Received October 16, 2018, accepted October 31, 2018, date of publication November 9, 2018,
date of current version December 18, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2879844

Digital Image Encryption Algorithm Based on Elliptic Curve Public Cryptosystem

XIAOQIANG ZHANG AND XUESONG WANG
School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China
Corresponding author: Xuesong Wang (wangxuesongcumt@163.com)

This work was supported by the Fundamental Research Funds for the Central Universities under Grant 2015QNA68.

ABSTRACT With the increasing use of media in communications, the content security of digital images attracts much attention in both the academia and the industry. Meanwhile, for the symmetric cryptosystem, the key transmission and management is burden on users. This paper proposes an asymmetric image encryption algorithm based on an elliptic curve cryptosystem (ECC). The sender and the recipient agree on an elliptic curve point based on the Diffie–Hellman public key sharing technique. First, to reduce the encryption times, the sender groups pixel values together and converts them into big integers. Second, the sender encrypts big integers with ECC and the chaotic system. Finally, the encrypted image is obtained from encrypted big integers. The proposed algorithm makes the key transmission and management relatively simple and secure. Simulation data show that the proposed algorithm exhibits both the strong security and the high efficiency.

INDEX TERMS Image encryption, big integer, elliptic curve cryptosystem (ECC), chaotic system.

I. INTRODUCTION

The digital image is an important means to deliver information in Internet, which is widely used in almost every field. Most of digital images involve business secrets and even national security. Internet development and multimedia easy distribution make the content security of images become an important issue for scientists and engineers.

The chaotic system is widely used in the field of image encryption for its extreme sensitivity to initial values and parameters, ergodicity, pseudo randomness, etc [1]. Many image encryption algorithms have been proposed based on chaotic system recently [2]–[7]. However, the low-dimensional chaotic sequences have the problems of short code period and low accuracy, which cannot guarantee the algorithm security. Researchers pay more attention to the high-dimensional chaotic encryption algorithms [8]–[13].

Elliptic Curve Cryptosystem (ECC) is an excellent asymmetric encryption algorithm, which depends on the difficulty of the Elliptic Curve Discrete Logarithm Problem (ECDLP). To encrypt the color image, Manish et al. proposed an image encryption algorithm based on DNA encoding and ECC [14]. The plain image is encoded using DNA encoding theory, and then the image encryption operation is performed by ECC. The encryption algorithm works well, but it can be improved. When all the pixels are black (pixel value 0), the corresponding encrypted image is itself. Generally speaking, a desirable image encryption algorithm should generate an unintelligible cipher image for any plain image. Wu *et al.* [15] proposed a color image encryption algorithm based on 4-dimensional cat map and ECC. The decrypted image of this algorithm is lossy for the compression technology used during the encryption process. Toughi *et al.* [16] proposed a color image encryption algorithm based on ECC and Advanced Encryption System. In their algorithm, ECC is used to generate the random sequence. Zhao and Zhang [17] proposed a color image encryption algorithm based on ECC and DNA encoding. In their algorithm, ECC is just used to encrypt the key information during the encryption process, not the pixel values of the plain image. Singh and Singh [18] proposed an image encryption algorithm based on ECC. In their algorithm, authors group the pixel values into some big integers to reduce the cryptographic operations, and then encrypt them with ECC. However, the quantity of cryptographic operations is also very large. Laiphrakpam and Khumanthem [19] proposed an image encryption algorithm based on ECC and chaotic system. In their algorithm, ECC is used to generate the random sequence to diffuse the plain image. Zhu and Zhang [20] proposed a mixed image element encryption algorithm based on ECC. In their algorithm, ECC is used to encrypt the filenames of mixed image elements. In total, most of these algorithms can be improved in terms of the security or efficiency.

To protect the content of digital images, this paper proposes an asymmetric image encryption algorithm based on ECC and chaotic system. Experimental results and algorithm analyses display that the proposed algorithm is desirable in terms of the security and efficiency.

The content structure of this paper is: Section II introduces ECC, chaotic system and secure hash algorithm in brief. A novel image encryption algorithm is designed with ECC in Section III. Three similar algorithms are described in Section IV. Experiments are carried out in Section V. Algorithm analyses are made in Section VI. Section VII makes the conclusions.

II. THEORETICAL PRINCIPLE

A. ECC

On the basis of the ECDLP, Koblitz and Miller proposed ECC in 1985. ECC provides a small and fast public key cryptosystem. As compared to the RSA cryptosystem, ECC provides the same level of security with smaller key size. An elliptic curve is the set of solutions (x, y) to

$$y^2 \equiv \left(x^3 + ax + b\right) \bmod p, \tag{1}$$

together with an extra point O, which is called the point at infinity [20]. In Eq. (1), a, b belong to the finite field $F_p = \{0, 1, \cdots, p-1\}$ and satisfy $\left(4a^3 + 27b^2\right) \bmod p \neq 0$, where p is a prime and more than 3.

To realize the ECC, two mathematical operations are described here, i.e., the point addition and point multiplication. In the ECC, these two operations are performed on the coordinate points of an elliptic curve.

(1) The point addition: to perform the addition of two coordinate points $P(x_1, y_1)$ and $Q(x_2, y_2)$ on an elliptic curve E, the following calculation is used,

$$P + Q = R(x_3, y_3), \tag{2}$$

where $x_3 = (\lambda^2 - x_1 - x_2) \bmod p$ and $y_3 = (\lambda(x_1 - x_3) - y_1) \bmod p$. If $P \neq Q$, $\lambda = (y_2 - y_1)/(x_2 - x_1)$; otherwise, $\lambda = (3x_1^2 + a)/2y_1$.

(2) The point multiplication: for any coordinate point $P(x_1, y_1)$ on an elliptic curve E, the point multiplication is defined by repeatedly performing $n-1$ times of the point addition operation for $P(x_1, y_1)$, i.e.,

$$n \times P = P + P + \cdots + P, \tag{3}$$

where "+" denotes the point addition operation. Many algorithms have been developed to perform the point multiplication swiftly [21].

B. CHAOTIC SYSTEM

The Piece-Wise Linear Chaotic Map (PWLCM) can be described by

$$x_{i+1} = F_q(x_i) = \begin{cases} \dfrac{x_i}{q} & 0 \leq x_i < q \\ \dfrac{x_i - q}{0.5 - q} & q \leq x_i < 0.5 \\ F_q(1 - x_i) & 0.5 \leq x_i < 1, \end{cases} \tag{4}$$

where $x_i \in [0, 1)$ and control parameter $q \in (0, 0.5)$ [22]. The PWLCM system has uniform invariant distribution and very excellent ergodicity, confusion and determinacy, so it can provide excellent random sequence to encrypt images.

