Optica Applicata, Vol. L, No. 3, 2020
DOI: 10.37190/oa200303

Image encryption based on permutation polynomials over finite fields

JIANHUA WU[1], HAI LIU[1], XISHUN ZHU[2*]

[1] School of Information Engineering, Nanchang University, Nanchang 330031, China

[2] Gongqing College, Nanchang University, Jiujiang 332020, China

[*] Corresponding author: zxs171039@aliyun.com

In this paper, we propose an image encryption algorithm based on a permutation polynomial over finite fields proposed by the authors. The proposed image encryption process consists of four stages: *i*) a mapping from pixel gray-levels into finite field, *ii*) a pre-scrambling of pixels' positions based on the parameterized permutation polynomial, *iii*) a symmetric matrix transform over finite fields which completes the operation of diffusion and, *iv*) a post-scrambling based on the permutation polynomial with different parameters. The parameters used for the polynomial parameterization and for constructing the symmetric matrix are used as cipher keys. Theoretical analysis and simulation demonstrate that the proposed image encryption scheme is feasible with a high efficiency and a strong ability of resisting various common attacks. In addition, there are not any round-off errors in computation over finite fields, thus guaranteeing a strictly lossless image encryption. Due to the intrinsic nonlinearity of permutation polynomials in finite fields, the proposed image encryption system is nonlinear and can resist known-plaintext and chosen-plaintext attacks.

Keywords: finite field, permutation polynomial, scrambling, diffusion, image encryption.

1. Introduction

Digital image with its visual and esthetic impact has become the main medium in modern information society. Image encryption, in particular, is urgently needed but it is a challenging task. It is quite different from text encryption due to some intrinsic properties of images such as bulky data capacity and high redundancy, which are generally difficult to handle by using traditional techniques. Many new image encryption schemes have been proposed in recent years. In [1], for example, image encryption is done by employing chaotic sequences. In [2] and [3], image encryption schemes based on local random phase encoding in fractional Fourier transform and gyrator transform domains, respectively, are proposed; in such schemes, random phase encoding is iteratively applied to different regions of an input image. REFREGIER and JAVIDI first proposed a dual random phase image encryption algorithm based on input plane and Fourier plane and its optical implementation [4]. Thanks to the fractional order, the free parameters can be used as a secondary key, thus increasing the key space. Some encryption techniques use fractional Fourier transform, discrete cosine transform, Arnold transform, discrete fractional random transform, multiple-parameter discrete fractional random transform, reality-preserving fractional discrete cosine transform, fractional Hartley transform, discrete cosine Stockwell transform, *etc*. [5–11]. Venturinit and Duhamel proposed a method of consolidation processing of arbitrary fractional order transform, providing a way to solve the real value encryption problem [12]. Since the image encryption algorithm based on real fractional transform can solve the problem of complex value output, many reality-preserving encryption algorithms are constantly emerging. For example, Lang first implemented a reality-preserving multi-parameter Fourier transform and applied it for image encryption [13]. Wu *et al*. [14] proposed an image encryption algorithm based on a reality-preserving fractional discrete cosine transform, in which a real-valued output facilitates efficient storage and transmission. However, these image encryption systems are a linear one which are relatively weak in security in comparison to nonlinear encryption systems. Consequently, Zhou *et al*. proposed a nonlinear color image encryption algorithm based on the reality preserving fractional Mellin transform [15]. The algorithm guarantees both the real value of the transformed image and the nonlinearity of the encryption system. In [16], a new image compression-encryption algorithm is proposed, which accomplishes encryption and compression simultaneously.

The above encryption algorithms have a common problem that the transmission data still needs to be quantized. It is obvious that the decrypted image will not be free of distortion and hence the quality of the image will be degraded.

For this purpose, researchers further proposed various transforms based on finite fields. Lima and others have proposed the finite field fractional Fourier transform in [17], fractional cosine and sine transforms over finite fields in [18], cosine transforms over finite fields of characteristic 2 [19, 20] successively. Subsequently, image encryption based on the fractional Fourier transform over finite fields [21] and image encryption based on the finite field cosine transform [22] are proposed by Lima and others. The most attractive property of this approach is that round-off errors are avoided and the decrypted image is rigorously equal to the original one.

The permutation polynomial also has the same encryption effect. Permutation polynomials over finite fields are reversible mappings. Since the 1950s, due to the need of cryptography, the study of permutation polynomials over finite fields has attracted extensive attention from mathematicians and engineering technicians. In [23], Wan and Lidl have given the permutation polynomials of the form $x^r f(x^{(q-1)/d})$ and the corresponding criteria. In [24], Wang found two new permutation polynomials based on Hermite discriminant and combinatorial theory. Cao and Qiu deeply studied Dickson polynomial [25]. When studying Kloosterman sums over F_{2^m}, Helleseth and Zinoviev found a special type of permutation polynomials [26]. In [27], Yuan and Ding attributes this polynomial to the form like $(x^{2^i} + x + \delta)^s + x$. The permutation property of this formal polynomial has attracted much attention from some researchers [28–30]. Akbary *et al*. proposed a new method [31], which is called AGW, to construct some new forms of permutation polynomials. Subsequently, Zeng *et al.* used this method to construct the permutation polynomial of the form like $(x^{p^i} - x + \delta)^{s_1} + (x^{p^i} - x + \delta)^{s_2} + L(x)$ [32, 33]. As is known, the S-box is a basic component of the symmetric encryption algorithm. The construction of S-boxes requires that the permutation polynomials have low difference properties [34]. The box component of Advanced Encryption Standard chooses a 4-differential replacement. Due to its properties of cryptography over finite fields, the polynomial permutation is widely applied to image encryption.

On the basis of the above discussions, a novel image encryption algorithm based on permutation polynomial over finite fields is proposed in this paper. In this algorithm, the plaintext image is firstly mapped into finite field. Then, a pre-scrambling of pixels' positions is conducted based on the permutation polynomial proposed and parameterized by the authors. This pre-scrambling behaves similarly as Arnold transforms. Subsequently, an operation of diffusion is carried out by a symmetric matrix transform over finite field. Finally, a post-scrambling is completed based on the same permutation polynomial with different parameters. Theoretical analysis and simulation results show that the new parameterized permutation polynomials ensure a sufficiently large key space. All the encryption steps over finite fields form a strictly lossless image encryption. In addition, the proposed image encryption scheme has a high efficiency and feasibility against various common attacks. At the same time, the intrinsic nonlinearity of the permutation polynomial guarantees the nonlinearity of the encryption algorithm, so this algorithm possesses the ability to resist known-plaintext and chosen-plaintext attacks.

The rest of this paper is organized as follows. In Section 2, some related work is given as fundamentals. Then the proposed image encryption and decryption algorithm based on permutation polynomials are described in detail in Section 3. Next in Section 4, the main aspect related to performance of image cryptosystem is discussed and simulation results are shown. Finally, some concluding remarks are drawn in the last section.

2. Related work

2.1. Basic knowledge of permutation polynomials

Let p be a prime, n be a positive integer, and F_{p^n} be the finite field with p^n elements. A polynomial $f(x)$ in $F_{p^n}[x]$ is said to be a permutation polynomial over F_{p^n} if it induces a permutation from F_{p^n} to F_{p^n}. Permutation polynomials over finite fields have been an important subject of study for a long time and have wide applications in coding theory, cryptography and sequence designs [35].

When $p = 2$, a polynomial $f(x)$ in $F_{p^n}[x]$ can be written as

$$f(x) = \sum_{i=0}^{2^n-2} a_i x^i, \qquad a_i \in F_{2^n} \tag{1}$$

where i can be written as $i = \sum_{j=0}^{n-1} b_j 2^j,\ b_j = 0 \text{ or } 1$.

Denote $W_2(i) = \sum_{j=0}^{n-1} b_j$ as weight of i. Then the algebraic degree of $f(x)$ can be denoted by $\deg f(x) = \max\{W_2(i)|a_i \neq 0\}$.

