Signal Processing 92 (2012) 1069–1078

Contents lists available at SciVerse ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Elliptic curve ElGamal based homomorphic image encryption scheme for sharing secret images

Li Li [a], Ahmed A. Abd El-Latif [b], Xiamu Niu [a,b,*]

[a] School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China
[b] School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150080, China

ARTICLE INFO

Article history:
Received 20 May 2011
Received in revised form
21 August 2011
Accepted 24 October 2011
Available online 7 November 2011

Keywords:
Image encryption
Elliptic curve ElGamal
Additive homomorphism

ABSTRACT

This paper proposes an encryption scheme with a new additive homomorphism based on Elliptic Curve ElGamal (EC-ElGamal) for sharing secret images over unsecured channel. The proposed scheme enables shorter key and better performance than schemes based on RSA or ElGamal. It has a lower computation overhead in image decryption comparing with the method that uses other additively homomorphic property in EC-ElGamal. Elliptic curve parameters are selected to resist the Pohlig–Hellman, Pollard's-rho, and Isomorphism attacks. Experimental results and analysis show that the proposed method has superior performance to RSA and ElGamal.

1. Introduction

Image is one of the most important information representation styles and widely used in most of the applications. Images are often exchanged between two parties over the insecure network. In order to protect the images from interception, the shared images should be encrypted before transmission. Furthermore, because of the limited bandwidth, the encrypted images for sharing are combined together to obtain a new image. In the literature, some schemes based on visual secret sharing [1,2] has emerged. However, the shared images are operated in the plaintext form, which is not secure enough. To facilitate the combination directly over the encrypted images, it is necessary to utilize the cryptosystem with homomorphic property where a specific algebraic operation performed on the plaintext data is equivalent to the decryption of the same (probably different) algebraic operation performed on the ciphertext data [3,4]. Homomorphic property of public key cryptosystems has been employed in various security scenarios, such as secret images sharing [5] based on RSA [6], secure electronic voting system [7,8] based on ElGamal [9], secure data aggregation in wireless sensor network [10–12] based on Elliptic Curve Cryptography (ECC) [13], secure distortion computation [14] based on Paillier [15] and some other works [16,17]. RSA and ElGamal have the multiplicative homomorphism while ECC and Paillier have the additive homomorphism. Moreover, additive homomorphic property has a wide application, such as pixel average for encrypted image resolution reduction and privacy protection in video surveillance by obtaining the difference image [18].

RSA and ElGamal cryptosystems are the most extensively used encryption methods. Both RSA and ElGamal cryptosystems require high computation overhead to perform exponential operations, while ECC needs only additions and multiplications. Besides, the least key length to achieve the minimum security requirement for ECC is much smaller than RSA and ElGamal as can be seen in Table 1 [19]. Encryption methods based on RSA or

* Corresponding author at: School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China. Tel.: +86 451 86402861.
E-mail addresses: lili.isec2008@gmail.com (L. Li), ahmed_rahiem@yahoo.com (A.A. Abd El-Latif), xm.niu@hit.edu.cn (X. Niu).

0165-1684/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2011.10.020

Table 1
Least key length to achieve the minimum security requirement.

Cryptosystem	Key length
RSA	1024 bits
ElGamal	1024 bits
ECC	160 bits

ElGamal have high computation overhead and large space consumption, so it is not suitable for the real-time and bandwidth-limited applications (e.g., image transmission, video streaming, and video surveillance). Since the method in Ref. [5] is based on RSA, it requires at least 1024 bits key to achieve the minimum security according to Table 1.

The additively homomorphic property in EC-ElGamal is first used in Ref. [10], where the decryption is the same as solving the elliptic curve discrete logarithm problem (ECDLP) and needs brute force method which means high computation overhead. Thus, it is not efficient for decryption on the receiver side, especially for huge data such as image applications.

In this paper, we present a new encryption scheme for sharing secret images by exploiting the additive homomorphism of EC-ElGamal. The scheme uses shorter key than long key for RSA or ElGamal which decreases the computation overhead greatly and uplifts the encryption efficiency. In addition, the elliptic curve parameters are selected to resist the Pohlig–Hellman and Pollard's-rho attacks and be immune to Isomorphism attacks.

This paper is organized as follows. Section 2 gives a review of homomorphism property and ECC cryptosystems. In Section 3, the proposed scheme is introduced. Experimental results along with performance analysis of the proposed scheme as well as comparison with other schemes are discussed in Section 4. Finally, Section 5 gives the conclusion and the future work.

2. Preliminary

2.1. Homomorphic encryption

In Ref. [4] it gives a definition for homomorphic public-key cryptosystem. Let $En_{k1}(m)$ be the encryption of plaintext m taken from the set of plaintexts M using public key $k1$ and $De_{k2}(c)$ be the decryption of ciphertext c using private key $k2$. A cryptosystem is homomorphic if the encryption and decryption functions satisfy Eq. (1) where m_1, m_2 are taken from M.

$$f_1(m_1,m_2)=De_{k2}(f_2(En_{k1}(m_1),En_{k1}(m_2))) \tag{1}$$

The operation $f_1(\cdot)$ on the plaintext is the same as the decryption of the operation $f_2(\cdot)$ on the corresponding encrypted ciphertext according to Eq. (1). If $f_1(\cdot)$ is an addition operator, the scheme is said to be additively homomorphic, and multiplicatively homomorphic if $f_1(\cdot)$ is a multiplicative operator. If $f_1(\cdot)$ and $f_2(\cdot)$ are the same operators, the cryptosystem is algebraically homomorphic.

2.2. Elliptic curve

In this work, we adopt elliptic curves over prime finite field F_p. In Ref. [13] it gives the definition for elliptic curve over F_p. The characters in upper case represent points on the elliptic curves, and those in lower case represent integers.

