International Journal of Geo-Information

MDPI

Article

A Novel Invariant Based Commutative Encryption and Watermarking Algorithm for Vector Maps

Yu Li [1,2,3], Liming Zhang [1,2,3,*], Xiaolong Wang [1,2,3], Xingang Zhang [4] and Qihang Zhang [1,2,3]

1 Faculty of Geomatics, Lanzhou Jiaotong University, Lanzhou 730070, China; 0219754@stu.lzjtu.edu.cn (Y.L.); 0219777@stu.lzjtu.edu.cn (X.W.); 12201870@stu.lzjtu.edu.cn (Q.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
4 School of Geography and Ocean Science, Nanjing University, Nanjing 210008, China; DG21270060@smail.nju.edu.cn
* Correspondence: zlm@lzjtu.edu.cn; Tel.: +86-1363-932-8110

Abstract: Commutative encryption and watermarking (CEW) is an emerging method that combines encryption technology with digital watermarking technology. It has the dual capability of secure transmission and copyright protection. However, the existing CEW methods for vector maps have good robustness in resisting geometric attacks but poor resistance to vertex attacks (e.g., addition, deletion, etc.). To solve this problem, here we propose a novel invariant-based CEW algorithm for vector maps, which consists of permutation-based encryption scheme and coordinates-based watermarking scheme. In the encryption scheme, the encryption key is generated via the Gaussian distribution method combined with the SHA-512 hash method; then, the double random position permutation strategy is applied to the vector map encryption. In watermarking embedding scheme, the original watermark image is scrambled via logistic chaotic encryption before embedding, and the coordinates of all the vertices are normalized. Then, the scrambled watermark image is embedded into the normalized coordinates. Results show that: proposed method is more robust to conventional attacks (e.g., vertex addition and deletion, reordering and data format conversion) and geometric attacks (e.g., scaling and translation). In addition, compared with the existing CEW methods for vector maps, the proposed method has higher security and stronger robustness against vertex attacks.

Keywords: commutative encryption and watermarking; vector map; secure transmission; copyright protection

Citation: Li, Y.; Zhang, L.; Wang, X.; Zhang, X.; Zhang, Q. A Novel Invariant Based Commutative Encryption and Watermarking Algorithm for Vector Maps. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 718. https://doi.org/10.3390/ijgi10110718

Academic Editors: Giedrė Beconytė and Wolfgang Kainz

Received: 24 August 2021
Accepted: 22 October 2021
Published: 25 October 2021

1. Introduction

Vector maps are one of the most important geospatial data [1,2], which play a vital role in economic development and national security [3,4]. Currently, vector maps have been widely used in navigation, cadastral management, urban planning and many other fields [5]. However, the development of information-sharing technology makes it easier to be leaked and illegally copied in the process of storage, transmission and application. To address the growing urgent issues of data security, a series of laws, rules and regulations have been issued [6]. For example, the "Surveying and Mapping Law of the People's Republic of China" was revised in 2017 to strengthen the protection of geospatial data (e.g., vector maps) [7]. Correspondingly, the security protection algorithm for vector maps needs to be developed.

Many technologies have been developed to ensure the secure transmission and copyright security of vector maps (e.g., encryption technology and digital watermarking technology). Encryption is an effective method to protect data from being illegal intercepted and tampered with, and it can provide secure protection for vector maps in the process of storage and transmission [8]. For example, Wang et al. [9] encrypted vector maps using the

double random position permutation method combined with a four-dimensional quadratic autonomous hyperchaotic system. This method can achieve the secure transmission of vector maps but cannot provide copyright protection. Digital watermarking can protect the data copyright via embedding a watermark with copyright information into the data [10]. In [11], the watermark information is embedded into the directions of polyline objects to protect the copyright of vector maps. However, this method takes no account of the secure transmission of vector maps which is not like encryption technology. Therefore, it is necessary to combine encryption technology and digital watermarking technology for the secure transmission and copyright security of vector maps [12,13].

Existing research has found that commutative encryption and watermarking (CEW) is a viable method to combine encryption technology and watermarking technology [14,15]. The CEW utilizes the respective advantages of encryption technology and digital watermarking technology to provide multiple security protection in various application scenarios [16]. For a CEW algorithm, the key is how to avoid the mutual interference between encryption and watermarking, and to achieve commutativity [17]. Some CEW algorithms have been proposed to achieve the secure transmission and copyright protection of multimedia data (e.g., image data, video data and audio data). These algorithms are mainly divided into three categories: partial encryption based CEW, homomorphic encryption based CEW and invariant based CEW [18].

(1) Partial encryption based CEW refers to the method that the data is divided into two different datasets to manipulate the encryption and watermarking respectively. Cancellaro [19] used the tree structured Harr transform to divide the image into two parts. Then, the watermark was embedded into the low-level coefficients and the high-level coefficients were encrypted simultaneously. Jiang [20] divided the protected image into an encryption domain and a watermarking domain by orthogonal transformation, which ensures the mutual independence between encryption and watermarking. However, the disadvantage of such schemes is that it does not effectively balance encryption security and watermark robustness.

(2) Homomorphic encryption based CEW refers to the method that the commutativity is accomplished by taking advantage of homomorphism. For example, Li et al. [21] designed a CEW model by the use of the additive homomorphism of Paillier algorithm, which can encrypt the overall image. However, such schemes can only support the simple homomorphic watermarking operation, which have a higher security but a low watermark robustness.

(3) Invariant based CEW fulfills the commutativity by embedding the watermark information into a certain feature space that has no impact on encryption operation. Such methods usually use scrambling encryption to encrypt the data, and the watermark information is embedded via shifting histogram [22] or modifying the pixel value [23]. This type of method not only encrypts the overall data, but also balances the security of encryption and the robustness of watermarking. However, the existing methods are applicable to raster data but cannot be applied to vector maps due to the differences in data representation and storage structure. As mentioned above, it is obvious that the invariant based CEW is superior to the other two schemes in terms of balancing encryption security and watermark robustness. Based on this, Ren et al. [24] analyzed the organization structure and storage structure of vector maps and proposed a CEW method by deriving two invariants: the sum of inner angles and the storage direction of two adjacent objects. This method has high robustness against geometric attacks. However, the watermark embedding positions can be affected by the number of vertices of each object, so this method has little resistance to the vertex addition and deletion.

To sum up, the above CEW methods still have the following shortcomings: (1) few algorithms can be applied directly to vector maps, and (2) few algorithms for vector maps are resistant to vertex attacks. Hence, this paper proposes a novel invariant based CEW method for vector maps to solve the above problems. To construct a CEW scheme for vector maps, the organization structure and storage structure of the vector maps are

taken fully into account. In addition, in light of few algorithms for vector maps that are resistant to vertex attacks, we build on the permutation encryption, and combine it with the watermarking scheme to achieve better robustness to vertex attacks.

The remainder of this paper is organized as follows. Section 2 details vector maps data, permutation encryption of vector maps, the principle of the logistic chaotic map and the normalization process of vector maps. Section 3 introduced the basic idea and the process of the algorithm proposed in this paper. Section 4 verifies the effectiveness and versatility of proposed algorithm by implementing several experiments. Finally, Section 5 concludes the paper and gives directions for further work.

2. Related Works

2.1. Vector Map Data

A vector map consists of a number of data layers, and each layer consists of attribute data and geometry data [25,26], as shown in Figure 1a. The attribute data presents the storage information such as ID, Text and Name. The geometry data (e.g., points, polylines and polygons) is used to determine the geolocation information of vector maps, as shown in Figure 1b. Points are the most fundamental element of geometry data, polylines are made up of a series of vertices, and polygons are composed of closed polylines. Points are usually used to describe simple objects like the position of parking lot, while polylines and polygons represent the complex objects, such as roads, rivers, buildings and lakes.

