International Journal of
Geo-Information

MDPI

Article

Vector Map Encryption Algorithm Based on Double Random Position Permutation Strategy

Xiaolong Wang [1,2,3], Haowen Yan [1,2,3,*] and Liming Zhang [1,2,3]

[1] Faculty of Geomatics, Lanzhou Jiaotong University, Lanzhou 730070, China; 0219777@stu.lzjtu.edu.cn (X.W.); zlm@lzjtu.edu.cn (L.Z.)

[2] National Local Joint Engineering Research Center of Technologies and Application for National Geographic State Monitoring, Lanzhou 730070, China

[3] Gansu Provincial Engineering Laboratory for National Geographic State Monitoring, Lanzhou 730070, China

* Correspondence: yanhw@mail.lzjtu.cn; Tel.: +86-1360-931-0452

Abstract: Encryption of vector maps, used for copyright protection, is of importance in the community of geographic information sciences. However, some studies adopt one-to-one mapping to scramble vertices and permutate the coordinates one by one according to the coordinate position in a plain map. An attacker can easily obtain the key values by analyzing the relationship between the cipher vector map and the plain vector map, which will lead to the ineffectiveness of the scrambling operation. To solve the problem, a vector map encryption algorithm based on a double random position permutation strategy is proposed in this paper. First, the secret key sequence is generated using a four-dimensional quadratic autonomous hyperchaotic system. Then, all coordinates of the vector map are encrypted using the strategy of double random position permutation. Lastly, the encrypted coordinates are reorganized according to the vector map structure to obtain the cipher map. Experimental results show that: (1) one-to-one mapping between the plain vector map and cipher vector map is prevented from happening; (2) scrambling encryption between different map objects is achieved; (3) hackers cannot obtain the permutation key value by analyzing the pairs of the plain map and cipher map.

Keywords: vector map; chaotic system; scrambling encryption; global objects

check for updates

Citation: Wang, X.; Yan, H.; Zhang, L. Vector Map Encryption Algorithm Based on Double Random Position Permutation Strategy. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 311. https://doi.org/10.3390/ijgi10050311

Academic Editors: Giedrė Beconytė and Wolfgang Kainz

Received: 25 February 2021
Accepted: 1 May 2021
Published: 7 May 2021

1. Introduction

Vector maps, the most important part of national basic geographic information [1–3], are indispensable resources in economic development and national security [4–6], which have been widely applied in many fields [7–9], such as resources and environments [10,11], disaster and emergency management [12], economic and social development [13], and health and life health [14]. Vector maps are of great value because collection, processing, and storage of such data rely on expensive surveying instruments, global navigation systems (e.g., Multi-GNSS, GPS, and BeiDou), and a large amount of physical labor resources [15–17]; therefore, vector maps generally cannot be freely used without their owners' permission.

However, the rapid development of science and technology in recent years has made it easy to acquire, access, spread, copy, and store vector maps [18], leading addressing copyright issues of vector maps to become increasingly urgent. Some countries and militaries have implemented a series of laws, rules, regulations, and institutions to solve this increasingly urgent issue [19]. For example, in 2005, a guideline was issued by the U.S. advising the use of effective standards and regulations to protect geographic information from piracy. In 2007, a regulation was published by Russia to regulate their geographic information. In 2017, China revised its Surveying and Mapping Law to protect its geographic information [20]. Germany, U.K., Japan, India, and some other countries have also issued laws and regulations on geographic information protection [21]. However, copyright protection for vector maps (one type of the most important geographic information) not only requires laws and regulations but also needs technical support.

Fast advancement of vector map copyright protection techniques has been witnessed in recent decades, which can be divided into two types: accountability and precaution. The accountability includes digital watermarking [22–24], digital fingerprinting [25–27], and blockchain [28–30]. Digital watermarking is used to identify the copyright of the vector map, digital fingerprinting performs well in tracing the original pirate, and blockchain is enabled by integrating several core technologies, such as cryptographic hash, digital signature (based on asymmetric cryptography), and distributed consensus mechanism, which can be applied for protecting data copyright and managing patents [31]. The precaution includes user access control [32–34] and vector map encryption. User access control can strictly manage the service period and environment of vector maps through confirming the legitimacy of the users. On vector map encryption, cryptography theory is applied to encrypt vector maps for ensuring the security of the maps in the ciphertext. Once the legal validity of the users is confirmed, the valid vector maps in plaintext state can be provided to the legitimate users. Furthermore, any unauthorized users are not allowed to use, extract, and modify vector maps. This study focuses on vector map encryption.

Vector map encryption is an effective technique to protect vector maps from piracy. The existing encryption algorithms are mainly divided into three categories (Figure 1): the encryption algorithms based on traditional cryptography, the encryption algorithms based on chaotic systems, and the selective encryption algorithms.

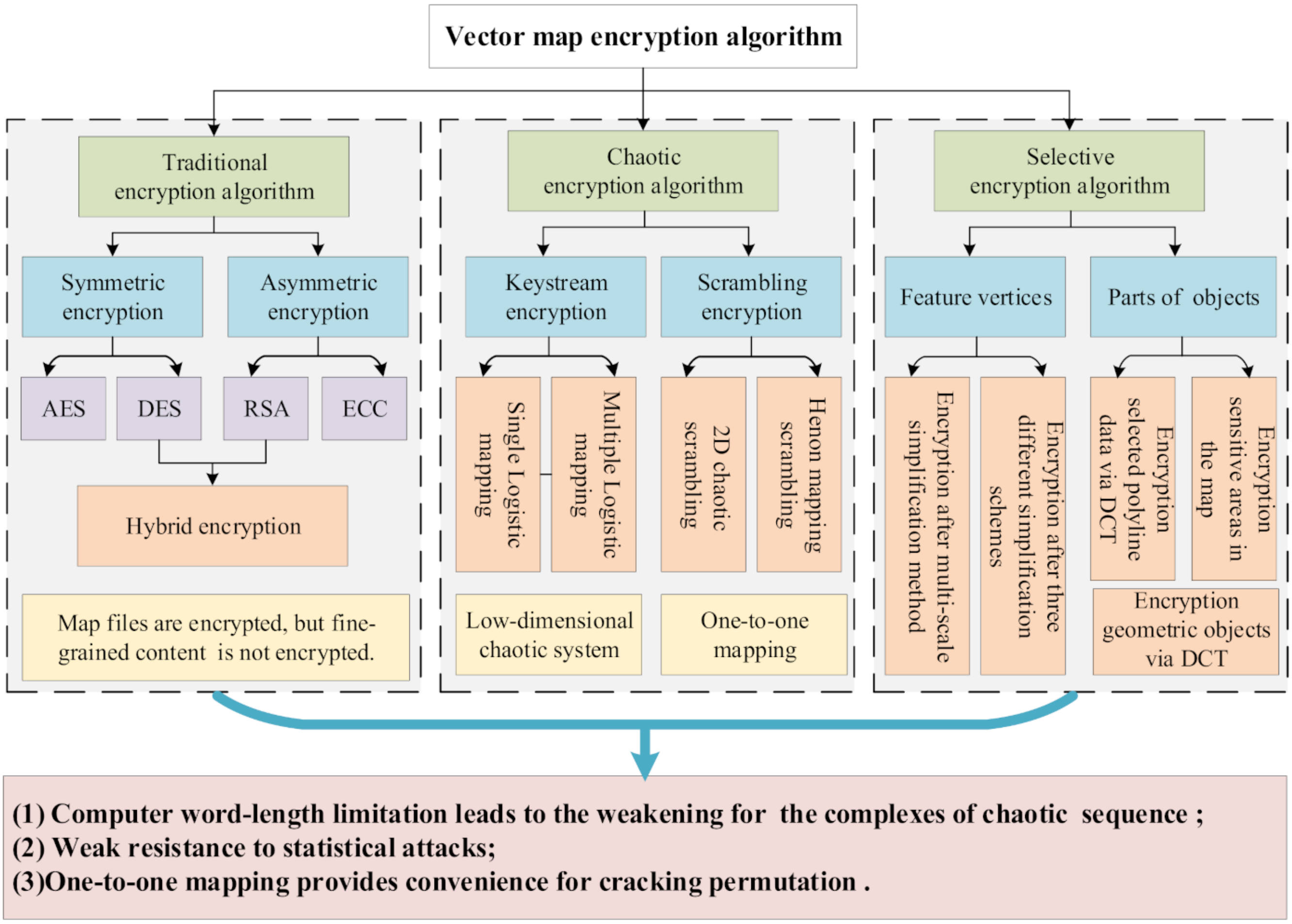

Vector map encryption algorithm
Traditional encryption algorithm
Symmetric encryption
Asymmetric encryption
AES
DES
RSA
ECC
Hybrid encryption
Map files are encrypted, but fine-grained content is not encrypted.
Chaotic encryption algorithm
Keystream encryption
Scrambling encryption
Single Logistic mapping
Multiple Logistic mapping
2D chaotic scrambling
Henon mapping scrambling
Low-dimensional chaotic system
One-to-one mapping
Selective encryption algorithm
Feature vertices
Parts of objects
Encryption after multi-scale simplification method
Encryption after three different simplification schemes
Encryption selected polyline data via DCT
Encryption sensitive areas in the map
Encryption geometric objects via DCT
(1) Computer word-length limitation leads to the weakening for the complexes of chaotic sequence ;
(2) Weak resistance to statistical attacks;
(3)One-to-one mapping provides convenience for cracking permutation .