C. SECURE HASH ALGORITHM

Secure Hash Algorithms (SHA) are a kind of hash functions, which is released by the National Institute of Standards and Technology. SHA is mainly applied to the integrity security services [23]. SHA-256 is a popular SHA, which outputs the message digest with the length of 256 bits.

The proposed algorithm uses SHA-256 to generate the initial value and control parameter of PWLCM. For the plain image, the proposed algorithm adopts SHA-256 to generate its 256-bit hash value K, which is divided into 8-bit blocks, i.e.,

$$K = k_1, k_2, \cdots, k_{32}. \tag{5}$$

The initial value $x_0 \in [0, 1)$ and control parameter $q \in (0, 0.5)$ of PWLCM can be calculated by

$$x_0 = \frac{k_1 \oplus k_2 \oplus \cdots k_{16}}{255}, \tag{6}$$

$$q = \frac{k_{17} \oplus k_{18} \oplus \cdots k_{32}}{510}, \tag{7}$$

where $\oplus$ denotes the exclusive OR (XOR) operation in the binary system.

III. PROPOSED IMAGE ENCRYPTION ALGORITHM

For easy description, the sender and recipient are Alice and Bob respectively. The following subsections describe the core technology of the proposed algorithm.

A. BOB'S KEY GENERATION

Bob generates his public key Q and private key d to encrypt the plain image and decrypt the encrypted image. The detailed steps are described as follows.

Step 1: choosing an elliptic curve

Bob chooses an elliptic curve $E(F_p)$ with the l-bit key length by setting the values of the parameters a, b and p. He selects a point N on $E(F_p)$ as the base point, whose order is T.

Step 2: generating the private and public keys

Bob randomly selects an integer d in $[1, T-1]$ as the private key and calculates the public key $Q = dN$.

Step 3: publishing the public key and chosen elliptic curve

Bob publishes the public key Q, the base point N, the order T and the parameters a, b and p of the chosen elliptic curve.

B. ALICE'S ENCRYPTION PROCESS

The image encryption flowchart of the proposed algorithm is shown in Fig. 1. Specific encryption steps are described as follows.

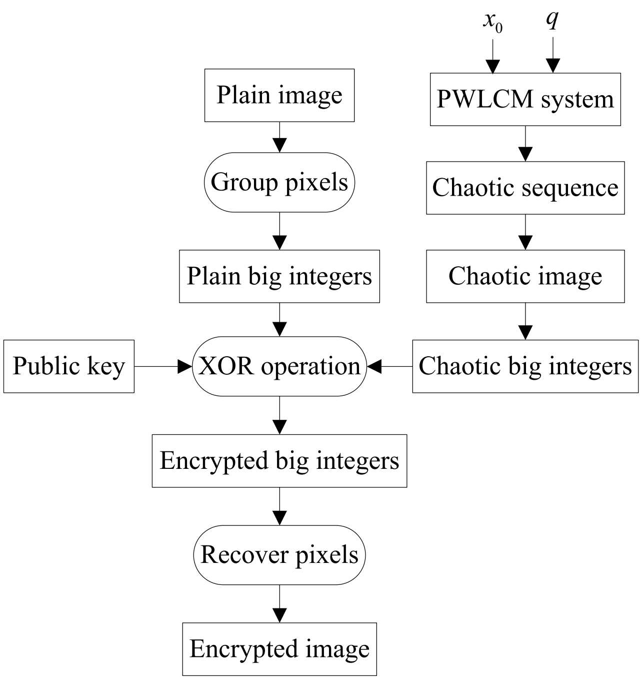

FIGURE 1. **The encryption flowchart of the proposed algorithm.**

Step 1: chaotic sequence generation

Let the plain image be I with the size $m \times n$. The initial value x_0 and control parameter q of PWLCM can be generated with the method described in Subsection II.C. Alice iterates Eq. (4) $m \times n$ times with x_0 and q, and obtains a chaotic sequence $X = \{x_i\}_{m \times n}$.

Step 2: chaotic image generation

If the computing precision of the computer is 10^{16}, Alice computes

$$y_i = mod\left(x_i \times 10^{16}, 256\right), \tag{8}$$

where mod(•) denotes the modulus after division, and $y_i \in Y_{m \times n}$. According to the element positions, Alice converts Y into a chaotic image C with the size $m \times n$.

Step 3: grouping pixels

For the gray image, the pixel value can be expressed with 8 bits in binary form. The key length of $E(F_p)$ is l bits. Therefore, Alice groups $l/8 - 1$ pixel values together to form a big integer with the $l - 8$ bit length. E.g., $l = 512$ bits, Alice groups $512/8 - 1 = 63$ pixel values together. The first big integer b_1 with the 504 bit length is

$$b_1 = I_{1,1}^b I_{1,2}^b \cdots I_{1,63}^b, \tag{9}$$

where $I_{1,1}^b, I_{1,2}^b, \cdots, I_{1,63}^b$ are the pixel values of I in the binary form. E.g., 63 pixel values are $I_{1,1} = 124, I_{1,2} = 45, \cdots, I_{1,63} = 245$ in the decimal form. Their binary forms are $I_{1,1}^b = 01111100$, $I_{1,2}^b = 00101101, \cdots, I_{1,63}^b = 11110101$ respectively. Therefore, $b_1 = I_{1,1}^b I_{1,2}^b \cdots I_{1,63}^b = 0111110000101101 \cdots 11110101$. b_1 is viewed as a big integer in the binary form, and its value in the decimal form can be calculated by $0 \times 2^{503} + 1 \times 2^{502} + 1 \times 2^{501} + \cdots + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$. For the plain image I, Alice can obtain $(m \times n)/63$ plain big integers $\{b_1, b_2, \cdots, b_{(m \times n)/63}\}$ in total. If mn isn't the multiple of 63, Alice can add several pixel values to meet this requirement. Similarly, for the chaotic image C, Alice can also obtain the chaotic big integers $\{r_1, r_2, \cdots, r_{(m \times n)/63}\}$.