Denote by #S the cardinality of a set S. For a function f: $F_{2^n} \rightarrow F_{2^n}$, its differential spectrum is defined by

$$\{\delta_f(a, b)|(a, b) \in F_{2^n}^* \times F_{2^n}\} \tag{2}$$

where $\delta_f(a, b)$ for any $a \in F_{2^n}^*$, $b \in F_{2^n}$ is given by

$$\delta_f(a, b) = \#\{x \in F_{2^n}|f(x+a) + f(x) = b\} \tag{3}$$

where # denotes cardinality of a set, which is the number of elements for a finite set. In addition, the differential uniformity of f, denoted by Δ_f, is defined by

$$\Delta_f = \max_{a \neq 0,\, b \in F_{2n}} \delta_f(a, b) \tag{4}$$

If f is a function with differential uniformity Δ_f, then it is called a differentially Δ_f-uniform function. Note that if the equation $f(x+ a) + f(x) = b$ has one solution x, then it has $x + a$ for a second solution. Thus $\delta_f(a, b)$ is always even. Functions used in block ciphers should have a low differential uniformity to allow a high resistance against the differential cryptanalysis [33]. In this sense, a deferentially 2-uniform function, called an almost perfect nonlinear function, is optimal.

Denote the inverse matrix C^{-1} of the matrix $C = [a_{ij}]$ over finite fields as

$$C^{-1} = \frac{1}{|C|}\begin{bmatrix} A_{11} & A_{21} & \dots & A_{n1} \\ A_{12} & A_{22} & \dots & A_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ A_{1n} & A_{2n} & \dots & A_{nn} \end{bmatrix} \tag{5}$$

where A_{ij} is the algebraic co-factor of a_{ij} that can be gotten by removing the row and column where a_{ij} resides, and $|C|$ is the value of determinant of the matrix C.

2.2. Some useful lemmas about permutation polynomials

Lemma 1 [35]. Let F_q be of characteristic p. Then $f \in F_q[x]$ is a permutation polynomial of F_q, if and only if the following two conditions hold:

1) $f^{q-1}(x) \bmod x^q - x$ has a degree $q - 1$;

2) for each integer t with $1 \leq t \leq q-2$ and $t = 0 \bmod p$,
the reduction of $f(x)^t \bmod x^q - x$ has a degree less than or equal to $q - 2$.

Lemma 2. The polynomial $f(x) = x^{59} + x^{209} + x^{254}$ is a differentially 4-uniform permutation over F_{2^8}.

Proof: By Lemma 1, the polynomial $f(x) = x^{59} + x^{209} + x^{254}$ permutes F_{2^8} if and only if the following two conditions hold:

1) The following condition holds:

$$\begin{aligned} f^{255}(x)\bmod(x^{256}-x) &\equiv (x^{59}+x^{209}+x^{254})^{255}\bmod(x^{256}-x) \\ &\equiv \sum_{i+j+k=255} x^{59i+209j+255k}\bmod(x^{256}-x) \\ &\equiv x^{255}+\ldots \end{aligned}$$

Therefore, there are positive integers i, j, k, l that satisfy the following equations:

$$\begin{cases} 59i + 209j + 254k - 256l = 255 \\ i + j + k = 255 \end{cases} \tag{6}$$

We can get $i = 1$, $j = 64$, and $k = 190$ that is a solution of Eq. (6) and there are totally 129 solutions. Since $129 \equiv 1 \bmod 2$, we have that $f^{255}(x)\bmod(x^{256}-x)$ has degree 255.

2) And there are positive integers i, j, k, l that satisfy the following equations:

$$\begin{cases} 59i + 209j + 254k - 256l = 255 \\ i + j + k < 255 \end{cases} \tag{7}$$

each set of i, j, k consists of numbers from $\{1, 2a, 4b, 8c, 16d, 32e, 64f, 128g\}$ without repetition, where $a, b, c, d, e, f, g \in \{0, 1\}$. From the remainder theorem, there is not a suitable set of i, j, k which satisfies Eq. (7).

Then the polynomial $f(x) = x^{59} + x^{209} + x^{254}$ is a permutation polynomial over F_{2^8}.

We consider the number of solutions of the following equation with $a \in F^*_{2^8}$, and $b \in F_{2^8}$

$$\begin{aligned} 0 &= f(x) + f(x+a) + b \\ &= x^{59} + x^{209} + x^{254} + (x+a)^{59} + (x+a)^{209} + (x+a)^{254} \end{aligned} \tag{8}$$

By enumeration, we can get at most six solutions for Eq. (8). Furthermore, let N_i denote the number of the pairs $(a, b) \in F^*_{2^8} \times F_{2^8}$ such that

$$\#\{x \in F_{2^8} \mid f(x+a) + f(x) = b\} = i$$

for $i = 0, 2, 4, 6$, then the differential spectrum of $f(x)$ is given as follows:

$$N_0 = 39375 = 2^{15} + 2^{12} + 2^{11} + 2^9 - 2^6 + 2^4 + 1$$

$$N_2 = 19890 = 2^{14} + 2^{11} + 2^{10} + 2^9 - 2^6 - 2^4 + 2$$

$$N_4 = 5295 = 2^{12} + 2^{10} + 2^7 + 2^6 - 2^4 - 1$$

$$N_6 = 720 = 2^9 + 2^8 - 2^6 + 2^4$$

Lemma 3. The function $f(x) = x^d$ is a permutation polynomial over F_{2^n} if and only if d and $2^n - 1$ are relatively prime:

$$\gcd(d, 2^n - 1) = 1 \tag{9}$$

where $\gcd(\cdot,\cdot)$ returns the greatest common divisor of two arguments. When it returns 1, the two arguments are relatively prime; $\gcd(\cdot,\cdot)$ is often abbreviated as $(\cdot,\cdot)$.

3. The proposed encryption and decryption algorithm

In this part, we propose an algorithm of four-stage image encryption and decryption: 1) a mapping from pixel gray-levels into finite field, 2) a pre-scrambling of pixels' positions based on the parameterized permutation polynomial, 3) a symmetric matrix transform over finite fields which completes the operation of diffusion and, 4) a post -scrambling based on the same permutation polynomial with different parameters, as shown in Fig. 1. The decryption process is the inverse version of the encryption one, which takes the encrypted results as the inputs and outputs the decrypted images. The encrypted results may be distorted after transmission through the channel.

3.1. Mapping the pixel values into finite field

Given a gray plain image P with pixel values $i \in \{0, 1, ..., 255\}$, the mapping from integers into finite field F_{2^8} is completed by h: $0 \to 0$, $i \to \omega^i$, $i = 1, 2, ..., 255$, with $\omega \in F_{2^8}$ being a 255-order solution of an irreducible polynomial.

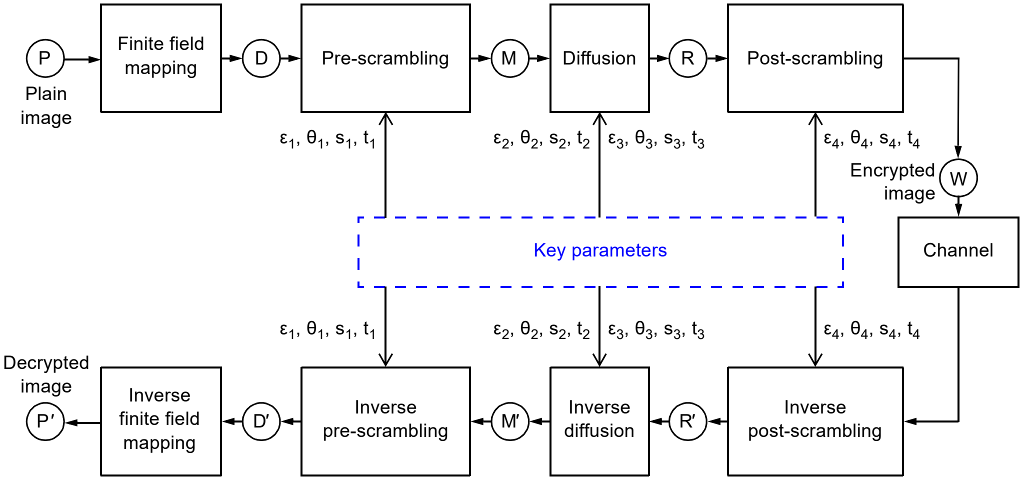

Fig. 1. The illustrative diagram of the proposed algorithm.