Figure 1. Categories of the vector map encryption algorithms.

(1) The encryption algorithms based on traditional cryptography are applied to encrypt vector map files, which are mainly classified into symmetric encryption and asymmetric encryption. ① The symmetric encryption chiefly includes data encryption standard (DES) [35] and advanced encryption standard (AES) [36]. ② The asymmetric encryption includes RSA algorithms [37] and elliptic curve cryptosystems (ECC) [38]. Both algorithms have their own pros and cons. The symmetric has high efficiency, but key management is difficult; the asymmetric encryption key management is easy, but encryption and decryption speed is slow. Thus, Zhang Shanshan [39] used the efficiency advantages of symmetric encryption to encrypt original map files and then combined the key management advantages of asymmetric encryption to encrypt the key in symmetric encryption. The security of the encryption algorithm and the speed of encryption and decryption are improved utilizing the respective advantages of the two types of algorithms. However, this algorithm [39] only encrypts the entire map files, it does not encrypt vector map fine-grained content (e.g., polyline objects, polygon objects, vertices, etc.).

(2) The encryption algorithms based on chaotic systems are able to encrypt map fine-grained content, which includes keystream encryption algorithms and scrambling encryption algorithms. ① The keystream encryption algorithms mainly include: (a) the vector maps that are encrypted utilizing the keystream, which is mapped from a random sequence generated by logistic mapping [40]; and (b) the binary sequence of non-linear combination that is generated utilizing multiple logistic generators, which is performed by the modular operation to obtain the keystream, and then the vector map is encrypted by the keystream [41]. ② The scrambling encryption algorithms include: (a) the scrambling encryption algorithm between different objects that is realized based on a two-dimensional chaotic sequence [42]; and (b) a scrambling algorithm that destroys the adjacent coordinate correlation and storage order of the vector maps [9]. Although the above algorithms can encrypt vector maps, there are still some shortcomings. Firstly, the original map is scanned from the first coordinate to the last one and is shuffled by using a key sequence one by one. This sequential one-to-one mapping provides great convenience for cracking the permutation through analyzing the pairs of the original map and cipher map. Furthermore, the limitation of computer word-length leads to the weakening of the dynamic characteristics for low-dimensional chaotic systems, which seriously affects the security of chaotic encryption.

(3) Selective encryption is an algorithm that encrypts feature vertices and encrypts parts of map objects (polylines and polygons). ① The algorithms for encrypting feature vertices include the geometric objects that are extracted from vector maps. The backbone object is computed from geometric objects. The backbone object is selectively simplified by the multi-scale simplification algorithm to obtain feature vertices of backbone object. The feature vertices are encrypted by the AES algorithm and the key. The remaining vertices and encrypted features vertices are randomly scrambled by a set of random Gaussian numbers [43]. The algorithms for encrypting feature vertices also include the feature vertices that are extracted by using three different simplification algorithms, after which the feature vertices are then encrypted [44]. ② The encryption algorithms for selecting parts of map objects include: (a) the geometric objects of the vector map selected through geometric transform, which are encrypted in the DCT domain [45]; (b) the polyline data that are selected from vector map, which is performed by perceptual encryption via DCT transformation [46]; and (c) the data in sensitive areas, which are encrypted based on Euclidean average distance [47]. The above algorithms encrypt the partial objects of the vector map. However, some selective encryption algorithms are weak against statistical attacks because they do not scramble vertices. Others are resistant to statistical attacks, but such algorithms adopt one-to-one mapping to encrypt vector maps. Hackers can conveniently obtain permutation index values by analyzing the pairs of the plain map and cipher map.

In a word, the existing research still has the following shortcomings (Figure 1). (1) The limitation of computer word-length leads to the weakening of the dynamic characteristics for low-dimensional chaotic systems, which seriously affects the complexes of chaotic sequence [40,41]. (2) The single index sequence is used to shuffle one by one, which provides great convenience for cracking permutation [9,42]. (3) Selection encryption causes weak resistance to statistical attacks [44–47], and there exists great convenience for cracking permutation [43]. To solve the above problems, this paper proposes a vector map encryption algorithm based on a double random position permutation strategy. To begin with, a four-dimensional quadratic autonomous hyperchaotic system (4-D hyperchaotic system) is utilized to generate the key sequence. Then, double random position permutation (DRPP) is used to encrypt global objects from vector maps. Finally, the encrypted objects are reorganized according to the map structure to get the cipher map.

The remainder of this paper is organized as follows. Section 2 describes some principles and encryption schemes in detail. Experimental results and performance of the proposed algorithm are discussed and evaluated in Section 3. Finally, a conclusion is drawn in Section 4.