Step 4: ECC encryption

After Alice receives Bob's public key Q, the base point N and the order T, she calculates the points $C_1(x_1, y_1) = uN$ and $C_2(x_2, y_2) = uQ$, where the integer u randomly selects in $[1, T - 1]$. She calculates

$$c_1 = [(b_1 \times x_2) \bmod p] \oplus r_1, \tag{10}$$

$$c_i = b_i \oplus c_{i-1} \oplus r_i, \quad i = 2, 3, \cdots, (m \times n)/63, \tag{11}$$

where c_i is the encrypted big integer.

If the computing precision of the computer is 10^{16}, Alice groups the initial value x_0 and control parameter q of PWLCM together to form a big integer t_1 by

$$t_1 = x_0 \times 10^{32} + q \times 10^{16}. \tag{12}$$

Meanwhile, she calculates the ciphertext e_2 of t_1 by

$$e_2 = (t_1 \times y_2) \bmod p. \tag{13}$$

Step 5: recovering pixels

If $l = 512$, Alice divides c_i, $i = 1, 2, \cdots, (m \times n)/63$ into 63 parts, and converts them into pixel values $\{p_1^i, p_2^i, \cdots, p_{63}^i\}$ as shown in Fig. 2. According to the pixel positions, she rebuilds these pixel values into an encrypted image E with the size $m \times n$.

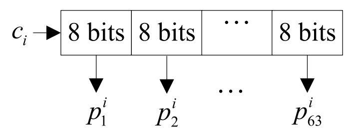

FIGURE 2. **Converting an encrypted big integer into pixel values.**

Step 6: sending the ciphertext

Alice sends the encrypted image E, C_1 and e_2 to Bob.

C. BOB'S DECRYPTION PROCESS

Bob performs the following steps after he receives Alice's ciphertext E, C_1 and e_2. The image decryption flowchart of the proposed algorithm is shown in Fig. 3.

Step 1: chaotic sequence generation

Bob calculates the point $C_2(x_2, y_2) = dC_1$ with his private key d. After that he calculates

$$t_2 = \left(e_2 \times y_2^{-1}\right) \bmod p, \tag{14}$$

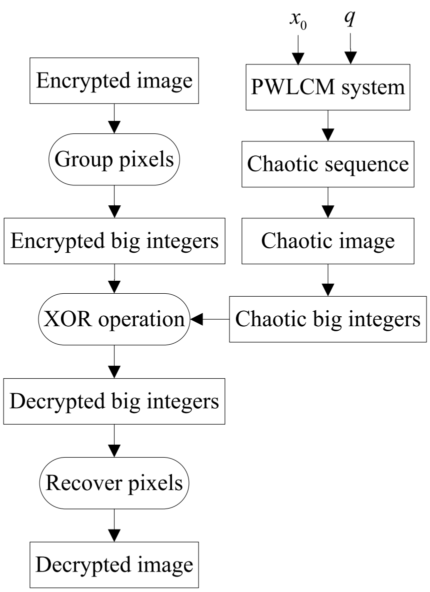

FIGURE 3. The decryption flowchart of the proposed algorithm.

where y_2^{-1} is the inverse element of y_2, i.e., $\left(y_2 \times y_2^{-1}\right)$ mod $p = 1$. Bob can obtain the initial value x_0 and control parameter q of PWLCM by

$$q = t_2/10^{16} - floor\left(t_2/10^{16}\right), \tag{15}$$

$$x_0 = \left(t_2 - q \times 10^{16}\right)/10^{32}, \tag{16}$$

where $floor(\bullet)$ is a function of round toward negative infinity. After that, he iterates Eq. (4) $m \times n$ times with x_0 and q, and obtains a chaotic sequence $X = \{x_i\}_{m\times n}$.

Step 2: chaotic image generation

Bob can get $Y = \{y_i\}_{m\times n}$ with Eq. (8). According to the element positions, he converts Y into a chaotic image C with the size $m \times n$.

Step 3: grouping pixels

Bob groups $l/8 - 1$ pixel values together to form a big integer number with the $l - 8$ bit length. For the encrypted image E, Bob can obtain $(m \times n)/63$ encrypted big integers $\left\{c_1, c_2, \cdots, c_{(m\times n)/63}\right\}$. For the chaotic image C, Alice can also obtain the chaotic big integers $\left\{r_1, r_2, \cdots, r_{(m\times n)/63}\right\}$.

Step 4: ECC decryption

Bob computes

$$d_1 = \left[(c_1 \oplus r_1) \times x_2^{-1}\right] \bmod p, \tag{17}$$

$$d_i = c_i \oplus c_{i-1} \oplus r_i, \quad i = 2, 3, \cdots, (m \times n)/63, \tag{18}$$

where d_i is the decrypted big integer, and x_2^{-1} is the inverse element of x_2.

Step 5: recovering pixels

If $l = 512$, Bob divides d_i, $i = 1, 2, \cdots, (m \times n)/63$ into 63 parts, and converts them into pixel values. According to the pixel positions, he rebuilds these pixel values into a decrypted image D with the size $m \times n$.

IV. EXISTING SIMILAR ALGORITHMS

A. SINGH'S ALGORITHM

Singh and Singh [18] proposed an image encryption algorithm based on ECC (short for Singh's algorithm), the corresponding encryption flowchart is shown in Fig. 4. The main encryption steps are described as follows.

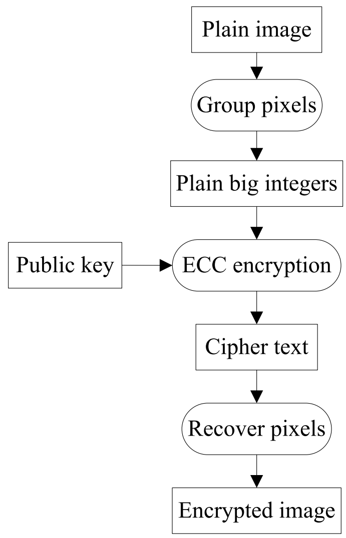

FIGURE 4. The encryption flowchart of Singh's algorithm.

Step 1: Alice groups the pixels of the plain image into the plain big integers, and saves them as P_m.

Step 2: Alice selects a random k and compute kN, kP_b, where P_b is the Bob's public key and N is the base point.

Step 3: Alice encrypts each value of P_m with kP_b, and stores the result as the cipher text P_c.

Step 4: Alice converts each value of P_c to values ranging from 0 to 255, and groups them according to the size of the plain image. In this way, the encrypted image can be obtained.

B. LAIPHRAKPAM'S ALGORITHM

Laiphrakpam and Khumanthem [19] proposed an image encryption algorithm based on ECC and chaotic system (short for Laiphrakpam's algorithm). The main encryption steps are described as follows.