Definition. Let p be a prime and $p>3$. The elliptic curve $y^2=x^3+ax+b$ over F_p, is the set of solutions (x, y) to the congruence Eq. (2) where $a, b\in F_p$ are constants such that $4a^3+27b^2\neq 0 \pmod p$, together with a special point O called the point at infinity.

$$y^2\equiv x^3+ax+b(\mathrm{mod}\,p) \tag{2}$$

The elliptic curve $y^2=x^3+ax+b$ over F_p could be represented as $E_p(a, b)$. $E_p(a, b)$ is an abelian group [13]. Suppose the points $P=(x_P, y_P)$, $Q=(x_Q, y_Q)$ are on $E_p(a, b)$. $R=P+Q=(x_R, y_R)$. The addition of points on $E_p(a, b)$ is defined as in Eqs. (3)–(6) to be make $E_p(a, b)$ into an abelian group [13]. If $P=Q$, then $R=2P$. Additionally, the negative of point P is computed as $-P=(x_P, -y_P)$.

$$x_R=(\lambda^2-x_P-x_Q)\mathrm{mod}\,p \tag{3}$$

$$y_R=(\lambda(x_P-x_R)-y_P)\mathrm{mod}\,p \tag{4}$$

$$\lambda=((y_Q-y_P)/(x_Q-x_P))\mathrm{mod}\,p,\ \text{if}\ P\neq Q \tag{5}$$

$$\lambda=((3x_P^2+a)/(2y_P))\mathrm{mod}\,p,\ \text{if}\ P=Q \tag{6}$$

Multiples of the points are computed by repeated doubling or additions of two points [20]. For example, $12G=2(2(G+2G))$, it performs 3 doublings and 1 addition of points on the curve.

2.3. EC-ElGamal

Security of ECC depends on the elliptic curve discrete logarithm problem (ECDLP), and there is no subexponential-time method solving ECDLP so far [20].

Definition of ECDLP [20]. Given a point $C\in E_p(a, b)$ (with base point G), the ECDLP is the problem of finding an integer $m\in F_p$ such that $C=mG$ if such an m exists.

Eqs. (7) and (8) prove the additively homomorphic property in $C_i=m_iG$ where $m_i\in F_p$, $i=1,2\ldots,n$. Since the decryption of m is not an easy work which is equivalent to solving ECDLP, the additively homomorphic property of $C=mG$ does not have practical application. $De(x)$ means the decryption of x.

$$\begin{aligned} C_1+C_2+\cdots+C_n &= m_1G+m_2G\cdots+m_nG \\ &= (m_1+m_2+\cdots+m_n)G \\ &= En(m_1+m_2+\cdots+m_n) \end{aligned} \tag{7}$$

$$\begin{aligned} De(C_1+C_2+\ldots+C_n) &= De(En(m_1+m_2+\cdots+m_n)) \\ &= m_1+m_2+\cdots+m_n \end{aligned} \tag{8}$$

In Ref. [20], it introduces the analog of ElGamal cryptosystem based on ECC, which is known as EC-ElGamal. For

the elliptic curve $E_p(a, b)$ with base point G, generate random integer k and r where k is the private key. Compute public key point $K=kG$. For the plaintext point M, two ciphertext points C' and C'' are generated as in Eq. (9). The decryption from C' and C'' are computed as in Eq. (10) where M' is the decrypted version of M. The encryption and decryption results are points on the same elliptic curve.

$$\text{Encryption}: C' = M + rK, \quad C'' = rG \tag{9}$$

$$\text{Decryption}: M' = C' - kC'' \tag{10}$$

In Eq. (9), encryption of the same plaintext point will generate different ciphertext points using different r; therefore, EC-ElGamal is probabilistic and has better performance than the method in Ref. [5].

3. Proposed scheme

Here, we exploit a new additive homomorphism in EC-ElGamal cryptosystem and propose an encryption system for sharing secret images based on it.

3.1. Additive homomorphism for EC-ElGamal

For the plaintext point M_i, its ciphertext points $C_i=(M_i+r_iK, r_iG)$ where $K=kG$. The exploited additive property of EC-ElGamal is proven to be homomorphic as in Eqs. (11) and (12) where $M_1, M_2, \ldots, M_n$ are the plaintext points on the same elliptic curve $E_p(a, b)$ with base point G and $r_1, r_2, \ldots, r_n$ are integers. $r_1+r_2+\cdots+r_n$ is denoted by r' for short.

As can be seen from Eq. (11), the addition of each ciphertext part is performed piece-wisely as in the third line of Eq. (11). Line 4 in Eq. (11) is satisfied according to the commutative property in an abelian group. All the resultant points in each step in Eq. (11) are on the same elliptic curve according to the closure property in Abelian group. Line 3 in Eq. (12) is satisfied since $K=kG$. The encryption and decryption in EC-ElGamal are based on the addition between two points.

$$\begin{aligned}
&C_1+C_2+\cdots+C_n\\
&=(M_1+r_1K,r_1G)+(M_2+r_2K,r_2G)+\cdots+(M_n+r_nK,r_nG)\\
&=(M_1+r_1K+M_2+r_2K+\cdots+M_n+r_nK,r_1G+r_2G+\cdots+r_nG)\\
&=(M_1+M_2+\cdots+M_n+r_1K+r_2K+\cdots r_nK,r_1G+r_2G+\cdots+r_nG)\\
&=(M_1+M_2+\cdots+M_n+(r_1+r_2+\cdots r_n)K,(r_1+r_2+\cdots+r_n)G)\\
&=(M_1+M_2+\cdots+M_n+r'K,r'G)
\end{aligned} \tag{11}$$

$$\begin{aligned}
De(C_1+C_2+\cdots+C_n) &= De(M_1+M_2+\cdots+M_n+r'K,r'G)\\
&= M_1+M_2+\cdots+M_n+r'K-kr'G\\
&= M_1+M_2+\cdots+M_n
\end{aligned} \tag{12}$$

(1) *Discussion about addition of points on different elliptic curves*

Each elliptic curve $E_p(a, b)$ defines an abelian group *Group*, hence different elliptic curves define different abelian groups. According to the closure property in abelian group, if point P and Q belong to the same Abelian group *Group*, $P+Q$ is also in *Group* by the defined addition rules in Ref. [13]. Assume P and Q belong to different abelian groups *Group*1 and *Group*2, respectively, $R=P+Q$ belong to neither *Group*1 nor *Group*2. It can be proven by contradiction method. If $R\in$*Group*1, $Q=R-P=R+(-P)$, then $Q\in$ *Group*1 which is contradicted with the condition $Q\in$ *Group*2. Thus R does not belong to *Group*1. Similarly, it can be proven that R does not belong to *Group*2. Therefore, the addition of points on different elliptic curves does not satisfy the closure property in an abelian group which is contradicted with the elliptic curve definition. Thus we only consider the addition of points on the same elliptic curve, i.e. in the same abelian group. And it is necessary to transfer any plaintext into the same elliptic curve before encryption using EC-ElGamal.