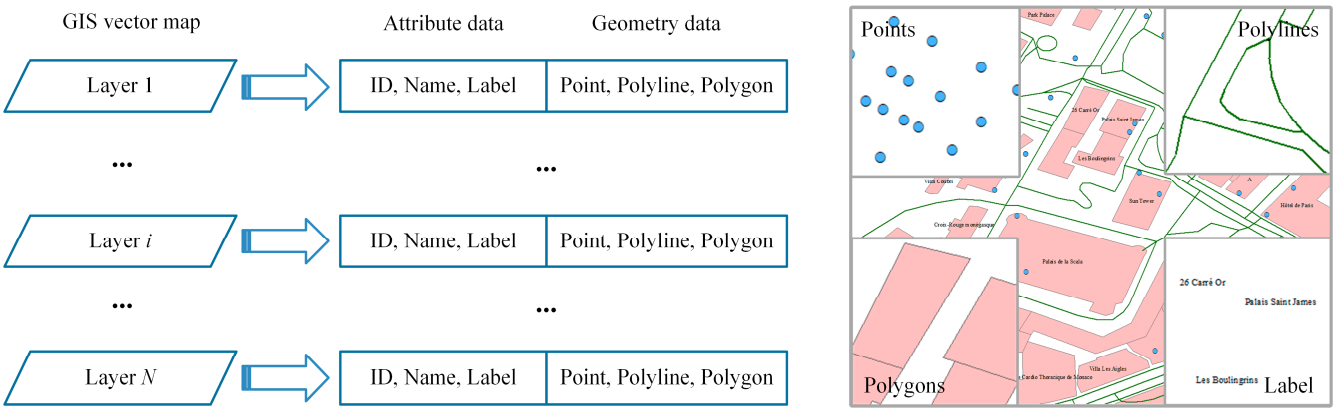

Figure 1. The model and components of the GIS vector map. (**a**) Model of the GIS vector map, (**b**) data components of the GIS vector map.

The organizational form of vector map is represented by Formula (1), (2) and (3), and the number of vertices is represented by N_V.

$$L = \{P_i | i \in [1, |L|]\} \tag{1}$$

$$P_i = \{v_{i,j} | j \in [1, |P_i|]\} \tag{2}$$

$$v_{i,j} = (x_{i,j}, y_{i,j}) \tag{3}$$

where L represents the data layers of the vector map, $|L|$ is the number of layers, P_i denotes the ith object of vector map, $|P_i|$ is the number of vertices on the ith object, $x_{i,j}$ and $y_{i,j}$ are the x-coordinate and y-coordinate of each vertex, respectively.

2.2. Double Random Position Permutation Encryption of Vector Maps

Permutation encryption is an encryption method that breaks the original spatial order and the correlation between adjacent objects [8,27]. Permutation encryption of vector maps refers to adjusting the order of vertices according to certain rules to achieve the scrambling effect. Due to the scrambling of point objects can only reorder the spatial data storage but

has no effect on the data structure, and thus polyline objects and polygon objects are the main permutation targets in vector maps. There are two methods to scramble the vertices: permutation within objects (Figure 2b) and permutation between objects (Figure 2c). It can be found from Figure 2b that permutation within objects achieves only local permutations within the same object though the organization structure of each object is disrupted. As can be seen from Figure 2c that the global permutations between objects is achieved, which increases the complexity of encryption results. Hence, the permutation between objects has a better scrambling effect.

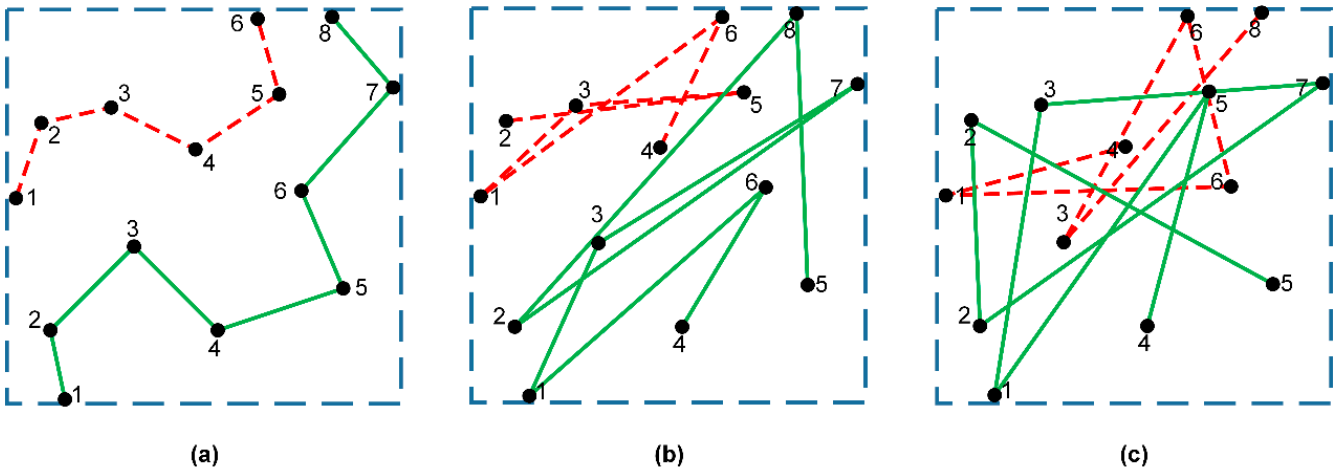

Figure 2. Scrambling of the GIS vector map data. (**a**) Original vector map data, (**b**) scrambling within objects, (**c**) scrambling between objects.

Double random position permutation (DRPP) is a viable permutation encryption method. Figure 3 shows the DRPP scrambling process of vector maps. To ensure the security of scrambling, two different key sequences are used to increase the difficulty of decryption. The key sequence K_1 is used to map the coordinates of vector maps at first, and then it is mapped to the other random location by the key sequence K_2.

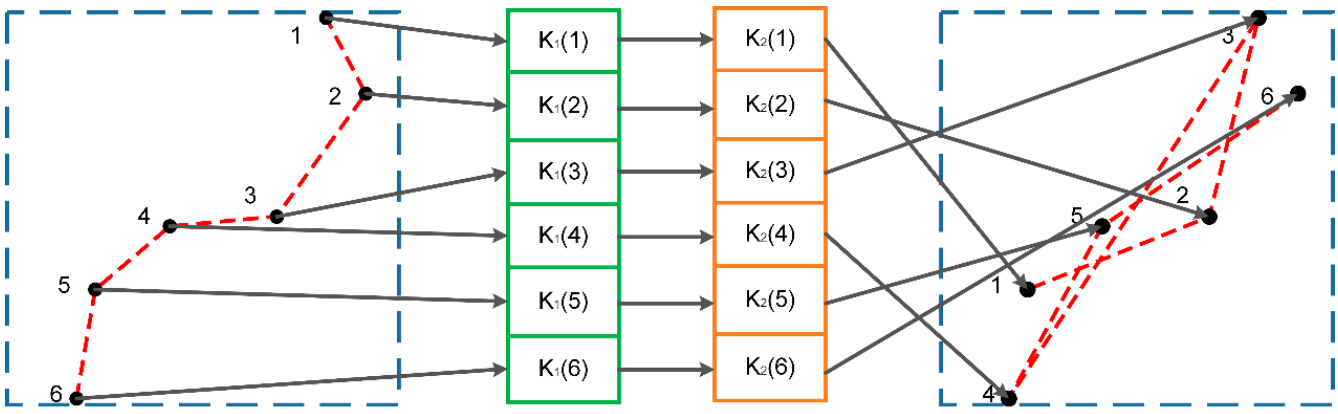

Figure 3. Double random position permutation.

2.3. Logistic Chaotic Map

A logistic chaotic map is simple and widely used in performing image encryption with less computation [28]. The definition of 1D logistic chaotic map is as follows:

$$x_{n+1} = \mu x_n(1 - x_n), n = 0, 1, 2, \quad (4)$$

To improve the security of encryption, a 2D logistic chaotic map [29] with a better chaotic property than 1D logistic chaotic map was proposed. It can be defined through the following equations:

$$\begin{cases} x_{n+1} = \mu\alpha_1 x_n(1-x_n) + \gamma y_n \\ y_{n+1} = \mu\alpha_2 y_n(1-y_n) + \gamma x_n \end{cases}, \ (x, y \in (0,1)) \tag{5}$$

when $\mu = 4$, $0.65 \leq \alpha_1, \alpha_2 \leq 0.9$ and $0 < \gamma < 1$ are set, the 2D logistic chaotic map reaches a chaotic state if $x, y \in (0,1)$.