2. Methods

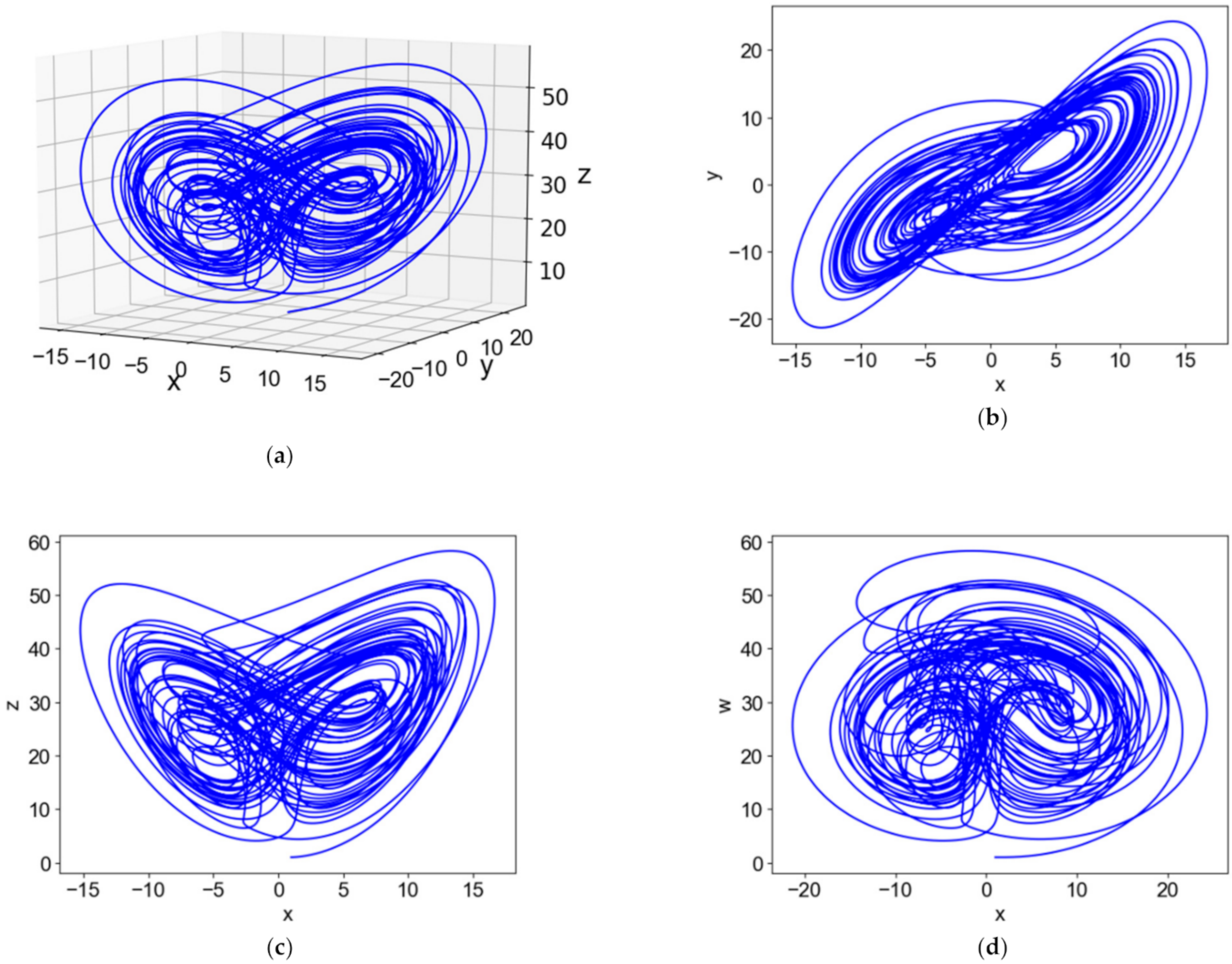
2.1. Four-Dimension Hyperchaotic System

Shannon [48] proposed that different content encryption with unique keys is an excellent encryption strategy. The chaotic system is very sensitive to initial values and parameters, and a slight change in initial values or parameters may lead to completely different chaotic dynamics. The different content encryption with unique keys can be achieved by a random key generated chaotic sequence. However, the limitation of computer word-length leads to the weakening of the dynamic characteristics for low-dimensional chaotic systems, which seriously affects the security of chaotic encryption [49].

To ensure the complexity of the chaotic sequence and to improve the security of the encryption algorithm, a four-dimensional quadratic autonomous hyperchaotic system (4-D hyperchaotic system, as shown in Equation (1)) is utilized to encrypt the vector map, which means that an attacker cannot decrypt the chaotic system by reconstructing the attractor in phase space. In addition, it is also very difficult to obtain initial values and parameters via brute force attacks, and it is given by [50]:

$$\begin{cases} \dot{x} &= a(y-x) \\ \dot{y} &= bx - y + ew - xz \\ \dot{z} &= xy + x^2 - cz \\ \dot{w} &= -dy \end{cases} \tag{1}$$

where x, y, z, w are state variables, and a, b, c, d, e are system parameters. When a = 10, b = 28, c = 8/3, d = 1, e = 16, this system is hyperchaotic, and its attractor is shown Figure 2.



(**a**)

(**b**)

(**c**)

(**d**)

Figure 2. Phase portraits of the 4-D hyperchaotic system with system parameters of a = 10, b = 28, c = 8/3, d = 1, and e = 16 and initial values of (1, 1, 1, and 1). (**a**) Projection on x-y-z plane; (**b**) projection on x-y plane; (**c**) projection on x-z plane; (**d**) projection on x-w plane.

2.2. Generating the Initial Values and System Parameters for the 4-D Hyperchaotic System

Based on the SHA-512 hash value of the original map file and external key, the initial values and system parameters of the 4-D hyperchaotic system are computed, and the steps are as follows.

Step1: Perform SHA-512 hash function on the original map file and the external key to get a 512-bit hash key H_k and initial key U_k; next, divide H_k and U_k into 64 8-bit groups, respectively, and then convert 64 8-bit groups of H_k into their decimal values h_{k1}, h_{k2}, ... , h_{k64}; convert 64 8-bit groups of U_k into 16 decimal values e_1, e_2, e_3, e_4.

Step2: Add 16 e_1, e_2, e_3, e_4 in Step 1 to get U_{k_sum}, and obtain U_{k_index} via Formula (1), and then choose sequence according to U_{k_index}.

$$U_{k\_index} = mod(U_{k\_sum}, 16) + 1, U_{k\_index} \in [1, 16] \quad (2)$$

Step3: Four parameters p_1, p_2, p_3, p_4 are obtained via

$$\begin{cases} p_1 &= e_1 + \frac{1}{|L|}((h_{k1} \oplus h_{k2} \oplus h_{k3} \oplus \cdots \oplus h_{k16}) + (h_{k17} \oplus h_{k18} \oplus h_{k19} \oplus \cdots \oplus h_{k32})) \\ p_2 &= e_2 + \frac{1}{|L|}((h_{k33} \oplus h_{k34} \oplus h_{k35} \oplus \cdots \oplus h_{k48}) + (h_{k49} \oplus h_{k50} \oplus h_{k51} \oplus \cdots \oplus h_{k64})) \\ p_3 &= e_3 + \frac{1}{|L|}((h_{k1} \oplus h_{k2} \oplus h_{k3} \oplus \cdots \oplus h_{k16}) + (h_{k17} + h_{k18} + h_{k19} + \cdots + h_{k32})) \\ p_4 &= e_4 \times |L| \times \frac{sum(h_{k17}, h_{k18}, h_{k19}, \cdots, h_{k32})}{max(h_{k17}, h_{k18}, h_{k19}, \cdots, h_{k32})} \end{cases} \tag{3}$$

where $x \oplus y$ is the bit-exclusive-or (bit-xor) operation between x and y; $sum(h_{k17}, h_{k18}, k_{19}, \ldots, h_{k32})$ is used to get the sum of $h_{k17}, h_{k18}, h_{k19}, \ldots, h_{k32}$; $max(h_{k17}, h_{k18}, h_{k19}, \ldots, h_{k32})$ is used to find the maximum value of $h_{k17}, h_{k18}, h_{k19}, \ldots, k_{32}$; $e_1, e_2, e_3, e_4 \in (0, +\infty)$ are four initial keys; and $|L|$ is the sum of the number of vertices of all objects.

Step4: Four parameters u_x, u_y, u_z, u_w are calculated via bringing the obtained p_1, p_2, p_3, p_4 in Step 3 into the following Equation:

$$\begin{cases} u_x &= mod((p_1 + p_2 + p_3) \times 10^5, 512)/512 \\ u_y &= mod((p_2 + p_3 + p_4) \times 10^5, 512)/512 \\ u_z &= mod((p_1 + p_2 + p_3 + p_4) \times 10^5, 512)/512 \\ u_w &= mod((p_1 + p_4) \times 10^5, 512)/512 \end{cases} \tag{4}$$

where *floor(x)* gives the greatest integer less than or equal to x. In this encryption algorithm, parameters u_x, u_y, u_z, u_w are used as initial values of the 4-D hyperchaotic system.

2.3. Double Random Position Permutation

Scrambling encryption is an encryption technology that destroys the correlation and storage order of adjacent data, and vector map is a kind of graphic data organized by map objects one by one, whose vertices in each object have obvious location characteristics and order characteristics. Therefore, scrambling encryption can disrupt its adjacent correlation and spatial order, thus achieving the goal of encrypting vector maps. There are two types of scrambling for vector maps: global object scrambling and scrambling in the same object. Global object scrambling refers to global scrambling between all objects (Figure 3c), while scrambling in the same object refers to scrambling within the same object (Figure 3b). Obviously, global object scrambling has a better effect.

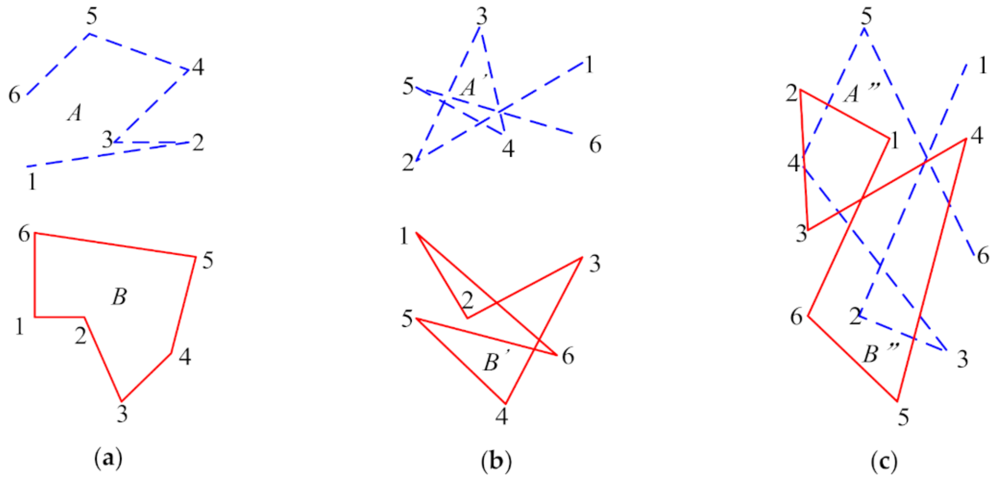


Figure 3. The vector map scrambling principle. (**a**) Original data; (**b**) scrambling in same object; (**c**) global object scrambling.