For $C_i=(M_i+r_iK, r_iG)$, if M_i is not a point on $E_p(a, b)$, the points M_i+r_iK and r_iG do not belong to the same elliptic curve but they establish another new elliptic curve $E_{p'}(a', b')$. Thus C_i could be decrypted correctly using Eq. (10). However, if $M_1, M_2, \ldots, M_n$ do not belong to the same elliptic curve, $C_1, C_2, \ldots, C_n$ establish different elliptic curves which do not establish an abelian group according to discussion in the above paragraph. In this case Eqs. (11) and (12) are not satisfied; therefore, the image pixel value is first transferred on to the points on the same elliptic curve before using the exploited additive homomorphism in EC-ElGamal.

(2) *An example*

Take $p=11$, $a=1$, $b=6$, i.e., the elliptic curve $y^2\equiv x^3+x+6 \pmod{11}$. And $G=(2, 7)$, $M_1=(5, 2)$, $M_2=(8, 3)$ are two points on the curve $E_{11}(1, 6)$. Assume the private key $k=6$, we obtain the public key point $K=6G=2(G+2G)$ by Eqs. (3)–(6). The following is the detailed computation steps.

To solve 2G, compute $\lambda=((3x_G^2+a)/(2y_G)) \bmod p=((3(2^2)+1)/(2*7)) \bmod 11=(13/14) \bmod 11=(2/3) \bmod 11=(1/7) \bmod 11=8$. The last step in the preceding equation involves computing the multiplicative inverse of 7 in Z_{11}.

$$\begin{aligned}
x_{2G} &= (\lambda^2-x_G-x_G) \bmod p=(8^2-2-2) \bmod 11=5\\
y_{2G} &= (\lambda(x_G-x_{2G})-y_G) \bmod p=(8(2-5)-7)\\
&\quad \bmod 11=(-31) \bmod 11=2
\end{aligned}$$

To solve $3G=G+2G$, compute $\lambda=((y_G-y_{2G})/(x_G-x_{2G})) \bmod p=((7-2)/(2-5)) \bmod 11=(-5/3) \bmod 11=((-5)*4) \bmod 11=(-20) \bmod 11=2$

$$\begin{aligned}
x_{3G} &= (\lambda^2-x_G-x_{2G}) \bmod p=(2^2-2-5) \bmod 11=8\\
y_{3G} &= (\lambda(x_G-x_{3G})-y_G) \bmod p=(2(2-8)-7)\\
&\quad \bmod 11=(-19) \bmod 11=3.
\end{aligned}$$

To solve $K=6G=2(3G)$, compute $\lambda=((3x_{3G}^2+a)/(2y_{3G})) \bmod p=((3(8^2)+1)/(2*3)) \bmod 11=(193/6) \bmod 11=(6/6) \bmod 11=1$

$$\begin{aligned}
x_K &= (\lambda^2-x_{3G}-x_{3G}) \bmod p=(1^2-8-8)\\
&\quad \times \bmod 11=(-15) \bmod 11=7\\
y_K &= (\lambda(x_{3G}-x_K)-y_{3G}) \bmod p=(1(8-7)-3) \bmod 11=9
\end{aligned}$$

Similarly, we compute the following additions using Eqs. (3)–(6) accordingly. The random number r_1 for M_1 is 5, r_2 for M_2 is 7. Here $De(x)$ denotes the decryption of x using Eq. (10).

$$M_1+M_2=(5,2)+(8,3)=(3,6)$$

$$C_1=(M_1+r_1K,r_1G)=((5,2)+5(7,9),5(2,7))=((7,9),(3,6))$$
$$C_2=(M_2+r_2K,r_2G)=((8,3)+7(7,9),7(2,7))=((7,9),(7,2))$$
$$C_1+C_2=((7,9),(3,6))+((7,9),(7,2))$$
$$=((7,9)+(7,9),(3,6)+(7,2))=((2,4),(2,4))$$
$$De(C_1+C_2)=C_1-k*C_2=(2,4)-6*(2,4)$$
$$=(2,4)-(7,2)=(2,4)+(7,-2)$$
$$=(2,4)+(7,9)=(3,6)$$

Thus, $M_1+M_2=De(C_1+C_2)$.

$M_3=(1, 2)$ is not a point on the curve $E_{11}(1, 6)$ since $2^2 \pmod{11} \neq (1^3+1+6) \bmod 11$, assume random number r_3 for M_3 is 2. $M_1+M_3=(5, 9)$ is also not a point on curve $E_{11}(1, 6)$. Similarly using the above points K and G to encrypt M_3, we obtain $C_3=(M_3+r_3K, r_3G)=((1, 9), (5, 2))$. $De(C_3)=(1, 9)-6(5, 2)=(1, 2)$.

$$C_1+C_3=((3,2),(7,2)),De(C_1+C_3)=(3,9).$$
$$M_1+M_3\neq De(C_1+C_3).$$

The above example gives an illustration to show that the additive homomorphism is only satisfied for the points on the same elliptic curve.

3.2. EC-ElGamal based image encryption scheme

(1) *EC-ElGamal based system for secret image sharing*
Assume user A and B share secret images between each other over unsecured channel. The shared images are first encrypted to protect the images from eavesdropping. Besides, the shared images may need to be combined together at the intermediate node due to the bandwidth limitation. In order to facilitate the direct operation on the encrypted images, homomorphic cryptosystem should be adopted. Therefore, in this paper, we propose such a cryptosystem based on EC-ElGamal as shown in Figs. 1 and 2. *ECP*1 and *ECP*2 are obtained from the original images in the preprocessing. The encrypted images *EnECP*1 and *EnECP*2 are generated from *ECP*1 and *ECP*2 by using EC-ElGamal encryption method separately and added point by point to obtain *AEnECP*. *AEnECP* is transmitted over unsecured channel. To implement the secret images sharing between A and B, A extracts B's original image $P2$ by first decrypting the combined image *AEnECP*, then subtracting the EC form of its original image $P1$ i.e., *ECP*1 from *De(AEnECP)*.

