_{22}}{\mu}c_2c_3 + a_{23}c_3\right) \\ \qquad + \left(\frac{a_{12}}{\mu} + \frac{a_{13}}{\mu}\right) \times \left(\frac{a_{22}}{\mu}c_2c_3 + a_{23}c_3\right) \times \left(a_{32}c_2 + \frac{a_{33}}{\mu}c_2c_3\right) \\ x_2(2) = \frac{a_{22}}{\mu}\left(\frac{a_{22}}{\mu}c_2c_3 + a_{23}c_3\right) \times \left(a_{32}c_2 + \frac{a_{33}}{\mu}c_2c_3\right) + a_{23}\left(a_{32}c_2 + \frac{a_{33}}{\mu}c_2c_3\right) \\ x_3(2) = a_{32}\left(\frac{a_{22}}{\mu}c_2c_3 + a_{23}c_3\right) + \frac{a_{33}}{\mu}\left(\frac{a_{22}}{\mu}c_2c_3 + a_{23}c_3\right) \times \left(a_{32}c_2 + \frac{a_{33}}{\mu}c_2c_3\right) \end{cases}. \tag{60}$$

Since only valid initial conditions are considered, by substituting Eqs. (59)-(60) into Eq. (48), one has

$$\begin{cases} m_6(2) = \operatorname{mod}\left(\left\lfloor \begin{array}{l} \left(\begin{array}{l} \frac{a_{11}}{\mu}\left(\frac{a_{12}}{\mu}+\frac{a_{13}}{\mu}\right)c_2c_3\left(\frac{a_{22}}{\mu}c_2c_3+a_{23}c_3\right) \\ +\left(\frac{a_{12}}{\mu}+\frac{a_{13}}{\mu}\right)\times\left(\frac{a_{22}}{\mu}c_2c_3+a_{23}c_3\right)\times\left(a_{32}c_2+\frac{a_{33}}{\mu}c_2c_3\right) \end{array} \right) \\ \times\left(\frac{a_{22}}{\mu}\left(\frac{a_{22}}{\mu}c_2c_3+a_{23}c_3\right)\times\left(a_{32}c_2+\frac{a_{33}}{\mu}c_2c_3\right)+a_{23}\left(a_{32}c_2+\frac{a_{33}}{\mu}c_2c_3\right)\right) \\ \times\left(a_{32}\left(\frac{a_{22}}{\mu}c_2c_3+a_{23}c_3\right)+\frac{a_{33}}{\mu}\left(\frac{a_{22}}{\mu}c_2c_3+a_{23}c_3\right)\times\left(a_{32}c_2+\frac{a_{33}}{\mu}c_2c_3\right)\right)\Big/2^{27} \end{array} \right\rfloor, 2^8 \right) \\ m_7(2) = \operatorname{mod}\left(\left\lfloor \begin{array}{l} \left(\begin{array}{l} \frac{a_{11}}{\mu}\left(\frac{a_{11}}{\mu}c_1c_2+\left(\frac{a_{12}}{\mu}+\frac{a_{13}}{\mu}\right)c_2c_3\right)\times\left(\frac{a_{22}}{\mu}c_2c_3+a_{23}c_3\right) \\ +\left(\frac{a_{12}}{\mu}+\frac{a_{13}}{\mu}\right)\times\left(\frac{a_{22}}{\mu}c_2c_3+a_{23}c_3\right)\times\left(a_{32}c_2+\frac{a_{33}}{\mu}c_2c_3\right) \end{array} \right) \\ \times\left(\frac{a_{22}}{\mu}\left(\frac{a_{22}}{\mu}c_2c_3+a_{23}c_3\right)\times\left(a_{32}c_2+\frac{a_{33}}{\mu}c_2c_3\right)+a_{23}\left(a_{32}c_2+\frac{a_{33}}{\mu}c_2c_3\right)\right) \\ \times\left(a_{32}\left(\frac{a_{22}}{\mu}c_2c_3+a_{23}c_3\right)+\frac{a_{33}}{\mu}\left(\frac{a_{22}}{\mu}c_2c_3+a_{23}c_3\right)\times\left(a_{32}c_2+\frac{a_{33}}{\mu}c_2c_3\right)\right)\Big/2^{27} \end{array} \right\rfloor, 2^8 \right) \end{cases}. \tag{61}$$