Assume the size of the original map is $|P_i|$. In the traditional sequential scrambling process shown in Figure 4, scan the original map from the first coordinate (left) to the last (right) one, and scramble them by use of the index sequence D one by one. This sequence one-to-one mapping provides great convenience for cracking the scrambling operation via analyzing the pairs of the original map and cipher map. To solve this problem, a double random position permutation (DRPP) [51] is illustrated and introduced in Figure 5. Two index sequences are applied. Firstly, index D1 is used to pick up the coordinate to be shuffled from the original map. Secondly, index D2 is utilized to map it to another random location. Finally, the scrambled map is obtained.

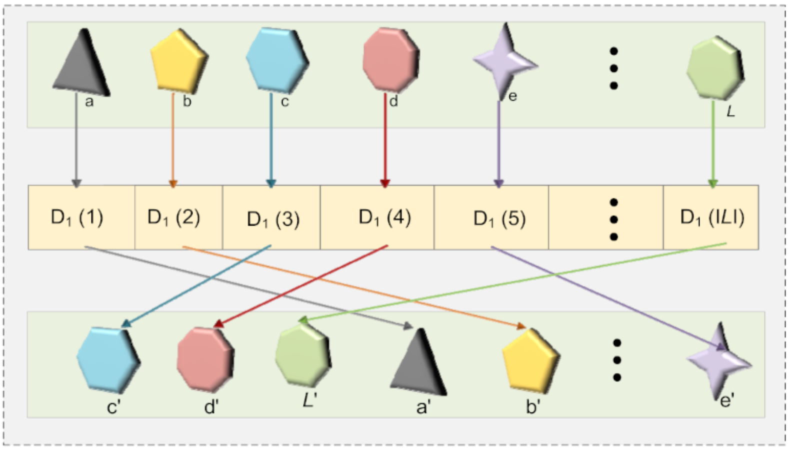


Figure 4. The traditional sequential scrambling process.

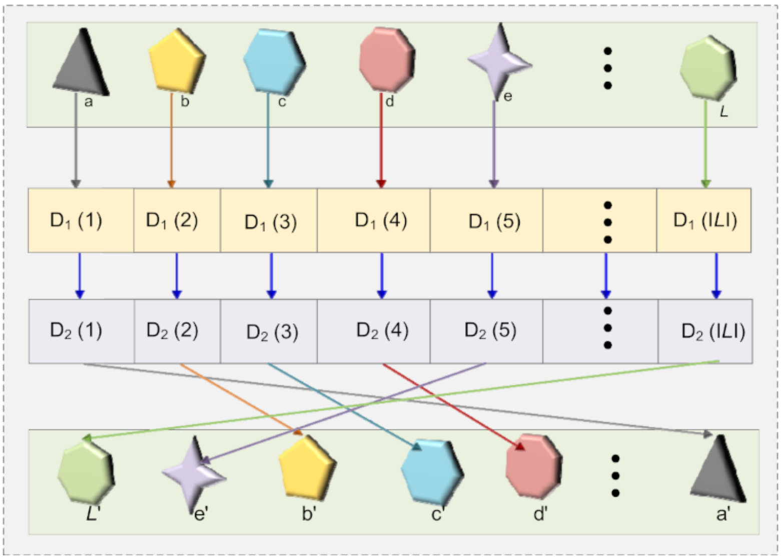


Figure 5. Double random position permutation.

2.4. Scrambling Encryption

Vector map objects with different lengths and irregular characteristics are regularized into a "one-dimensional matrix" form, the objects of "one-dimensional matrix" form are shuffled via DRPP, and then the objects of the "one-dimensional matrix" form are reconstructed into a vector map form, as shown in Figure 6. First of all, the object information of the original map is obtained, after which the object information is regularly processed. Furthermore, the regularized data are shuffled by use of the key sequence generated via the 4-D hyperchaotic system. Finally, the scrambled data are reconstructed to obtain a cipher map according to the original data format structure. The detailed steps are shown in Step1–Step5.

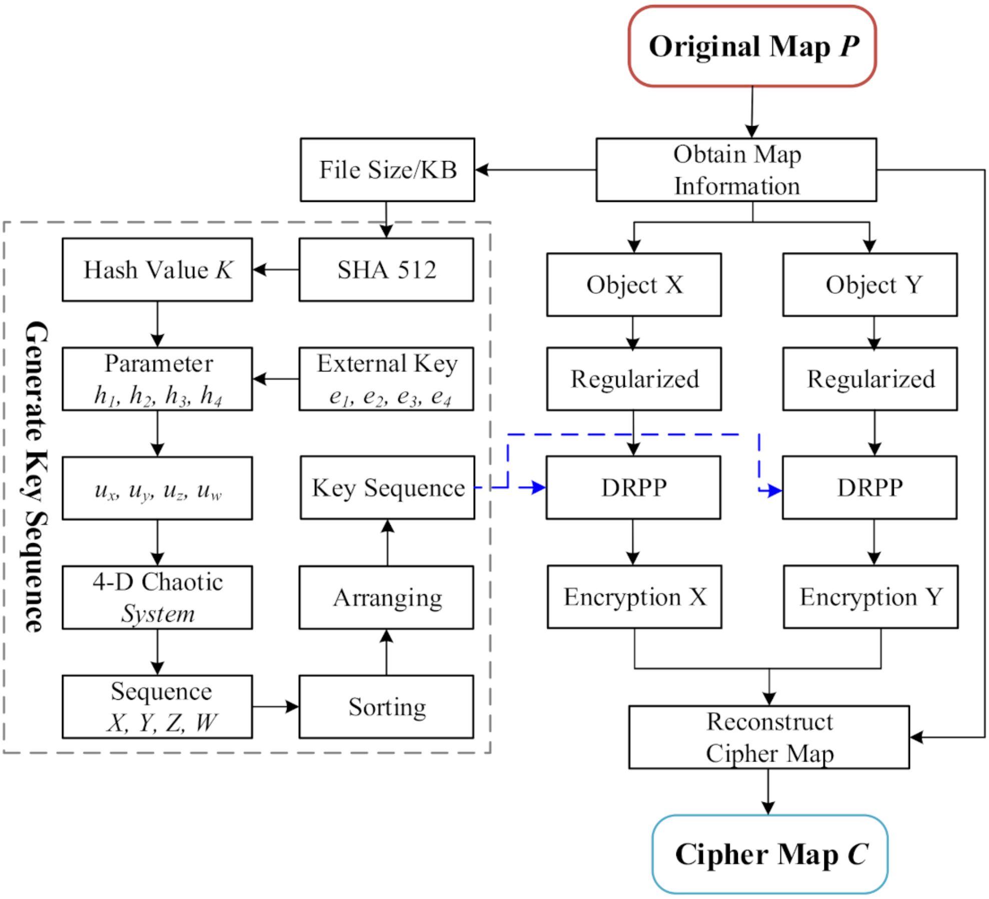


Figure 6. The diagram of the proposed encryption scheme.

Step1: As decrypted in Section 2.2, parameters u_x, u_y, u_z, u_w are obtained by utilizing 512-bit hash value H_k of the original map file and external key U_k. Consider the obtained u_x, u_y, u_z, u_w in Equation (3) as initial values, put them into the 4-D hyperchaotic system, and iterate t_0+$|L|$ ($t_0 \geq 1000$) times. To avoid negative influence, previous t_0 values are removed, and four chaotic sequences X, Y, Z, W sized of $1 \times |L|$ are obtained. Subsequently, four sequences $Sort\_X_1$, $Sort\_Y_1$, $Sort\_Z_1$, $Sort\_W_1$ and the corresponding index sequences $Sort\_D_X$, $Sort\_D_Y$, $Sort\_D_Z$, $Sort\_D_W$ are obtained according to their arrangement in ascending order of chaotic sequences X, Y, Z, W, and these processes can be expressed as

$$\begin{cases} [Sort\_X_1, Sort\_D_X] & = sort(X) \\ [Sort\_Y_1, Sort\_D_Y] & = sort(Y) \\ [Sort\_Z_1, Sort\_D_Z] & = sort(Z) \\ [Sort\_W_1, Sort\_D_W] & = sort(W) \end{cases} \quad (5)$$

where $|L|$ is the sum of the number for vertices of all objects.