3.2. Pre-scrambling of pixels' positions based on permutation polynomial

Because neighboring pixels of natural images have a strong correlation in common, an image encryption scheme should be able to reduce these correlations. In this encryption stage, an algorithm similar to a cat map is taken to scramble the locations of image pixels.

The proposed permutation polynomial is chosen and parameterized with parameters ε_1, θ_1, and α_1, as:

$$f^{\alpha_1}_{\varepsilon_1,\,\theta_1}(m) \;=\; \left[(m+\varepsilon_1)^{59} + (m+\varepsilon_1)^{209} + (m+\varepsilon_1)^{254} + \theta_1\right]^{\alpha_1} \tag{10}$$

where $\varepsilon_1, \theta_1 \in \mathrm{F}_{2^8}$, and $0 < \alpha_1 = s_1/t_1 < 255$, and s_1, t_1 are positive integers with $(s_1, 255) = 1$, $(t_1, 255) = 1$, and $(s_1, t_1) = 1$.

Given an array $D(m, n)$ on F_{2^8}, $m, n = 0, 1, ..., 255$, position scrambling is implemented by:

$$M(u, v) = D(m, n), \qquad u, v = 0, 1, ..., 255 \tag{11}$$

where $u = f^{\alpha_1}_{\varepsilon_1,\,\theta_1}(m)$ and $v = f^{\alpha_1}_{\varepsilon_1,\,\theta_1}(n)$. $\varepsilon_1, \theta_1, s_1,$ and t_1 in Eq. (10) are used as cipher keys.

3.3. Diffusion based on a symmetric matrix transform

In order to further scramble and diffuse the encryption results, a new symmetric reversible matrix is constructed over finite field. The constructed matrix is not only symmetric but also reversible and can be parameterized. The property of symmetry and reversibility plays a very important role in the process of image decryption.

First, a matrix $T = [t_{ij}]_{256\times 256}$ is constructed, where

$$t_{ij} \;=\; \begin{cases} \mu^{i+j}, & i+j \le 255 \\ \mu^{510-i-j}, & i+j > 255 \end{cases} \qquad i, j = 0, 1, ..., 255 \tag{12}$$

where $\mu \in \mathrm{F}_{2^8}$ and its multiplicative order is 255.

From (12), it is known that T can be expanded to:

$$T \;=\; \begin{bmatrix} 1 & \mu & \mu^2 & \dots & 1 \\ \mu & \mu^2 & \mu^3 & \dots & \mu^{254} \\ \mu^2 & \mu^3 & \mu^4 & \dots & \mu^{253} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \mu^{254} & \mu^{253} & \dots & 1 \end{bmatrix} \tag{13}$$

Then, T is parameterized with parameters ε_2, α_2, θ_2 and ε_3, α_3, θ_3 as:

$$T_1 \;=\; T^{\alpha_2}_{\varepsilon_2,\,\theta_2} \;=\; \left[f^{\alpha_2}_{\varepsilon_2,\,\theta_2}(t_{ij})\right]_{256\times 256} \tag{14}$$

and

$$T_2 = T^{\alpha_3}_{\varepsilon_3,\,\theta_3} = \left[f^{\alpha_3}_{\varepsilon_3,\,\theta_3}(t_{ij})\right]_{256\times 256} \tag{15}$$

respectively, where

$$f^{\alpha_2}_{\varepsilon_2,\,\theta_2}(t_{ij}) = \left[(t_{ij}+\varepsilon_2)^{59}+(t_{ij}+\varepsilon_2)^{209}+(t_{ij}+\varepsilon_2)^{254}+\theta_2\right]^{\alpha_2} \tag{16}$$

and

$$f^{\alpha_3}_{\varepsilon_3,\,\theta_3}(t_{ij}) = \left[(t_{ij}+\varepsilon_3)^{59}+(t_{ij}+\varepsilon_3)^{209}+(t_{ij}+\varepsilon_3)^{254}+\theta_3\right]^{\alpha_3} \tag{17}$$

with ε_2, ε_3, θ_2, $\theta_3 \in \mathrm{F}_{2^8}$, and $0 < \alpha_2 = s_2/t_2 < 255$, and $0 < \alpha_3 = s_3/t_3 < 255$, and s_2, t_2, s_3, t_3 being positive integers with $(s_2, 255) = 1$, $(s_3, 255) = 1$, $(t_2, 255) = 1$, $(t_3, 255) = 1$, $(s_2, t_2) = 1$ and $(s_3, t_3) = 1$.

Given an array M, the operation of diffusion is implemented by:

$$R = T_1 \times M \times T_2^{\mathrm{T}} \tag{18}$$

where T_2^{T} means the transpose of T_2, and ε_2, θ_2, s_2, t_2 in (16) and ε_3, θ_3, s_3, t_3 in (17) are used as cipher keys.

3.4. Post-scrambling of diffusion results based on permutation polynomial

In this encryption stage, a post-scrambling similar to pre-scrambling is performed to further scramble the encryption results.

The same permutation polynomial is chosen and parameterized with parameters ε_4, θ_4, α_4 as:

$$f^{\alpha_4}_{\varepsilon_4,\,\theta_4}(m) = \left[(m+\varepsilon_4)^{59}+(m+\varepsilon_4)^{209}+(m+\varepsilon_4)^{254}+\theta_4\right]^{\alpha_4} \tag{19}$$

where ε_4, $\theta_4 \in \mathrm{F}_{2^8}$, and $0 < \alpha_4 = s_4/t_4 < 255$, and s_4, t_4 are positive integers with $(s_4, 255) = 1$, $(t_4, 255) = 1$, and $(s_4, t_4) = 1$.

Given an array $R(m, n)$ on F_{2^8}, m, $n = 0, 1, ..., 255$ the post-scrambling is implemented by:

$$W(u, v) = R(m, n), \quad u, v = 0, 1, ..., 255 \tag{20}$$

where

$$u = f^{\alpha_4}_{\varepsilon_4,\,\theta_4}(m)$$

$$v = f^{\alpha_4}_{\varepsilon_4,\,\theta_4}(n)$$

and ε_4, θ_4, s_4, t_4 in (19) are used as cipher keys.

3.5. Decryption process

In order to recover the plain image, the inverses of the diffusion, pre-scrambling and post-scrambling are performed in the decryption process. Firstly, the inverse of the post-scrambling is completed by

$$R'(m, n) = W(u, v), \quad m, n = 0, 1, ..., 255 \tag{21}$$

where $m = (f_{\varepsilon_4, \theta_4}^{\alpha_4})^{-1}(u)$ and $n = (f_{\varepsilon_4, \theta_4}^{\alpha_4})^{-1}(v)$.

Next, the inverse diffusion is carried out by:

$$M' = (T_1)^{-1} \times R' \times (T_2^{\mathrm{T}})^{-1} \tag{22}$$

Subsequently, the inverse pre-scrambling is implemented by

$$D'(m, n) = M'(u, v), \quad m, n = 0, 1, ..., 255 \tag{23}$$

where $m = (f_{\varepsilon_1, \theta_1}^{\alpha_1})^{-1}(u)$ and $n = (f_{\varepsilon_1, \theta_1}^{\alpha_1})^{-1}(v)$.