5.2.3. *Security analysis by using DCA-TMNCIC*

According to Eq. (57) and Eq. (61), two nonlinear equations are obtained in the first iteration, and the highest degree of the product term is 3. In the second iteration, four nonlinear equations are obtained, yet the highest degree of product term reaches 9. If the solution conditions are not satisfied, the third iteration is required so as to obtain six nonlinear equations. However, the highest degree of the product term will reach 21 at this time. Hence, it is concluded that more iterations will cause a higher degree of the product term, thereby increasing the workload of deciphering the secret keys in 3-D SCSC-NNS.

According to the results given by Eq. (57) and Eq. (61), one cannot set appropriate values of the initial conditions to realize the divide-and-conquer attack. If Eq. (57) and Eq. (61) can be expressed as

$$\begin{aligned} m_l(k) &= \operatorname{mod}\left(\left\lfloor f_l^{(k)}\left(c_1, c_2, c_3, a_{11}, a_{12}, a_{13}, a_{22}, a_{23}, a_{32}, a_{33}, 2^{27}, \mu\right)\right\rfloor, 2^8\right) \\ &= \operatorname{mod}\left(\left\lfloor c_q \times F_l^{(k)}\left(c_v, c_u, a_{11}, a_{12}, a_{13}, a_{22}, a_{23}, a_{32}, a_{33}, 2^{27}, \mu\right)\right\rfloor, 2^8\right) \end{aligned}, \tag{62}$$

where $k = 1, 2, 3, \cdots$, $l = 4, 7, 2^{27}, \mu$ are constants, c_v, c_u are invariant constants, and $q, v, u \in \{1, 2, 3\}$, $q \neq v \neq u$, then c_q can be used to realize the divide-and-conquer attack. However, this is actually impossible. For example, according to Eq. (57) and Eq. (61), there are only two sets of valid initial conditions, namely $c_1 = 0, c_2 \neq 0, c_3 \neq 0$ and $c_1 \neq 0, c_2 \neq 0, c_3 \neq 0$. The following process only needs to examine all the possibilities of these two sets of valid initial conditions, and other invalid initial conditions need not be considered.

In the first case, let $c_1 \neq 0, c_2 \neq 0, c_3 \neq 0$. According to Eq. (57) and Eq. (61), the requirement of Eq. (62) cannot be met.

In the second case, let $c_1 = 0, c_2 = 1, c_3 \neq 0$. According to Eq. (57) and Eq. (61), one gets

