Multimedia Tools and Applications (2019) 78:33865–33884
https://doi.org/10.1007/s11042-019-08171-2

Fast encryption scheme for 3D models based on chaos system

Xingyuan Wang[1] • Mingxiao Xu[1] • Yong Li[1]

Received: 4 March 2019 / Revised: 20 August 2019 / Accepted: 2 September 2019 /
Published online: 9 October 2019

Abstract
With the development of multimedia applications, the application of 3D models becomes more and more popular, and its security has become an urgent problem to be solved. 3D objects have more complex spatial structures than 1D and 2D objects. Most of the previous work is to encrypt 3D objects directly, and this kind of algorithm is often tedious and the encryption time is long. Therefore, a new fast image encryption scheme based on chaos theory is proposed in this paper. In this scheme, the 3D object is transformed into 2D object (similar to image format), and then encrypted. The encryption process is divided into two stages: confusion stage and diffusion stage. In the confusion phase, we introduce random points. In the diffusion phase, we split the floating-point data, the integer part is encrypted by XOR, and the decimal part is scrambled only. The experimental results show that the scheme can encrypt and decrypt the 3D model correctly. The numerical results in security analysis are close to the ideal value, which shows that the scheme can resist common attacks and has high security.

Keywords Chaos System · 3D models · Scrambling-diffusion · Floating point data

1 Introduction

With the development of multimedia applications, 3D models are almost everywhere in the field of digital images due to their complex spatial information [8, 18, 20]. Today, 3D models have been widely used in various fields, such as computer-aided design, virtual reality, digital visualization, and 3D printing [10]. Among them, 3D printing based on Stereolithography (STL) file format [4] is widely used in medical institutions, aerospace, education, e-commerce, military and other fields. Therefore, 3D models plays an important role in the modern big data

✉ Xingyuan Wang
xywang@dlmu.edu.cn

✉ Mingxiao Xu
1206512593@qq.com

[1] School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China

era. However, an unauthorized cryptanalysis is a major threat to multimedia messaging [1, 13, 32, 36, 42]. So how to effectively protect the security of 3D models in the communication becomes a problem that needs to be solved.

Chaos is sensitive to initial values and parameters, ergodicity and deterministic inherent randomness. This makes him have some characteristics of cryptography and provides a new solution for multimedia encryption [41]. However, some chaotic encryption systems are pointed out to be insecure. For example, M. Preishuber et al. [24] pointed out that deliberately chosen insecure encryption systems still pass the type of tests used to support the security of chaotic image security systems. Therefore, it is very important to carry out a comprehensive evaluation of chaotic cryptography.

In this paper, a fast encryption scheme of 3D models based on chaotic system is proposed, which converts 3D model into 2D object similar to image format. For the difference between 3D models encryption and image encryption, The additional work in this article is as follows:

(1) The coordinate point data in 3D models is a floating point type. We split the array of floating point numbers into an array of integers and an array of decimals during the diffusion process. The integer part is XOR-encrypted, and the fractional part is only scrambled. Finally, a new array of floating point numbers is formed according to the same position addition operation to complete its diffusion.

(2) There are some redundant and duplicate text information in the STL file that stores 3D models. As shown in Section 3.1, text information such as “facet normal”, “outer loop”, “vertex”, and so on exists in the STL file. It exists only to facilitate reading without having any effect on the structural information of the 3D models. Therefore, only the key information (coordinate point data) is extracted for encryption in order to achieve the effect of fast encryption.

(3) We generated a set of random points using a one-dimensional (1D) chaotic map and added it to a scrambled data. The addition of random points not only hides the original coordinate quantity information, but also plays a certain interference effect on the original data.

The remainder of the article is divided into six parts. Section 2 introduces related works; Section 3 introduces preparation works; Section 4 introduces the encryption algorithm; Section 5 analyzes the experimental results; Section 6 analyzes the security; Section 7 summarizes the full text and suggests further work.

2 Related works

The study of cryptography is a hot spot from beginning to end [11, 12, 28]. Since Matthews proposed the chaotic encryption algorithm in 1989 [21], many related studies have been proposed. These include image encryption [19, 31, 33–35, 39, 43, 44], voice encryption and text encryption [2, 3, 23, 26, 27]. In recent years, some 3D models encryption algorithms based on chaos theory have also been proposed. Rey [25] proposes 3D models encryption method using 2D Arnold map and 3D celluar automata; Jin et al. [14] propose a 3D models encryption method using 3D Lu mapping. 3D objects mean higher levels of representation or semantics than 1D and 2D data [15]. The proposed algorithm directly encrypts the 3D objects of 3D models, which makes the encryption process tedious and increases the encryption time. Therefore, in this paper, 3D models is transformed into two-dimensional objects, and then processed to achieve the effect of fast encryption.

In addition, Fridrich [9] proposed a chaotic-based image encryption scheme should consist of two processes: permutation (relative position change of elements) and diffusion (modification of element values). Various image encryption systems based on this method are proposed, which demonstrates the security of the method and has excellent encryption effect [6, 7, 38, 40, 45]. Because of the similarity between 3D models data and image data, the encryption scheme based on scrambling-diffusion approach is adopted in this paper.

3 Preparation

3.1 Introduction to STL Files in 3D models

An STL file is a tessellated (triangulated) surface model [5, 16, 29]. It's an interface protocol developed by 3D SYSTEM in 1988. It is divided into ASCII code and binary. Among them, ASCII files and binary files can be easily converted by professional software, such as Siemens NX 12.0. The ASCII format file is easy to read and test. So it is selected as the research object. The STL file format is as follows:

```
0 solid ...
1 facet normal nx ny nz
2       outer loop
3           vertex v1x v1y v1z
4           vertex v2x v2y v2z
5           vertex v3x v3y v3z
6       endloop
7   endfacet
...
n endsolid ...
```

Due to the standardization of its format, the extraction of coordinate data in STL file is very convenient. Therefore, it was chosen as the research object. In fact, for other formats of 3D models files, the reading method will change. Though the encryption process will not change much.

3.2 Sort scrambling introduction

The algorithm uses the pseudo-random sequence generated by the chaotic system to sort and scramble, as shown in Eq. (1):

$$\begin{cases} index = \mathrm{sort}(w) \\ p_1 = p(index) \end{cases} \tag{1}$$

w is a random sequence generated by the chaotic system, p is the original data sequence. sort(w) is a function to find the index mapping between a sequence w and its sortedversion w^* in the ascending order. Then p is reordered according to the sequence index to generate p_1. For example, when$w = [0.45, 0.21, 0.8, 0.7, 0.9]$, Use the sort($w$) function to get $w^* = [0.21, 0.45, 0.7, 0.8, 0.9]$, $index = [2\ 1\ 4\ 3\ 5]$. if $p = [1, 2, 3, 4, 5]$, reorder by $index$ to get $p_1 = [2, 1, 4, 3, 5]$.

3.3 The 1D chaotic map

In this algorithm, we use the 1D chaotic map proposed by May to perform early scrambling and generate random numbers [22, 37]. The 1D chaotic map can be defined as Eq. (2).

$$x_{n+1} = \mu x_n (1 - x_n) \tag{2}$$

In this model, when $x_n \in (0, 1)$ and $\mu \in (3.5699456, 4]$, the 1D chaotic map is in chaotic state that the resulting sequence $\{x_n, n = 0, 1, 2, 3\ldots\}$ is aperiodic, non-converged, and very sensitive to initial values. In order to avoid the periodic window, this article uses the parameters in the $\mu \in (3.99, 4]$ range.