Finally, the plain image is recovered based on the inverse finite field mapping:

$$h^{-1}: 0 \to 0, \omega^i \to i, i = 1, 2, ..., 255$$

4. Simulations and security analysis

In this section, the simulation is carried out to evaluate the proposed encryption algorithm. In our experiments, we use the following six 8-bit (256-level) gray images as plain images: *Barb*, *Cameraman*, *Peppers*, *Goldhill*, *Boat* and *Baboon* of size 256 × 256, as shown in the first column of Fig. 2. The secret keys ε_1, ε_2, ε_3, ε_4, θ_1, θ_2, θ_3, θ_4, s_1, s_2, s_3, s_4, t_1, t_2, t_3 and t_4 are set to ω^{11}, ω^{50}, ω^{50}, ω^{127}, ω^{15}, ω^{18}, ω^{26}, ω^{11}, 191, 47, 103, 181, 193, 41, 101 and 127, respectively. Certainly, these key parameters can be set to other values inside their scopes. The encrypted images are shown in the second column of Fig. 2. The third Column shows the decrypted images with correct keys. It can be

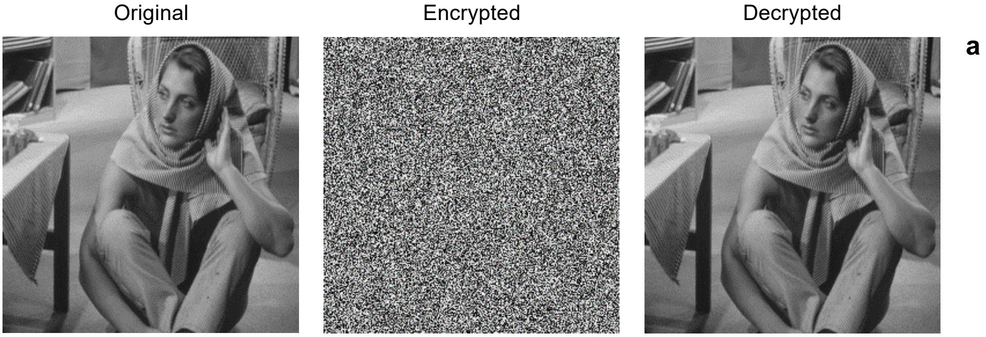

Fig. 2. Encryption results and decryption ones with correct keys: original, encrypted and decrypted *Barb* (**a**), *Peppers* (**b**), *Cameraman* (**c**), *Baboon* (**d**), and *Boat* (**e**).

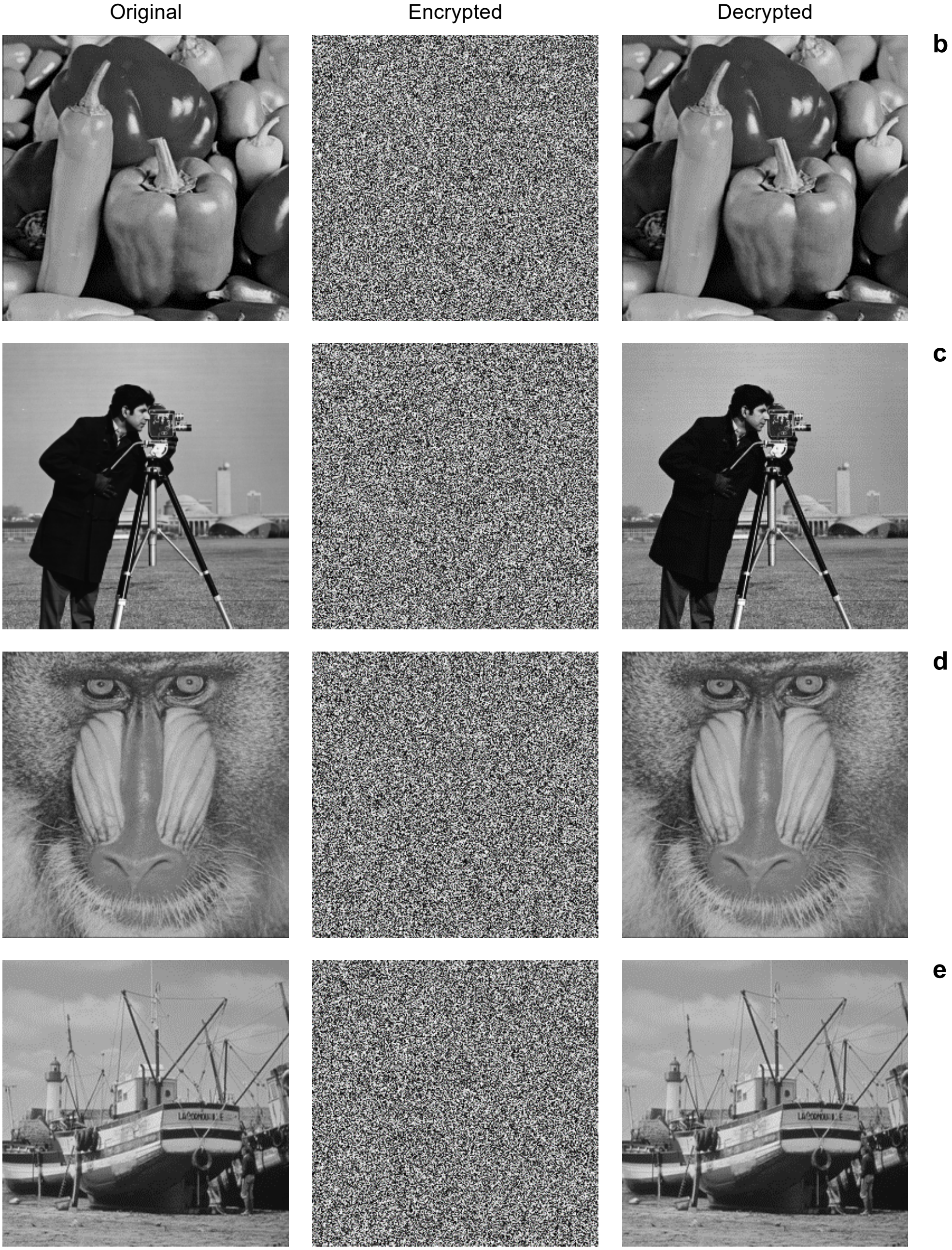

Fig. 2. Continued.

seen that the encrypted images are visually cluttered and unidentifiable, and the decrypted images are visually the same with original ones. In fact, the mean square error (MSE) between each plain image and its recovered version is zero.

4.1. Statistical analysis

4.1.1. Histogram

Histogram is commonly employed to reveal pixel distributions of an image. A uniform histogram often implies a great randomness. The histograms of plain and corresponding encrypted images are shown in Fig. 3. Obviously, histograms of ciphered images are very flat and show a nearly uniform distribution. Thus, the proposed cryptosystem