$$\begin{cases} m_6(1) = \operatorname{mod}\left(\left\lfloor\left(\left(\frac{a_{12}}{\mu}+\frac{a_{13}}{\mu}\right)c_3\right)\times\left(\frac{a_{22}}{\mu}c_3+a_{23}c_3\right)\times\left(a_{32}+\frac{a_{33}}{\mu}c_3\right)\Big/2^{27}\right\rfloor, 2^8\right) \\ m_7(1) = \operatorname{mod}\left(\left\lfloor\left(\left(\frac{a_{12}}{\mu}+\frac{a_{13}}{\mu}\right)c_3\right)\times\left(\frac{a_{22}}{\mu}c_3+a_{23}c_3\right)\times\left(a_{32}+\frac{a_{33}}{\mu}c_3\right)\Big/2^{27}\right\rfloor, 2^8\right) \\ m_6(2) = \operatorname{mod}\left(\left\lfloor \begin{array}{l} \left(\frac{a_{11}}{\mu}\left(\frac{a_{12}}{\mu}+\frac{a_{13}}{\mu}\right)c_3\left(\frac{a_{22}}{\mu}c_3+a_{23}c_3\right)+\left(\frac{a_{12}}{\mu}+\frac{a_{13}}{\mu}\right)\times\left(\frac{a_{22}}{\mu}c_3+a_{23}c_3\right)\times\left(a_{32}+\frac{a_{33}}{\mu}c_3\right)\right) \\ \times\left(\frac{a_{22}}{\mu}\left(\frac{a_{22}}{\mu}c_3+a_{23}c_3\right)\times\left(a_{32}+\frac{a_{33}}{\mu}c_3\right)+a_{23}\left(a_{32}+\frac{a_{33}}{\mu}c_3\right)\right) \\ \times\left(a_{32}\left(\frac{a_{22}}{\mu}c_3+a_{23}c_3\right)+\frac{a_{33}}{\mu}\left(\frac{a_{22}}{\mu}c_3+a_{23}c_3\right)\times\left(a_{32}+\frac{a_{33}}{\mu}c_3\right)\right)\Big/2^{27} \end{array}\right\rfloor, 2^8\right) \\ m_7(2) = \operatorname{mod}\left(\left\lfloor \begin{array}{l} \left(\frac{a_{11}}{\mu}\left(\left(\frac{a_{12}}{\mu}+\frac{a_{13}}{\mu}\right)c_3\right)\times\left(\frac{a_{22}}{\mu}c_3+a_{23}c_3\right)+\left(\frac{a_{12}}{\mu}+\frac{a_{13}}{\mu}\right)\times\left(\frac{a_{22}}{\mu}c_3+a_{23}c_3\right)\times\left(\frac{a_{33}}{\mu}c_3\right)\right) \\ \times\left(\frac{a_{22}}{\mu}\left(\frac{a_{22}}{\mu}c_3+a_{23}c_3\right)\times\left(a_{32}+\frac{a_{33}}{\mu}c_3\right)+a_{23}\left(a_{32}+\frac{a_{33}}{\mu}c_3\right)\right) \\ \times\left(a_{32}\left(\frac{a_{22}}{\mu}c_3+a_{23}c_3\right)+\frac{a_{33}}{\mu}\left(\frac{a_{22}}{\mu}c_3+a_{23}c_3\right)\times\left(a_{32}+\frac{a_{33}}{\mu}c_3\right)\right)\Big/2^{27} \end{array}\right\rfloor, 2^8\right) \end{cases}. \tag{63}$$

However, according to Eq. (63), it is impossible to extract all c_3 , so that the requirement of Eq. (62) cannot be met.

In the third case, let $c_1 = 0, c_2 \neq 0, c_3 = 1$. According to Eq. (57) and Eq. (61), it is obvious that the requirement of Eq. (62) cannot be met.

In summary, no matter how one selects the valid initial conditions, Eq. (57) and Eq. (61) cannot be guaranteed to meet the requirement of Eq. (62). Therefore, Eq. (57) and Eq. (61) cannot satisfy the basic conditions of the divide-and-conquer attack. Thus, it can be concluded that 3-D SCSC-NNS is safe against the combinational effect of chosen-ciphertext attack and divide-and-conquer attack.

6. n-D SCSC-SM and its security analysis

In Section 5, the design method of 3-D SCSC-NNS was described. However, this design method is suitable only for the case of $n = 3, 4$, with low dimensionality. For the case of $n = 5, 6, 7, 8$, with high dimensionality, the difficulty and complexity of designing the chaotic cipher will be greatly increased. As a remedy, in this section an improved scheme, n-D SCSC-SM ($n = 3, 4, 5, 6, 7, 8$), is proposed, which can effectively resist DCA-TMNCIC.