Step 1: Alice generates the parameters x_0, μ of Logistic map with the shared key $kN : (x, y)$, where k is the hash value obtained by SHA-512 on the plain image, and N is the base point.

Step 2: for the plain image $I_{m\times n}$, Alice generates a chaotic sequence $CS[i]$, $i = 1, 2, \cdots, mn/128$.

Step 3: Alice converts $CS[i]$ into big integers $L[i] = floor(CS[i] \times 10^{32})$, $i = 1, 2, \cdots, mn/128$.

Step 4: Alice calculates $PM[i] : (x_{pm}, y_{pm}) = L[i] \bullet kG$, $i = 1, 2, \cdots, mn/128$, where $\bullet$ is the point multiplication.

Step 5: Alice converts $PM[i]$, $i = 1, 2, \cdots, mn/128$ into 128-byte values with the given method of base conversion. After that, she can obtain a random sequence $RS[i]$, $i = 1, 2, \cdots, mn$.

Step 6: Alice scrambles the pixel position for r rounds with Arnold's transformation, where r is calculated by $kG : (x, y)$.

Step 7: Alice diffuses the plain image by

$$CI[i] = SI[i] \oplus RS[i], \quad i = 1, 2, \cdots, mn. \tag{19}$$

where $SI[i]$ and $CI[i]$ are the pixel values of the scrambled and encrypted images respectively.

C. ZHU'S ALGORITHM

Zhu and Zhang [20] proposed an image encryption algorithm based on ECC (short for Zhu's algorithm). The main encryption steps are described as follows.

Step 1: Alice segments the plain image and $K - 1, K \geq 2$ camouflaged images into $m_K \times n_K$ image blocks. Here the sizes of these images are $m \times n$.

Step 2: Alice randomly chooses $(K \times m \times n)/(m_K \times n_K)$ unique big integers in $[0, p - 1]$ as the filenames of mixed image elements.

Step 3: Alice computes $C_1(x_1, y_1) = uN$ and $C_2(x_2, y_2) = uQ$, where u is a random big integer in $[1, T - 1]$, T is the period of the base point N, and Q is Bob's public key.

Step 4: Alice computes $c_1 = (a_{ij} \times x_2) \bmod p$ and $c_2 = (a_{i,j+1} \times y_2) \bmod p$, where a_{ij} and $a_{i,j+1}$ are the filenames of mixed image elements.

Step 5: Alice sends C_1, c_1, c_2 and the set of mixed image elements to Bob.

V. EXPERIMENTS

The first elliptic curve in our experiments is the 256-bit standard elliptic curve given by ECC Brainpool (short for ECC-256), where the parameters p, a and b are listed in Tab. 1. The base point N, its period T, Bob's private key d and his public key Q are listed in Tab. 1 too.

TABLE 1. ECC-256 parameters in decimal form.

	Values
p	76884956397045344220809746629001649093037950200943055203735601445031516197751
a	56698187605326110043627228396178346077120614539475214109386828188763884139993
b	17577232497321838841075697789794520262950426058923084567046852300633325438902
N	(63243729749562333355292243550312970334778175571054726587095381623627144114786,38218615093753523893122277964030810387585405539772602581557831887485717997975)
T	76884956397045344220809746629001649092737531784414529538755519063063536359079
d	29546760483825510980134987192545409393663261554963700603763073905806892897511
Q	(67902213922733409072953238763567458148598344264448689990560812698883302693898,71867995946314228918141084084305933376070327237154860810714661890498171903426)

The second elliptic curve in our experiments is the 512-bit standard elliptic curve given by ECC Brainpool (short for ECC-512) [18], where the parameters p, a and b are listed in Tab. 2. The base point N, its period T, Bob's private key d and his public key Q are listed in Tab. 2 too.

TABLE 2. ECC-512 parameters in decimal form.

	Values
p	894896220765023255165660281515915342216260964409835451134459718720005701041355243991793430419195694276544653038642734593796389430992392853607053460781 6947
a	629486055797306322766642130647637932407471577062274622713691044545030191428127609802799096840798396269115185367856387783422183402743971823806572584426413 8
b	245789008328967059274849584342077916531909009637501918328323668736179176583263496463525128488282611559800773506973771797764811498834995234341530862286627
N	(6792059140424575174435640431269195087843153390102528814680230127320474825798530775456474462728667949363715224107745326865824846179460139288742968443515 22,6592244555240112873324748381429610341312712940326266331327445066687010545415256461097707483288650216992613090185042957716318301180159234788504307628509330)
T	894896220765023255165660281515915342216260964409835451134459717200057010413418528378981730643524959857451398370029280583094215613882043973354392115544169
d	195336213603604152096989093281052901489014454978653357772417570947908950389989732125499873509607620241560401056087989094206465431983075623592397662843361
Q	(2676768453394755195814069676970956501372728066923265402024558860509945130668915321409219227316917046333509396850812227710907993174996639554683299332304 91,1131597161747597035291971628695155819410433149663549688536900815073558383463833647344578137300949803313877481626288812490119296372063156315616609671333 72)

For ECC-256 and ECC-512, their key lengths are 256 and 512 bits respectively. We grouped 31 and 63 pixel values together to form a big integer respectively in our experiments.

FIGURE 5. Airfield.

The experimental purpose is to encrypt the plain image "Airfield" with the size 512×512 in Fig. 5 with the proposed algorithm. The computer configuration used in our experiments is: M-5Y71 CPU, 1.20 GHz processor, 8GB RAM, and Eclipse version 4.7.3a. For the plain image, its 256-bit hash value is K=0xb40608c3183bc9e3d9933de 8b4db1366fa970574cd8c12d8d9ad0e7dcf37c0fe. Therefore, the initial value and control parameter of PWLCM are x_0=0.960784313725490 and q=0.150980392156863.

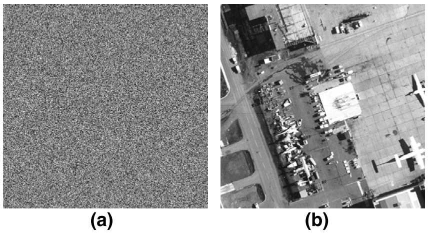

FIGURE 6. Encrypted and decrypted images for ECC-256. (a) Encrypted image. (b) Decrypted image.