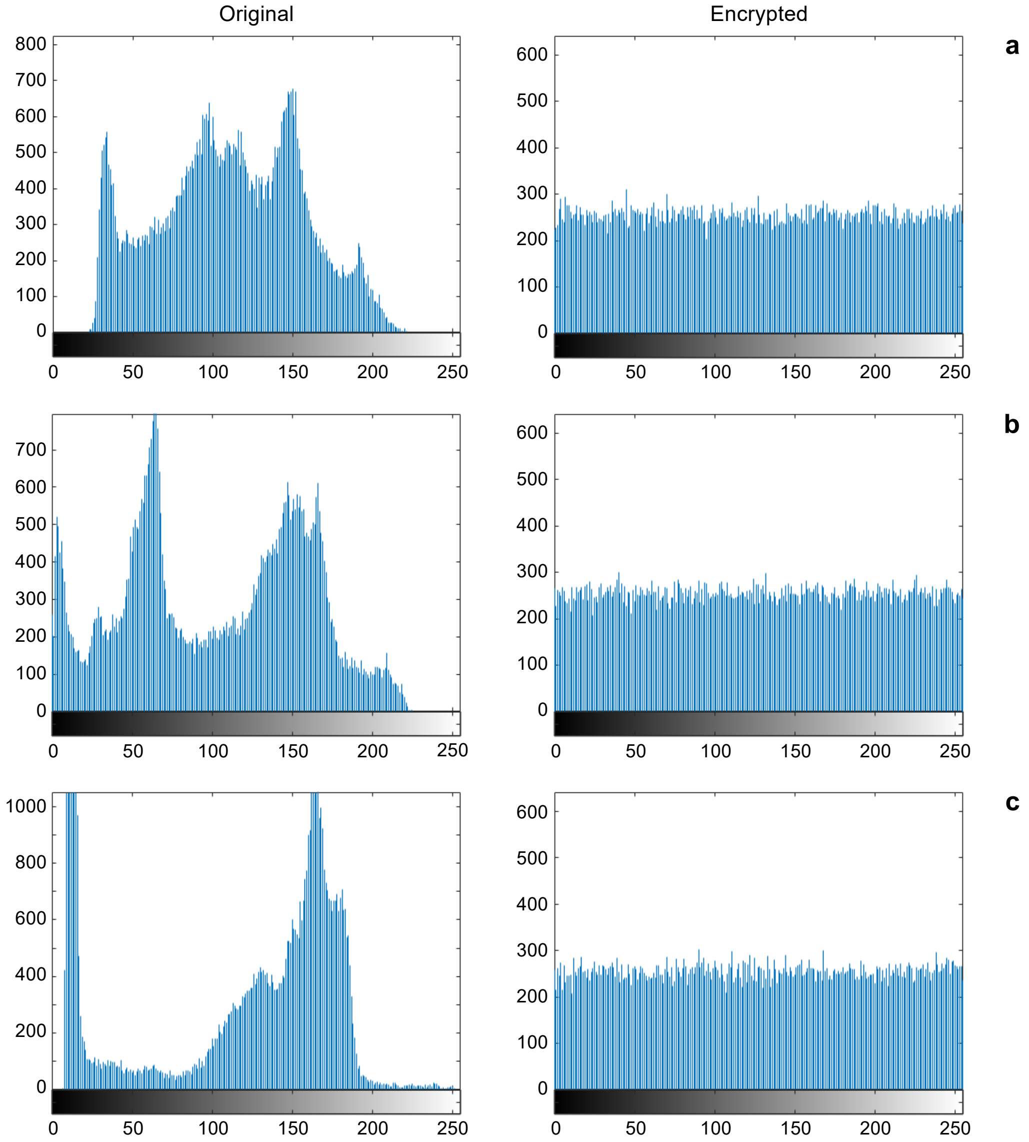

Fig. 3. Histograms of original and encrypted images: *Barb* (**a**), *Peppers* (**b**), *Cameraman* (**c**), *Baboon* (**d**), and *Boat* (**e**).

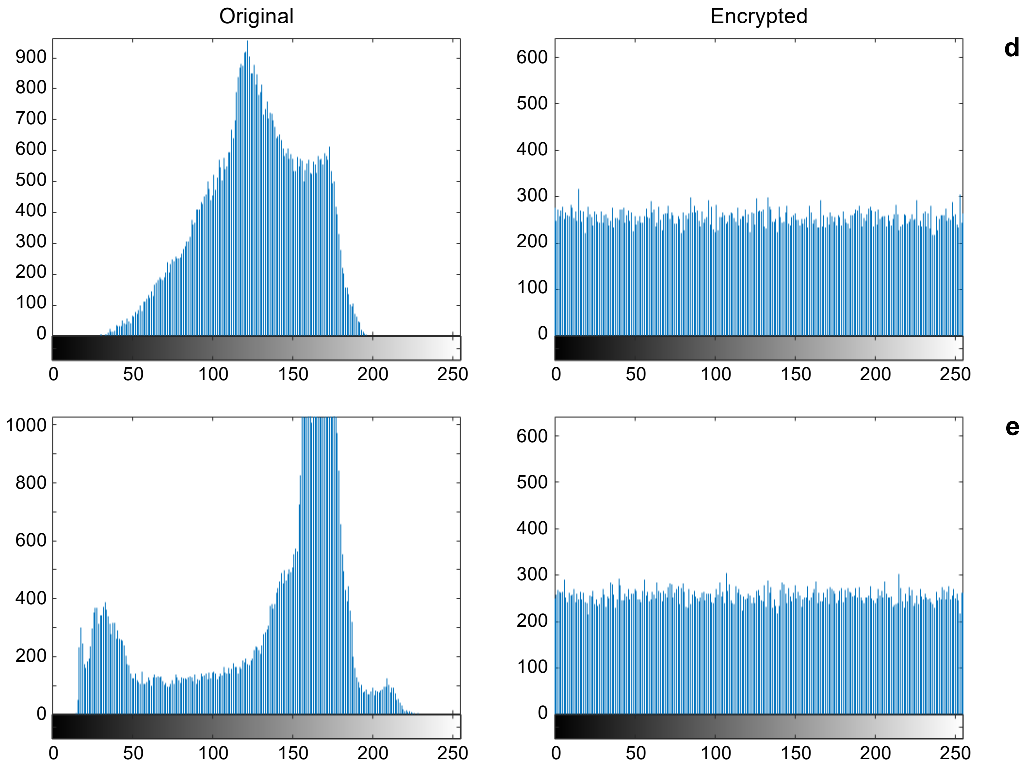

Fig. 3. Continued.

has an ability of resisting histogram analysis attacks because the encrypted results of different images have similar histograms. In a word, the encryption system fulfils the task of histogram uniformization.

4.1.2. Information entropy

The information uncertainty can be described by information entropy. The entropy H of an image can be expressed as:

$$H = -\sum_{i=0}^{L-1} p(m_i)\log_2 p(m_i) \tag{24}$$

where m_i is the i-th gray value for an L-level gray image, $p(m_i)$ represents the probability of occurrence of m_i. The H values are listed in Table 1, from which it can be seen

Table 1. Information entropies of images before and after encryption.

	Goldhill	*Boat*	*Peppers*	*Barb*	*Baboon*	*Cameraman*
Original	7.4415	7.0932	7.5612	7.3893	6.9730	7.0097
Encrypted	7.9972	7.9975	7.9973	7.9973	7.9969	7.9970

that the entropies of all encrypted images are very close to the theoretically maximum value 8, regardless of those of original images. Hence the encrypted results have a sufficiently large randomness in pixel values and the proposed algorithm is secure against the entropy analysis attacks.

4.2. Correlation between adjacent pixels and joint distribution analysis

In order to check the security and efficiency, the correlation and joint distribution between adjacent pixels are analyzed. The correlation coefficients of neighboring pixels in an image are computed by:

$$r_{xy} = \frac{\operatorname{cov}(x, y)}{\sqrt{D(x)}\,\sqrt{D(y)}} \tag{25}$$

where

$$\operatorname{cov}(x, y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y))$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N}x_i$$

$$E(y) = \frac{1}{N}\sum_{i=1}^{N}y_i$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2$$

$$D(y) = \frac{1}{N}\sum_{i=1}^{N}(y_i - E(y))^2$$

and x_i and y_i are gray values of two adjacent pixels in an image, N is the number of pixel pairs randomly selected from the image. It is common that the original image has correlation coefficients close to 1. It is expected that the correlation coefficients between adjacent pixels in encrypted images are as close to 0 as possible. Experimental results of correlation coefficients in three directions (vertical, horizontal and diagonal)

Table 2. Correlation coefficients between two adjacent pixels.