6.1. *Design of n-D SCSC-SM*

Consider a 4-D SCSC-SM. The iterative equation is given by

$$\begin{cases} x_1(k+1) = a_{11}x_1(k) + a_{12}x_2(k) + a_{13}x_3(k) + a_{14}x_4(k) \\ x_2(k+1) = a_{21}p(k) + a_{22}x_2(k) + a_{23}x_3(k) + a_{24}x_4(k) \\ x_3(k+1) = a_{31}p(k) + a_{32}x_2(k) + a_{33}x_3(k) + a_{34}x_4(k) \\ x_4(k+1) = a_{41}p(k) + a_{42}x_2(k) + a_{43}x_3(k) + a_{44}x_4(k) + \varepsilon\sin(\sigma p(k)) \end{cases}. \tag{64}$$

The ciphertext $p(k)$ in Eq. (64) is derived, as

$$p(k) = \bmod\left(\lfloor \sin(x_1(k)) \times \xi \rfloor, 2^8\right) \oplus m(k) \to m(k) \oplus p(k) = \bmod\left(\lfloor \sin(x_1(k)) \times \xi \rfloor, 2^8\right), \tag{65}$$

where $k = 0, 1, 2, 3, 4 \cdots$, $\xi = 10^8$ denotes a constant, $m(k)$ denotes the corresponding plaintext, $x_i(k)(i = 1, 2, 3, 4)$ denotes the chaotic variable, and $a_{11} = 0.1033$, $a_{12} = -0.6367$, $a_{13} = 0.4133$, $a_{14} = -0.0067$, $a_{21} = -0.4833$, $a_{22} = -0.2033$, $a_{23} = 0.5667$, $a_{24} = 0.1467$, $a_{31} = -0.12$, $a_{32} = -0.58$, $a_{33} = 0.37$, $a_{34} = 0.49$, $a_{41} = 0.02$, $a_{42} = -0.44$, $a_{43} = 0.63$, $a_{44} = 0.09$, $\varepsilon = 5.9 \times 10^8$, $\sigma = 3.3 \times 10^{10}$ denote the secret keys.

With the same method, a more general n-D SCSC-SM ($n = 3, 4, 5, 6, 7, 8$) can be designed, and the general form of its mathematical expression is

$$\begin{cases} x_1(k+1) = f_1(x_1(k), x_2(k), \cdots, x_n(k)) \\ x_2(k+1) = f_2(p(k), x_2(k), \cdots, x_n(k)) \\ \qquad \vdots \\ x_{n-1}(k+1) = f_{n-1}(p(k), x_2(k), \cdots, x_n(k)) \\ x_n(k+1) = f_n(p(k), x_2(k), \cdots, x_n(k)) \end{cases}, \tag{66}$$

where $n = 3, 4, 5, 6, 7, 8$. The ciphertext $p(k)$ in Eq. (66) is derived as

$$p(k) = \bmod\left(\lfloor \sin(x_i(k)) \times \xi \rfloor, 2^8\right) \oplus m(k) \to m(k) \oplus p(k) = \bmod\left(\lfloor \sin(x_i(k)) \times \xi \rfloor, 2^8\right), \tag{67}$$

where $k = 0, 1, 2, 3, 4 \cdots$, $i = 1, 2, \cdots, 8$, and $m(k)$ denotes the corresponding plaintext. Note that the value range of the constant ξ is $10^5 \leq \xi \leq 10^{14}$, and the value of ξ has great impact on the statistical characteristics of the chaotic sequence and the sensitivity of the secret keys. A larger ξ results in a higher sensitivity of the secret keys.