2.4. Normalization of Vector Maps

The min-max normalization method [30] refers to linearly maps the original data to $(0,1)$. Assume that x_i is an original value, x_{min} and x_{max} are the minimum x and maximum x of original values, respectively, and Nx_i is the normalized value. The min-max normalization method is defined using Formula (6).

$$Nx_i = \frac{x_i - x_{min}}{x_{min} - x_{max}} \tag{6}$$

Correspondingly, the normalized values can be renormalized by Formula (7) to recover the original coordinates.

$$x_i = x_{min} + (x_{max} - x_{min}) \cdot Nx_i \tag{7}$$

3. Proposed CEW Method for Vector Maps

3.1. Basic Idea

The permutation-based encryption scheme has no effect on vertex coordinates, which means a permutation-based encryption scheme can be combined with a coordinates-based watermarking scheme to construct a CEW scheme. Therefore, the proposed algorithm consists of two parts, i.e., permutation-based encryption scheme and coordinates-based watermarking scheme. The detailed process of the proposed algorithm is shown in Figure 4.

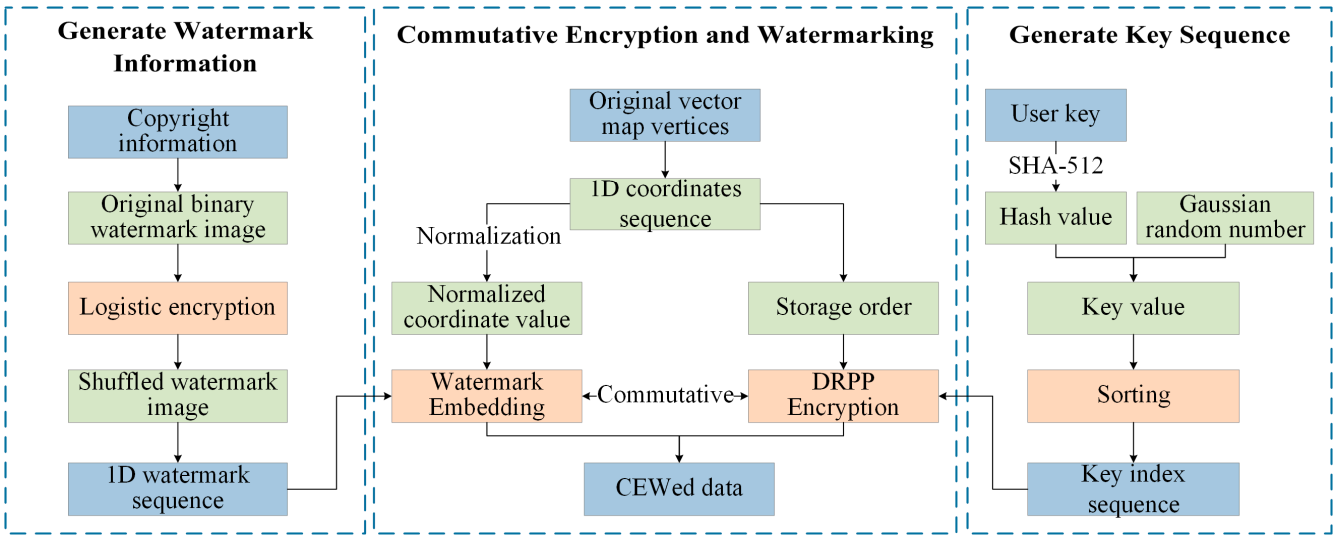

Figure 4. The diagram of the proposed CEW scheme.

3.2. Permutation-Based Encryption Scheme

In the encryption scheme, the SHA-512 hash method and Gaussian distribution are used to generate the encryption key, then all vertices are scrambled via DRPP. The detailed encryption process is as follows. It should be pointed out that the following content only gives the encryption method of the X-coordinate, and the operation process of the Y-coordinate is the same as that of the X-coordinate.

(1). Generation of the encryption key

Step 1: According to DRPP, two initial keys U_k and H_k are obtained by performing the SHA-512 hash method on the key input by users and the file information of the vector map respectively.

Step 2: To enhance the security of two initial keys, two sets of Gaussian random numbers Gu and Gh are calculated to scramble initial keys U_k and H_k. Then, two encryption key sequences K_{E_1} and K_{E_2} of vector maps can be obtained. The detailed operation is as follows:

$$Gu = \left\{ g_{u_{i,j}} \middle| 0 < i < |L|, 0 < j \le |P_i| \right\},\ Gh = \left\{ g_{h_{i,j}} \middle| 0 < i < |L|, 0 < j \le |P_i| \right\} \tag{8}$$

$$g_{u_{i,j}} = \frac{i \times j}{|P_i|} \times \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}},\ g_{h_{i,j}} = \frac{i \times j}{|P_i|} \times \frac{1}{\sqrt{2\pi}} e^{-\frac{h^2}{2}} \Bigg) \tag{9}$$

$$K_{E_1} = \frac{n_1 \times u}{Gu},\ K_{E_2} = \frac{n_2 \times h}{Gh} \tag{10}$$

where u and h are the values the initial keys U_k and H_k, respectively; n_1 and n_2 are the length of the initial keys U_k and H_k, respectively.

Step 3: Two index sequences $Sort\_K_{E_1}$ and $Sort\_K_{E_2}$ are obtained via sorting two encryption keys K_{E_1} and K_{E_2} in ascending order respectively.

(2). Encryption of vector maps

Step 1: Read the storage order of all vertices from the vector map and form a one-dimensional sequence.

Step 2: Reorder all vertices according to the key $Sort\_K_{E_1}$, and the reordered vertices are stored into Cx_i according to Formula (11).

$$Cx_i = x_i\left(Sort\_K_{E_1}(i)\right), i \in [1, N_V] \tag{11}$$

Step 3: According to Formula (12), Cx_i is mapped to a random position of Sx_i by using the other key $Sort\_K_{E_2}$, and the Sx_i is the encryption result of x_i.

$$Sx_i = Cx_i\left(Sort\_K_{E_2}(i)\right), i \in [1, N_V]) \tag{12}$$

Step 4: After all vertices are scrambled, the encrypted vector map is obtained.

(3). Decryption of vector maps

The decryption process is an inverse process of encryption. Two key sequences are generated based on the key input by users and the information of vector map file, and the encrypted x-coordinates are obtained from the encrypted vector maps. Then, the encrypted x-coordinate sequence and the key sequences are transformed into one-dimensional sequence. Finally, the decrypted vector maps are obtained based on the principle of DRPP.

3.3. Coordinates-Based Watermarking Scheme

The coordinates of the vector maps are not changed in the above permutation encryption process, which makes it possible to achieve the commutativity between the watermarking process and the encryption process in this paper. Thus, a coordinate-based watermarking scheme is proposed to achieve the commutativity in this section. Firstly, the 2D logistic chaotic map is used to shuffle the original watermark image, and all vertex coordinates are normalized; then the shuffled watermark is embedded into the normalized coordinates according to the mapping relation between each coordinate and each bit of the watermark. The detailed watermark embedding and extraction process are as follows.

(1). Generation of the watermarking information

Firstly, a binary watermark image with copyright is selected. Then, the watermark image is shuffled by using the 2D logistic chaotic encryption algorithm, aiming to reduce the correlation between pixels and enhancing the security of the watermark. Finally, the shuffled watermark is mapped into a one-dimensional sequence $W = \{w_i | w_i = 0, 1\}$, where $0 \leq i < N_W$, and N_W is the length of the one-dimensional watermark sequence.

(2). Watermark embedding

Step 1: Read the coordinates of all vertices to form a one-dimensional coordinate sequence, and then normalize all coordinates.

Step 2: To ensure each bit of the watermark can be embedded into at least one coordinate value, each vertex is taken as an embedding position, and a mapping relationship between each coordinate and each bit of the watermark is established by Formula (13).

$$index = (Nx_i \times 10^n) mod N_W, i \in [1, N_V] \tag{13}$$

where n denotes the magnification of normalized coordinate; $index$ is the watermark bit to be embedded.

Step 3: The shuffled watermark is embedded into the normalized coordinates Nx_i using the quantization method, and the watermarked coordinates Wx_i are calculated by Formula (14).

$$Wx_i = \begin{cases} Nx_i - \frac{R}{2}, \ if \ W(index) = 0 \text{ and } Nx_i mod \ R > R/2 \\ Nx_i + \frac{R}{2}, \ if \ W(index) = 1 \text{ and } Nx_i mod \ R \leq R/2 \\ Nx_i, \quad else \end{cases} \tag{14}$$

where $i \in [1, N_V]$, and R is the quantization step.

Step 4: The watermarked coordinates are renormalized to derive an integral and watermarked vector map.