Images	Horizontal	Vertical	Diagonal
Plain *Barb*	0.9319	0.9565	0.9267
Encrypted *Barb*	0.0032	0.0016	–0.0024
Plain *Peppers*	0.9635	0.9718	0.9468
Encrypted *Peppers*	0.0006	–0.0030	–0.0049

for *Barb* and *Peppers* are shown in Table 2. As can be seen, the correlation coefficients of plain images are considerably close to 1 in the horizontal, vertical and diagonal directional, while those of the encrypted images are close to 0. Figure 4 exhibits graphically the joint distribution $z(x, y)$ of adjacent pixels of image *Barb* in three directions, in which two horizontal coordinates, x and y, indicate values of a pair of adjacent pixels, and the longitudinal coordinate z indicates the number of occurrences of such pair of

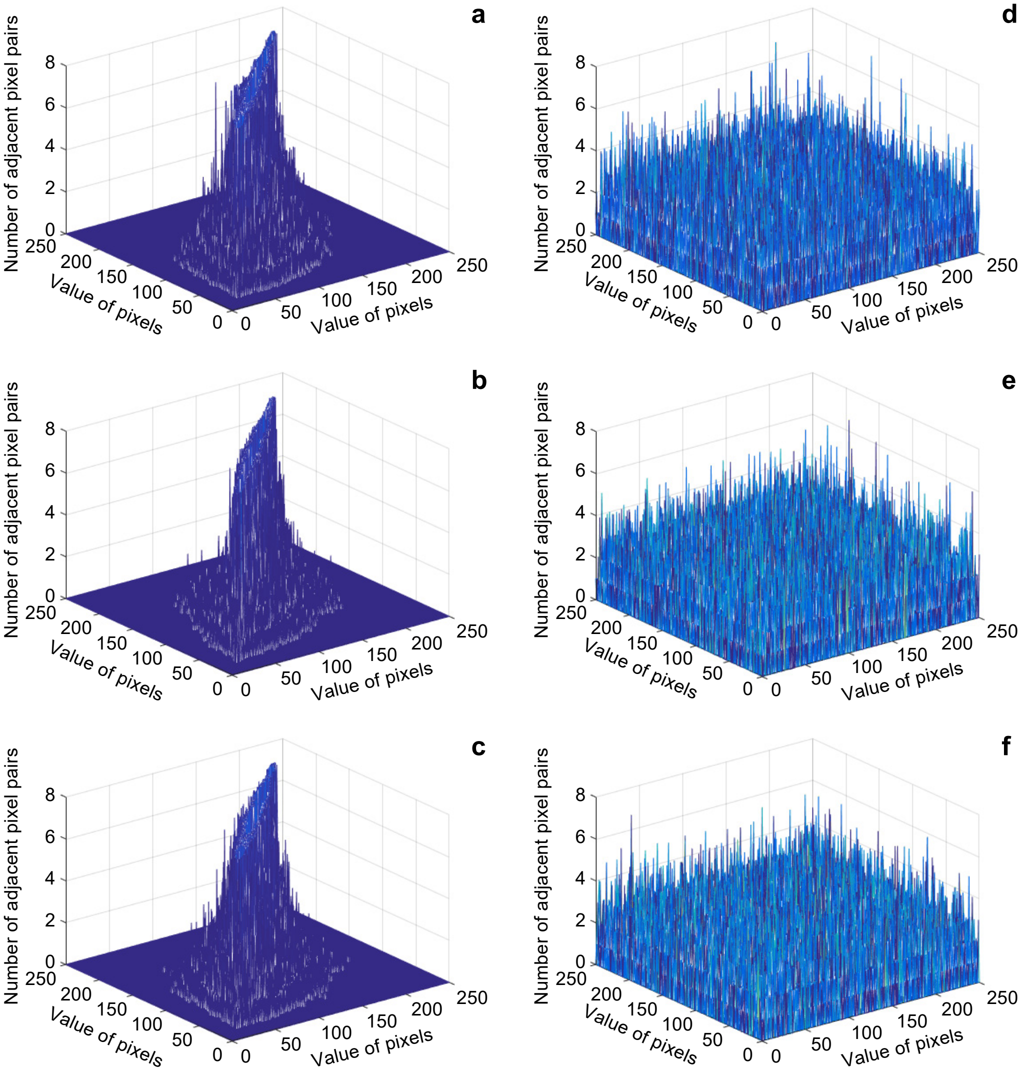

Fig. 4. Joint distribution of adjacent pixels in three directions of the original and encrypted *Barb*: horizontal (**a**), vertical (**b**), and diagonal (**c**) directions of *Barb*; horizontal (**d**), vertical (**e**), and diagonal (**f**) directions of encrypted *Barb*.

pixels taking value of (x, y). As x and y can take any possible values in F_{2^8}, the possible number of values of (x, y) is 256^2. From Figs. 4**a**–4**c**, it can be seen that adjacent pixels of the original image in three directions (horizontal, vertical, diagonal) distribute jointly near the diagonal of the coordinate plane, while those of the encrypted images fill in the whole plane in Figs. 4**d**–4**f**. Hence, the adjacent pixels have a much stronger correlation in original images than in encrypted results. Therefore, the algorithm is very effective for decorrelation and is secure against correlation analysis attacks.

4.3. Resistance to differential attack

The ability of encryption systems to resist differential attacks depends on plaintext sensitivity. The differential attack is that the attackers find a relationship between two encrypted images encrypted from two plain-images, which are different only by one bit. To measure the ability of the encryption algorithms to resist differential attacks, two criteria are used: the number of pixels change rate (NPCR) and the unified average changing intensity (UACI), defined as:

$$\mathrm{NPCR} = \frac{\sum_{m,\,n} C(m,\,n)}{W \times H} \tag{26}$$

$$\mathrm{UACI} = \frac{1}{W \times H} \frac{\sum_{m,\,n} \left| M_1(m,\,n) - M_2(m,\,n) \right|}{g_{\max}} \tag{27}$$

where $W \times H$ is the size of encrypted images M_1 and M_2, $g_{\max}$ is the biggest gray level, $M_1(m, n)$ and $M_2(m, n)$ represent the pixel values at (m, n) of the cipher images, corresponding to plain images with and without a one-bit change, respectively. $C(m, n)$ is a bipolar indicator defined as:

$$C(m,\,n) = \begin{cases} 0, & M_1(m,\,n) = M_2(m,\,n) \\ 1, & M_1(m,\,n) \neq M_2(m,\,n) \end{cases} \tag{28}$$

For some 256-gray-scale plain images, we consider a plain-image with a pixel changed by one bit. Specifically, in our experiments, the least significant bits of the pixels at (128, 128) of *Goldhill*, *Boat*, *Barb*, *Baboon*, *Peppers* and *Cameraman* are inverted. The resulting NPCR and UACI values are shown in Table 3. It is clear that in

Table 3. Results of NPCRs and UACIs.

	Goldhill	*Boat*	*Barb*	*Baboon*	*Peppers*	*Cameraman*
NPCR	99.6216%	99.6094%	99.6140%	99.5773%	99.6262%	99.5758%
UACI	33.3647%	33.3875%	33.2734%	33.4729%	33.4292%	33.5893%

all cases the resulting NPCRs and UACIs are very close to $\text{NPCR}_{\text{expected}}$ (= 99.6094%) and $\text{UACI}_{\text{expected}}$ (= 33.4635%) [36], respectively. From Table 3, one can conclude that the proposed algorithm is not vulnerable to differential attacks. In another word, the proposed cryptosystem satisfies a dynamic property similar as the avalanche effect.

4.4. Key space and key sensitivity analysis

4.4.1. Key space

As is known, the key space should be large enough to overcome the brute-force attack. In our algorithm, ε_1, ε_2, ε_3, ε_4, θ_1, θ_2, θ_3, θ_4, s_1, s_2, s_3, s_4, t_1, t_2, t_3 and t_4 are used as the secret keys. The key space is interpreted as follows.

1) For each set of ε_i, $\theta_i \in \text{F}_{2^8}$, $i = 1, 2, 3, 4$ obviously its key space is $2^8 \times 2^8 = 2^{16}$.