6.2. *Security analysis of 4-D SCSC-SM*

Take the 4-D SCSC-SM as an example and consider the chosen-ciphertext attack. According to Eq. (64), by setting $p(k) = 0$, one gets

$$\begin{cases} x_1(k+1) = a_{11}x_1(k) + a_{12}x_2(k) + a_{13}x_3(k) + a_{14}x_4(k) \\ x_2(k+1) = a_{22}x_2(k) + a_{23}x_3(k) + a_{24}x_4(k) \\ x_3(k+1) = a_{32}x_2(k) + a_{33}x_3(k) + a_{34}x_4(k) \\ x_4(k+1) = a_{42}x_2(k) + a_{43}x_3(k) + a_{44}x_4(k) \end{cases}. \tag{68}$$

6.2.1. *Situations of the first iteration*

By substituting $k = 0$ into Eq. (68), the first iteration results in

$$\begin{cases} x_1(1) = a_{11}x_1(0) + a_{12}x_2(0) + a_{13}x_3(0) + a_{14}x_4(0) \\ x_2(1) = a_{22}x_2(0) + a_{23}x_3(0) + a_{24}x_4(0) \\ x_3(1) = a_{32}x_2(0) + a_{33}x_3(0) + a_{34}x_4(0) \\ x_4(1) = a_{42}x_2(0) + a_{43}x_3(0) + a_{44}x_4(0) \end{cases}. \tag{69}$$

By substituting $x_1(0) = c_1, x_2(0) = c_2, x_3(0) = c_3, x_4(0) = c_4$ into Eq. (69), one has

$$\begin{cases} x_1(1) = a_{11}c_1 + a_{12}c_2 + a_{13}c_3 + a_{14}c_4 \\ x_2(1) = a_{22}c_2 + a_{23}c_3 + a_{24}c_4 \\ x_3(1) = a_{32}c_2 + a_{33}c_3 + a_{34}c_4 \\ x_4(1) = a_{42}c_2 + a_{43}c_3 + a_{44}c_4 \end{cases}. \tag{70}$$

The corresponding set of fifteen initial conditions is given by

$$(x_1(0), x_2(0), x_3(0), x_4(0)) \in \left\{ \begin{array}{l} (c_1,0,0,0), (0,c_2,0,0), (0,0,c_3,0), (0,0,0,c_4), (c_1,c_2,0,0), \\ (c_1,0,c_3,0), (c_1,0,0,c_4), (0,c_2,c_3,0), (0,c_2,0,c_4), (0,0,c_3,c_4), \\ (c_1,c_2,c_3,0), (c_1,c_2,0,c_4), (c_1,0,c_3,c_4), (0,c_2,c_3,c_4), (c_1,c_2,c_3,c_4) \end{array} \right\}, \tag{71}$$

where $c_i \neq 0 (i = 1, 2, 3, 4)$.

According to the fifteen initial conditions given in Eq. (71), with the chosen-ciphertext attack, the corresponding plaintext $m_i(k) (i = 1, 2, \cdots, 15)$ can be obtained by substituting $p_i(k) = 0 (i = 1, 2, \cdots, 15)$ into Eq. (65). Note that the fifteen initial conditions in Eq. (71) are all valid initial conditions in the first iteration. Therefore, the first iteration results corresponding to the fifteen initial conditions are substituted into Eq. (65), yielding

$$\begin{cases} m_1(1) = \bmod\left(\lfloor \sin(a_{11}c_1) \times \xi \rfloor, 2^8\right) \\ m_2(1) = \bmod\left(\lfloor \sin(a_{12}c_2) \times \xi \rfloor, 2^8\right) \\ \quad\vdots \qquad\qquad\qquad \vdots \\ m_{14}(1) = \bmod\left(\lfloor \sin(a_{12}c_2 + a_{13}c_3 + a_{14}c_4) \times \xi \rfloor, 2^8\right) \\ m_{15}(1) = \bmod\left(\lfloor \sin(a_{11}c_1 + a_{12}c_2 + a_{13}c_3 + a_{14}c_4) \times \xi \rfloor, 2^8\right) \end{cases}, \tag{72}$$