3.4 The Logistic-dynamic coupled logistic map lattice (LDCML) Model

Traditional coupled Logistic map (CML) model proposed by Kaneko [17]:

$$xn + 1(i) = (1 - e)(f(xn(i))) + (e/2)f(xn(i - 1)) + f(xn(i + 1)) \tag{3}$$

In this model, the parameter $e(0 \le e \le 1)$ is coupling coefficient; the parameter $n(n = 1, 2, 3\ldots)$ is time sequence. The parameter $i(1 \le i \le L)$ is the number of lattices, $i + 1$ or $i - 1$ is the lattices adjacent to i.The function is the 1D logical map. The boundary conditions are that $i + 1 = 1$ when $i = L$ and $i - 1 = L$ when $i = 1$, which can ensure that lattice is in the range of $(1, L)$.

However, due to the simple structure of the CML spatiotemporal chaotic system, its chaos is only determined by the parameter μ and parameter e. The most important thing is that when $e < 0.3$ or $\mu < 3.8$, a lot of crystal lattice in CML will be reduced or even disappeared. The LDCML proposed by Wang, Feng et al. [30] is as follows:

$$xn + 1(i) = (1 - L(e))(f(xn(i))) + (L(e)/2)f(xn(i - 1)) + f(xn(i + 1)) \tag{4}$$

In LDCM, the meaning and boundary conditions of i, n, $f(x)$ are the same as CML. The difference is that the coupling coefficient is e in CML and the coupling coefficient is $L(e)$ in LDCM, where $L(e) = \mu_0 e(1 - e)$. In order to obtain the best dynamic, the μ_0 in the logic map $L(e)$ is 3.99.

The paper [17] studied the Kolmogorov-Sinai entropy, bifurcation diagram, information entropy, spatiotemporal behavior and mutual information of the LDCML system to prove that LDCML is more advanced than the traditional CML model. As shown in Fig. 1, this paper takes the number of 3 lattices, and its initial value and parameters use the data of Eq. (4) in this experiment, and generate its bifurcation diagram in 3 lattices. It can be found that it has good chaotic characteristics.

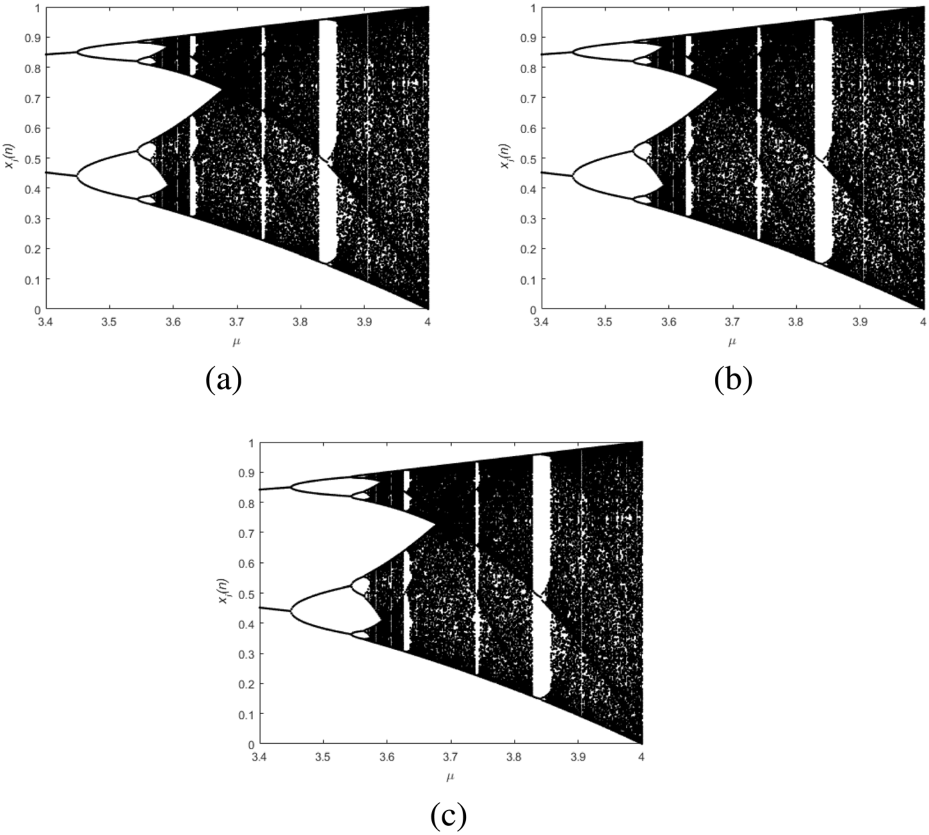


Fig. 1 Bifurcation diagram of lattices of LDCML: **a** Bifurcation diagram of the first lattice of LDCML, **b** Bifurcation diagram of the Second lattice of LDCML, **c** Bifurcation diagram of the Third lattice of LDCML

4 Algorithm description

4.1 Reading and writing of STL files

The main method of reading in this paper is to first detect the file format of the STL. If it is a binary format, it can be converted into ASCII format using professional processing software. Since the normal vector coordinates can be calculated according to Eq. (5), in this paper, the while loop in Matlab are used to extract only the coordinate points according to their file format, and an $r \times 3$ matrix is generated to encrypt it.

$$\begin{cases} nx = (v1y - v3y)(v2z - v3z) - (v1z - v3z)(v2y - v3y) \\ ny = (v1z - v3z)(v2x - v3x) - (v2z - v3z)(v1x - v3x) \\ nz = (v1x - v3x)(v2y - v3y) - (v2x - v3x)(v1y - v3y) \end{cases} \tag{5}$$

n_x, n_y, n_z refers to the normal vector coordinate; $v1_x$, $v1_y$, $v1_z$ refers to the x, y, z coordinate points of the $i(i = 1, 2, 3)$th points of the triangular face.

After the decryption is completed, we write the data of the coordinate points to the file to generate a ciphertext file. In this article, we only write coordinate point data and write all

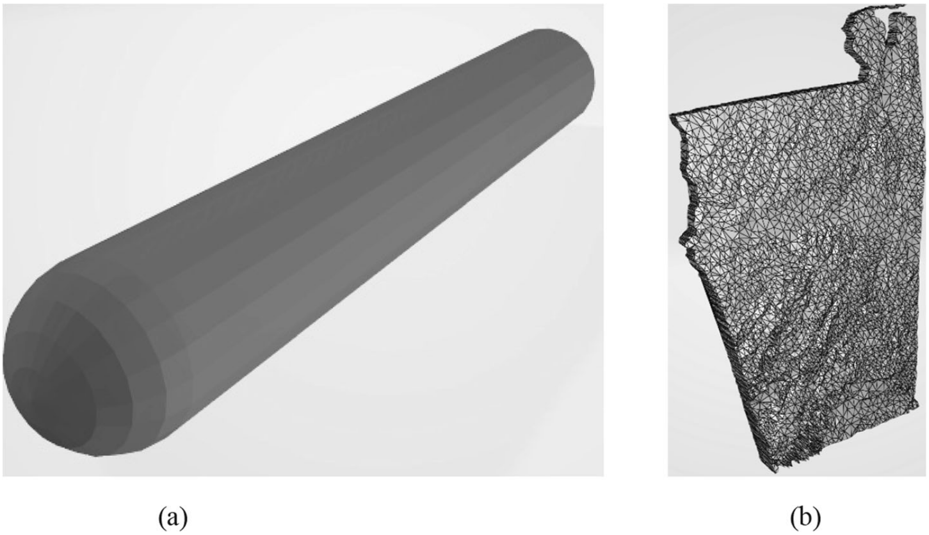


Fig. 2 3D models showing an angle Encryption: **a** Cylindrical model, **b** Terrain model

normal vectors to zero. Even if the attacker uses the Eq. (5) to calculate the normal vector coordinates and successfully read the STL file, the result is a meaningless, messy model.