In the above watermark embedding process, the watermark bit to be embedded is determined by the mapping relationship between the coordinates and the watermark bits. Therefore, every bit of the watermark can be multiply embedded into the normalized coordinates. In other words, each bit of the watermark can be embedded into at least one coordinate value. Based on this, the embedded watermark bit can still be determined by other coordinates even if some vertices are added or deleted. This ensures the embedded watermark is not easily destroyed when the watermarked maps are subject to vertex attacks.

(3). Watermark extraction

The process of watermark extraction is the inverse process of watermark embedding. The detailed steps are listed as follows.

Step 1: The watermarked coordinates are obtained and normalized. Moreover, the embedding position is identified by the relationship between the watermarked coordinates and the watermark bits.

Step 2: A one-dimensional sequence W' is defined with an initial value 0 and length N_W. The value of W' is determined by formula (15).

$$W'(index) = \begin{cases} W'(index) + 1, if \ Wx_i \bmod R > R/2 \\ W'(index) - 1, if \ Wx_i \bmod R \leq R/2 \end{cases} \tag{15}$$

where $i \in [1, N_V]$, $index$ and R have the same meaning as the watermark embedding process.

Step 3: In accordance with Formula (16), the watermark W' is rewritten using the majority rule.

$$W'(index) = \begin{cases} 1, \ if \ W'(index) > 0 \\ 0, \ if \ W'(index) \leq 0 \end{cases} \tag{16}$$

Step 4: The one-dimensional sequence W' is reconstructed as a two-dimensional watermark image based on the size of the original watermark image.

4. Experiments and Results

Several experiments were conducted to verify the effectiveness and versatility of the proposed algorithm. All experiments were implemented under Windows 10 64-bit PC with Intel(R) Core(TM) i5-10500, 3.10GHz CPU and 8GB RAM, running in the Python 3.7 platform. In the experiments, we downloaded the shapefile data of different sizes, regions and object types from OpenStreetMap (http://download.geofabrik.de/, accessed on 10 January 2021) as original GIS vector maps, and selected a binary image with 32×64 pixels as the watermark image. Figure 5a–c shows the original GIS vector maps, and the detailed information of these vector maps is listed in Table 1. Figure 6a,b shows the original watermark and the shuffled watermark. It is important to note that $\alpha_1 = 0.89$, $\alpha_2 = 0.9$, $\gamma = 1$ are used as initial values of the 2D logistic chaotic map in the process of the watermark scrambling.

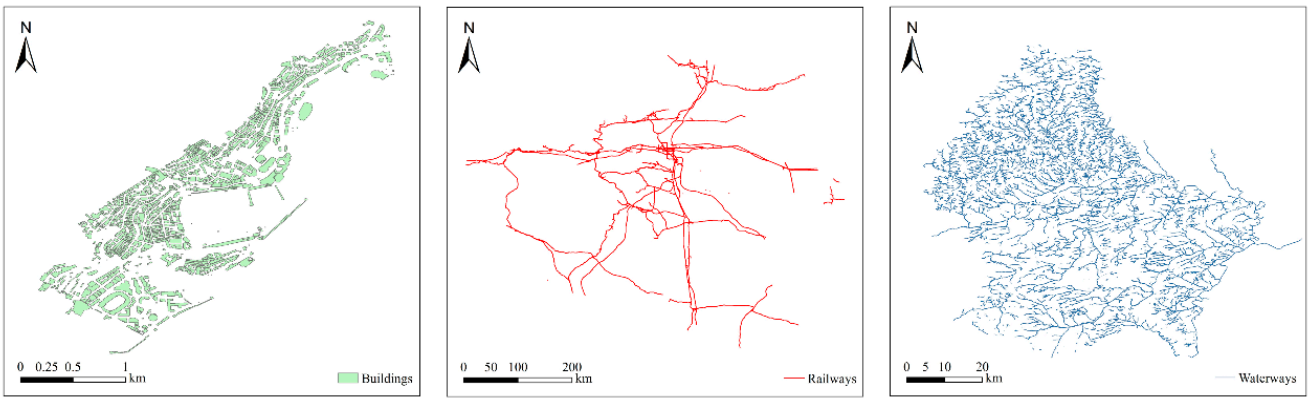

Figure 5. Original vector maps data: (**a**) buildings, (**b**) roads, (**c**) waterways.

Table 1. The detailed information of the GIS vector map data.

Map	Region	Object Type	Size (KB)	Vertices Number	Objects Number
Buildings	Monaco	Polygon	243	11029	1209
Railways	Henan, China	Polyline	1557	67123	9279
Waterways	Luxembourg	Polyline	3564	186920	11797

Figure 6. Watermark information: (**a**) original watermark, (**b**) shuffled watermark.

4.1. Visualization Experiments

The original vector maps are performed CEW operation using the proposed algorithm. The experimental results are shown in Figures 7 and 8. Figure 7 displays the commutative encryption-watermarked (CEWed) maps, Figure 7a–c display the encrypted-watermarked (E-Wed) maps obtained by performing encryption operation at first and watermarking operation afterwards. Figure 7d–f show the watermarked-encrypted (W-Eed) maps, which are obtained by performing watermarking operation at first and then encryption operation. Figure 8a–c are the decrypted-watermarked (D-Wed) maps obtained by decrypting the CEWed maps, respectively. As shown in Figures 7 and 8, the information of the original

maps cannot be found from the CEWed maps, and there is no difference between the D-Wed map and the original map visually.

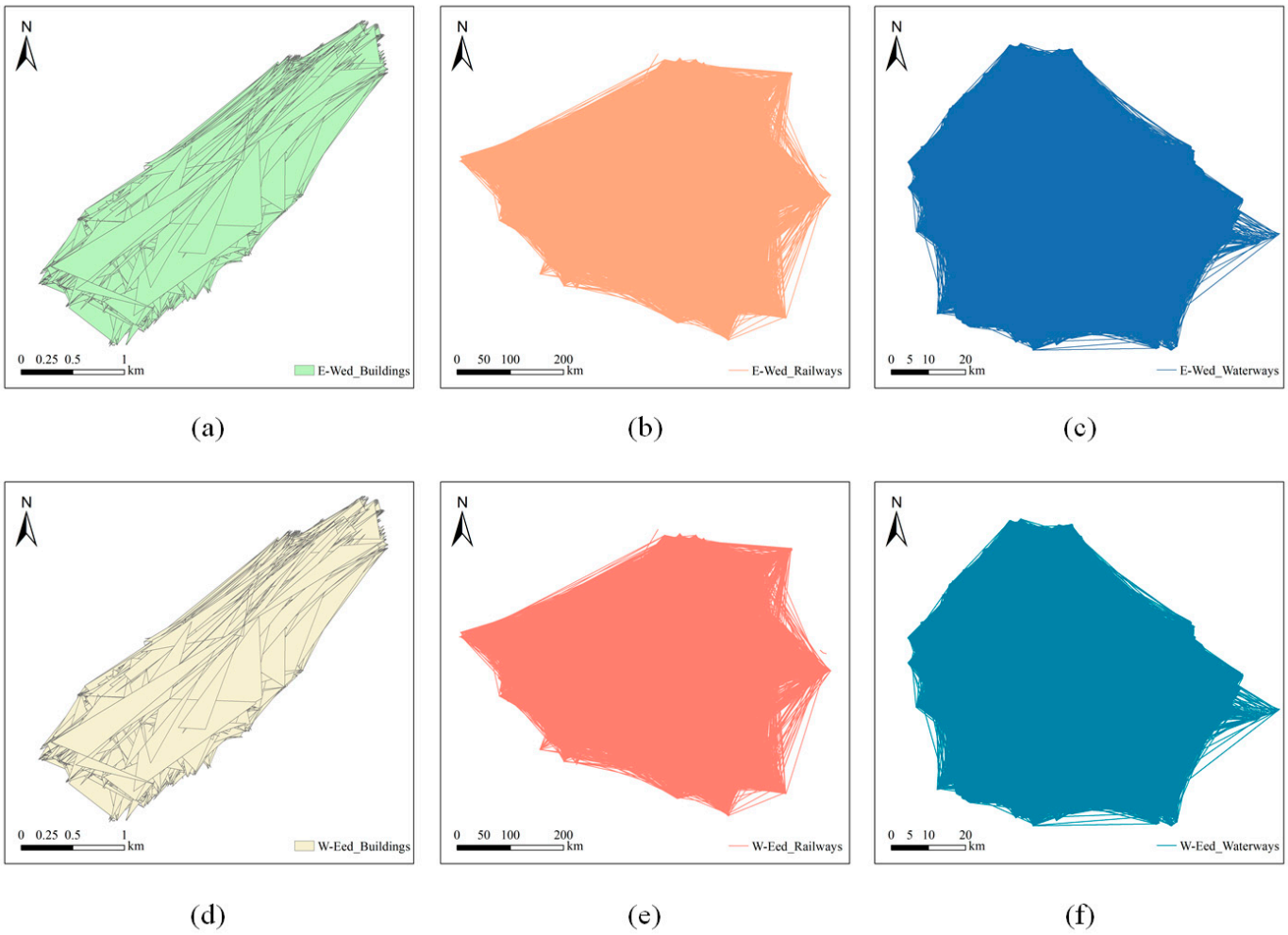

Figure 7. The visualization results of CEWed maps. (**a**) E-Wed buildings, (**b**) E-Wed railways, (**c**) E-Wed waterways, (**d**) W-Eed buildings, (**e**) W-Eed railways, (**f**) W-Eed waterways.