Step2: To improve the correlation between the encryption scheme and the original map, four sequences $Sort\_X_1$, $Sort\_Y_1$, $Sort\_Z_1$, $Sort\_W_1$ are divided into six groups—namely: A1 = ($Sort\_D_X$, $Sort\_D_Y$), A2 = ($Sort\_D_X$, $Sort\_D_Z$), A3 = ($Sort\_D_X$, $Sort\_D_W$), A4 = ($Sort\_D_Y$, $Sort\_D_Z$), A5 = ($Sort\_D_Y$, $Sort\_D_W$), A6 = ($Sort\_D_Z$, $Sort\_D_W$).

Step3: To begin with, three variables H_sum, H_{x_index}, and H_{y_index} need to be defined. The hash value H_k of the original map file is obtained according to the SHA-512 hash function, it converts hexadecimal character in the hash value into a decimal number, and then all the decimal numbers converted from the hexadecimal characters are added to gain the H_sum value in order to reduce the correlation between the x coordinate and the y

coordinate. Find the H_{x_index} according to Equation (5), and calculate the H_{y_index} according to Equation (6).

$$H_{x\_index} = mod(H\_sum, 6) + 1, H_{x\_index} \in [1, 6] \quad (6)$$

$$H_{y\_index} = fllor(mod(\frac{p_1 + p_2 + p_3 + p_4}{4} \times 10^6, 6)) + 1, H_{y\_index} \in [1, 6] \quad (7)$$

Step4: According to H_{x_index} and H_{y_index}, pick out one group index sequence from Step 2. If H_{x_index} (or H_{y_index}) = i, group Ai is chosen.

Step5: DRPP is operated on coordinates according to group Ai, and the obtained confusing sequences are denoted as $S\_x_{i,j}$ and $S\_y_{i,j}$. One example is given to show this process. If group A1 and group A2 are selected, the detailed encryption operation is as

$$C\_x_{i,j} = x_{i,j}(D_X(i)), S\_x_{i,j}(D_Y(i)) = C\_x_{i,j} \quad (8)$$

$$C\_y_{i,j} = y_{i,j}(D_X(i)), S\_y_{i,j}(D_Z(i)) = C\_y_{i,j} \quad (9)$$

For the x-coordinate, using the index sequence of group A1, the index sequence $D_X(i)$ is utilized to select the coordinate to be shuffling from $x_{i,j}$, storing it into $C\_x_{i,j}$. Then, the index sequence $D_Y(i)$ is used to map $C\_x_{i,j}$ into a random position of $S\_x_{i,j}$, and the DRPP of $x_{i,j}$ is achieved; for the y-coordinate, using the index sequence of group A2, the index sequence $D_X(i)$ is utilized to select the coordinate to be shuffling from $y_{i,j}$, storing it into $C\_y_{i,j}$. Then, the index sequence $D_z(i)$ is used to map $C\_y_{i,j}$ into a random position of $S\_y_{i,j}$, and the DRPP of $y_{i,j}$ is finished. Therein, $i \in [1, |L|]$.

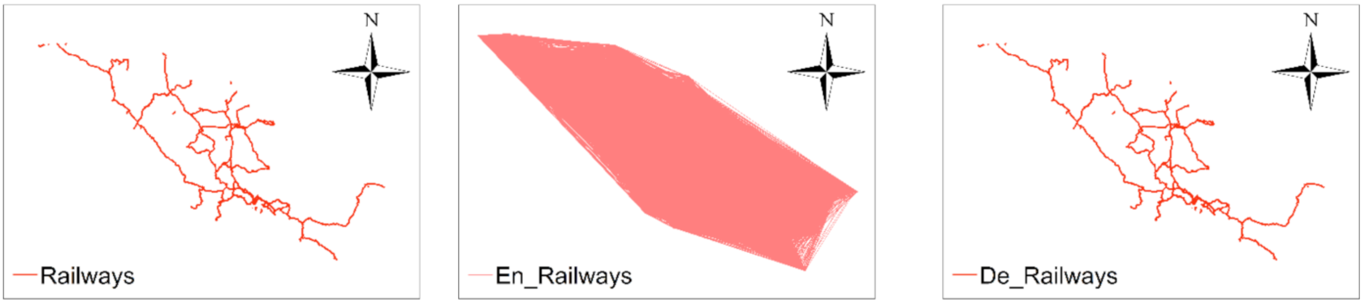
2.5. Decryption Processing

The decryption process could be obtained by manipulating the encryption process inversely, as shown in Figure 6. First, the key sequences must be generated over a key generator before decrypting the encrypted map. Then, the map objects obtained from the cipher map are a regularized "one-dimensional matrix", after which they are restored according to DRPP. Lastly, according to the arrangement and organization of vector map, the objects in the form of a "one-dimensional matrix" are reorganized into the form of a vector map, the decrypted map is obtained, and then decryption is achieved.

3. Experimental Results and Performance Analysis

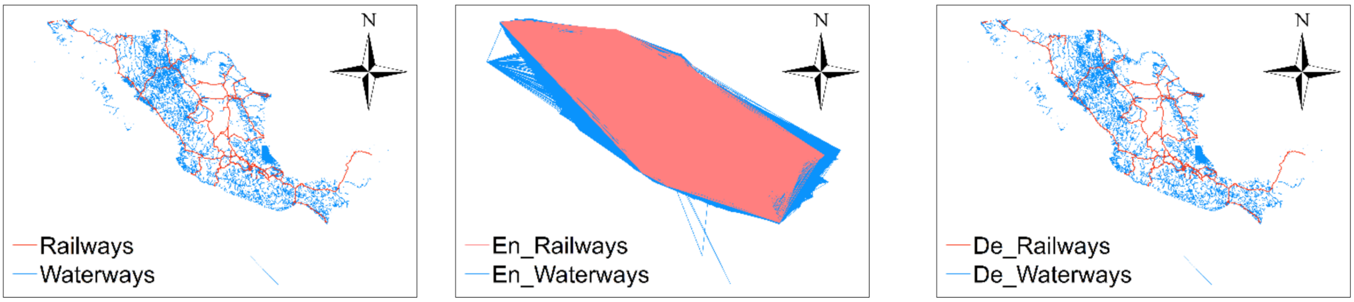
3.1. Encryption and Decryption Visualization

This algorithm was implemented using the Python language, and the experiments were achieved on a PC with Intel® Core™ i7-10750H CPU @ 2.60 GHZ, 16.00GB of RAM, and Windows 10 64-bit. The experimental results are shown in Figures 7–9. Figure 7a is the original map with only one layer, Figure 7b is the cipher map with only one layer, and Figure 7c is the decrypted map with only one layer; Figure 8a–c is the original map, the cipher map, and the decrypted map with two layers, respectively; and Figure 9a–c is the original map, the cipher map, and the decrypted map with three layers, respectively. A polygon is a set of connected polylines used to represent objects such as lakes, boundaries, and buildings. Thus, the original district data in polygon format can be converted to polyline format for encryption. After decryption, the decrypted district data in polyline format are converted to polygon format for restoration.



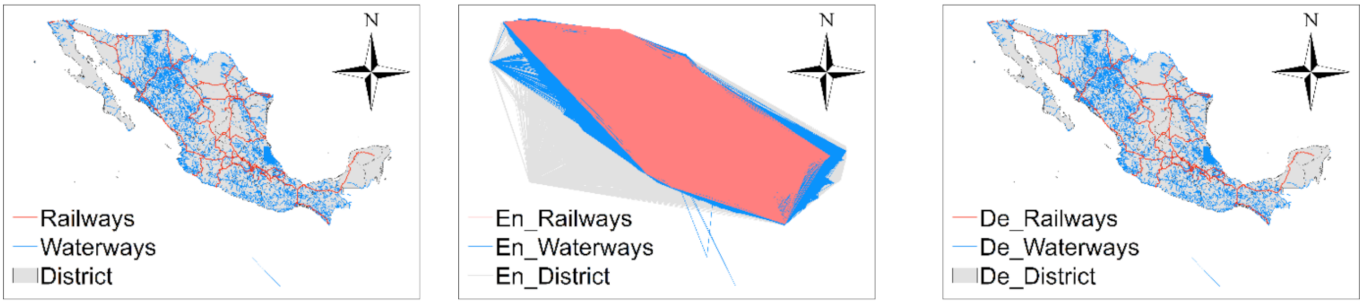
(**a**) (**b**) (**c**)

Figure 7. The map with one layer (railways). (**a**) The original map; (**b**) the cipher map; (**c**) the decrypted map.