As shown in Fig. 2. This paper selects two 3D models: the first is a simpler cylindrical model, and the second is a topographic map of a certain area of the United States. The reason for selecting the Terrain model is that it has complex detail structure, which is more helpful to verify the effect of the algorithm. The data type is all positive floating point numbers with a total of 29,757 vertex information.

4.2 Initial key generation

The keys used in this article are as follows:

$$Key = \left(x_1, x_2, x_3, u, u_0, e, bitxor\_value, T\right) \tag{6}$$

where $x1, x2, x3$ is the initial value corresponding to 3 lattice in the LCDML system, u, $u0$, e is the parameter of the Eq. (4) chaotic system, and $bitxor\_value$ is the value used for the pseudo-random sequence required for the diffusion. T is the number of discarded values when the transient is removed.

In this paper, the plaintext data P uses hash-256 to generate a 256-bit key and extracts the first 240-bit generated key K. The key K is divided into six groups of 40-bit subkeys (K_1, K_2, K_3, K_4, K_5, K_6), and is converted into a decimal fraction (b_1, b_2, b_3, b_4, b_5, b_6)between [0, 1]. Where (b_1, b_2, b_3) is the initial value of the 3 cells of the LDCML system coefficient $u = 3.99 + 0.01 \times b_4$, $u_0 = 3.99 + 0.01 \times b_5$, $e = 0.01 + 0.09 \times b_1$.

The method of generation of $bitxor\_value$ is shown in Fig. 3. Its value is used in XOR encryption phase, and the function is to limit the value after the diffusion of plaintext data to a certain range. First, the largest integer in plaintext P is counted and converted to binary. Then the maximum integer within this binary bit is calculated according to its binary bits, and finally it is added 1.

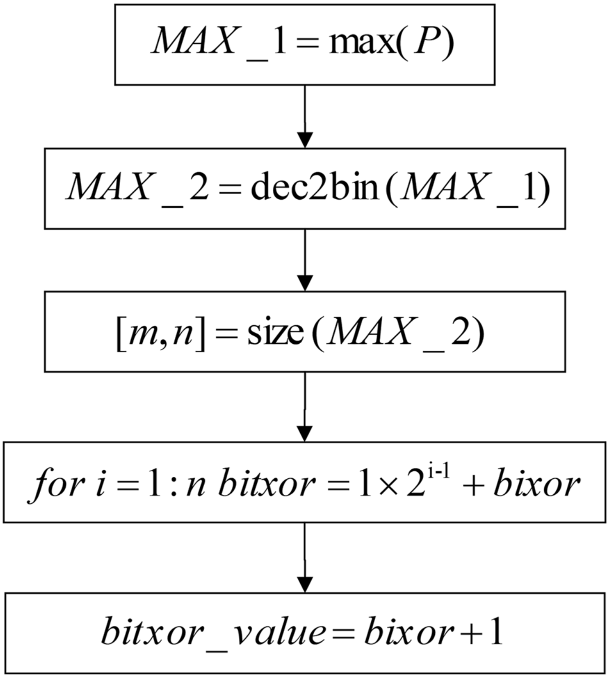

$MAX\_1 = \max(P)$
$MAX\_2 = \text{dec2bin}(MAX\_1)$
$[m,n] = \text{size}(MAX\_2)$
$for\ i = 1:n\ bitxor = 1 \times 2^{i\text{-}1} + bixor$
$bitxor\_value = bixor + 1$

Fig. 3 *Bitxor_value* generation process

4.3 Encryption algorithm

Step 1: Generate *Key*. Bring K into Eq. (6) and generate *Key*. The specific steps are discussed in Section 4.2.

Step 2: Generate A. Use Eq. (7) to rewrite x_1 to generate x_0. Bring x_0, u_0 into Eq. (2) and iterate $r \times 3 + \mathrm{T}$ times. Take the values of the last $r \times 3$ iterations to form A.

$$x_0 = abs(x_1 - \mathrm{mod}(sum(floor(P)), 100)/1000) \tag{7}$$

Step 3: Permutation. Use A and scramble the plaintext P according to Eq. (1) to generate P_1.

Step 4: Generate and insert random points. Bring A into Eq. (8) and generate Z_1. The matrix that converts Z_1 into a matrix with a size of $r_1 \times 3$ is connected to the end of P_1 to generate P_2.

$$\begin{cases} random = \mathrm{floor}((r \times 3 \times x_1)/100 + 1) \times 3 \times 3 \\ Z_0 = A(1, random + 1 : random \times 2) \\ Z_1 = \mathrm{floor}(bitxor\_value \times A) + \mathrm{roundn}(Z_0, -4) \end{cases} \tag{8}$$

where floor represents round down for the number; roundn represents the first 4 decimal places of the Z_0 sequence.

Step 5: Generate sequence (B_1, B_2, B_3). The (b_1, b_2, b_3) in the K is taken as the initial value of the LDCML system. Bring u, u_0, e into Eq. (4) and iterate $r \times 3 + random + T$ times. Take the values of the last $r \times 3 + random$ iterations to form (B_1, B_2, B_3).

Step 6: Secondary permutation. Use B_1 and scramble P_2 according to Eq. (1) to generate P_3.

Step 7: Diffusion. Firstly, P_3 is divided into an integer part P_4 and a fractional part P_5. Secondly, use $bitxor\_value$ and B_2 into Eq. (9) for XOR operation on P_4; P_5 is scrambled using the Eq. (1). Finally, the integer part and the fractional part are added to obtain P_6.

$$\begin{cases} T_i = \mathrm{mod}\left(\left\lfloor P\mathrm{i} \times B_2 \times 10^{14} \right\rfloor, bitxor\_value\right) \\ T' = \begin{cases} T'_1 = p1 \oplus T_1 \\ T'_{1+1} = p_{i+1} \oplus T'_i \oplus T_i \end{cases} \end{cases} \tag{9}$$

Step 8: Loop or output ciphertext. Judge whether $round < L$ is true. If it is established, P_6 is ciphertext V. If it is not true, the Step 6 - Step 7 are carried out again by re-entering the cycle.

The meaning of variables in encryption can be found in Table 1. To put it simply, the encryption process of this algorithm is as follows: Firstly, the parameters Key for chaos system are generated in accordance with Section 4.2. Secondly, the 1D chaotic system is used to scramble its plaintext. Thirdly, the LDCML system is used to scramble and diffusion it again. Finally, determine if you need to cycle again and finally generate ciphertext. Figure 4 shows the schematic block diagram in the encryption process.

4.4 Decryption algorithm

The decryption process is the inverse of the encryption algorithm. I will not go into details here. It is worth noting that:

(1) The rewrite operation should be located before the last unwinding of the outer loop, and the sum of the integer parts of the data at this time is the same.

(2) Only need to calculate the value of *random* when decrypting, without calculating random data.

(3) The algorithm only uses the permutation operation on the fractional part, the value will not be lost during the encryption and decryption process, which ensures that the decrypted model is identical to the original model.