(2) *Preprocessing*
Before using EC to encrypt the image pixels, the pixels are transferred into the form of points on elliptic curve $E_p(a, b)$ using the method in Ref. [20]. Given a plaintext unit *plainm* (one pixel value or combination value of several pixels), compute $x=plainm*L+j$, where $plainm*L+L<p$, $0\leq j<L$. Then, compute y satisfying Eq. (2). The failure probability of finding y (i.e., the square root of $f(x)=x^3+ax+b \bmod p$) is $1/2^L$. In practice, $L=30$ is enough according to Ref. [20]. $M=(x, y)$ is the point on $E_p(a, b)$ corresponding to plaintext unit *plainm*. In order to generate small data expansion, the length of *plainm* should be chosen close to the length of p, therefore, multiple image pixels should be combined together to obtain *plainm*. Fig. 3 shows an illustration for the image pixel combination. *plainm* is computed as $plainm=p1\|p2\|\ldots\|pm$, where $\|$ denotes the concatenation of $p1$, $p2$, …, pm which are the binary form of pixel values in the image. After the preprocessing step, all the image pixels are combined into blocks and then transferred to the corresponding EC points *ECP*1 or *ECP*2 as shown in Fig. 1.

(3) *Postprocessing*
After decryption, it is necessary to convert the decrypted points back into the image pixels. For each

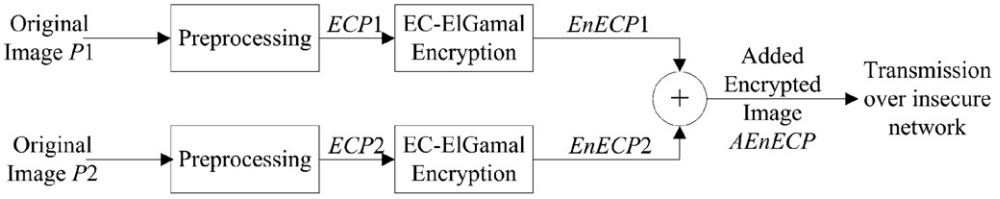

Fig. 1. Encryption and addition for two original images.

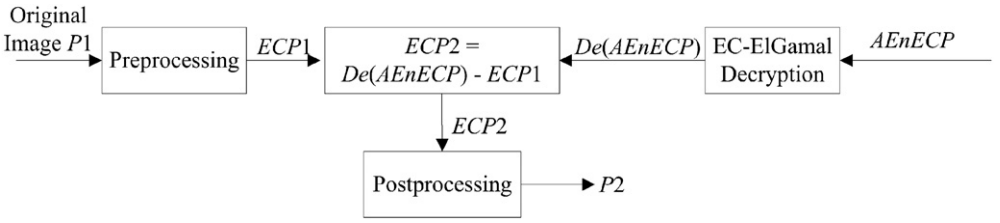

Fig. 2. Decryption and Extraction of the original image.

decrypted point (x, y), using the parameter L obtained in preprocessing, its corresponding pixels value $plainm' = [x/L]$, where $[x/L]$ indicates the maximal integers not greater than x/L. Then extract the original m pixels from $plainm'$.

(4) *Solving data expansion*
Since the proposed scheme is based on elliptic curve, it will generate four numbers modulo p which results in an expansion factor 4 (the size of the encrypted data divided by that of original data). To solve the data expansion using EC-ElGamal encryption, the random number r is chosen the same for one image but different for different images. Therefore, only one copy of rG for an encrypted image is saved. The expansion factor decreases nearly to 2.

The value y could be computed from x in Eq. (2), so y needs not be transmitted. But there exist two y values (y and $-y$) corresponding to one x, thus we replace y with a random number i having a byte size to further decrease the expansion factor to be less than 2. The random number i is computed as Eq. (13) to obtain a unique y. If $y < p/2$, substitute a random even number for y, else, substitute a random odd number for y. It is necessary to assign i randomly. Otherwise, if i is 0 for y and 1 for $-y$, it will generate many same pixel values 0 and 1.

$$y = i, \text{ where } \begin{cases} i \bmod 2 = 0, & y < p/2 \\ i \bmod 2 = 1, & y \geq p/2 \end{cases}, \quad i \in [0,255] \qquad (13)$$

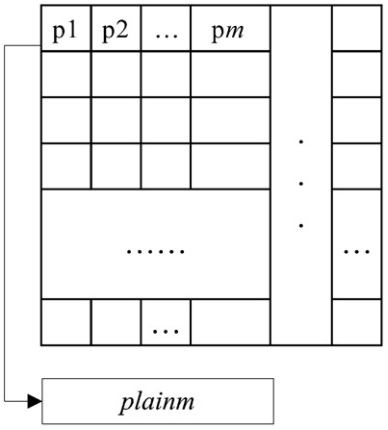

Fig. 3. Image pixel combination.

4. Experimental results and analysis

4.1. Parameter selection

Elliptic curve parameters should be selected in such a way that resists the attacks against ECC and with low computation overhead.

4.1.1. Length of p, plainm and L

All the computations are performed over finite field F_p. The length of p is regarded as the key length in ECC. To achieve the minimum security requirement, the length of p is 160 bits according to Table 1. In addition to the length of p, the parameters L and length of *plainm* in the preprocessing step should be determined before encryption.

The experiments are implemented based on LibTommath library [21] which realizes big number operation. The operation unit *mp_digit* in LibTommath takes up 28 bits i.e., 3.5 bytes. The length of plaintext unit *plainm* is determined satisfying the following two conditions. First, it should be close to but less than the length of p i.e. 160 bits. Second, it should have integral bytes from the aspect of computation, thus, the length of *plainm* is $3.5*i$ bytes where i is an even number. By combining the above two conditions, we obtain $i=6$ is the best choice, and thus 168 bits for the length of *plainm*. Besides, the failure probability $1/2^{30}$ is considered secure when embedding the *plainm* onto the elliptic curve point, thus L is chosen to be 30 which means L takes up 5 bits. Totally, L and *plainm* takes up $168+5=173$ bits. Since the parameters L, *plainm* should satisfy the condition $plainm*L+L<p$, *plainm*'s bit length plus L's bit length are less than p's bit length and thus p's length is at least 174 bits. Therefore, 174 bits is the least length of p where it achieves the minimum security requirements and satisfies the computation condition.

4.1.2. Values of p, a, b, G

Let n be the order of the base point $G \in E(\mathrm{F}_p)$, $\#E(\mathrm{F}_p)$ be the order of the elliptic curve over finite field F_p (i.e., the number of points on $E(\mathrm{F}_p)$). As stated in Ref. [13], the parameters p, a, b, and G are generated randomly satisfying some conditions to resist some known attacks as shown in Table 2.