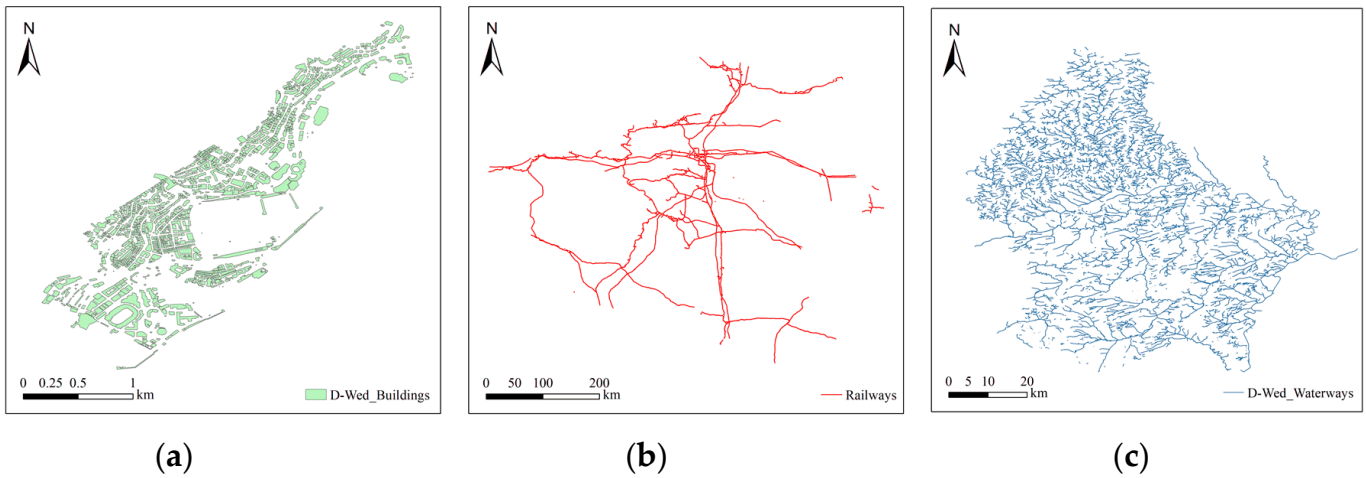

Figure 8. The visualization results of D-Wed maps: (**a**) D-Wed buildings, (**b**) D-Wed railways, (**c**) D-Wed waterways.

4.2. Commutativity Experiments

Commutativity refers to the mutual independence between the encryption and the watermarking algorithm, which is the key of a CEW algorithm. The commutativity requires that the operation order of encryption and watermark embedding has no impact on the final results, and the embedded watermark can be extracted before or after decryption.

To verify the commutativity between encryption and watermark embedding, Root-mean-square error (RMSE) is introduced to evaluate the difference between the E-Wed map and the W-Eed map. RMSE is calculated by Formula (17).

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\left(x_i' - x_i\right)^2 + \left(y_i' - y_i\right)^2\right)} \tag{17}$$

where (x_i, y_i) and (x_i', y_i') are the coordinates of two maps, respectively; N is the number of vertices. The value of RMSE is closer to 0, the error between the two maps is smaller. Table 2 displays the RMSE between the E-Wed maps and the W-Eed maps. Moreover, railways map is taken as an example, and the coordinates of some vertices in the E-Wed railways map and the W-Eed railways map are listed in Table 3. It can be found from Tables 2 and 3 that all vertices in both maps are identical, and their RMSEs are 0, which shows the E-Wed maps has the same content as the W-Eed maps. This is a good indication that the encryption operation and watermark embedding operation are commutative.

Table 2. The difference between the E-Wed map and the W-Eed map.

Comparison Result	Buildings	Railways	Waterways
Number of consistent vertices	11029	67123	186920
Number of inconsistent vertices	0	0	0
RMSE	0	0	0

Table 3. The coordinates of the E-Wed railways map compared with the W-Eed railways map.

The Coordinates of the E-Wed Railways Map	The Coordinates of the W-Eed Railways Map
(112.24909600028025, 34.69927299974921)	(112.24909600028025, 34.69927299974921)
(113.81205260004754, 34.242683800217264)	(113.81205260004754, 34.242683800217264)
(115.71832759963911, 31.882133599885776)	(115.71832759963911, 31.882133599885776)
...	...
(112.29290890012567, 33.053237600434954)	(112.29290890012567, 33.053237600434954)
(114.10714130013093, 32.103603399777114)	(114.10714130013093, 32.103603399777114)
(114.04892300039626, 35.42289510022874)	(114.04892300039626, 35.42289510022874)

To verify the commutativity between decryption and watermark extraction, normalized correlation coefficient (NC) is introduced to measure the difference between the original watermark and the extracted watermark. The definition of NC is shown in Formula (18).

$$NC = \frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N} XNOR\left(w_{i,j}, w_{i,j}'\right) \tag{18}$$

where $w_{i,j}$ and $w_{i,j}'$ denote the original watermark image and the extracted watermark image respectively, $M \times N$ is the size of the watermark image, and $XNOR(\bullet)$ is the operator of exclusive NOR. In this experiment, we extracted the watermark from the CEWed maps and the D-Wed maps respectively, and the NC values are displayed in Table 4. It is apparent that the NC values are always 1, which shows that the decryption operation and the watermark extraction operation are commutative.

Table 4. The NC value of the extracted watermark.

Type	Map	NC
CEWed map	Buildings	1
	Railways	1
	Waterways	1
D-Wed map	Buildings	1
	Railways	1
	Waterways	1

4.3. Encryption Security Analysis

(1). Key Space

The key space is an important evaluation indicator of key security. The key space should be large enough to resist exhaustive attacks, and the size of the key space should be greater than the standard requirement (i.e., 2^{100}) [24]. The keys of proposed algorithm consist of the following two keys: the 512 bits hash value generated by user's input key and the map file respectively. Therefore, the key length of proposed algorithm is 1024 bits, i.e., the size of key space is 2^{1024}, which is greater than 2^{100}. This shows that the proposed algorithm has strong resistance to exhaustive attacks.

(2). Key Sensitivity

Key sensitivity refers to the fact that the encrypted data will be changed dramatically and the decryption data cannot be obtained correctly when the key is modified even with minimal changes [31]. The cypher data must be highly sensitive to the key.

To analyze the effect of key change on encryption results, the initial key k_1 and the modified key k_2 are used to encrypt the maps, respectively. The initial keys are $k_1 = (U_k = 1234567890,\ H_k = abcdefghijklmn)$, and the modified keys are $k_2 = (U_k = 1234567891,\ H_k = abcdefghijklmn)$. Figure 9 shows the results of overlaying the encrypted maps with k_1 and the encrypted maps with k_2. It can be found that the encrypted maps with the modified keys are obviously different from the encrypted maps with the initial keys. Besides, to analyze the effect of different keys on the decryption results, we encrypted the original vector maps using k_1 and decrypted the maps using k_2. Figure 10 shows the maps obtained by decrypting with k_2, and the correct decryption results have been given in Figure 8a–c. It is apparent that the encrypted maps cannot be decrypted correctly even if the decryption key is slightly different from the encryption key. This shows that even a slight change of key can significantly affect the decryption result. Therefore, the proposed algorithm has strong key sensitivity.