Table 1 Table of variables in encryption algorithm

Variables	Meaning
Key	The key used by the encryption process
r	The number of coordinate points in 3D models
P; $P_{i(i=1,2,3,4,5,6)}$	Plaintext matrix; Transition matrix in encryption process
V	Ciphertext matrix
A;$B_{i(i=1,2,3)}$	Random series
Z_0, Z_1	Random points, added to 3D models to hide vertex quantity information
random	Number of random points
round	Number of cycles that have been executed

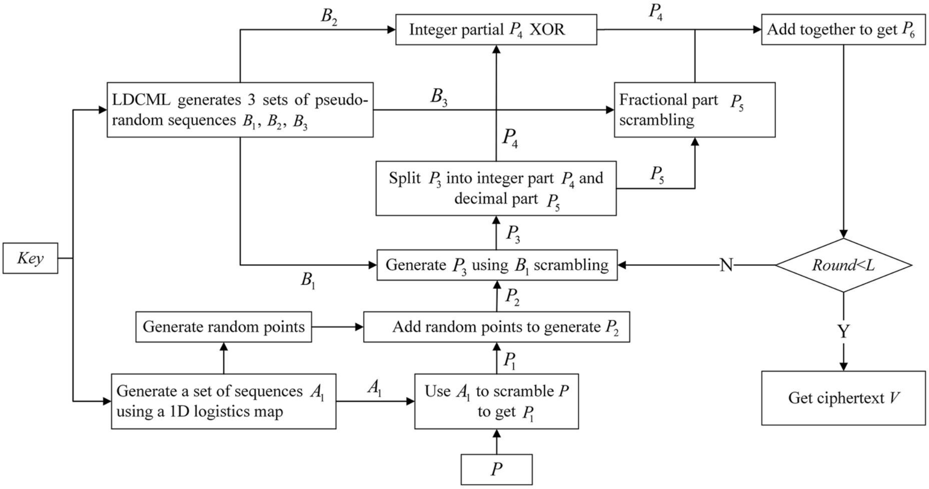

B_2
Integer partial P_4 XOR
P_4
Add together to get P_6
LDCML generates 3 sets of pseudo-random sequences B_1, B_2, B_3
B_3
Fractional part P_5 scrambling
P_4
Split P_3 into integer part P_4 and decimal part P_5
P_5
P_3
Key
Generate P_3 using B_1 scrambling
N
Round<L
B_1
P_2
Generate random points
Add random points to generate P_2
Y
P_1
Generate a set of sequences A_1 using a 1D logistics map
A_1
Use A_1 to scramble P to get P_1
Get ciphertext V
P

Fig. 4 Encryption flow chart

5 Analysis of experimental results

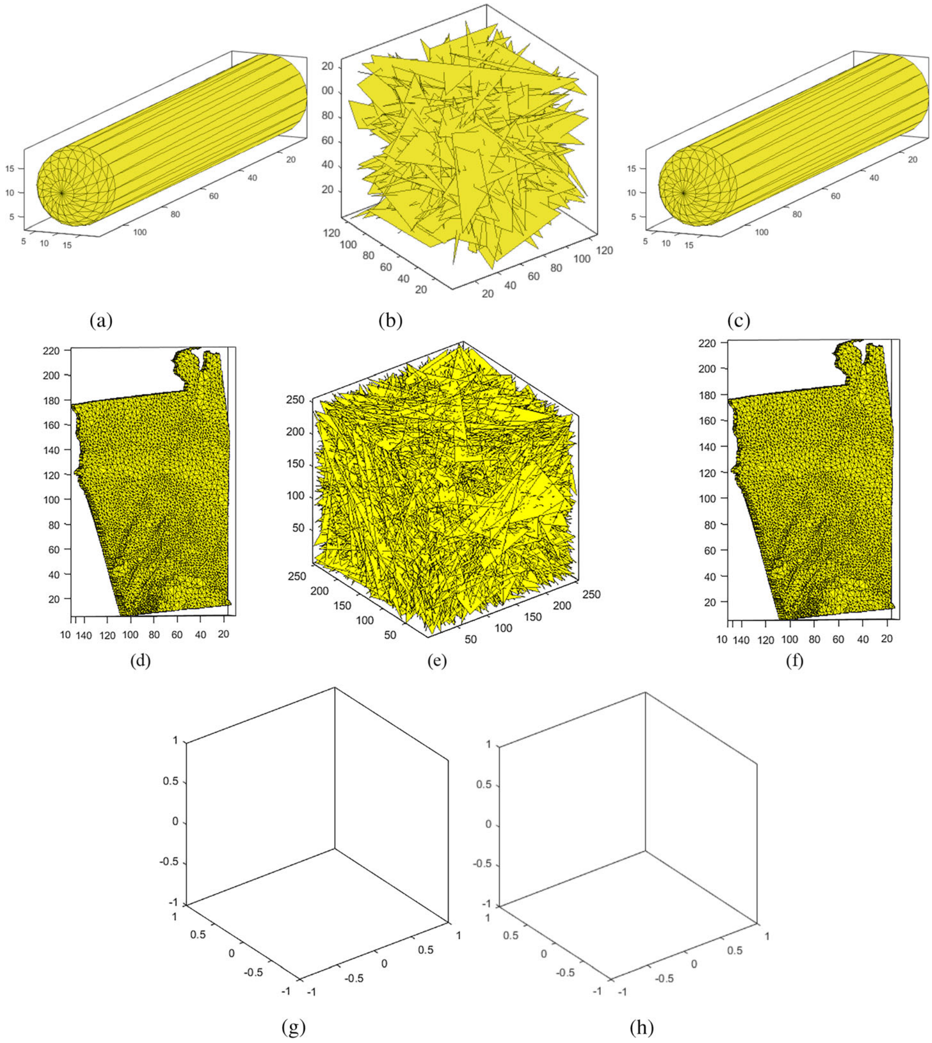
5.1 Experimental results

In the article, according to the plain text, Give a key value: K = 28011745815b1a545e7d064b5998ba73392f11b8d144c82a9ef4e64ddd47. Other parameters: $bitxor\_value$=256, T=1000, $random$=1260. All results will then be displayed in MATLAB, where 3D models of the ciphertext file is read, and 3D models generated by the decryption is shown in Fig. 5. It can be found that 3D models obtained after encryption has completely lost the appearance characteristics of the plaintext. The naked eye observation shows that clear text and ciphertext have the same appearance characteristics. In addition, we subtract the original model from the decrypted model. If there is a difference between the original 3D model and the decrypted 3D model, the stereo structure will be displayed; The original 3D model is completely different from the decrypted model, then the blank space will be displayed. As shown by Fig. 5g and h, the original 3D model of decrypted 3D model is exactly the same as the is obtained.

5.2 Key space analysis

In order to improve the security of the encryption algorithm, there should be enough key space to defend against brute force attacks. All key spaces include the keys used in the proliferation and scrambling process. The valid keys of this algorithm are as follows:

The key is shown in Eq. (6). The key K in the algorithm includes 240 bits. The coefficients of the LDCML system are completely dependent on the key K and the key space is 2^{240}. Also because of the introduction of $bitxor\_value$, T. Therefore, the key space should be greater than 2^{240}, which is greater than the theoretical requirement of 2^{100}. So in theory, the algorithm in this paper can resist brute force attacks.



(a) (b) (c)

(d) (e) (f)

(g) (h)

Fig. 5 Encryption effect display: **a** Plaintext cylindrical model, **b** Ciphertext cylindrical model, **c** Decrypted cylindrical model **d** Plaintext terrain model, **e** Ciphertext terrain model, **f** Decrypted terrain model, **g** Fig. 5(a)-Fig. 5(c), (g) Fig. 5(f)–Fig. 5(f)

5.3 Algorithm complexity analysis

For the encryption algorithm, in addition to considering the security of encryption, it is also necessary to analyze the computational complexity. Assume that the size of the data is $r \times 3$. The most time consuming step in this algorithm, the least squares system needs to generate a set of pseudo-random sequences of length $r \times 3 + \mathrm{T}$ and 3 groups of length $r \times 3 + random + T$ and the time complexity is about $O(4 \times r \times 3)$.