2) Considering $255 = 3 \times 5 \times 17$, for each s_i/t_i, $i = 1, 2, 3, 4$ with $(s_i, 255) = 1$, $(t_i, 255) = 1$, $(s_i, t_i) = 1$,

$$\begin{aligned}\#s_i &= 256 - \#\{0\} - \#\{x|(x, 3) \neq 1, (x, 5) \neq 1, (x, 17) \neq 1\} \\ &\quad + \#\{x|(x, 3 \times 5) \neq 1, (x, 5 \times 17) \neq 1, (x, 3 \times 17) \neq 1\} - \#\{255\} \\ &= 256 - 1 - 5 \times 17 - 3 \times 17 - 3 \times 5 + 17 + 3 + 5 - 1 = 128 = 2^7 \end{aligned} \tag{29}$$

3) Given $s_i = 2 \times 7 \times 11 < 255$, which is the product of maximum numbers of least primes. Thus, the number of choices of t_i is at least,

$$\begin{aligned}\#t_i &= 128 - \#\{x|(x, 2) \neq 1, (x, 7) \neq 1, (x, 11) \neq 1\} \\ &\quad + \#\{x|(x, 2 \times 7) \neq 1, (x, 7 \times 11) \neq 1, (x, 2 \times 11) \neq 1\} - \#\{2 \times 7 \times 11\} \\ &= 128 - 2^6 - 2^4 - 12 + 9 + 4 + 3 - 1 = 49 = 2^{5.6147} \end{aligned} \tag{30}$$

4) From 1), 2) and 3), it can be seen that the key space for each set of ε_i, θ_i, s_i, t_i, $i = 1, 2, 3, 4$, is at least $2^{16} \times 2^7 \times 2^{5.6147} = 2^{28.6147}$. The key space for the proposed encryption algorithm is at least $(2^{28.6147})^4 \cong 2^{114.46}$, which is a sufficiently large key space against brute-force attacks.

4.4.2. Key sensitivity

One of the main factors that measure the security of an encryption system is the sensitivity to its keys. Figure 5 shows the decrypted results of *Goldhill* with a key minimally different from the right key. On specifics, the 16 cipher keys, ε_1, ε_2, ε_3, ε_4, θ_1, θ_2, θ_3, θ_4, s_1, s_2, s_3, s_4, t_1, t_2, t_3 and t_4 are changed, *one at a time*, from their correct values into ω^{12}, ω^{51}, ω^{51}, ω^{127}, ω^{16}, ω^{19}, ω^{27}, ω^{12}, 193, 49, 104, 182, 197, 43, 103 and 131. Taking *Goldhill* as an example, the resulting 16 decrypted images are shown in Fig. 5, from which we can see that the quality of decrypted images drops dramatically when any

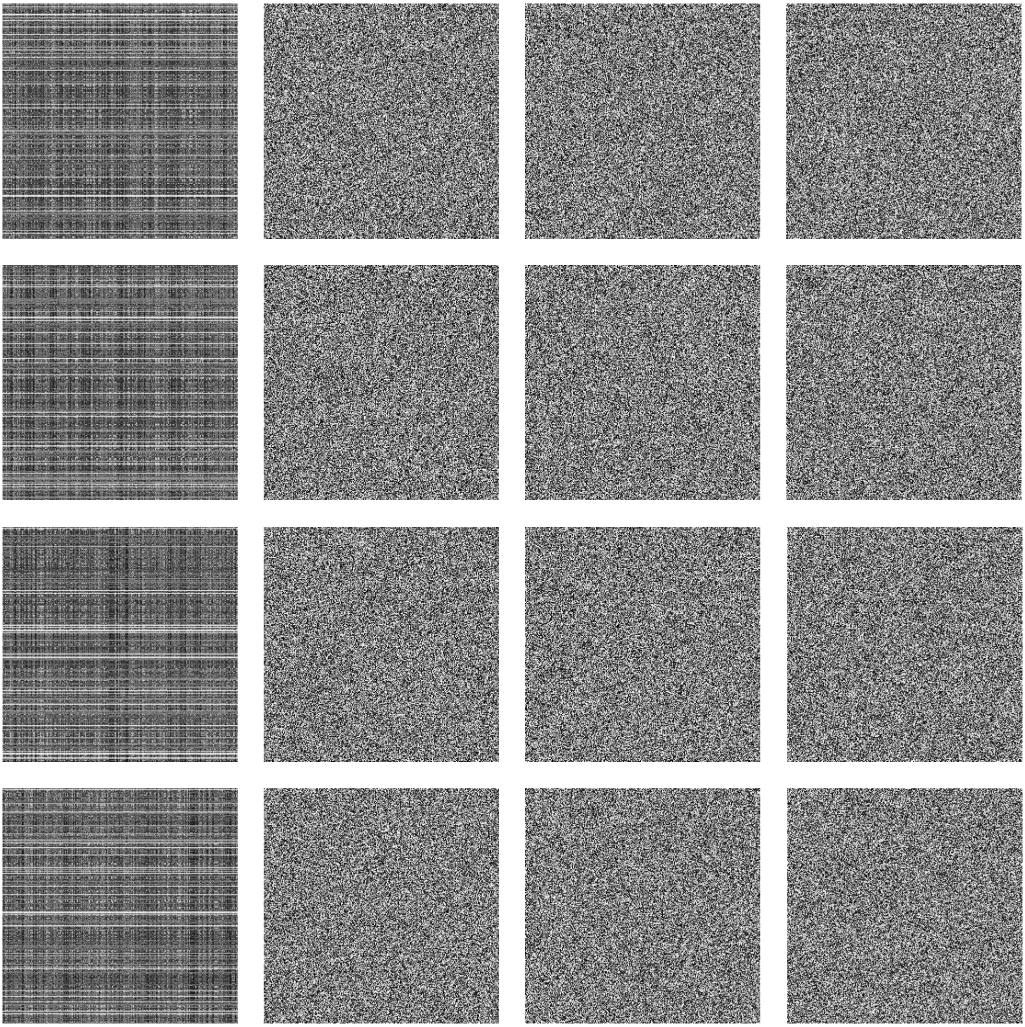

Fig. 5. Decrypted *Goldhill* with a deviation to one specific key: from top to bottom, the first row shows decrypted images with a deviation to ε_1, ε_2, ε_3, ε_4, respectively; the second row, to θ_1, θ_2, θ_3, θ_4, respectively; the third row, to s_1, s_2, s_3, s_4, respectively; the last row, to t_1, t_2, t_3, t_4, respectively.

key has a small numerical change in its scopes. It is obvious that the proposed scheme has a sufficiently large key sensitivity.

5. Conclusion

In this paper, we have described a procedure to encrypt digital images based on permutation polynomial over finite fields. First, mapping the pixel values of a plain image into finite fields achieves an initial scrambling. Then, a pre-scrambling based on polynomial permutation is carried out to shuffle the pixels' positions. Next, a symmetric matrix transform over finite fields completes the image diffusion. A post-scrambling is then performed to further scramble the encryption results. Since the proposed encryption method does not introduce any round-off errors, it is highly suitable in scenarios where such errors must be avoided. Hence it is a strictly lossless image encryption scheme and, in this sense, is superior to most of encryption algorithms based on transforms such as complex Fourier transform or Mellin transform, cosine transform, or reality-preserving fractional transform. Furthermore, the intrinsic nonlinearity of permutation polynomials over finite fields makes the proposed cryptosystem a nonlinear one and able to resist known-plaintext and chosen-plaintext attacks. In the future, extension of our encryption method to color images and other digital media, such as audio and video, is to be developed. Encryption for arbitrary sizes of images will also be considered in the future research work.

Acknowledgments – This work is partly supported by the National Natural Science Foundation of China (Grant No. 61662047) and the Science and Technology Program of Jiangxi Provincial Education Department (Grant No. GJJ171747).

References

[1] Ye G., *Image scrambling encryption algorithm of pixel bit based on chaos map*, Pattern Recognition Letters **31**(5), 2010, pp. 347–354, DOI: 10.1016/j.patrec.2009.11.008.

[2] Liu Z., Xu L., Dai J., Liu S., *Image encryption by using local random phase encoding in fractional Fourier transform domains*, Optik **123**(5), 2012, pp. 428–432, DOI: 10.1016/j.ijleo.2011.04.022.

[3] Liu Z., Yang M., Liu W., Li S., Gong M., Liu W., Liu S., *Image encryption algorithm based on the random local phase encoding in gyrator transform domains*, Optics Communications **285**(19), 2012, pp. 3921–3925, DOI: 10.1016/j.optcom.2012.05.061.

[4] Refregier P., Javidi B., *Optical image encryption using input plane and Fourier plane random encoding*, Optics Letters **20**(7), 1995, pp. 767–769, DOI: 10.1364/OL.20.000767.