4.4. Watermark Security Analysis

(1). Imperceptibility

Imperceptibility refers to the fact that the watermark information has no impact on the precision of the original vector maps, and it depends on the magnification of normalized coordinates (i.e., n) and the quantization step (i.e., R). In this experiment, the average distortion ($AveD$) and the maximum distortion ($MaxD$) are chosen to assess the imperceptibility of the watermarked maps. The $AveD$ and $MaxD$ are defined as follows:

$$AveD = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\left(x_i' - x_i\right)^2 + \left(y_i' - y_i\right)^2} \tag{19}$$

$$MaxD = \max\left(\sqrt{\left(x_i' - x_i\right)^2 + \left(y_i' - y_i\right)^2}\right) \tag{20}$$

where N is the number of vertices, $max(\bullet)$ is the maximum function, (x_i, y_i) and (x_i', y_i') are the coordinates of the original map and the watermarked map, respectively. In general, the accuracy of vector maps should not be lower than the minimum accuracy (10^{-4} m) [11].

Table 5 displays the relationship between *AveD*, *MaxD*, *NC* and n when $R = 8 \times 10^{-8}$, and Table 6 lists the relationship between *AveD*, *MaxD*, *NC* and R when $n = 8$. It can be found from Table 5 that the *NC* values are 1 when n ranges from 4 to 8, and the *AveD* is the lowest when $n = 8$. Besides, as can be seen from Table 6, the *NC* values are always 1 when the value of R is between 8×10^{-15} and 3×10^{-8}. Thus, to balance the imperceptibility of the watermark and the data precision, we embedded the watermark information into the 8th decimal of the normalized coordinates (i.e., n = 8), and the quantification step is set to 5×10^{-10} (i.e., $R = 5 \times 10^{-10}$).

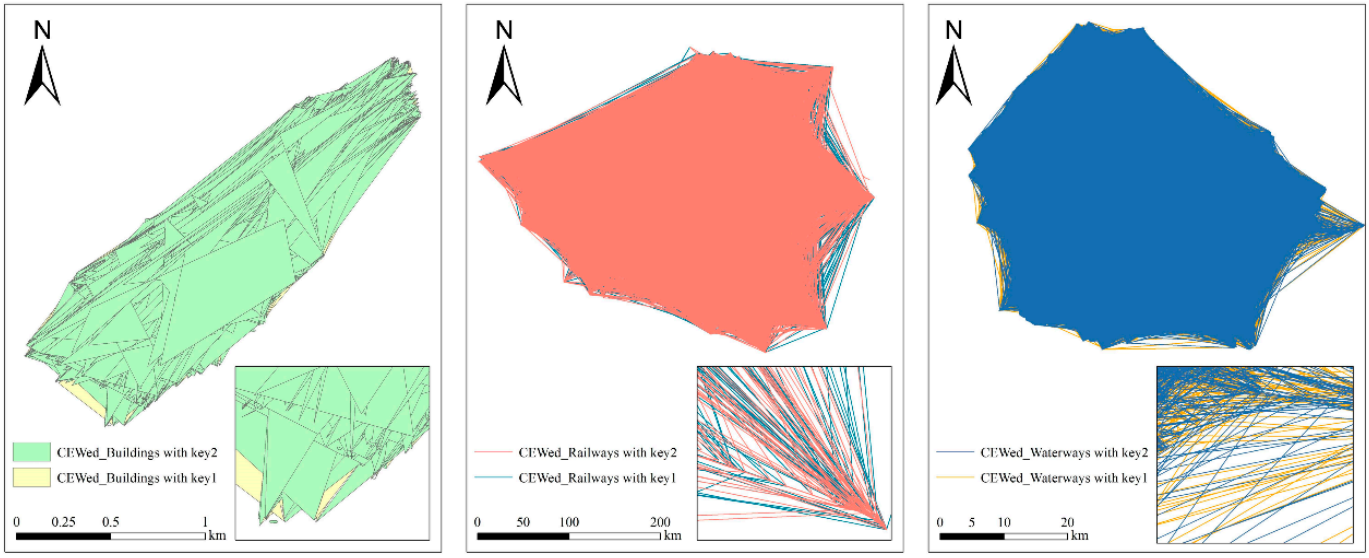

Figure 9. The encrypted maps using different keys: (**a**) encrypted buildings using k_1 and k_2, (**b**) encrypted railways using k_1 and k_2, (**c**) encrypted waterways using k_1 and k_2.

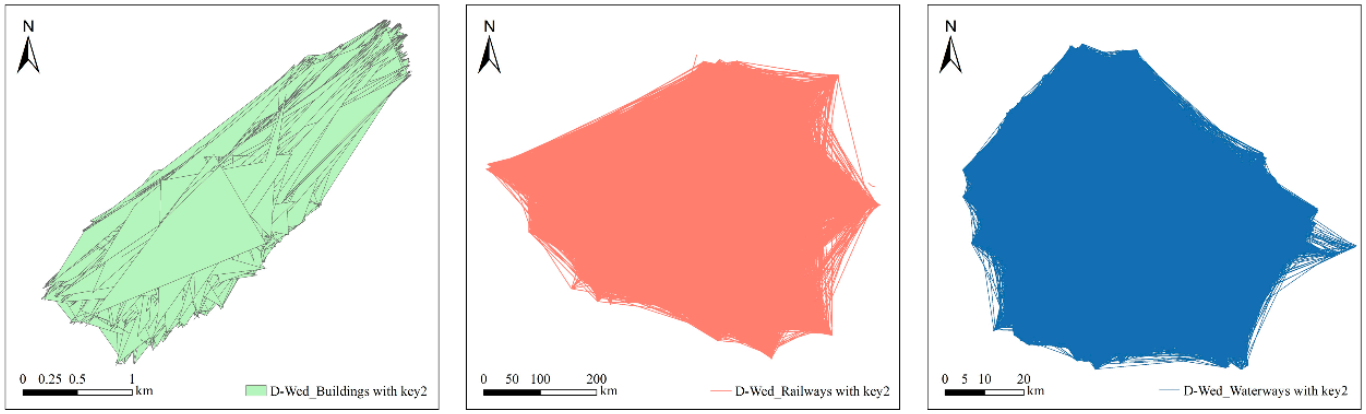

Figure 10. The D-Wed maps with modified key: (**a**) D-Wed buildings with k_2, (**b**) D-Wed railways with k_2, (**c**) D-Wed waterways with k_2.

Table 5. The relationship between *AveD*, *MaxD*, *NC* and n ($R = 8 \times 10^{-8}$).

n	*AveD*	*MaxD*	*NC*
2	1.8977×10^{-9}	3.1703×10^{-9}	0.8359
3	1.8972×10^{-9}	3.1703×10^{-9}	0.9038
4	1.8963×10^{-9}	3.1703×10^{-9}	1
5	1.8960×10^{-9}	3.1703×10^{-9}	1
6	1.8974×10^{-9}	3.1703×10^{-9}	1
7	1.9007×10^{-9}	3.1703×10^{-9}	1
8	1.8943×10^{-9}	3.1703×10^{-9}	1
9	1.8850×10^{-9}	3.1703×10^{-9}	0.9995

Table 6. The relationship between *AveD*, *MaxD*, *NC* and *R* (n = 8).

R	*AveD*	*MaxD*	*NC*
9×10^{-16}	4.6048×10^{-17}	3.5527×10^{-15}	0.51514
1×10^{-15}	2.1373×10^{-14}	3.5527×10^{-14}	0.52002
2×10^{-15}	2.1232×10^{-14}	3.5527×10^{-14}	1
3×10^{-10}	7.1163×10^{-10}	1.1889×10^{-9}	1
5×10^{-10}	1.1903×10^{-9}	1.9814×10^{-9}	1
9×10^{-9}	2.1455×10^{-8}	3.5667×10^{-8}	1
1×10^{-8}	2.3858×10^{-8}	3.9628×10^{-8}	1
2×10^{-8}	4.7059×10^{-8}	7.9257×10^{-8}	0.8979
3×10^{-8}	7.1305×10^{-8}	1.1889×10^{-7}	0.8750

(2). Watermark Robustness

Watermark robustness refers to the ability to reconstruct watermark information from the attacked vector maps. According to the practical application of the vector maps, the watermarked maps should be resistant to conventional attacks (e.g., vertex addition and deletion, reordering and data format conversion, etc.) and geometric attacks (e.g., rotation, scaling and translation, etc.). In this experiment, *NC* is used to verify the watermark robustness. In general, the *NC* value is closer to 1, the difference between the extracted watermark and the original watermark is smaller.