Specifically, when the algorithm is running in windows10 operating system, 2.80GHz CPU frequency, 8G memory and Matlab2016a running software, the encryption time of 89,271 data is 0.261795 s. Obviously, the time performance of the algorithm is excellent.

6 Security analysis

The security analysis method in this paper adopts key sensitivity analysis, histogram analysis, correlation analysis, information entropy analysis and differential attack. The terrain map of Fig. 2b is selected as the test object in this section.

6.1 Key sensitivity analysis

In this algorithm, the initial value of the lattice and the coefficients of the DCML system are generated by the key K. Therefore, the analysis of the key sensitivity is also an indirect analysis. The sensitivity of the LDCML system to the initial state. Of course, the sensitivity of the key is also related to the design of the algorithm. In this section, the sensitivity to the key may be to quantitatively describe the relationship between the two sets of data using NPCR and UACI, calculated as Eq. (10) and Eq. (11), respectively.

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\% \tag{10}$$

$$UACI = \frac{1}{W \times H}\left[\sum_{i,j} \frac{\left|c_1(i,j) - c_2\left(i,j\right)\right|}{bitxor\_value}\right] \times 100\% \tag{11}$$

where c_1 and c_2 denote two ciphertext data, W and H denote the length and width of the matrix. When the NPCR reaches about 1 and the UACI reaches about 0.3346, the algorithm meets the safety standards.

For the key K, the first 120-bit average allocation constitutes the initial value of the three cells of the LDCML system, the [120, 160] bits are part of the coefficient u, and the [161, 200] bits are part of the u_0, [201, 240] The bit is part of u_0. Therefore, the analysis of key sensitivity can be divided into 8 cases. 40th, 80th, 120th, 160th, 200th, and 240th. The last digit of each group is incremented by 1, and the sum of 1 is also used for the sum of the sums. According to Fig. 6, the NPCR and UACI calculated by different key change methods are shown in Table 1. It can be found that the algorithm is fully compliant with security standards.

Before decrypting, change the key one by one as shown in Fig. 6. As shown in Fig. 7, only minor changes have been made to each key and the decrypted 3D model has changed dramatically. It can be seen that the decrypted 3D models are all stacked triangles, losing the original structure of plaintext. On the other hand, according to Eq. (10) and Eq. (11), we carry out numerical analysis of the test results. As shown in Table 2, both NPCR and UACI are close to ideal values. The above conclusions show that the key sensitivity of this algorithm is very good.

6.2 Histogram analysis

The histogram of the data is an important indicator of whether an algorithm can resist statistical analysis attacks [17, 30]. The histogram describes the distribution of pixel values for an image. If the distribution is not uniform, a certain amount of information can be obtained through statistical analysis attacks. By analyzing the statistical properties of ciphertext data, it is easy to

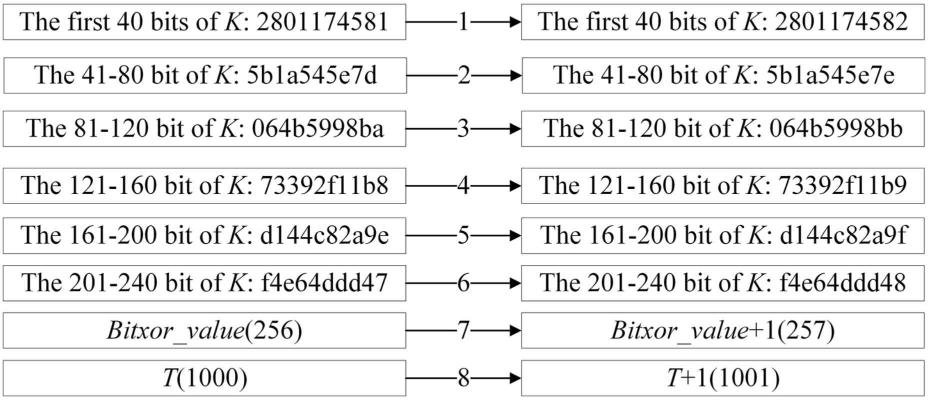

The first 40 bits of K: 2801174581
1
The first 40 bits of K: 2801174582
The 41-80 bit of K: 5b1a545e7d
2
The 41-80 bit of K: 5b1a545e7e
The 81-120 bit of K: 064b5998ba
3
The 81-120 bit of K: 064b5998bb
The 121-160 bit of K: 73392f11b8
4
The 121-160 bit of K: 73392f11b9
The 161-200 bit of K: d144c82a9e
5
The 161-200 bit of K: d144c82a9f
The 201-240 bit of K: f4e64ddd47
6
The 201-240 bit of K: f4e64ddd48
Bitxor_value(256)
7
Bitxor_value+1(257)
T(1000)
8
T+1(1001)

Fig. 6 Key change mode

select ciphertext attacks. The source of information in this article is a floating point number with a maximum of less than 256. The statistics of the histogram in this paper are divided into integers, which is 256. Specifically, these 256 data ranges $s_1 = [0, 1)$, $s_2 = [1, 2)$, $s_3 = [2, 3)$, ..., $s_{254} = [253, 254)$, $s_{255} = [254, 255)$ and $s_{256} = [255, 256)$.

As shown in Fig. 8. This paper tests the histograms of the *x*, *y*, and *z* values of 3D models, that is, the histograms of the three columns in the matrix. It can be found that the distribution of the values in the encrypted data histogram is more averagely.

6.3 Correlation analysis

In three kinds of coordinate data of plaintext x, y, z, it is found through experiments that adjacent coordinate values often have strong correlation. The reason is that STL files are often generated with bottom-up or top-down vertex values. In order to avoid statistical information being used for attacks, it is necessary to reduce the correlation between adjacent data in the ciphertext [17]. In this experiment, 2000 values are compared in plaintext and ciphertext respectively.

As shown in Fig. 9, The distribution of the points in the encrypted graph is more uniform, which indicates that the correlation of coordinate points in the STL file is weaker. The more uniform the distribution of the points, the lower the correlation between the coordinate point.

$$\begin{cases} E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \\ D(x) = \frac{1}{N}\sum_{i+1}^{N} (x_i - E(x))^2 \\ \operatorname{cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))(y_i - E(y)) \\ r_{xy} = \frac{\operatorname{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \end{cases} \tag{12}$$

where *x* and *y* are two adjacent values and *N* is the total number of (x, y) that exists from the matrix. $E(x)$ is the expected and $D(x)$ is the variance.

We use Eq. (12) to digitize the results to better illustrate the results. The correlation coefficients of the plaintext and ciphertext data are given in Table 3. As can be found from the table, there is a great correlation between adjacent data of the original plaintext data. In