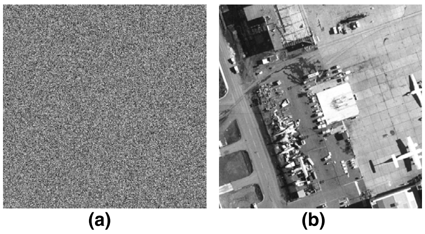

FIGURE 7. Encrypted and decrypted images for ECC-512. (a) Encrypted image. (b) Decrypted image.

For the proposed algorithm, the encrypted images with ECC-256 and ECC-512 are shown in Figs. 6(a) and 7(a). Bob can recover the plain images with his private key from the encrypted images. The corresponding decrypted images are shown in Figs. 6(b) and 7(b). The correlation coefficient between the plain image and decrypted image is defined by

$$r_{x,y} = \frac{E\left(\left(x - E\left(x\right)\right)\left(y - E\left(y\right)\right)\right)}{\sqrt{D\left(x\right)D\left(y\right)}}, \tag{20}$$

$$E\left(x\right) = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{21}$$

$$D\left(x\right) = \frac{1}{N}\sum_{i=1}^{N} \left(x_i - E\left(x\right)\right)^2, \tag{22}$$

where x, y denote the pixel values of the plain and decrypted images respectively, $E\left(\bullet\right)$ is the expectation function and $D\left(\bullet\right)$ is the variance function. The Peak Signal-to-Noise Ratio (PSNR) between the plain and decrypted images is defined by

$$PSNR = 10\lg\left(\frac{255^2 \times m \times n}{\sum_{i=1}^{m}\sum_{j=1}^{n}\left[I(i,j) - D(i,j)\right]^2}\right), \tag{23}$$

where I, D denote the plain and decrypted images with the size $m \times n$ respectively. Because the proposed algorithm is lossless, the values of the correlation coefficient and PSNR are 1 and ∞ respectively.

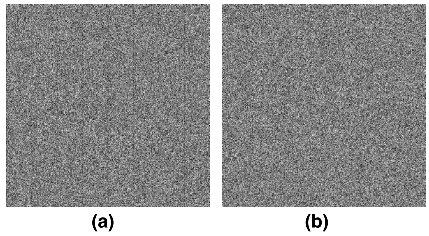

FIGURE 8. Encrypted images for Singh's and Laiphrakpam's algorithms. (a) Encrypted image1. (b) Encrypted image2.

For Singh's and Laiphrakpam's algorithms, their encrypted images are shown in Figs. 8(a) and (b) respectively.

For Zhu's algorithm, the ciphertext is the set of mixed image elements, not the encrypted image. We added three camouflaged images, i.e., "Aerial", "Base" and "Islands" as shown in Figs. 9(a)-(c) respectively. The set of mixed image blocks is shown in Fig. 10.

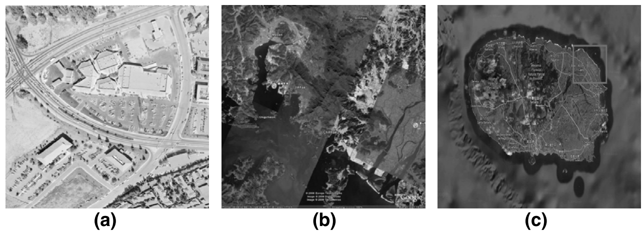

FIGURE 9. Camouflaged images. (a) Aerial. (b) Base. (c) Islands.

FIGURE 10. The set of mixed image blocks.

Experimental results show that the encrypted images of the proposed, Singh's and Laiphrakpam's algorithms appear to be really noisy so that people cannot obtain any information from them. Therefore, the proposed algorithm has excellent encryption effect.

VI. ALGORITHM ANALYSES

For an excellent image encryption algorithm, it can resist several commonly used attacks, such as the brute-force attack and differential attack. This paper analyzes the performance of the proposed algorithm in terms of the key space, histogram, correlation, differential attack, information entropy, encryption speed, and known plaintext attack.

A. KEY SPACE ANALYSIS

For a desirable encryption algorithm, its key space should be large enough to resist the brute-force attack. The keys of the proposed algorithm is the public key and private key of ECC. The public key can be published, but the private key should be protected secretly. The algorithm security depends a lot on the size of the key used. The bigger the key size, the more it is difficult to perform the brute-force attack. In theory, the security of ECC depends on the ECDLP difficulty. The ECDLP is one of the most difficult problems in mathematics, and there isn't an effectively deciphered method at present.

For the proposed and Singh's algorithms, they are designed with ECC and big integers. Singh's algorithm used ECC-512 in their experiments, so its key spaces are 2^{512}. In our experiments, we used both ECC-256 and ECC-512. Therefore, their key spaces are 2^{256} or 2^{512}, which are large enough to resist the brute-force attack. However, the key space for ECC-256 is obviously smaller than the key space for ECC-512.

For Laiphrakpam's algorithm, it is designed with Logistic map, Arnold's transformation and ECC. Authors used ECC-512 in their experiments, so its key space is 2^{512}, which are large enough to resist the brute-force attack.

For Zhu's algorithm, it is designed with both ECC and mixed image elememts. They used an 8-bit elliptic curve in their experiment just for verifying their algorithm. However, according to the security requirements, users can choose the suitable elliptic curve in practice.

B. HISTOGRAM ANALYSIS

The histogram can reflect the statistical feature of pixel value distribution in the image. For a desirable encryption algorithm, the histogram of the encryption image is always uniform [24]. Fig. 11 is the histogram of the plain image. For the proposed algorithm with ECC-256 or ECC-512, Figs. 12 and 13 show the histograms of their encrypted images respectively. For Singh's and Laiphrakpam's algorithms, the histograms of the encrypted images are shown in Figs. 14 and 15 respectively. The experimental results show that the histograms of encrypted images are

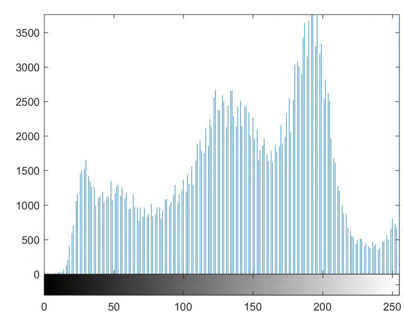

FIGURE 11. The histogram of the plain image.

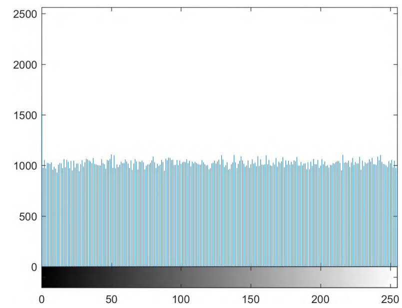

FIGURE 12. The histogram of the encrypted image for the proposed algorithm with ECC-256.