(**a**) (**b**) (**c**)

Figure 8. The map with two layers (railways and waterways). (**a**) The original map; (**b**) tThe cipher map; (**c**) the decrypted map.



(**a**) (**b**) (**c**)

Figure 9. The map with three layers (railways, waterways, and district). (**a**) The original map; (**b**) the cipher map; (**c**) the decrypted map.

Science a vector map contains many layers, each layer contains many objects (polylines and polygons), and each object is composed of a large number of vertices. The scrambling effect is shown in Figures 7b, 8b and 9b, where the cipher map is completely shuffled and distorted. Human vision cannot distinguish the cipher map, and quantitative values are required for evaluation. Therefore, the correlation between adjacent coordinates is applied for quantitative evaluation in Section 3.2.

3.2. Correlation of the Adjacent Coordinates

A vector map is a kind of graphic data organized one by one according to objects (polylines and polygons), and the vertices in each object have obvious location order. Thus, it is necessary to analyze the correlation between the original map, the cipher map, and the decrypted map. The calculation of the correlation between adjacent coordinates is shown in Equation (9).

$$\left.\begin{array}{l} E(x) = \frac{1}{N}\sum\limits_{i=1}^{N} x_i \\ D(x) = \frac{1}{N}\sum\limits_{i=1}^{N} (x_i - E(x))^2 \\ \mathrm{cov}(x,y) = \frac{1}{N}\sum\limits_{i=1}^{N} ((x_i - E(x))(y_i - E(y)) \\ r_{xy} = \frac{\mathrm{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \end{array}\right\} \quad (10)$$

The correlation statistics of the original map, cipher map, and decrypted map are shown in Table 1. It can be found that the correlation of the original map is high, while the correlation of the cipher map is close to zero, and the correlation coefficient of the decrypted map is the same as the original map. This shows that the algorithm in this paper can disrupt the correlation between all objects, and that the decrypted map is the same as the original map.

Table 1. Correlation of the original map, the cipher map, and the decrypted map.

Data	Correlation					
	Original Map		Cipher Map		Decrypted Map	
	X-Coordinates	*Y*-Coordinates	*X*-Coordinates	*Y*-Coordinates	*X*-Coordinates	*Y*-Coordinates
Railways	0.7486	0.7572	−0.0013	−0.0008	0.7486	0.7572
Waterways	0.8384	0.8498	0.0003	0.0004	0.8384	0.8498
District	0.9660	0.9618	−0.0065	−0.0009	0.9660	0.9618

3.3. Key Space

For a perfect vector map encryption algorithm, the key space should be as large as possible to resist all kinds of violent attacks. In this paper, the keys include the following: (1) 512-bit hash value H_k of the map file; (2) the initial keys e_1, e_2, e_3, e_4; and (3) the initial values and parameters of 4-D hyperchaotic system (it is mainly generated by calculating the hash value and the given initial keys e_1, e_2, e_3, e_4). Supposing that the computation accuracy of the computer is 10^{-14}, the size of key space will be much greater than $10^{56} > 2^{168}$, which is larger than 2^{100}. If the 512-bit hash value H_k and other parameters are considered, the key space may be even larger to resist any brute force attack.

3.4. Key Sensitivity

A security vector map encryption algorithm must be sensitive to the key. To guarantee the security of the encryption scheme, the key sensitivity has to be analyzed. The key sensitivity refers to a minor change in the key that will lead to a completely different decryption. Of course, when the key sensitivity is higher, the security of the encryption algorithm is better. To test the key sensitivity, the correct keys were utilized to encrypt the original map, and then the cipher map attempted to decrypt using the slightly modified key; the results are illustrated in Figure 10.

The original key is (10, 28, 8/3, 1, 16, u_x, u_y, u_z, u_w), and the changed key is (10 + 10^{-14}, 28, 8/3, 1, 16, u_x, u_y, u_z, u_w). The original maps are shown in Figure 10a,e, the corresponding maps are shown in Figure 10b,f. The decrypted maps for the modified key are shown in Figure 10c,g, while the decrypted maps for the correct key are shown in Figure 10d,h.

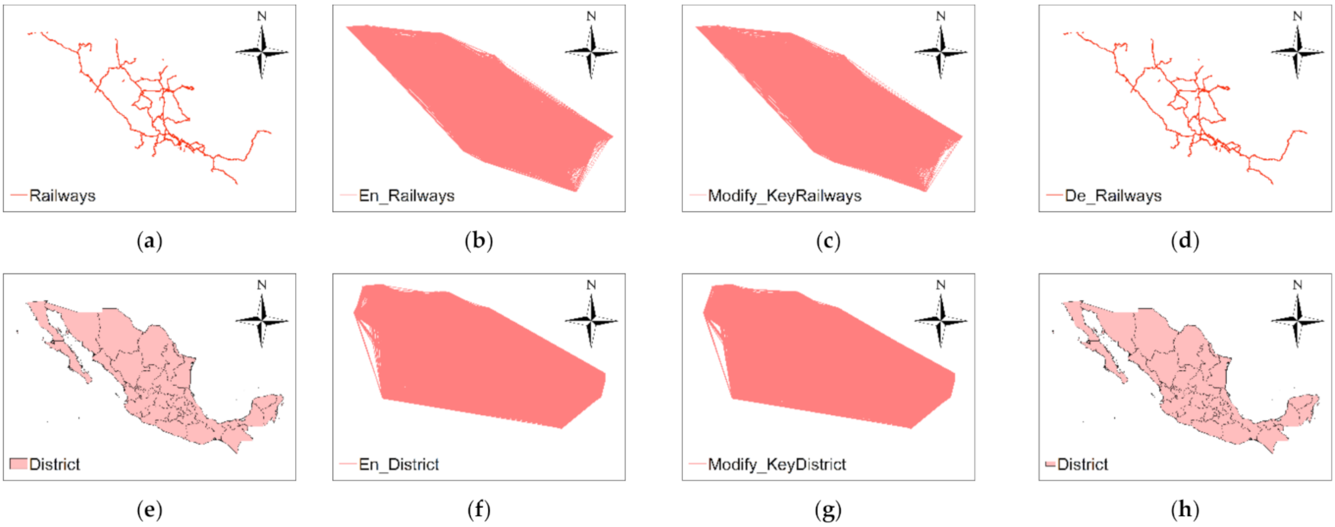


Figure 10. Key sensitivity. (**a**) The original "Railways" data; (**b**) cipher map using original key; (**c**) decrypted map using incorrect key; (**d**) decrypted map using correct key; (**e**) the original "District" data; (**f**) cipher map using original key; (**g**) Decrypted map using incorrect key; (**h**) Decrypted map using correct key.

3.5. Comparison of Different Scrambling Times

To test whether the scrambling algorithm needed only one scrambling to achieve a better scrambling effect, the "Railways" data were used, different scrambling times to test the proposed algorithm were set, and the results are shown in Figure 11 and Table 2.