6.2.2. *Situations of the second iteration*

By substituting $k = 1$ into Eq. (68), the second iteration result is obtained as

$$\begin{cases} x_1(2) = a_{11}x_1(1) + a_{12}x_2(1) + a_{13}x_3(1) + a_{14}x_4(1) \\ x_2(2) = a_{22}x_2(1) + a_{23}x_3(1) + a_{24}x_4(1) \\ x_3(2) = a_{32}x_2(1) + a_{33}x_3(1) + a_{34}x_4(1) \\ x_4(2) = a_{42}x_2(1) + a_{43}x_3(1) + a_{44}x_4(1) \end{cases}. \tag{73}$$

Since the fifteen initial conditions in Eq. (71) are all valid initial conditions in the second iteration, the second iteration results corresponding to the fifteen initial conditions are substituted into Eq. (65), yielding

$$\begin{cases} m_1(2) = \bmod\left(\lfloor \sin((a_{11}^2)c_1) \times \xi \rfloor, 2^8\right) \\ m_2(2) = \bmod\left(\lfloor \sin((a_{11}a_{12}+a_{12}a_{22}+a_{13}a_{32}+a_{14}a_{42})c_2) \times \xi \rfloor, 2^8\right) \\ \quad\vdots \qquad\qquad\qquad\qquad\qquad \vdots \\ m_{14}(2) = \bmod\left(\left\lfloor \begin{array}{l} \sin(a_{11}(a_{12}c_2+a_{13}c_3+a_{14}c_4)+a_{12}(a_{22}c_2+a_{23}c_3+a_{24}c_4) \\ +a_{13}(a_{32}c_2+a_{33}c_3+a_{34}c_4)+a_{14}(a_{42}c_2+a_{43}c_3+a_{44}c_4)) \times \xi \end{array} \right\rfloor, 2^8\right) \\ m_{15}(2) = \bmod\left(\left\lfloor \begin{array}{l} \sin(a_{11}(a_{11}c_1+a_{12}c_2+a_{13}c_3+a_{14}c_4) + a_{12}(a_{22}c_2+a_{23}c_3+a_{24}c_4) \\ +a_{13}(a_{32}c_2+a_{33}c_3+a_{34}c_4)+a_{14}(a_{42}c_2+a_{43}c_3+a_{44}c_4)) \times \xi \end{array} \right\rfloor, 2^8\right) \end{cases}. \tag{74}$$

6.2.3. *Security analysis by using DCA-TMNCIC*

According to the results given by Eq. (72) and Eq. (74), one cannot set an appropriate value of the initial condition $c_i (i = 1, 2, 3, 4)$ to realize the divide-and-conquer attack.

Note that Eq. (72) and Eq. (74) can be further expressed as

$$m_l(k) = \mathrm{mod}\left(\left\lfloor \sin\left(f_l^{(k)}(c_1, c_2, c_3, c_4, a_{11}, a_{12}, a_{13}, a_{14}, a_{22}, a_{23}, a_{24}, a_{32}, a_{33}, a_{34}, a_{42}, a_{43}, a_{44})\right) \times \xi \right\rfloor, 2^8\right), \qquad (75)$$

where $k = 1, 2, 3, \cdots, l = 1, 2, \cdots, 15$ and ξ is a constant. According to Eq. (75), since the initial conditions are involved in the sine function, there is an implicit relationship between $m_l(k)$ and $c_i (i = 1, 2, 3, 4)$. Thus, it is impossible to extract $c_i (i = 1, 2, 3, 4)$ from $\sin(\cdot)$, set an appropriate value of $c_i (i = 1, 2, 3, 4)$ and use the divide-and-conquer attack to decipher the secret keys. In the case of a failed divide-and-conquer attack, only using the chosen-ciphertext attack cannot obtain all the information of the secret key expressions, so that the secret keys cannot be further deciphered. Therefore, it can be concluded that 4-D SCSC-SM is secure against the combinational effect of chosen-ciphertext attack and divide-and-conquer attack. Similarly, the same conclusion can be drawn for n-D SCSC-SM $(n = 3, 5, 6, 7, 8)$.