In the experiment of vertex attacks, vertex addition and deletion are performed on the watermarked maps, respectively. To ensure the reliability of the results, vertices are randomly added and deleted at certain ratios. we added the vertex from 10% to 100%, deleted the vertex from 10% to 60%, and the *NC* value are displayed in Tables 7 and 8. It can be clearly seen that the embedded watermark can still be extracted successfully, and the *NC* values are still higher than the threshold of 0.8 even if the vertices are increased by 1× or 60% vertices are deleted. This shows that the proposed algorithm is highly resistant to vertex addition attack and vertex deletion attack.

Table 7. The robustness results of vertex addition attack.

Vertex Addition Ratio (%)	*NC*		
	Buildings	**Railways**	**Waterways**
10	0.9921	1.0	1.0
20	0.9767	1.0	1.0
40	0.9541	1.0	1.0
60	0.9482	1.0	1.0
80	0.9014	1.0	1.0
100	0.8984	0.9902	1.0

Table 8. The robustness results of vertex deletion attack.

Vertex Deletion Ratio (%)	*NC*		
	Buildings	**Railways**	**Waterways**
10	0.9878	1.0	1.0
20	0.9677	1.0	1.0
30	0.9541	1.0	1.0
40	0.9434	1.0	1.0
50	0.9346	0.9981	1.0
60	0.8945	0.9812	1.0

Geometric attack is a common operation for vector map data. In the experiment, rotation, scaling and translation operation were performed on the watermarked maps. To ensure the reliability of the results, the watermarked maps were scaled from 0.2 to 6.0, translated from 10 to 100 m and rotated from 45° to 315°. The *NC* results are listed in Tables 9–11, respectively. The *NC* values show the proposed algorithm has good robustness in resisting scaling and translation attacks, whereas it is not resistant to rotation attack.

Table 9. The robustness results of scaling attack.

Attack Type	Attack Strength	*NC*		
		Buildings	**Railways**	**Waterways**
Scaling	0.2	1.0	1.0	1.0
	0.4	1.0	1.0	1.0
	0.6	1.0	1.0	1.0
	0.8	1.0	1.0	1.0
	2.0	1.0	1.0	1.0
	4.0	1.0	1.0	1.0
	6.0	1.0	1.0	1.0

Table 10. The robustness results of translation attack.

Attack Type	Attack Strength	*NC*		
		Buildings	**Railways**	**Waterways**
Translation	10	1.0	1.0	1.0
	20	1.0	1.0	1.0
	30	1.0	1.0	1.0
	40	1.0	1.0	1.0
	50	1.0	1.0	1.0
	60	1.0	1.0	1.0
	70	1.0	1.0	1.0
	80	1.0	1.0	1.0
	90	1.0	1.0	1.0
	100	1.0	1.0	1.0

Table 11. The robustness results of rotation attack.

Attack Type	Attack Strength	*NC*		
		Buildings	**Railways**	**Waterways**
Translation	45	0.5332	0.5215	0.4932
	90	0.4937	0.4824	0.5308
	135	0.5254	0.5107	0.4907
	180	0.5146	0.5137	0.5063
	225	0.5254	0.5005	0.5049
	270	0.5366	0.5303	0.5093
	315	0.5229	0.5234	0.5171

To guarantee that the vector maps can be used in different GIS platforms, the proposed algorithm must be robust in resisting format conversion attack. In this experiment, the watermark information is embedded into the vector maps in the shapefile format, and then the watermarked maps are transformed into dwg format, e00 format or gdb format. It is worth noting that the vector maps need to be converted to shapefile format before extracting the watermark. Table 12 reflects the results of watermark extraction, it is obvious that the watermark can be extracted successfully and that the NC values are always 1, which shows that the extracted watermark is not affected by data format conversion.

Table 12. The robustness results of format conversion attack.

Data Format	*NC*		
	Buildings	**Railways**	**Waterways**
dwg	1.0	1.0	1.0
e00	1.0	1.0	1.0
gdb	1.0	1.0	1.0

Reordering refers to rearrange the storage order of vector map data. The watermark embedding and extraction operations are implemented based on the coordinates of the vector maps, which is not affected by the storage order. In other words, the watermark embedding positions will not change if embedding occurs after the reordering of the vertices, and the embedded watermark can be correctly extracted by the original embedding positions. To further illustrate the reliability of the above theory, the storage order of objects was randomly disrupted and reordered in the watermarked maps. Results found that the *NC* values between the extracted watermark and the original watermark are always 1, which confirms that the reordering of objects has no effect on the extraction of the watermark information.

(3). Robustness comparisons

To further highlight the better performance of our method, the Railways map was used to implement the comparison with the existing methods [2,13,24]. We compared their robustness in vertex attacks, geometric attacks and data format conversion attack. Figure 11 displays the comparison results. Figure 11a–b reflect that the proposed algorithm is more robust in resisting vertex addition attack and vertex deletion attack. As shown in Figure 11c–f, the proposed algorithm has the same robustness as the existing methods [2,13,24] in resisting data format conversion, scaling and translation attacks, whereas its robustness in resisting rotation attack is not as good as the existing methods [2,13,24].

The above results show that the proposed algorithm is resistant to vertex attacks, scaling and translation attacks, data format conversion attack. However, the normalized coordinates are invariant to translation and scaling, but not to rotation. As a result, the proposed algorithm has no resistance to rotation attack.

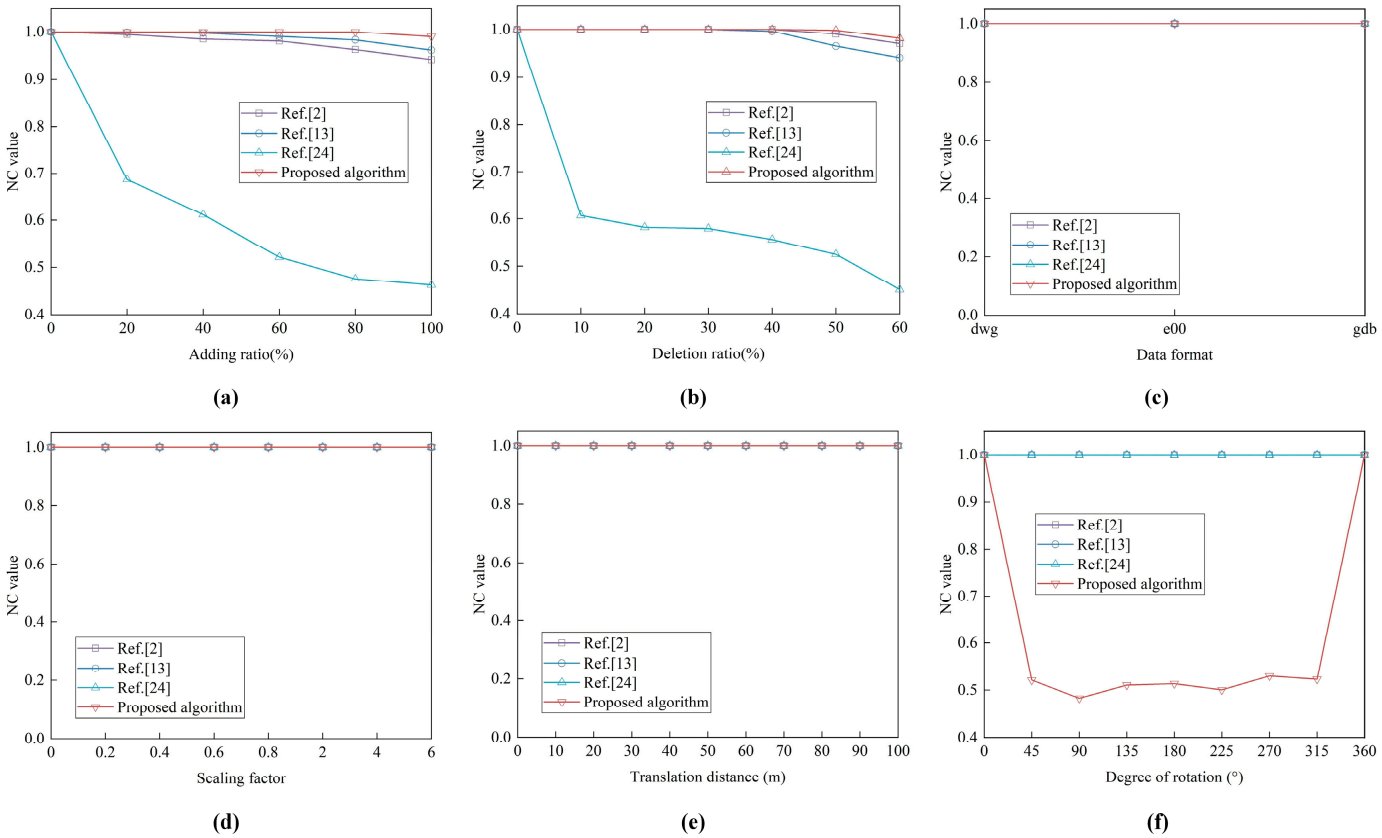

Figure 11. Comparison results for vertex attacks: (**a**) comparison results for vertex addition, (**b**) comparison results for vertex deletion, (**c**) comparison results for data format conversion attack, (**d**) comparison results for scaling attack, (**e**) comparison results for translation attack, (**f**) comparison results for rotation attack.