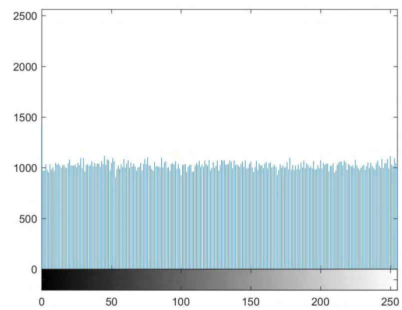

FIGURE 13. The histogram of the encrypted image for the proposed algorithm with ECC-512.

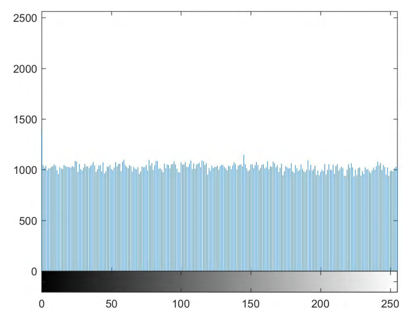

FIGURE 14. The histogram of the encrypted image for Singh's algorithm.

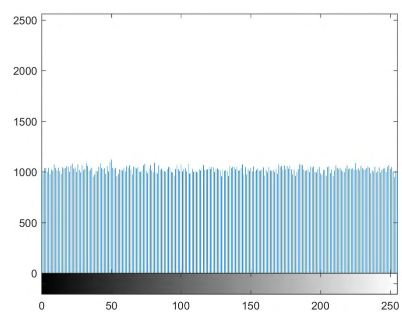

FIGURE 15. The histogram of the encrypted image for Laiphrakpam's algorithm.

uniformly distributed for the proposed, Singh's and Laiphrakpam's algorithms, which are totally different from the histogram of the plain image.

C. CORRELATION ANALYSIS

To evaluate the performance of pixel correlation, we carry out some simulations. The correlation coefficient of adjacent pixels is calculated by Eqs. (20)-(22), where x, y denote the adjacent pixels respectively.

For the proposed algorithm, 5,000 pairs of adjacent pixels are randomly chosen from Figs. 5 and 7(a). Fig. 16 reflects their horizontal, vertical and diagonal relevance for adjacent pixels respectively. For the Singh's and Laiphrakpam's algorithms, 5,000 pairs of adjacent pixels are also randomly chosen from Figs. 8(a) and (b). The values of correlation coefficients are listed in Tab. 3.

TABLE 3. Correlation coefficients.

Image	Horizontal	Vertical	Diagonal
Plain image	0.9402	0.9423	0.9032
Encrypted image of the proposed algorithm with ECC-256	0.0012	0.0044	0.0046
Encrypted image of the proposed algorithm with ECC-512	-0.0013	0.0025	0.0014
Encrypted image of Singh's algorithm	0.0023	0.047	0.0026
Encrypted image of Laiphrakpam's algorithm	-0.0043	0.0014	0.0048

Experimental results show that the correlation coefficients of the plain image are close to 1, while the correlation coefficients of encrypted images for the proposed, Singh's and Laiphrakpam's algorithms are close to 0. Therefore, the proposed algorithm can destroy the correlation between adjacent pixels well and protect the content of the plain image.

D. DIFFERENTIAL ATTACK ANALYSIS

The differential attack is used to check the plaintext sensitivity for an image encryption algorithm [25]. Therefore, if we make a slight change to the plain image, a desirable image encryption algorithm can spread this influence over the whole encryption process. To evaluate the ability to resist the differential attack, the Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI) are defined by

$$NPCR = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f(i,j)}{m \times n} \times 100\%, \tag{24}$$

$$UACI = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} \left|I'(i,j) - I''(i,j)\right|}{255 \times m \times n} \times 100\%, \tag{25}$$

where $I'(i,j)$ is the encryption image for the plain image, $I''(i,j)$ is the encryption image for the modified image, and $f(i,j)$ is defined by

$$f(i,j) = \begin{cases} 0 & I'(i,j) = I''(i,j) \\ 1 & I'(i,j) \neq I''(i,j). \end{cases} \tag{26}$$

In the experiment, the original pixel value of $I(1,1)$ in the plain image is 212. To test the ability to resist the differential attack, this pixel value is changed to 200. For the proposed, Singh's and Laiphrakpam's algorithms, both *NPCR* and *UACI* values are given in Tab. 4. The dada in Tab. 4 show

TABLE 4. NPCR and UACI values.

Algorithm	NPCR	UACI
The proposed algorithm with ECC-256	99.95%	33.11%
The proposed algorithm with ECC-512	99.62%	33.28%
Singh's algorithm	0	0
Laiphrakpam's algorithm	99.47%	33.18%

that for the proposed algorithm and Laiphrakpam's algorithm, a slight change to the plain image will result in a great change in the encrypted image. However, both *NPCR* and *UACI* are 0 for Singh's algorithm. Therefore, the proposed algorithm has an excellent ability to resist the differential attack.

E. INFORMATION ENTROPY ANALYSIS

Information entropy can reflect the indeterminacy of image information. For an ideal encryption image, its information entropy is close to 8 [26]. For the gray image, the information entropy is defined by

$$H(m) = -\sum_{i=0}^{255} P(m_i)\log_2 P(m_i), \tag{27}$$

where m_i is the ith gray level for the digital image with 256 gray levels, and $P(m_i)$ is the emergence probability of m_i. For the plain image, its entropy value is 7.1206. For the encrypted images, the entropy values for the proposed, Singh's and Laiphrakpam's algorithms are given in Tab. 5. From Tab. 5, we can conclude that the entropy values of encrypted images for the proposed, Singh's and Laiphrakpam's algorithms are close to 8. Therefore, the proposed algorithm can effectively resist the statistical attack.

TABLE 5. The values of information entropy.