[5] Liu Z., Dai J., Sun X., Liu S., *Triple image encryption scheme in fractional Fourier transform domains*, Optics Communications **282**(4), 2009, pp. 518–522, DOI: 10.1016/j.optcom.2008.10.068.

[6] Yuen C.H., Wong K.W., *A chaos-based joint image compression and encryption scheme using DCT and SHA-1*, Applied Soft Computing **11**(8), 2011, pp. 5092–5098, DOI: 10.1016/j.asoc.2011.05.050.

[7] Vaish A., Kumar M., *Color image encryption using singular value decomposition in discrete cosine Stockwell transform domain*, Optica Applicata **48**(1), 2018, pp. 25–38, DOI: 10.5277/oa180103.

[8] Singh H., *Nonlinear optical double image encryption using random-optical vortex in fractional Hartley transform domain*, Optica Applicata **47**(4), 2017, pp. 557–578, DOI: 10.5277/oa170406.

[9] Wang Z., Zhang Y., Gong Q., Li S., Qin Y., *Fully-phase optical image encryption in diffractive-imaging scheme with QR-code-based random illumination*, Optica Applicata **47**(2), 2017, pp. 233–243, DOI: 10.5277/oa170206.

[10] Zhou N., Dong T., Wu J., *Novel image encryption algorithm based on multiple-parameter discrete fractional random transform*, Optics Communications **283**(15), 2010, pp. 3037–3042, DOI: 10.1016/j.optcom.2010.03.064.

[11] Yin F., He Q., Liu Z., *A known-plaintext attack on iterative random phase encoding in fractional Fourier domains*, Optica Applicata **47**(1), 2017, pp. 131–139, DOI: 10.5277/oa170112.

[12] Venturini I., Duhamel P., *Reality preserving fractional transforms [signal processing applications]*, [In] *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Canada, May 17–21, 2004, Vol. 5, pp. 205–208, DOI: 10.1109/ICASSP.2004.1327083.

[13] Lang J., *Image encryption based on the reality-preserving multiple-parameter fractional Fourier transform and chaos permutation*, Optics and Lasers in Engineering **50**(7), 2012, pp. 929–937, DOI: 10.1016/j.optlaseng.2012.02.012.

[14] Wu J., Guo F., Zeng P., Zhou N., *Image encryption based on a reality-preserving fractional discrete cosine transform and a chaos-based generating sequence*, Journal of Modern Optics **60**(20), 2013, pp. 1760–1771, DOI: 10.1080/09500340.2013.858189.

[15] Zhou N., Wang Y., Gong L., Chen X., Yang Y., *Novel color image encryption algorithm based on the reality preserving fractional Mellin transform*, Optics and Laser Technology **44**(7), 2012, pp. 2270–2281, DOI: 10.1016/j.optlastec.2012.02.027.

[16] Chen R.L., Zhou Y., Luo M., Zhang A.D., Gong L.H., *Image compression-encryption algorithm combining compressive sensing with log operation*, Optica Applicata **48**(4), 2018, pp. 563–573, DOI: 10.5277/oa180403.

[17] Lima J.B., Campello de Souza R.M., *The fractional Fourier transform over finite fields*, Signal Processing **92**(2), 2012, pp. 465–476, DOI: 10.1016/j.sigpro.2011.08.010.

[18] Lima J.B., Campello de Souza R.M., *Fractional cosine and sine transforms over finite fields*, Linear Algebra and its Applications **438**(8), 2013, pp. 3217–3230, DOI: 10.1016/j.laa.2012.12.021.

[19] Lima J.B., Barone M., Campello de Souza R.M., *Cosine transforms over fields of characteristic 2*, Finite Fields and Their Applications **37**, 2016, pp. 265–284, DOI: 10.1016/j.ffa.2015.10.005.

[20] Lima J.B., da Silva E.S., Campello de Souza R.M., *Cosine transforms over fields of characteristic 2: fast computation and application to image encryption*, Signal Processing: Image Communication **54**, 2017, pp. 130–139, DOI: 10.1016/j.image.2017.03.007.

[21] Lima J.B., Novaes L.F.G., *Image encryption based on the fractional Fourier transform over finite fields*, Signal Processing **94**(1), 2014, pp. 521–530, DOI: 10.1016/j.sigpro.2013.07.020.

[22] Lima J.B., Lima E.A.O., Madeiro F., *Image encryption based on the finite field cosine transform*, Signal Processing: Image Communication **28**(10), 2013, pp. 1537–1547, DOI: 10.1016/j.image.2013.05.008.

[23] Wan D., Lidl R., *Permutation polynomials of the form $x^r f(x^{(q-1)/d})$ and their group structure*, Monatshefte für Mathematik **112**(2), 1991, pp. 149–163, DOI: 10.1007/BF01525801.

[24] Wang L., *On permutation polynomials*, Finite Fields and Their Applications **8**(3), 2002, pp. 311–322, DOI: 10.1006/ffta.2001.0342.

[25] Cao X., Qiu W., *On Dickson polynomials and difference sets*, Journal of Mathematical Research and Exposition **26**(2), 2006, pp. 219–226.

[26] Helleseth T., Zinoviev V., *New Kloosterman sums identities over F_{2m} image for all m*, Finite Fields and Their Applications **9**(20), 2003, pp. 187–193, DOI: 10.1016/S1071-5797(02)00028-X.

[27] Yuan J., Ding C., *Four classes of permutation polynomials of F_{2m}*, Finite Fields and Their Applications **13**(4), 2007, pp. 869–876, DOI: 10.1016/j.ffa.2006.05.006.

[28] Yuan J., Ding C., Wang H., Pieprzyk J., *Permutation polynomials of the form $(x^p - x + \delta)^s + L(x)$*, Finite Fields and Their Applications **14**(2), 2008, pp. 482–493, DOI: 10.1016/j.ffa.2007.05.003.

[29] Zeng X., Zhu X., Hu L., *Two new permutation polynomials with the form $(x^{2^k} + x + \delta)^s + x$ over F_{2n}*, Applicable Algebra in Engineering Communication and Computing **21**(2), 2010, pp. 145–150, DOI: 10.1007/s00200-010-0120-6.

[30] Li N., Helleseth T., Tang X., *Further results on a class of permutation polynomials over finite fields*, Finite Fields and Their Applications **22**, 2013, pp. 16–23, DOI: 10.1016/j.ffa.2013.02.004.

[31] Akbary A., Ghioca D., Wang Q., *On constructing permutations of finite fields*, Finite Fields and Their Applications **17**(1), 2011, pp. 51–67, DOI: 10.1016/j.ffa.2010.10.002.

[32] Zeng X., Zhu X., Li N., Liu X., *Permutation polynomials over F_{2m} of the form $(x^{2^i} + x + \delta)^{s_1} + (x^{2^i} + x + \delta)^{s_2} + x$*, Finite Fields and Their Applications **47**, 2017, pp. 256–268, DOI: 10.1016/j.ffa.2017.06.012.

[33] Li L., Wang S., Li C., Zeng X., *Permutation polynomials* $(x^{p^m} - x + \delta)^{s_1} + (x^{p^m} - x + \delta)^{s_2} + x$ *over* F_{p^n}, Finite Fields and Their Applications **51**, 2018, pp. 31–61, DOI: 10.1016/j.ffa.2018.01.003.

[34] Nyberg K., *Deferentially uniform mappings for cryptography*, The Workshop on the Theory and Application of Cryptographic Techniques, Springer, Berlin, Heidelberg, 1993, pp. 55–64.

[35] Lidl R., Niederreiter H., *Finite Fields*, Cambridge University Press, Cambridge, 1997.

[36] Liang Y., Liu G., Zhou N., Wu J., *Image encryption combining multiple generating sequences controlled fractional DCT with dependent scrambling and diffusion*, Journal of Modern Optics **62**(4), 2015, pp. 251–264, DOI: 10.1080/09500340.2014.964342.

Received May 5, 2019
in revised form August 28, 2019