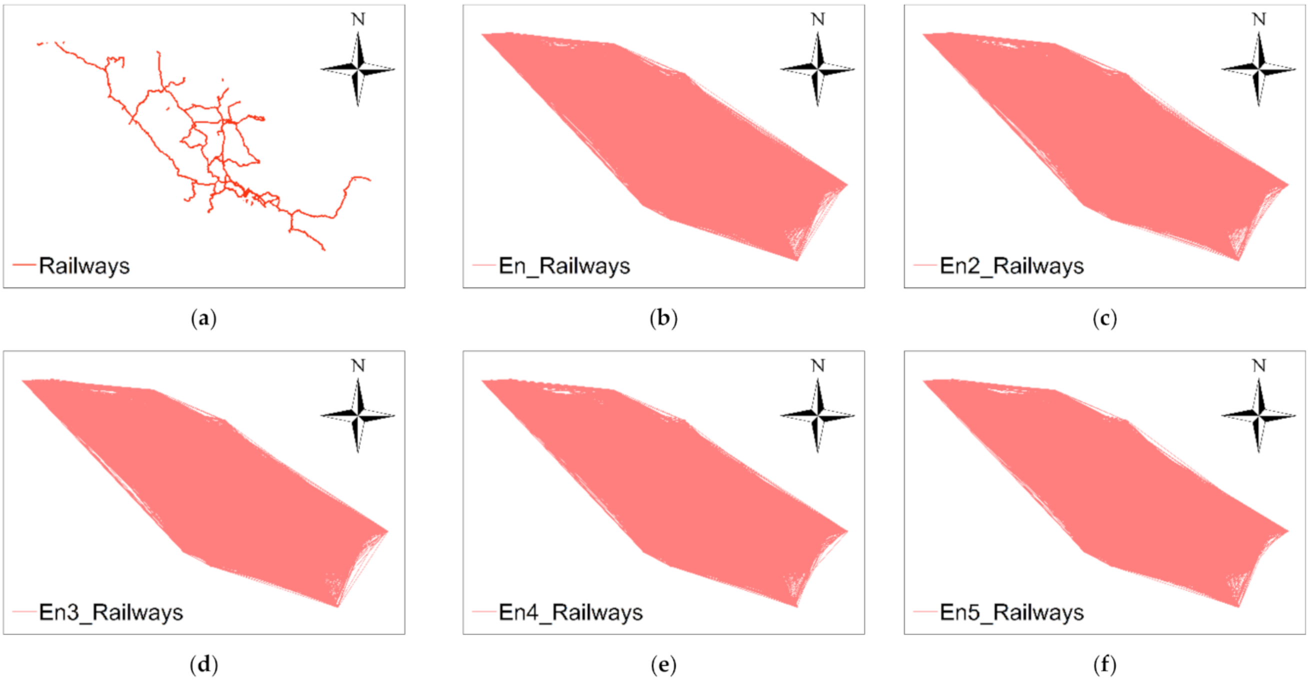


Figure 11. Comparison of different scrambling times. (**a**) The scrambling time is 0; (**b**) the scrambling time is 1; (**c**) the scrambling time is 2; (**d**) the scrambling time is 3; (**e**) the scrambling time is 4; (**f**) the scrambling time is 5.

Table 2. The correlation coefficient of different scrambling times.

Times	**1**		**2**		**3**		**4**		**5**	
Data	**X**	**Y**	**X**	**Y**	**X**	**Y**	**X**	**Y**	**X**	**Y**
Railways	−0.0013	−0.0008	−0.0015	0.0016	0.0070	0.0054	0.0047	−0.0013	−0.0016	−0.0036
Waterways	0.0003	0.0004	−0.0001	0.0016	0.0008	0.0019	−0.0001	−0.0008	0.0014	0.0028
District	−0.0065	−0.0009	0.0136	0.1128	−0.0036	−0.0052	−0.0018	−0.0053	−0.0028	0.0023

Figure 11 shows that the vector map was scrambled once, and the effect reached a good state. As the scrambling times increased, the scrambling effect change was not obvious. It can be seen in Table 2 that the correlation coefficients of scrambling one time and multiple scrambling are close to zero. Hence, the vector map only needed to be scrambled one time, which would achieve the goal of scrambling encryption, and computational costs would be saved.

3.6. Comparison with Existing Studies

To evaluate the effectiveness of the proposed encryption algorithm, Table 3 lists the results of comparison with other algorithms. As shown in Table 3, (1) compared with References [40,46,47], the proposed algorithm has the ability to resist statistical attacks; (2) compared with References [40,43,47], the scrambling of global objects was completely achieved; (3) compared with References [9,42,43,47], the proposed algorithm used double random position permutation, which completely avoided the one-to-one mapping relationship and improved security; (4) compared with References [9,40,42], the proposed algorithm uses a 4-D hyperchaotic system to generate the key sequence, which makes up for the shortcomings of low-dimensional chaotic systems limited by computer word-length.

Table 3. Comparison with existing algorithms.

Algorithms	**This Algorithm**	**Ref [9].**	**Ref [40].**	**Ref [42].**	**Ref [43].**	**Ref [46].**	**Ref [47].**
One-to-one mapping	×	√	√	√	√	/	√
Compensation for computer word-length limitation	√	×	×	×	/	/	/
Resistance to statistical attacks	√	√	×	√	√	×	×

4. Conclusions

This paper proposed an encryption algorithm for encrypting vector maps. The algorithm is based on a double random position permutation strategy to improve the security on vector map scrambling encryption. It utilizes the SHA-512 hash function and a 4-D hyperchaotic system to generate the key sequences. The key sequences are used for encrypting vector maps based on a double random position permutation strategy. Vector map objects first are processed in “one-dimensional matrix” form. Then the “one-dimensional matrix” is encrypted by the mapping relationship of two different key sequences. Finally, the encrypted map objects are reorganized according to the vector map structure to get the cipher map. The algorithm can increase the security of scrambling encryption compared with existing scrambling encryption algorithms.

The contributions of this paper are as follows: (1) one-to-one mapping during vector map scrambling is completely avoided; (2) it is difficult for attackers to obtain the permutation key value by analyzing the pairs of the plain map and cipher map; (3) compared with some existing algorithms, the proposed algorithm uses a 4-D hyperchaotic system to generate the key sequence, which makes up for the shortcomings of low-dimensional chaotic systems that are limited by computer word-length; and (4) this algorithm has the ability to resist statistical attacks. In sum, one-to-one mapping during vector map scrambling is completely avoided, and security of scrambling encryption has been improved. Meanwhile, the cipher map is completely shuffled and distorted, the algorithm in this paper can disrupt the correlation between all map objects, and the correlation of the decrypted map is the same as the original map. Moreover, a vector map only needs to be scrambled one time to achieve the goal of scrambling encryption, thus saving computational costs. Moreover, the key space is large enough to resist any brute force attack. The key sensitivity is so high that a minor change in the key cannot decrypt the encrypted vector map.

Although the proposed algorithm can encrypt the vector map, for point objects, the encryption effect is weak. Consequently, encryption and decryption algorithms for point objects will be our work in near future, as they are of interest not only to cartographers but also to researchers in the field of information security. In addition, transforming the algorithm into software that is available for public use is also a project that the authors are working on.

Author Contributions: Xiaolong Wang conceived and designed the experiments; Xiaolong Wang and Haowen Yan carried out the method; Xiaolong Wang performed the analysis and wrote the paper; Haowen Yan and Liming Zhang revised and edited the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was jointly funded by the National Natural Science Foundation of China (grant no. 41761080), the Industrial Support and Guidance Project of Universities in Gansu Province (grant no. 2019C-04), the National Natural Science Foundation of China (grant no. 41930101), and LZJTU EP 201806.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Qiu, Y.; Duan, H. A novel multi-stage watermarking scheme of vector maps. *Multimed. Tools Appl.* **2021**, *80*, 877–897. [CrossRef]
2. Qiu, Y.; Gu, H.; Sun, J. High-payload reversible watermarking scheme of vector maps. *Multimed. Tools Appl.* **2018**, *77*, 6385–6403. [CrossRef]
3. Wang, Q.S.; Zhu, C.Q.; Fu, H.J. The digital watermarking algorithm for vector geographic data based on point positioning. *Acta Geod. Cartogr. Sin.* **2013**, *42*, 310–316.
4. Li, H.; Zhu, H.; Hua, W. Key Technologies and Methods for Vector Geographic Data Security Protection. *Earth Sci.* **2020**, *45*, 4574–4588.
5. Zhu, C. Research Progresses in Digital Watermarking and Encryption Control for Geographical Data. *Acta Geod. Cartogr. Sin.* **2017**, *46*, 1609–1619.
6. Shekhar, S.; Huang, Y.; Djugash, J. Vector map compression: A clustering approach. In Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems, McLean, VA, USA, 8–9 November 2002; pp. 74–80.
7. Zhou, C. Prospects on pan-spatial information system. *Prog. Geogr.* **2015**, *34*, 129–131.
8. Wang, J. Development of geographic information system and developing geographic information system. *Eng. Sci.* **2009**, *11*, 10–16.
9. Li, A.; Wang, H.; Zhou, W. Scrambling encryption of vector digital map base on 2D chaos system. *J. China Univ. Min. Technol.* **2015**, *44*, 747–753.
10. Oladipo, J.O.; Aboyeji, O.S.; Akinwumiju, A.S.; Adelodun, A.A. Fuzzy logic interference for characterization of surface water potability in Ikare rural community, Nigeria. *J. Geovisualization Spat. Anal.* **2020**, *4*, 1. [CrossRef]
11. Biswas, K.; Chatterjee, A.; Chakraborty, J. Comparison of Air Pollutants Between Kolkata and Siliguri, India, and Its Relationship to Temperature Change. *J. Geovisualization Spat. Anal.* **2020**, *4*, 1–15. [CrossRef]
12. Xu, J.; Zhou, H.; Nie, G.; An, J. Plotting earthquake emergency maps based on audience theory. *Int. J. Disaster Risk Reduct.* **2020**, *47*, 101554. [CrossRef]
13. Du, M.; Zhang, X. Urban greening: A new paradox of economic or social sustainability? *Land Use Policy* **2020**, *92*, 104487. [CrossRef]
14. Guida, C.; Carpentieri, G. Quality of life in the urban environment and primary health services for the elderly during the Covid-19 pandemic: An application to the city of Milan (Italy). *Cities* **2021**, *110*, 103038. [CrossRef]
15. Yan, H.; Zhang, L.; Yang, W. A normalization-based watermarking scheme for 2D vector map data. *Earth Sci. Inform.* **2017**, *10*, 471–481. [CrossRef]
16. Yan, H.; Li, J.; Wen, H. A key points-based blind watermarking approach for vector geo-spatial data. *Comput. Environ. Urban Syst.* **2011**, *35*, 485–492. [CrossRef]
17. López, C. Watermarking of digital geospatial datasets: A review of technical, legal and copyright issues. *Int. J. Geogr. Inf. Syst.* **2002**, *16*, 589–607. [CrossRef]
18. Ren, N.; Wu, W.; Zhu, C.; Wang, D. An Accuracy Authentication Algorithm of Anti-Deleting Elements for Vector Geographic Data. *Geogr. Geo-Inf. Sci.* **2015**, *17*, 166–171.