4.2. Image encryption illustration

In this section, we test 100 standard images that are grayscale images (8 bits per pixel) with a resolution

Table 2
Conditions resisting known attacks.

Attack	Condition
Pohlig–Hellman &Pollard's rho	$\#\mathrm{E}(\mathrm{F}_p)=hn$, $h \leq 4$, n is prime and $n \geq 2^{160}$
Prime-field-anomalous curve	$\#\mathrm{E}(\mathrm{F}_p) \neq p$
Weil and Tate pairing	$(p^t-1) \bmod n \neq 0$, $1 \leq t \leq 20$
GHS Weil descent attack	p is prime or $p=2^j$ where j is a prime

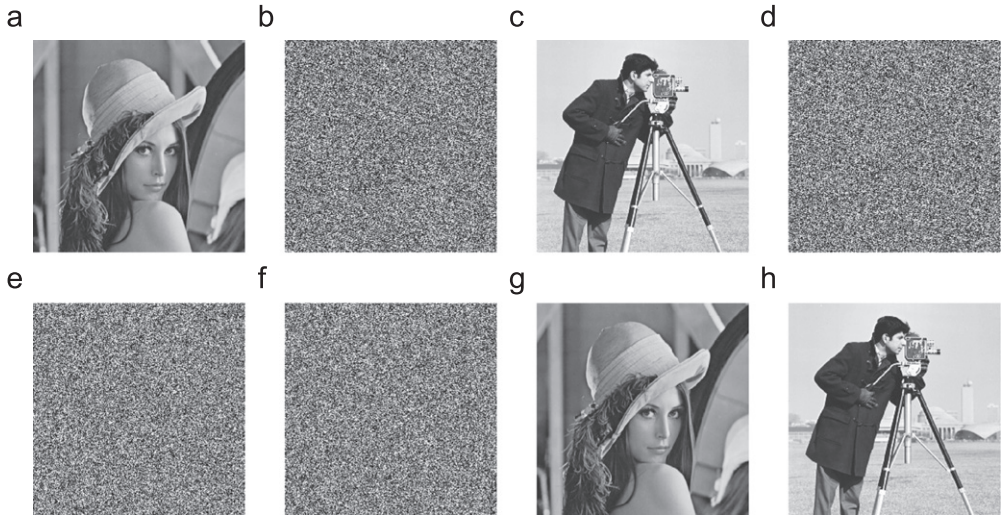

Fig. 4. An illustration of using the proposed encryption scheme. (a) Original *Lena*, (b) encrypted image of (a), (c) original *Cameraman*, (d) encrypted image of (c), (e) addition of images (a) and (c), (f) addition of images (b) and (d), (g) recovered image (a) from (c) and (f) and (h) recovered image (c) from (a) and (f).

256∗256 and size of 66 KB in BMP format. The values of parameters p, a, b, G, and k are listed in Table 6. Fig. 4 shows the visual illustration of the proposed scheme. The original images *Lena* and *cameraman* are shown in Fig. 4a and c, their encrypted images are shown in Fig. 4b and d. Original images illustrated in Fig. 4a and c are added directly as shown in Fig. 4e. Fig. 4f shows the image by adding the two encrypted images shown in Fig. 4b and d. Fig. 4g is the recovered image after subtracting the preprocessed image of Fig. 4c from decryption of Fig. 4f, and postprocessing. From the visual point of view, the proposed method has visual security for the encryption and homomorphic addition.

4.3. Key space analysis

Key space size is the total number of different keys that can be used in the encryption. For a secure image encryption, the key space should be large enough to make brute force attacks infeasible [19]. The proposed scheme has 2^{174} different combinations of the secret key. Thus, the key space of the proposed scheme is extensively large enough to resist the exhaustive brute-force attack.

4.4. Key sensitivity test

An efficient cryptosystem should be sensitive to the private key. Randomly change one bit of any private key $K1$ to obtain key $K2$. Fig. 5a and b shows the encrypted images by using $K1$ and $K2$ to encrypt the original *Lena* image. There is no difference from human vision; however, the difference ratio is 99.55%. The difference ratio (DR) is the number of different pixels between two images (NDP) divided by the number of total pixels per image (NTP) *as* illustrated in Eq. (14). Higher difference ratio indicates the proposed scheme is more sensitive to the key and thus higher security. The difference ratio for

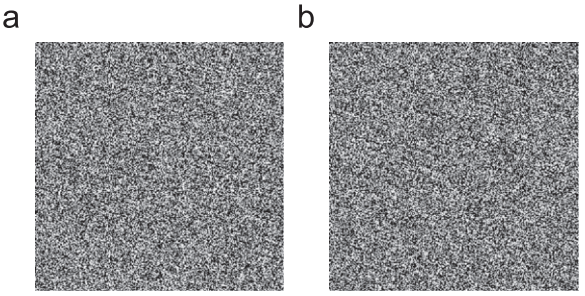

Fig. 5. Key sensitivity test. (a) Encrypted *Lena* using $K1$. (b) Encrypted *Lena* using $K2$.

100 images encrypted by $K1$ and $K2$ are statistically shown in Fig. 6. It shows that most of the encrypted images change with a difference ratio close to 99.55%, and even the least difference ratio could achieve 97.0%. This indicates that the scheme is very sensitive to the key.

$$DR = (NDP/NTP)*100\% \quad (14)$$

4.5. Histogram analysis

An image histogram illustrates how pixels in an image are distributed by graphing the number of pixels at each gray level. A good image encryption scheme should always generate a cipher image having uniform histogram for any plain image. We have calculated and analyzed the histograms of the several encrypted images as well as its original images that have widely different content. Fig. 7 gives the histograms for *Lena* and *Cameraman* images, respectively. It is shown that the distribution of the gray values in the encrypted images (Fig. 7b and d) is quite different from that of original images (Fig. 7a and c) and is nearly uniform implying a good statistical property.

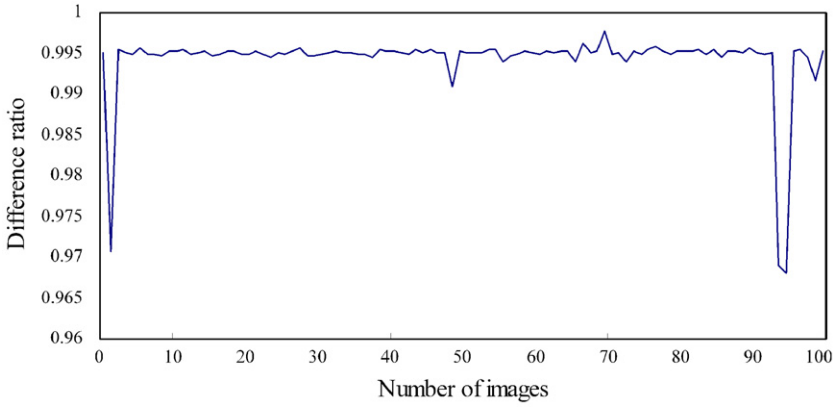

Fig. 6. Difference ratio for 100 images.