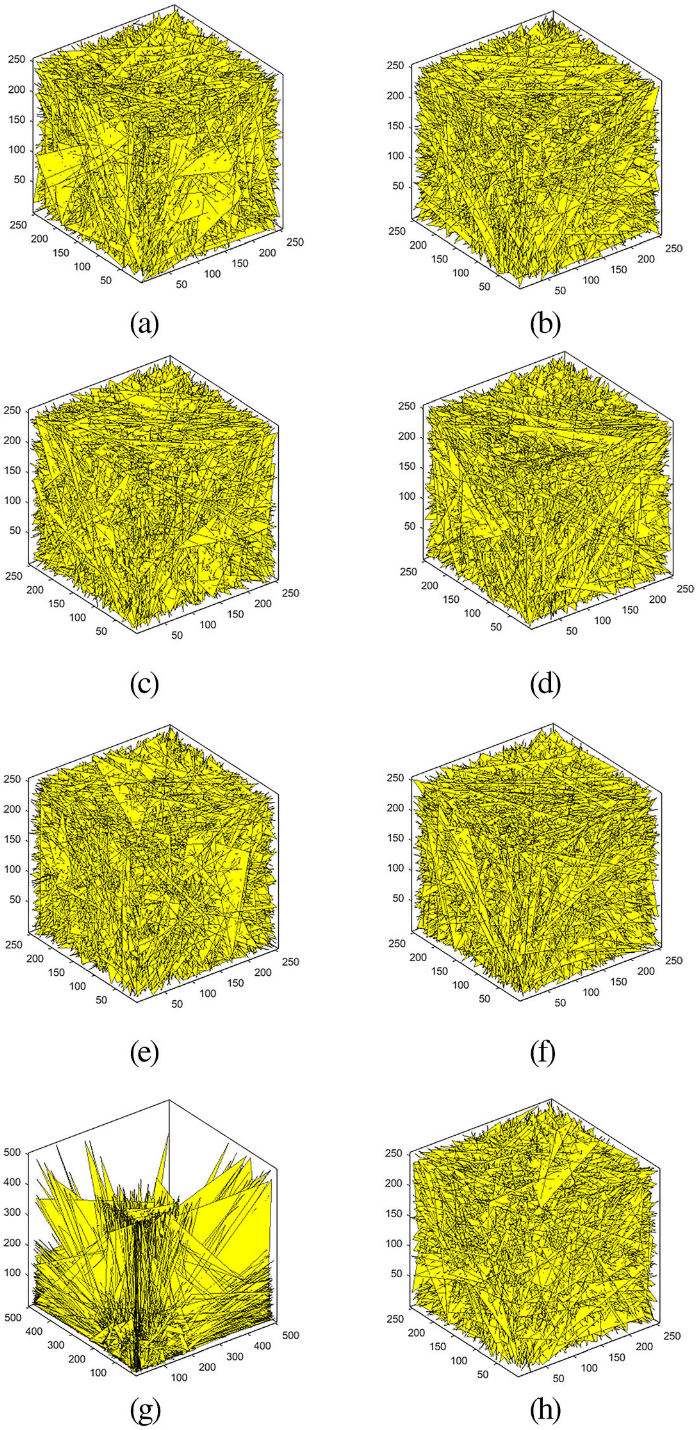
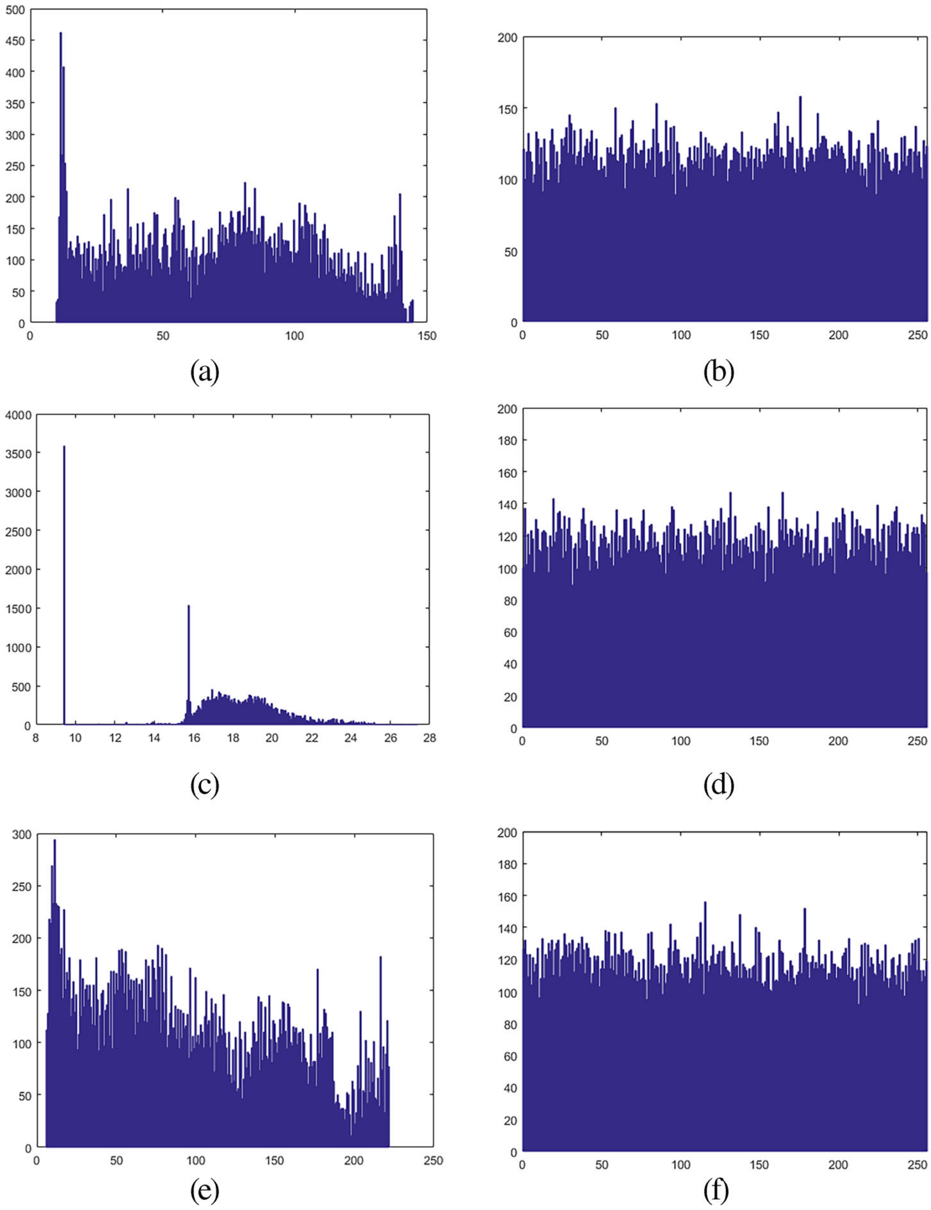


Fig. 7 Key change mode: **a** First change mode, **b** Second change mode, **c** Third change mode, **d** Fourth change mode, **e** Fifth change mode, **f** Sixth change mode, **g** Seventh change mode, **h** Eighth change mode

Table 2 Changes the NPCR and UACI values of the key

Shift gears	1	2	3	4	5	6	7	8
NPCR	0.9959	0.9959	0.9961	0.9962	0.9961	0.9958	0.9983	0.9963
UACI	0.3326	0.3322	0.3330	0.3329	0.3333	0.3326	0.3372	0.3321



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 8 Information entropy analysis: **a** Plaintext x direction histogra, **b** Ciphertext x direction histogram, **c** Plaintext y direction histogram, **d** Ciphertext y direction histogram, **e** Plaintext z direction histogram, **f** Ciphertext z direction histogram