7. Conclusions

In this study, a cryptanalysis method is studied, which combines chosen-ciphertext attack and DCA-TMNCIC for n-D SCSC-2 and n-D SCSC-3 $(n = 3, 4, 5, 6, 7, 8)$. Compared with the DCA-TSNCIC proposed in [Lin *et al.*, 2018], listed in Table 3, the attack intensity of DCA-TMNCIC is stronger. Note that the attack intensity grows with the numbers of valid initial conditions and nonlinear equations obtained in the same iteration. More valid initial conditions mean that in each iteration more nonlinear equations are obtained, consequently less iterations are needed to decipher the secret keys. Less iterations will make a lower degree of the product terms in the nonlinear equations and decrease the workload of deciphering the secret keys. However, since the whole solution process is to solve nonlinear equations, this task is more challenging. In general, when DCA-TMNCIC is used for security analysis of n-D SCSC-2 and n-D SCSC-3 $(n = 3, 4, 5, 6, 7, 8)$, the secret keys can be deciphered only after two or three iterations. In contrast, DCA-TSNCIC cannot decipher the secret keys of n-D SCSC-2 and n-D SCSC-3 $(n = 4, 5, 6, 7, 8)$ even with a larger number of iterations. On this basis, several new improved chaotic cipher schemes are proposed, including 3-D SCSC-NNS and n-D SCSC-SM $(n = 3, 4, 5, 6, 7, 8)$. These improved schemes can completely resist the divide-and-conquer attack, so as to ensure the security against the combinational effect of chosen-ciphertext attack and divide-and-conquer attack.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (No.2016YFB0800401), the National Natural Science Foundation of China (No. 61532020, 61671161). We thank Prof. Guanrong Chen of City University of Hong Kong for assistance and comments that greatly improved the manuscript.

References

Chen, B., Yu, S., Chen, P., Xiao, L. & Lü, J. [2020] "Design and Virtex-7-based implementation of video chaotic secure communications," *International Journal of Bifurcation and Chaos* **30**, 2050075.

Chen, J., Zhu, Z., Fu, C., Zhang, L. & Zhang, Y. [2015] "An efficient image encryption scheme using lookup table-based confusion and diffusion," *Nonlinear Dynamics* **81**, 1151–1166.

Chen, P., Yu, S., Chen, B., Xiao, L. & Lü, J. [2018] "Design and SOPC-based realization of a video chaotic secure communication scheme," *International Journal of Bifurcation and Chaos* **28**, 1850160.

Diab, H. [2018] "An efficient chaotic image cryptosystem based on simultaneous permutation and diffusion operations," *IEEE Access* **6**, 42227–42244.

Hu, G., Xiao, D., Wang, Y. & Li, X. [2017] "Cryptanalysis of a chaotic image cipher using latin square-based confusion and diffusion," *Nonlinear Dynamics* **88**, 1305–1316.

Huang, L., Cai, S., Xiao, M. & Xiong, X. [2018] "A simple chaotic map-based image encryption system using both plaintext related permutation and diffusion," *Entropy* **20**, 535.

Li, C., Lin, D., Lü, J. & Hao, F. [2018a] "Cryptanalyzing an image encryption algorithm based on autoblocking and electrocardiography," *IEEE Multimedia* **25**, 46–56.

Li, M., Lu, D., Wen, W., Ren, H. & Zhang, Y. [2018b] "Cryptanalyzing a color image encryption scheme based on hybrid hyper-chaotic system and cellular automata," *IEEE Access* **6**, 47102–47111.

Lin, Z., Yu, S., Feng, X. & Lü, J. [2018] "Cryptanalysis of a chaotic stream cipher and its improved scheme," *International Journal of Bifurcation and Chaos* **28**, 1850086.

Lin, Z., Yu, S., Lü, J., Cai, S. & Chen, G. [2015] "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* **25**, 1203–1216.

Matthews, R. [1989] "On the derivation of a chaotic encryption algorithm," *Cryptologia* **13**, 29–42.