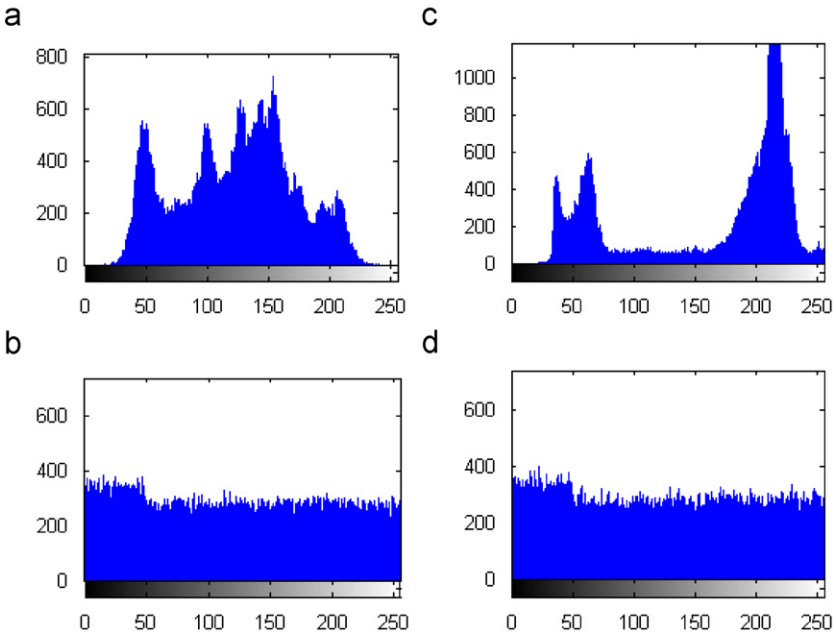

Fig. 7. Histogram of *Lena* and *Cameraman*. (a) Histogram of original image *Lena*. (b) Histogram of encrypted image *Lena*. (c) Histogram of original image *Cameraman*. (d) Histogram of encrypted image *Cameraman*.

4.6. Information entropy analysis

Entropy is a statistical measure of randomness in information theory. The entropy $H(m)$ is computed as in Eq. (15) where $p(m_i)$ represents the probability of symbol m_i and the entropy is expressed in bits.

$$H(m) = -\sum_{(i=0)}^{(2^N-1)} p(m_i) * \log_2 p(m_i) bits \quad (15)$$

Suppose the grayscale image has 2^8 gray levels with equal probability, $m=\{m_0, m_1, \ldots, m_{255}\}$. According to Eq. (15), we obtain its entropy value $H(m)=8$. Actually, the entropy value of a grayscale image is generally smaller than the ideal value 8 because the pixel values are seldom random. But for the encrypted grayscale image, their entropy should ideally be 8; otherwise there exists certain degree of predictability which threatens its security. The entropy values for 100 plain images and corresponding cipher images are given in Fig. 8. The vertical axis denotes the entropy values within [0,8], and the horizontal axis stands for image sequential number i within [0,99]. Curve with circles signifies entropy values for the 100 original images, and the other curve is for the 100 encrypted images. Most of the entropy values for the 100 cipher images are very close to the ideal value 8. This implies that the information leakage in the proposed encryption process is negligible and the encryption scheme is secure against the entropy based attack.

4.7. Correlation of adjacent pixels

For an ordinary image having definite visual content, each pixel is highly correlated with its adjacent pixels

Fig. 8. Entropy value for 100 images and corresponding encrypted images.

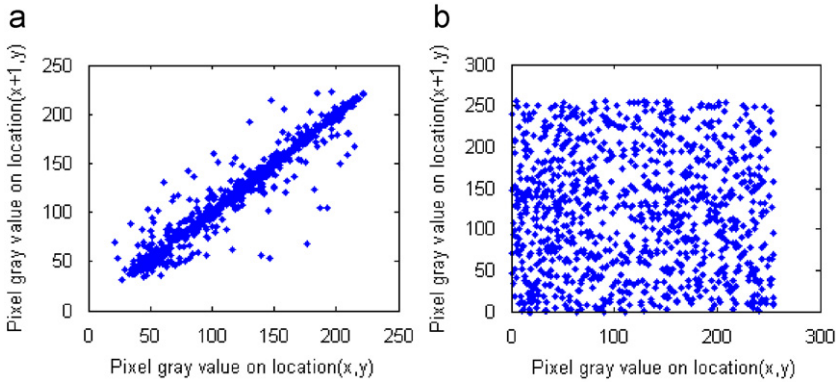

Fig. 9. Two horizontally adjacent pixels correlation in original image/encrypted image, respectively.

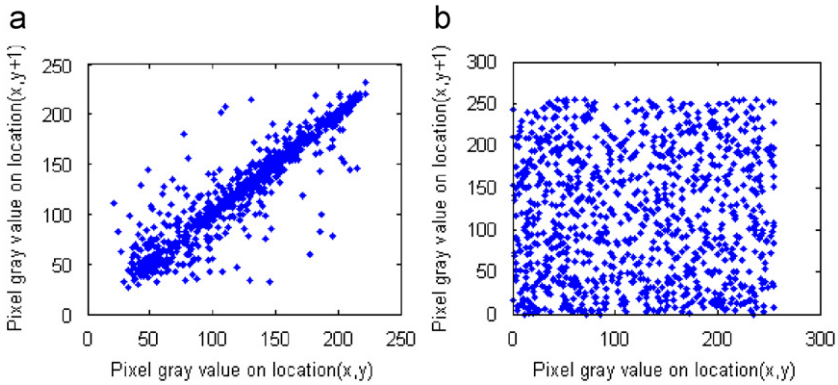

Fig. 10. Two vertically adjacent pixels correlation in original image/encrypted image, respectively.