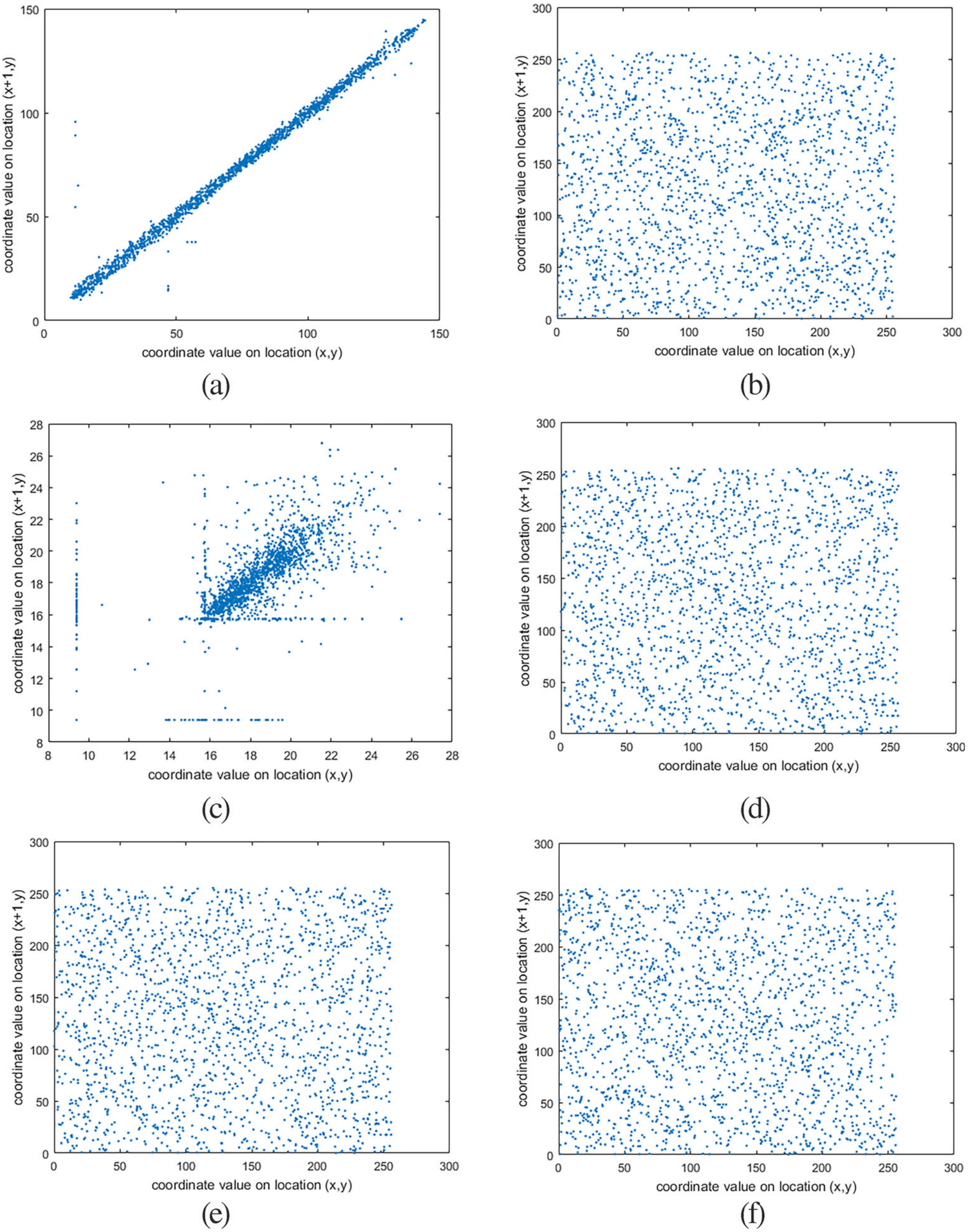


Fig. 9 Adjacent data correlation: **a** Plaintext x-direction correlation, **b** Plain text y-direction correlation, **c** Plain text z-direction correlation, **d** Ciphertext x-direction correlation, **e** Ciphertext y-direction correlation, **f** Ciphertext z-direction correlation

order to be able to resist statistical attacks, the correlation between adjacent data must be effectively reduced. The correlation between adjacent data of ciphertext data encrypted by the algorithm is very low, close to zero. Therefore, the algorithm in this paper can better resist statistical analysis attacks.

Table 3 Correlation coefficients between plaintext and ciphertext in different directions

Direction	Plaintext	Ciphertext
X direction	0.9923	−0.0254
Y direction	0.7365	−0.0097
Z direction	0.8950	0.0049

6.4 Information entropy analysis

Information entropy was proposed by Shannon in 1948 and it reflects the degree of confusion in the system. The higher the order of the system, the lower the information entropy of the system. The more disordered the system, the higher the information entropy of the system. The coordinate data of 3D models is essentially an information source, so it is analyzed by information entropy. The more uniform the distribution of values, the less effective information the information source has. Its definition is [30]:

$$H(s)=\sum_{i=0}^{n}p(s_i)\log_2\frac{1}{p(s_i)} \tag{13}$$

where s is the information source, i is the number of data, and n is the total number of data. The source of information in this article is a floating point number with a maximum of less than 256. This article is divided into integers, which is 256. Specifically, these 256 states are $s_1=[0,1)$, $s_2=[1,2)$, $s_3=[2,3)$, ..., $s_{254}=[253,254)$, $s_{255}=[254,255)$ and $s_{256}=[255,256)$.

In addition, $p(s_i)$ indicates the probability of occurrence of s_i. Therefore, in theory, the maximum information entropy of this information source is $\log_2 256=8$. Information entropy of plaintext is 6.8030, and Information entropy of ciphertext is 7.9980. The information entropy of the encrypted data is significantly improved and is close to the ideal value of 8. The ciphertext encrypted by the algorithm is difficult to disclose information, and can better resist statistical attacks.

6.5 Differential attack

Differential attack is a method of making minor changes to the plaintext, obtaining ciphertext before and after modifying the ciphertext, and performing data analysis on the data. Data analysis is performed on the data source to obtain a key. Therefore, a good data encryption system should be able to make small differences in the plaintext, so that the ciphertext changes greatly to resist different attacks. The data change rate (NPCR) and the uniform mean change intensity (UACI) are shown in Eq. (10) and Eq. (11), respectively [30]. This article adds 1 to the first data of 9.6790, which becomes 10.6790. Encrypt it with the same algorithm and key. Theoretically, the closer the values of NPCR and UACI are to 100% and 33.4635%, the better. NPCR and UACI after changing a value are 0.9958% and 0.3311%. It can be found that the algorithm can better resist differential attacks.

7 Conclusion

Chaos theory is one of the hotspots of modern cryptography research. In this paper, a fast encryption scheme of 3D models based on chaotic system is proposed. It's rapidity mainly comes from simplifying 3D objects to 2D objects for processing. In the era of big data wide application in 3D models, the scheme has certain practical significance and universality. The experimental results and security analysis show that the scheme has large key space, is sensitive to key and plaintext data, can resist common attack methods, and has good encryption effect.

In this paper, only the most commonly used 3D models file encryption and decryption processing. For other 3D model files, they often have more structural features. For example, polygons and texture. This scheme can not fully encrypt this kind of model. The future work will be to propose a general encryption scheme for all 3D model files to solve the above problems.

Acknowledgements This research is supported by the National Natural Science Foundation of China (No: 61672124), the Password Theory Project of the 13th Five-Year Plan National Cryptography Development Fund (No: MMJJ20170203), Liaoning Province Science and Technology Innovation Leading Talents Program Project (No: XLYC1802013), Key R&D Projects of Liaoning Province (No: 2019JH2/10300057). The authors declare that they have no conflict of interest.