Algorithm	Value
The proposed algorithm with ECC-256	7.9986
The proposed algorithm with ECC-512	7.9990
Singh's algorithm	7.9994
Laiphrakpam's algorithm	7.9988

F. ENCRYPTION SPEED ANALYSIS

For the image encryption algorithm based on ECC, the main time-consuming operation is the point multiplication. The XOR operation is very fast, so we omit it here. In our experiments, the size of the plain image is 512 × 512. To encrypt a number with ECC, we need to performing twice point multiplication operations to get the values of C_1 and C_2. For the proposed algorithm, there are twice point multiplication operations to get the encrypted big integer c_1 from the plain big integer b_1. For Singh's algorithm, it encrypts

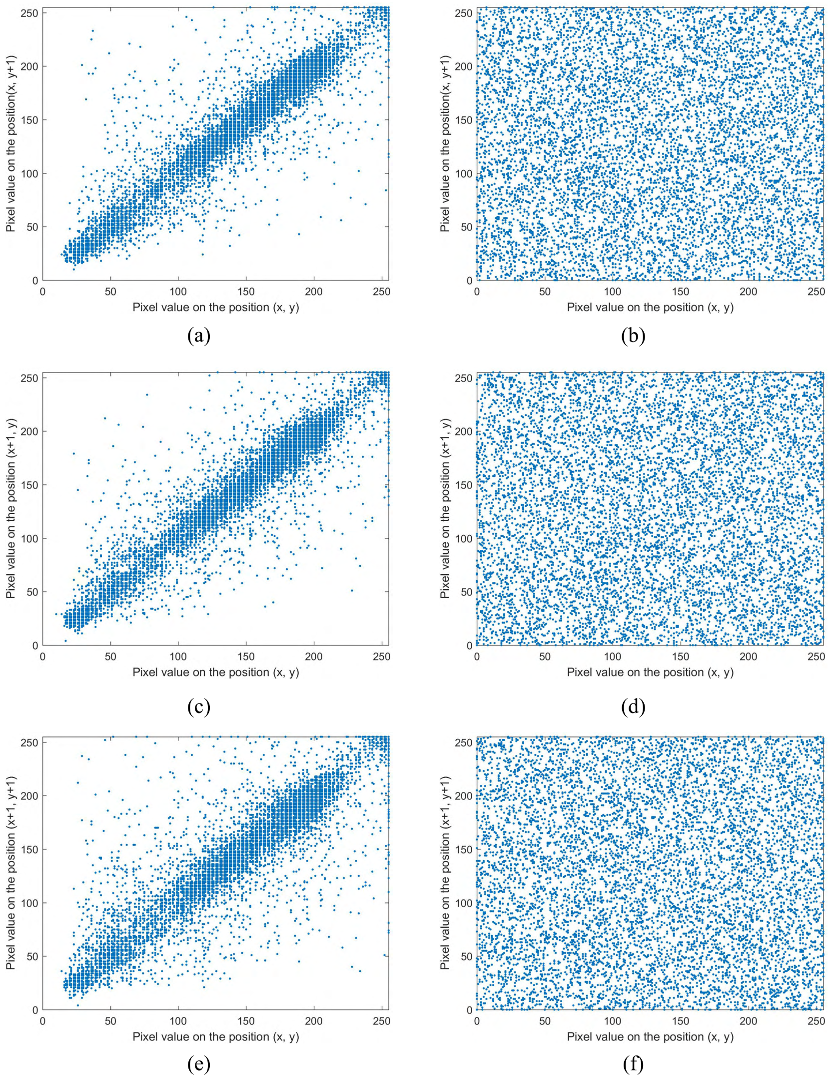

FIGURE 16. Adjacent pixel correlation of Airfield and its corresponding encrypted image for the proposed algorithm with ECC-512. (a) Horizontal direction in the plain image. (b) Horizontal direction in the encrypted image. (c)Vertical direction in the plain image. (d) Vertical direction in the encrypted image. (e) Diagonal direction in the plain image. (f) Diagonal direction in the encrypted image.

all the plain big integers with ECC, so its point multiplication operations is 8192 times. For Laiphrakpam's algorithm, Alice performs the point multipication operation $mn/128 = 2048$ times in Step 4. For Zhu's algorithm, we added three camouflaged images in our experiments. If the size of image blocks is 32×32, then there are 1024 mixed image elements in total. This algorithm should encrypt the filenames of these mixed image elements, so its point multipication operation is 2048 times. Finally, for the proposed, Singh's, Laiphrakpam's and Zhu's algorithms, the times of the point multipication operation for these algorithms are given in Tab. 6. The unit is times. Therefore, the quantity of the point multipication operation reduces obviously for the proposed algorithm in theory.

The execution time of encryption and decryption depends on various factor like programming skills, algorithm, hardware where the program is implemented, size of the image, etc. For the proposed, Singh's, Laiphrakpam's and Zhu's algorithms, the encryption time is given in Tab. 6 too. The unit is minute. The proposed algorithm runs the fastest of all these four algorithms. Although the times of the point multiplication times for both Laiphrakpam's and Zhu's

TABLE 6. Point multiplication times (unit: Times) and encryption time (unit: Minute).

Algorithm	Point multiplication times	Time
The proposed algorithm with ECC-256	2	0.91
The proposed algorithm with ECC-512	2	0.61
Singh's algorithm	8192	37.2
Laiphrakpam's algorithm	2048	2.64
Zhu's algorithm	2048	10.57

algorithms are the same, i.e., 2048 times, their encryption time is obviously different. The reason is that $L[i]$ is a very small big integer in Laiphrakpam's algorithm, which is less than 10^{32}. Therefore, the proposed algorithm is efficient, which can be suitable for ecrypting images in practice.

G. KNOWN PLAINTEXT ATTACK ANALYSIS

Even if two images are very similar, such as only one-bit difference, their hash values of SHA-256 are completely different [24]. The proposed algorithm adopts SHA-256 on the plain image to generate 256-bit hash value K, which is used to generate the initial value x_0 and control parameter q of PWLCM. These values are very sensitive to the plain image. The detailed description are given in Subsection II.C. Therefore, the encryption process has a strong relationship with plain image in the proposed algorithm. So, the proposed algorithm can resist the known plaintext attack.

VII. CONCLUSIONS

To protect the content of the digital image, this paper presents an asymmetric image encryption algorithm based on ECC and chaotic system. The proposed algorithm makes the key transmission and management relatively simple and secure. Experimental results and algorithm analyses show that the proposed algorithm is secure enough to resist the brute-force attack, differential attack, the known plaintext attack, and statistical attack. Comparing with three existing similar algorithms, the proposed algorithm is the fastest in terms of the encryption speed.

ACKNOWLEDGMENT

The authors would like to express their sincere thanks to six anonymous reviewers and the associate editor Dr. Wen Chen for their constructive comments.