either in horizontal or vertical direction. An ideal encryption technique should produce cipher images with no such correlation in the adjacent pixels (correlation coefficient ≈ 0) [22,23]. The visual testing of the correlation of adjacent pixels can be done by plotting the distribution of the adjacent pixels correlation in the plain image and its corresponding cipher image. The correlation coefficient of the adjacent pixels is calculated as Eq. (16) where $Avg(x)=mean\ (x_i)$ and x, y are gray values of two adjacent pixels in the image. For the proposed scheme, the correlation coefficients of 1000 randomly selected pairs of horizontally, vertically and diagonally adjacent pixels are computed. The corresponding distribution for the horizontal, vertical and diagonal directions are shown in Figs. 9–11. They demonstrate that the proposed encryption scheme shows good performance with balanced $0\sim1$ ratio. The values of correlation coefficient shown in Table 3 and Figs. 9–11 show that the two adjacent pixels in the plain images are highly correlated to each other , whereas the values obtained for cipher images are

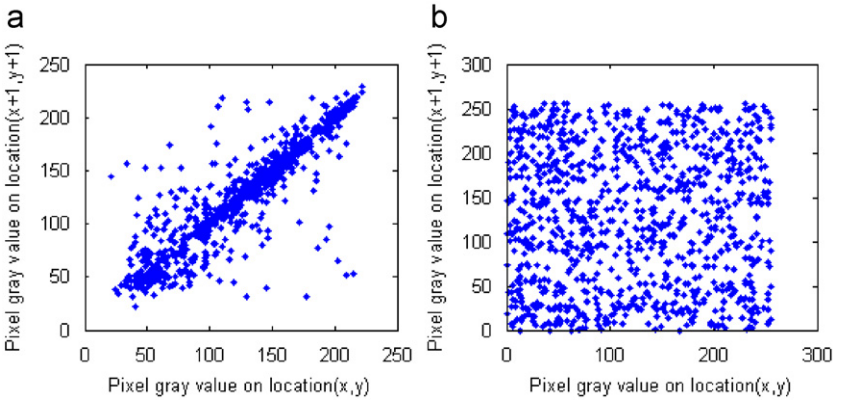

Fig. 11. Two diagonally adjacent pixels correlation in original image/encrypted image, respectively.

Table 3
Correlation coefficients of adjacent pixels in the two images for *Lena*.

Correlation coefficient	Original image	Encrypted image
Horizontal	0.9691	−0.0146
Vertical	0.9492	−0.0028
Diagonal	0.9288	−0.0240

Table 4
Parameters for RSA.

p	912F5842C71B66DFD96931A264919EED0C3711 A3005DD5F27FD8BE709C000C6A3EAC93786 A5E1955C5BA00EE543D25B9872B07 4876C92C1E6ACDC0C19F138011
q	D8398E1890D23AECC7528EB171A9CB 77EE086C99BB46FD41EA83302ECF0B2EB5FD7C 2B06712C3DB10CC56E694732100722BF50D25D 7B0EFDEBE1EC71A4B8A5CD
n	7AA096993064A0819E5FD698F136655 20A928BAD5E9216CB98F9D75601173815C963 6FD19BBC7F0A6530711250524287D8300601C 5225C625A95BCBFA8AC6493E3626FDFBC0E6F 4F5CD5F1A7E00222938D69D3FA44BBCA10D48 CC4BC140553B49C8748C67B5250E8B705C40 977C3636E5DFB0500F0FA4C2E14AF2E23E460829D
e	3A845C27BB45F959C1D00512AB1D510B4 1333BBBCA72F403D01810C05CB13C7D0C665B 1C53D4646B2006C2420B74CD3ED48512202DF 02AADD5CCF74310BF6B17037F7E7398308060B 0EE102AFA3572824DB755F692842E5264EB428C0 B7A906A914D861E8DA2FC480D5DFD86CDF397F681 DE6DAD8512ED0E13C5C9D0DCB04657
d	C3D8FAA178BD01AC36242A57077DF8E7

close to 0 (zero co-correlation).

$$r_{xy}=\left(\sum_{i=1}^{N}(x_i-Avg(x))(y_i-Avg(y))\right)\Big/\left(\left(\sum_{i=1}^{N}(x_i-Avg(x))^2\right)^{1/2}*\left(\sum_{i=1}^{N}(y_i-Avg(y))^2\right)^{1/2}\right) \quad (16)$$

Table 5
Parameters for ElGamal.

p	BFA40429207C500875DF402E66265B52D972C8534 139BED9B6FE3B3A4B5FF99A9C559E57A6B832A4 DC01D7737754B4990DBE332C65BB6C308AFBE17 31E656BB6F15D740DD2426789A53B6513B6343A9 B4D238F73839060F55354B16CF4A17D326642F015 4C24D6FFF91652D7CDCDD3A2C22B44EE42C183703 C9215EC76191855
k	C2C03A1DEEBA8BAA0677866805151EC9
g	9E46B5A21E174FEA46DC89E78F7B7CDB 875BC48834793DE7FAB90A1DBE7ADCAB4C6C7 0E5F6A65AA57EF30240376DECB9664E3C35EE 0279AF2D8191ED21DB1DB3CF14E736E533A9D 39E961801E6882185BF1673E5E31E3D1EAAA83 9992F79E95CAEE6217E31E7C5DD6C6CD413E 5F4A5A5585B717E59F3129C3BD70A0A31
y	563221C099EBFD234200761F323248B99 4FFDDB914D5DBFC0DF33B8A3B1A4F830803ED 14042AFC72FDA56C212D367025D04DCE321104 18DEED5B665E49556336F66ABB066C6A38C 22D429701A6D9C213CCACFCE466B7279FF7A1CC7FFA4140C0 9D39635B89EB5E6E465CD391DB39E761751521A99B927 16CF9D4627B611B3648

5. Comparison among several encryption schemes

The encryption and decryption performance of the proposed scheme is compared with RSA and ElGamal. All the simulations were done by VC++6.0 in a computer of Intel Core i3 CPU 530 of 2.93 GHz and 3.36 GB of RAM 1.17 GHz. The key length using RSA, ElGamal, and the proposed scheme are determined according to the discussions in the introduction. The key values are listed in Tables 4–6. The keys denotation for RSA in Table 4 and those for ElGamal in Table 5 are the same as the denotation in Ref. [3], respectively. The experiments are operated on *Lena* image (Fig. 4a) with original image size of 66 KB. In Table 7, the size of 372 KB, 479 KB, 478 KB are the size of encrypted images by applying the proposed scheme, ElGamal and RSA based methods without using any data expansion reduction method, and the size of 75 KB is the encrypted image size using the proposed data expansion reduction method after encryption. It is shown that the proposed scheme has a shorter key length while overall takes less encryption/

Table 6
Parameters for the proposed scheme.

p	37A925C980A8BC8BE6AB4F3ECF34279567CB806F6B5F
a	205E14A1
b	DE7EA83755
G's x coordinate	888EA0E68AAC5411398EBB5F34607D7CEDB4952EDF3
G's y coordinate	10D18D8456716F3CD0C1404246DA256C89F21752774
k	9F01BC57517872255A42A41FFDE74BAB

Table 7
Performance among RSA, ElGamal, and the proposed scheme.