19. Zhu, C.; Zhou, W.; Wu, W. *Research on the Policy and Law of China Geographic Information Security*; Science Press: Beijing, China, 2015; pp. 1–18, 65–71.
20. PRC NPC Web Site. Surveying and Mapping Law of the People's Republic of China [EB/OL]. Available online: http://www.npc.gov.cn/wxzl/gongbao/2000-12/05/content_5004576.htm (accessed on 10 January 2021).
21. Zhou, H.; Lv, Y. Research on Construction of Foreign Geographic Information Security Policies and Laws. *Bull. Surv. Mapp.* **2015**, 115–118. [CrossRef]
22. Zhou, Q.; Ren, N.; Zhu, C.; Zhu, A. Blind Digital Watermarking Algorithm against Projection Transformation for Vector Geographic Data. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 692. [CrossRef]
23. Vybornova, Y.D.; Sergeev, V.V. Method for protection of copyright on vector data. *Inform. Autom.* **2021**, *20*, 181–212.
24. Yang, C.; Zhu, C.; Wang, Y.; Rui, T.; Ding, K. A Robust Watermarking Algorithm for Vector Geographic Data Based on Qim and Matching Detection. *Multimed. Tools Appl.* **2020**, *79*, 30709–30733. [CrossRef]
25. Chen, J.; Zhang, L.; Jiang, M. A collusion-based vector spatial data fingerprinting scheme. *Sci. Surv. Mapp.* **2019**, *45*, 149–156. [CrossRef]
26. Lv, W.; Zhang, L.; Ma, L.; Chen, J. A digital fingerprinting algorithm for vector spatial data using BIBD. *Sci. Surv. Mapp.* **2017**, *42*, 134–139.
27. Chen, J.; Zhang, L.; Jiang, M. Digital fingerprint algorithm for vector spatial data using GD-PBIBD coding. *Bull. Surv. Mapp.* **2020**, 81–86+100. [CrossRef]
28. Sahoo, S.; Roshan, R.; Singh, V.; Halder, R. Bdmark: A blockchain-driven approach to big data watermarking. In *Asian Conference on Intelligent Information and Database Systems*; Springer: Singapore, 2020.
29. Sahoo, S.; Halder, R. Traceability and ownership claim of data on big data marketplace using blockchain technology. *J. Inf. Telecommun.* **2021**, *5*, 35–61. [CrossRef]
30. Sladić, G.; Milosavljević, B.; Nikolić, S. A Blockchain Solution for Securing Real Property Transactions: A Case Study for Serbia. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 35. [CrossRef]
31. Zheng, Z.; Xie, S.; Dai, H.N. Blockchain challenges and opportunities: A survey. *Int. J. Web Grid Serv.* **2018**, *14*, 352–375. [CrossRef]
32. Mao, J.; Zhu, C.; Zhang, X. A Fine-Granined access control model for vector geospatial data. *Geogr. Geo-Inf. Sci.* **2017**, *33*, 13–18.
33. Zhang, A.; Gao, J.; Ji, C. Multi-Granularity spatial -temporal access control model for Web GIS. *Trans. Nonferrous Met. Soc. China* **2014**, *24*, 2946–2953. [CrossRef]
34. Yu, G.; Li, R.; Lu, Z. Feature based spatial data access control model research. *Comput. Sci.* **2008**, *35*, 122–125+130.SS.
35. Standard, Data Encryption. *Federal Information Processing Standards Publication 46*; National Bureau of Standards: Gaithersburg, MA, USA; US Department of Commerce: Washington, DC, USA, 1997; Volume 23.
36. Daemen, J.; Rijmen, V. Reijndael: The Advanced Encryption Standard. *Dr. Dobb's J. Softw. Tools Prof. Program.* **2001**, *26*, 137–139.
37. Rivest, R.L.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [CrossRef]
38. Koblitz, N. Elliptic curve cryptosystems. *Math. Comput.* **1987**, *48*, 203–209. [CrossRef]
39. Zhang, S. *Research on the Algorithm of Encryption in Network Transmission of Vector Graphic Data*; Wuhan University: Wuhan, China, 2005.
40. Van, B.N.; Lee, S.H.; Kwon, K.R. Selective Encryption Algorithm Using Hybrid Transform for GIS Vector Map. *JIPS* **2017**, *13*, 68–82.
41. Zhao, Y.; Li, G.; Li, L. Electronic chart encryption method based on chaotic stream cipher. *J. Harbin Eng. Univ.* **2007**, *28*, 60–64.
42. Wang, H. *Scrambling Encryption Methods and Scrambling Performance Evaluation for Vector Geographic Data*; Nanjing Normal University: Nanjing, China, 2014.
43. Pham, G.N.; Ngo, S.T.; Bui, A.N. Vector Map Random Encryption Algorithm Based on Multi-Scale Simplification and Gaussian Distribution. *Appl. Sci.* **2019**, *9*, 4889. [CrossRef]
44. Bang, N.V.; Lee, S.H.; Moon, K.S. Encryption Algorithm using Polyline Simplification for GIS Vector Map. *J. Korea Multimed. Soc.* **2016**, *19*, 1453–1459. [CrossRef]
45. Giao, P.N.; Kwon, O.J.; Lee, S.H. Perceptual encryption method for vector map based on geometric transformations. In Proceedings of the 2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), Jeju, Korea, 13–16 December 2016.
46. Pham, N.G.; Lee, S.H.; Kwon, K.R. Perceptual Encryption Based on Features of Interpolating Curve for Vector Map. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2017**, *100*, 1156–1164. [CrossRef]
47. Jang, B.J.; Lee, S.H.; Lee, E.J. A crypto-marking method for secure vector map. *Multimed. Tools Appl.* **2017**, *76*, 16011–16044. [CrossRef]
48. Shannon, C.E. Communication theory of secrecy systems. *Bell Syst. Tech. J.* **1949**, *28*, 656–715. [CrossRef]
49. Wang, X.; Wang, X.; Zhao, J. Chaotic encryption algorithm based on alternant of stream cipher and block cipher. *Nonlinear Dyn.* **2011**, *63*, 587–597. [CrossRef]
50. Zarei, A.; Tavakoli, S. Hopf bifurcation analysis and ultimate bound estimation of a new 4-D quadratic autonomous hyper-chaotic system. *Appl. Math. Comput.* **2016**, *291*, 323–339. [CrossRef]
51. Chai, X.; Bi, J.; Gan, Z. Color image compression and encryption scheme based on compressive sensing and double random encryption strategy. *Signal Process.* **2020**, *176*, 107684. [CrossRef]