Mollaeefar, M., Sharif, A. & Nazari, M. [2017] "A novel encryption scheme for colored image based on high level chaotic maps," *Multimedia Tools and Applications* **76**, 607–629.

Niyat, A. Y., Moattar, M. H. & Torshiz, M. N. [2017] "Color image encryption based on hybrid hyper-chaotic system and cellular automata," *Optics and Lasers in Engineering* **90**, 225–237.

Parvin, Z., Seyedarabi, H. & Shamsi, M. [2016] "A new secure and sensitive image encryption scheme based on new substitution with chaotic function," *Multimedia Tools and Applications* **75**, 10631–10648.

Shafique, A. & Shahid, J. [2018] "Novel image encryption cryptosystem based on binary bit planes extraction and multiple chaotic maps," *The European Physical Journal Plus* **133**, 331.

Shahzadi, R., Anwar, S. M., Qamar, F., Ali, M. & Rodrigues, J. J. [2019] "Chaos based enhanced RC5 algorithm for security and integrity of clinical images in remote health monitoring," *IEEE Access* **7**, 52858–52870.

Song, C. & Qiao, Y. [2015] "A novel image encryption algorithm based on DNA encoding and spatiotemporal chaos," *Entropy* **17**, 6954–6968.

Wen, H. & Yu, S. [2019] "Cryptanalysis of an image encryption cryptosystem based on binary bit planes extraction and multiple chaotic maps," *The European Physical Journal Plus* **134**, 337.

Wen, H., Yu, S. & Lü, J. [2019] "Breaking an image encryption algorithm based on DNA encoding and spatiotemporal chaos," *Entropy* **21**, 246.

Wu, J., Liao, X. & Yang, B. [2018] "Cryptanalysis and enhancements of image encryption based on three-dimensional bit matrix permutation," *Signal Processing* **142**, 292–300.

Wu, X., Zhu, B., Hu, Y. & Ran, Y. [2017] "A novel color image encryption scheme using rectangular transform-enhanced chaotic tent maps," *IEEE Access* **5**, 6429–6436.

Xu, M. & Tian, Z. [2019] "A novel image cipher based on 3D bit matrix and latin cubes," *Information Sciences* **478**, 1–14.

Ye, G. & Huang, X. [2015] "An image encryption algorithm based on autoblocking and electrocardiography," *IEEE Multimedia* **23**, 64–71.

Ye, G., Jiao, K., Wu, H., Pan, C. & Huang, X. [2020a] "An asymmetric image encryption algorithm based on a fractional-order chaotic system and the rsa public-key cryptosystem," *International Journal of Bifurcation and Chaos* **30**, 2050233.

Ye, G., Pan, C., Dong, Y., Jiao, K. & Huang, X. [2020b] "A novel multi-image visually meaningful encryption algorithm based on compressive sensing and schur decomposition," *Transactions on Emerging Telecommunications Technologies* , e4071.

Zhang, W., Yu, H., Zhao, Y. & Zhu, Z. [2016] "Image encryption based on three-dimensional bit matrix permutation," *Signal Processing* **118**, 36–50.

Zhang, X., Wang, L., Zhou, Z. & Niu, Y. [2019] "A chaos-based image encryption technique utilizing hilbert curves and H-fractals," *IEEE Access* **7**, 74734–74746.

Zhang, Z. & Yu, S. [2019] "On the security of a latin-bit cube-based image chaotic encryption algorithm," *Entropy* **21**, 888.

Zhao, J., Wang, S., Chang, Y. & Li, X. [2015] "A novel image encryption scheme based on an improper fractional-order chaotic system," *Nonlinear Dynamics* **80**, 1721–1729.

Zhou, J., Zhou, N. & Gong, L. [2020] "Fast color image encryption scheme based on 3D orthogonal Latin squares and matching matrix," *Optics and Laser Technology* **131**, 106437.