Method	Encrypted image size (KB)	Key length	Encryption time (s)	Decryption time (s)
RSA	478	1024 bits	90.328	13.125
ElGamal	479	1024 bits	18.125	0.406
Proposed scheme	372	174 bits	7.484	0.906
	75		6.232	2.353

decryption time under the same security requirement. Since the decryption time is short, the proposed scheme is more efficient than the method in Ref. [10].

6. Conclusion and future work

In this paper, a homomorphic image encryption for sharing secret images based on EC-ElGamal is proposed. It achieves the same security level with smaller key size in contrast with RSA and ElGamal. The efficiency and security are exploited by the experimental results and analysis which concluded better performance than those methods based on RSA, ElGamal, and current work based on EC-ElGamal. Due to the close relation between videos and images, sharing secret videos will be further studied in the future work.

Acknowledgments

The authors thank the reviewers for their helpful comments. We also thank Prof. Mohamed Amin, Menoufia University, Egypt, for his effort and valuable comments that are greatly helpful to improve the clarity and quality of this work. This work is supported by the National Natural Science Foundation of China (Project number: 60832010).

References

[1] T.H. Chen, C.S. Wu, Efficient multi-secret image sharing based on Boolean operations, Signal Processing 91 (1) (2011) 90–97.
[2] C.N. Yang, A.G. Peng, T.S. Chen, MTVSS: (M)isalignment (T)olerant (V)isual (S)ecret (S)haring on resolving alignment difficulty, Signal Processing 89 (8) (2009) 1602-1624.
[3] C. Fontaine, F. Galand, A survey of homomorphic encryption for nonspecialists, EURASIP Journal on Information Security 2007 (2007), Article ID 13801, doi:10.1155/2007/13801.
[4] Z. Erkin, A. Piva, S. Katzenbeisser R.L. Lagendijk, J. Shokrollahi, G. Neven, M. Barni, Protection and retrieval of encrypted multimedia content: when cryptography meets signal processing. EURASIP Journal on Information Security 2007 (2007), Article ID 78943, doi:10.1155/2007/78943.
[5] N. Islam, W. Puech, R. Brouzet., A homomorphic method for sharing secret images, Digital Watermarking LNCS 5703 (2009) 121–135.
[6] R. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public key cryptosystems, Communications of the ACM 21 (2) (1978) 120–126.
[7] R. Cramer, R. Gennaro, B. Schoenmakers, A secure and optimally efficient multi-authority election scheme, European Transactions on Telecommunications 8 (5) (1997) 481–490.
[8] K. Peng, F. Bao, Efficient multiplicative homomorphic e-voting, Information Security LNCS 6531 (2010) 381–393.
[9] T. E1Gamal, A public key cryptosystem and a signature scheme based on discrete logarithms, IEEE Transactions on Information Theory 31 (4) (1985) 469–472.
[10] O. Ugus, D. Westhoff, R. Laue, A. Shoufan, S.A. Huss, Optimized implementation of elliptic curve based additive homomorphic encryption for wireless sensor networks, CoRR abs/0903.3900, 2009.
[11] J.M. Bahi, C. Guyeux, A. Makhoul, Efficient and robust secure aggregation of encrypted data in sensor networks, in: Proceedings of the Fourth International Conference on Sensor Technologies and Applications, Venice, Italy, IEEE, 18–25 July 2010, pp. 472–277.
[12] J.M. Bahi, C. Guyeux, A. Makhoul, Secure data aggregation in wireless sensor networks: homomorphism versus watermarking approach, Ad Hoc Networks, LNICST 49 (2010) 344–358.
[13] N. Koblitz, A. Menezes, S. Vanstone, The state of elliptic curve cryptography, Designs, Codes and Cryptography, 19, Kluwer Academic Publishers, 2000, pp. 173–193.
[14] S.D. Rane, W. Sun, A. Vetro, Secure distortion computation among untrusting parties using homomorphic encryption, in: Proceedings of the 16th IEEE International Conference on Image Processing, Cairo, Eygpt, IEEE, 7–10 November 2009, pp. 1485–1488.
[15] P. Paillier, Public-key cryptosystems based on composite degree residuosity classes, in: Advances in Cryptology—Eurocrypt '99, LNCS, vol. 1592, 1999, pp. 223–238.
[16] M. Kuribayashi, H. Tanaka, Fingerprinting protocol for images based on additive homomorphic property, IEEE Transactions on Image Processing 14 (12) (2005) 2129–2139.
[17] R. Gennaro, J. Katz, H. Krawczyk, T. Rabin, Secure network coding over the integers, in: Public Key Cryptography, LNCS, vol. 6056, 2010, pp. 142–160.
[18] M. Upmanyu, A.M. Namboodiri, K. Srinathan, C.V. Jawahar, Efficient privacy preserving video surveillance, in: Proceedings of the 12th IEEE International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009, pp. 1639–1646.
[19] P. Christof, Applied cryptography and data security. 〈http://www.crypto.ruhr-uni-bochum.de/en_lectures.html〉.
[20] N. Koblitz, A course in number theory and cryptography, second edition, Springer-Verlag, New York, 1994, pp. 178–185.
[21] 〈http://libtomcrypt.org/tfm/features.html[EB/OL]〉.
[22] M. Amin, O.S. Faragallah, A.A. Abd El-Latif, A chaotic block cipher algorithm for image cryptosystems, Communication in Nonlinear Science and Numerical Simulation 15 (2010) 3484–3497.
[23] X. Tong, M. Cui, Image encryption scheme based on 3D baker with dynamical compound chaotic sequence cipher generator, Signal Processing 89 (4) (2009) 480–491.