REFERENCES

[1] X. Huang and G. Ye, "An image encryption algorithm based on hyper-chaos and DNA sequence," *Multimedia Tools Appl.*, vol. 72, no. 1, pp. 57–70, 2014.

[2] X. Chai, X. Zheng, Z. Gan, D. Han, and Y. Chen, "An image encryption algorithm based on chaotic system and compressive sensing," *Signal Process.*, vol. 148, no. 7, pp. 124–144, 2018.

[3] R. Lan, J. He, S. Wang, T. Gu, and X. Luo, "Integrated chaotic systems for image encryption," *Signal Process.*, vol. 147, no. 6, pp. 133–145, 2018.

[4] R. Parvaz and M. Zarebnia, "A combination chaotic system and application in color image encryption," *Opt. Laser Technol.*, vol. 101, no. 5, pp. 30–41, 2018.

[5] L. Teng, X. Wang, and J. Meng, "A chaotic color image encryption using integrated bit-level permutation," *Multimedia Tools Appl.*, vol. 77, no. 6, pp. 6683–6896, 2018.

[6] L. Liu, S. Hao, J. Lin, Z. Wang, X. Hu, and S. Miao, "Image block encryption algorithm based on chaotic maps," *IET Signal Process.*, vol. 12, no. 1, pp. 22–30, 2018.

[7] C. Pak and L. Huang, "A new color image encryption using combination of the 1D chaotic map," *Signal Process.*, vol. 138, no. 9, pp. 129–137, 2017.

[8] Q. Ran, L. Wang, J. Ma, L. Tan, and S. Yu, "A quantum color image encryption scheme based on coupled hyper-chaotic Lorenz system with three impulse injections," *Quantum Inf. Process.*, vol. 17, no. 8, pp. 1–30, 2018.

[9] L. Gong, C. Deng, S. Pan, and N. Zhou, "Image compression-encryption algorithms by combining hyper-chaotic system with discrete fractional random transform," *Opt. Laser Technol.*, vol. 103, no. 7, pp. 48–58, 2018.

[10] N. Zhou, W. Chen, X. Yan, and Y. Wang, "Bit-level quantum color image encryption scheme with quantum cross-exchange operation and hyper-chaotic system," *Quantum Inf. Process.*, vol. 17, no. 6, pp. 1–24, 2018.

[11] H. Zhu, X. Zhang, H. Yu, C. Zhao, and Z. Zhu, "An image encryption algorithm based on compound homogeneous hyper-chaotic system," *Nonlinear Dyn.*, vol. 89, no. 1, pp. 61–79, 2017.

[12] X. Zhang, W. Nie, Y. Ma, and Q. Tian, "Cryptanalysis and improvement of an image encryption algorithm based on hyper-chaotic system and dynamic S-box," *Multimedia Tools Appl.*, vol. 76, no. 14, pp. 15641–15659, 2017.

[13] M. Kaur and V. Kumar, "Efficient image encryption method based on improved Lorenz chaotic system," *Electron. Lett.*, vol. 54, no. 9, pp. 562–563, 2018.

[14] M. Kumar, A. Iqbal, and P. Kumar, "A new RGB image encryption algorithm based on DNA encoding and elliptic curve Diffie–Hellman cryptography," *Signal Process.*, vol. 125, pp. 187–202, Aug. 2016.

[15] J. Wu, X. Liao, and B. Yang, "Color image encryption based on chaotic systems and elliptic curve ElGamal scheme," *Signal Process.*, vol. 141, pp. 109–124, Dec. 2017.

[16] S. Toughi, M. H. Fathi, and Y. A. Sekhavat, "An image encryption scheme based on elliptic curve pseudo random and advanced encryption system," *Signal Process.*, vol. 141, no. 12, pp. 217–227, 2017.

[17] Z. Zhao and X. Zhang, "ECC-based image encryption using code computing," *Amer. J. Eng. Technol. Res.*, vol. 11, no. 9, pp. 1399–1405, 2011.

[18] L. D. Singh and K. M. Singh, "Image encryption using elliptic curve cryptography," *Procedia Comput. Sci.*, vol. 54, pp. 472–481, 2015.

[19] D. S. Laiphrakpam and M. S. Khumanthem, "A robust image encryption scheme based on chaotic system and elliptic curve over finite field," *Multimedia Tools Appl.*, vol. 77, no. 11, pp. 8629–8652, 2018.

[20] G. Zhu and X. Zhang, "Mixed image element encryption algorithm based on an elliptic curve cryptosystem," *J. Electron. Imag.*, vol. 17, no. 2, pp. 1–5, 2008.

[21] X. Zhang, G. Zhu, W. Wang, and M. Wang, "Scalar multiplication algorithm of ECC based on precomputation and periodicity," (in Chinese), *J. Beijing Univ. Aeronaut. Astronaut.*, vol. 37, no. 11, pp. 1451–1455, 2011.

[22] X. Wang and D. Xu, "A novel image encryption scheme based on Brownian motion and PWLCM chaotic system," *Nonlinear Dyn.*, vol. 75, nos. 1–2, pp. 345–353, 2015.

[23] R. Guesmi, M. A. B. Farah, A. Kachouri, and M. Samet, "A novel chaos-based image encryption using DNA sequence operation and secure hash algorithm SHA-2," *Nonlinear Dyn.*, vol. 83, no. 3, pp. 1123–1136, 2015.

[24] X. Chai, Y. Chen, and L. Broyde, "A novel chaos-based image encryption algorithm using dna sequence operations," *Opt. Lasers Eng.*, vol. 88, pp. 197–213, Jan. 2017.

[25] A. Belazi, A. A. A. El-Latif, and S. Belghith, "A novel image encryption scheme based on substitution-permutation network and chaos," *Signal Process.*, vol. 128, pp. 155–170, Nov. 2016.

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

XIAOQIANG ZHANG received the B.Sc. and M.Sc. degrees from the Department of Information Engineering, North China University of Water Conservancy and Hydroelectric Power, Zhengzhou, China, in 2006 and 2008, respectively, and the Ph.D. degree from Beihang University in 2013. From 2012 to 2013, he has been a Visiting Scholar with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, Canada. He is currently an Associate Professor with the School of Information and Control Engineering, China University of Mining and Technology. His main research interests include multimedia security and signal processing.

XUESONG WANG received the Ph.D. degree from the China University of Mining and Technology in 2002. She is currently a Professor with the School of Information and Control Engineering, China University of Mining and Technology. Her main research interests include machine learning and image processing.