References

1. Alsmirat MA, Al-Alem F, Al-Ayyoub M et al (2019) Impact of digital fingerprint image quality on the fingerprint recognition accuracy. Multimed Tools Appl 78(3):3649–3688
2. Altaf M, Ahmad A, Khan FA et al (2018) Computationally efficient selective video encryption with chaos based block cipher. Multimed Tools Appl 77:27981–27995
3. Babaei M (2013) A novel text and image encryption method based on chaos theory and DNA computing. Nat Comput 12(1):101–107
4. Berman B (2012) 3-D printing: The new industrial revolution. Business Horizons 55(2):155–162
5. Brown AC, de Beer D (2013) Development of a stereolithography (STL) slicing and G-code generation algorithm for an entry level 3-D printer. Africon Institute of Electrical and Electronics Engineers 2013:1–5
6. Chai X, Chen Y, Broyde L (2017) A novel chaos-based image encryption algorithm using DNA sequence operations. Opt Lasers Eng 88:197–213
7. Chen G, Mao Y, Chui CK (2004) A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos, Solitons Fractals 21(3):749–761
8. Chen DY, Tian XP, Shen YT et al (2003) On visual similarity based 3D model retrieval. Computer Graphics Forum 22(3):223–232
9. Fridrich J (1998) Symmetric ciphers based on two-dimensional chaotic maps. International Journal of Bifurcation and Chaos 8(06):1259–1284
10. Gao Y, Dai Q, Zhang NY (2010) 3D model comparison using spatial structure circular descriptor. Pattern Recogn 43(3):1142–1151
11. Gupta B, Agrawal DP, Wang H et al (2018) Computer and cyber security: principles, algorithm, applications, and perspectives. CRC Press, Boca Raton
12. Gupta B, Agrawal DP, Yamaguchi S et al (2016) Handbook of research on modern cryptographic solutions for computer and cyber security. IGI Global, Hershey
13. Ibtihal M, Hassan N (2017) Homomorphic encryption as a service for outsourced images in mobile cloud computing environment. International Journal of Cloud Applications and Computing 7(2):27–40
14. Jin X, Zhu S, Xiao C et al (2017) 3D textured model encryption via 3D Lu chaotic mapping. SCIENCE CHINA Inf Sci 60(12):122107
15. Jolfaei A, Wu XW, Muthukkumarasamy V (2014) A 3D object encryption scheme which maintains dimensional and spatial stability. Institute of Electrical and Electronics Engineers Transactions on Information Forensics and Security 10(2):409–422
16. Kaminsky W, Snyder T, Stone-Sundberg J et al (2014) One-click preparation of 3D print files (*. stl,*. wrl) from*. cif (crystallographic information framework) data using Cif2VRML. Powder Diffract 29(S2):S42–S47
17. Kaneko K (1989) Pattern dynamics in spatiotemporal chaos. Physica D 34(1-2):1–41

18. Kusaka M, Sugimoto M, Fukami N et al (2015) Initial experience with a tailor-made simulation and navigation program using a 3-D printer model of kidney transplantation surgery. Transplant Proc 47(3):596–599
19. Li X, Zhou C, Xu N (2018) A Secure and Efficient Image Encryption Algorithm Based on DNA Coding and Spatiotemporal Chaos. International Journal Network Security 20(1):110–120
20. Liu AA, Nie WZ, Gao Y et al (2018) View-based 3-d model retrieval: a benchmark. Institute of Electrical and Electronics Engineers Transactions on Cybernetics 48(3):916–928
21. Matthews R (1984) On the derivation of a "Chaotic" encryption algorithm. Cryptologia 8(1):29–41
22. May RM (1976) Simple mathematical models with very complicated dynamics. Nature 261(5560):459–467
23. Parvees MYM, Samath JA, Bose BP (2018) Audio encryption-a chaos-based data byte scrambling technique. Int J Appl Syst Stud 8(1):51–75
24. Preishuber M, Hütter T, Katzenbeisser S et al (2018) Depreciating motivation and empirical security analysis of chaos-based image and video encryption. Institute of Electrical and Electronics Engineers Transactions on Information Forensics and Security 13(9):2137–2150
25. Rey AM (2015) A Method to encrypt 3D solid objects based on three-dimensional cellular automata. International Conference on Hybrid Artificial Intelligence Systems. Springer, Cham, pp. 427-438
26. Sadkhan SB, Mohammed RS (2015) Proposed random unified chaotic map as PRBG for voice encryption in wireless communication. Procedia Computer Science 65:314–323
27. Saraf KR, Jagtap VP, Mishra AK (2014) Text and image encryption decryption using advanced encryption standard. International Journal of Emerging Trends & Technology in Computer Science 3(3):118–126
28. Tang J, Zhang F (2017) A new code-based encryption scheme and its applications. International Journal of High Performance Computing and Networking 10(6):515–523
29. Ventola CL (2014) Medical applications for 3D printing: current and projected uses. Pharmacy and Therapeutics 39(10):704–711
30. Wang XY, Feng L, Wang SB et al (2018) Spatiotemporal chaos in coupled logistic map lattice with dynamic coupling coefficient and its application in image encryption. IEEE Access 6:39705–39724
31. Wang X, Feng L, Zhao H (2019) Fast image encryption algorithm based on parallel computing system. Inf Sci 486:340–358
32. Wang X, Gao S (2020) Image encryption algorithm for synchronously updating Boolean networks based on matrix semi-tensor product theory. Inf Sci 507:16–36
33. Wang XY, Li P, Zhang YQ et al (2018) A novel color image encryption scheme using DNA permutation based on the Lorenz system. Multimed Tools Appl 77(5):6243–6265
34. Wang XY, Liu CM (2017) A novel and effective image encryption algorithm based on Chaos and DNA encoding. Multimed Tools Appl 76(5):6229–6245
35. Wang X, Liu L, Zhang Y (2015) A novel chaotic block image encryption algorithm based on dynamic random growth technique. Opt Lasers Eng 66:10–18
36. Wang J, Wang C (2018) Full secure identity-based encryption scheme over lattices for wireless sensor networks in the standard model. International Journal of High Performance Computing and Networking 12(2):111–117
37. Wang C, Wang X, Xia Z et al (2019) Ternary radial harmonic Fourier moments based robust stereo image zero-watermarking algorithm. Inf Sci 470:109–120
38. Wang X, Zhang HL (2016) A novel image encryption algorithm based on genetic recombination and hyper-chaotic systems. Nonlinear Dynamics 83(1-2):333–346
39. Wang XY, Zhang YQ, Bao XM (2015) A novel chaotic image encryption scheme using DNA sequence operations. Opt Lasers Eng 73:53–61
40. Xu SJ, Wang JZ, Yang SX (2008) An improved image encryption algorithm based on chaotic maps. Chinese Physics B 17(11):4027
41. Yu C, Li J, Li X et al (2018) Four-image encryption scheme based on quaternion Fresnel transform, chaos and computer generated hologram. Multimed Tools Appl 77(4):4585–4608
42. Zhang J, Gao H (2019) A compact construction for non-monotonic key-policy attribute-based encryption. International Journal of High Performance Computing and Networking 13(3):321–330
43. Zhang YQ, Wang XY (2014) A symmetric image encryption algorithm based on mixed linear–nonlinear coupled map lattice. Inf Sci 273:329–351
44. Zhang YQ, Wang XY (2014) Analysis and improvement of a chaos-based symmetric image encryption scheme using a bit-level permutation. Nonlinear Dynamics 77(3):687–698
45. Zhang H, Wang XY, Wang SW et al (2017) Application of coupled map lattice with parameter q in image encryption. Opt Lasers Eng 88:65–74

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Wang Xingyuan received the Ph.D. degree in computer software and theory from Northeast University, China, in 1999. From 1999 to 2001, he was a Post-Doctoral Researcher at Northeast University. He is currently a Professor with School of Information Science and Technology, Dalian Maritime University, China. He has published three books and over 400 scientific papers in refereed journals and proceedings. His research interests include nonlinear dynamics and control, image processing, chaos cryptography, systems biology, and complex networks.

Xu Mingxiao Dalian Maritime University Graduate School of Information Science and Technology, the main research interests are: image processing, chaotic cryptography.

Li Yong graduated from the computer software of Dalian University of Technology in July 1982. He is currently a adjunct Professor with School of Information Science and Technology, Dalian Maritime University, China. Mainly engaged in information engineering and database, software engineering research.