5. Conclusions

A novel invariant-based CEW algorithm is proposed in this paper, aiming at providing the dual capability of secure transmission and copyright protection for vector maps. The permutation-based encryption scheme does not change the coordinates of vector maps, which is combined with the coordinates-based watermarking scheme to implement the commutativity between encryption operation and watermarking operation. Based on the proposed algorithm, the operation order of encryption and watermark embedding has no impact on the final results, and the embedded watermark can be extracted before or after decryption. The contributions of this paper are as follows:

(1). Proposed algorithm provides multiple protection for vector maps by the combination of encryption technology and watermarking technology compared to existing encryption algorithm and watermarking algorithm.

(2). Compared with the existing CEW methods for vector maps, proposed method has stronger robustness in resisting vertex attacks by using multiple embedding strategy for watermark. In addition, proposed method is resistant to geometric attacks (e.g., scaling and translation) and other conventional attacks (e.g., reordering and data format conversion).

(3). Compared with the existing CEW methods for vector maps, the introduction of double random position permutation strategy completely avoids one-to-one mapping during vector map scrambling and greatly enhances encryption security.

Proposed algorithm has a poor encryption effect on point objects, because the permutation of point objects can only change the storage order, which has no impact on the data structure. Therefore, the CEW algorithm for the point objects of vector maps will be our work in future research.

Author Contributions: Yu Li conceived and designed the experiments; Yu Li and Liming Zhang carried out the method; Yu Li and Xiaolong Wang completed the implementation of the computer code and the supporting algorithms; Yu Li performed the analysis and wrote the paper; Liming Zhang, Xiaolong Wang, Xingang Zhang and Qihang Zhang reviewed and edited the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work is 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), the Talent Innovation Venture Science and Technology Program of Lanzhou (grant no. 2016-RC-59), and Funded by LZJTU EP (grant no.201806).

Data Availability Statement: Experimental data are downloaded from OpenStreetMap (http://download.geofabrik.de/, accessed on 10 January 2021).

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

References

1. Yang, C.; Zhu, C.; Wang, Y.; Rui, T.; Zhu, J.; Ding, K. A Robust Watermarking Algorithm for Vector Geographic Data Based on Qim and Matching Detection. *Multimed. Tools Appl.* **2020**, *79*, 30709–30733. [CrossRef]
2. 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]
3. Zhang, L.; Yan, H.; Zhu, R.; Du, P. Combinational spatial and frequency domains watermarking for 2D vector maps. *Multimed. Tools Appl.* **2020**, *79*, 31375–31387. [CrossRef]
4. Zhang, L.; Yan, H.; Qi, J.; Zhang, Y. A blind watermarking algorithm for copyright protection of vector geospatial data under controllable errors based on DFT. *Geomat. Inf. Sci. Wuhan Univ.* **2015**, *40*, 990–994.
5. Peng, Y.; Lan, H.; Yue, M.; Xue, Y. Multipurpose watermarking for vector map protection and authentication. *Multimed. Tools Appl.* **2018**, *77*, 7239–7259. [CrossRef]
6. 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.
7. 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).
8. Li, A.; Wang, H.; Zhou, W. Scrambling encryption of vector digital map based on 2D chaos system. *J. China Univ. Min. Technol.* **2015**, *44*, 747–753.
9. 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. [CrossRef]
10. Zhu, C. Research Progresses in Digital Watermarking and Encryption Control for Geographical Data. *Acta Geod. Cartogr. Sin.* **2017**, *46*, 1609–1619.
11. Zhou, Q.; Ren, N.; Zhu, C.; Tong, D. Storage feature-based watermarking algorithm with coordinate values preservation for vector line data. *KSII Trans. Internet Inf. Syst.* **2018**, *12*, 3475–3496.
12. Benrhouma, O.; Mannai, O.; Hermassi, H. Digital images watermarking and partial encryption based on DWT transformation and chaotic maps. Systems, Signals & Devices (SSD). In Proceedings of the 2015 IEEE 12th International Multi-Conference on Systems, Signals & Devices (SSD15), Mahdia, Tunisia, 16–19 March 2015.
13. Jang, B.J.; Lee, S.H.; Lee, E.J.; Lim, S.; Kwon, K.R. A crypto-marking method for secure vector map. *Multimed. Tools Appl.* **2017**, *76*, 16011–16044. [CrossRef]
14. Puech, W.; Rodrigues, J.M.; Develay-Morice, J.E. A new fast reversible method for image safe transfer. *J. Real-Time Image Process.* **2007**, *2*, 55–65. [CrossRef]
15. Jiang, L. The identical operands commutative encryption and watermarking based on homomorphism. *Multimed. Tools Appl.* **2018**, *77*, 30575–30594. [CrossRef]
16. Zhang, X. Commutative reversible data hiding and encryption. *Secur. Commun. Netw.* **2013**, *6*, 1396–1403. [CrossRef]
17. Li, M.; Xiao, D.; Zhu, Y.; Zhang, Y.; Sun, L. Commutative fragile zero-watermarking and encryption for image integrity protection. *Multimed. Tools Appl.* **2019**, *78*, 22727–22742. [CrossRef]
18. Schmitz, R. Use of SHDM in commutative watermarking encryption. *EURASIP J. Inf. Secur.* **2021**, *2021*, 1. [CrossRef]
19. Cancellaro, M.; Battisti, F.; Carli, M.; Boato, G.; De Natale, F.G.; Neri, A. A Commutative Digital Image Watermarking and Encryption Method in the Tree Structured Haar Transform Domain. *Signal Process. Image Commun.* **2011**, *26*, 1–12. [CrossRef]
20. Jiang, L.; Xu, Z.Q.; Xu, Y.Y. Commutative encryption and watermarking based on orthogonal decomposition. *Multimed. Tools Appl.* **2014**, *70*, 1617–1635. [CrossRef]
21. Li, Y.; Zhang, L.; Wang, H.; Wang, X. Commutative Encryption and Watermarking Algorithm for High-Resolution Remote Sensing Images Based Homomorphic Encryption. *Laser Optoelectron. Prog.* Available online: http://kns.cnki.net/kcms/detail/31.1690.tn.20210816.1546.060.html (accessed on 24 October 2021).

22. Schmitz, R.; Li, S.; Grecos, C.; Zhang, X. Towards robust invariant commutative watermarking-encryption based on image histograms. *IJMDEM* **2014**, *5*, 36–52. [CrossRef]
23. Schmitz, R.; Li, S.; Grecos, C.; Zhang, X. A new approach to commutative watermarking-encryption. In Proceedings of the IFIP International Conference on Communications and Multimedia Security, Canterbury, UK, 3–5 September 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 117–130.
24. Ren, N.; Zhu, C.; Tong, D.; Chen, W.; Zhou, Q. Commutative encryption and watermarking algorithm based on feature invariants for secure vector map. *IEEE Access* **2020**, *8*, 221481–221493. [CrossRef]
25. Schmidt, S.; Barron, C. Mapping impervious surfaces precisely—a GIS-based methodology combining vector data and high-resolution airborne imagery. *J. Geovis. Spat. Anal.* **2020**, *4*, 1–10. [CrossRef]
26. Banesh, D.; Petersen, M.R.; Ahrens, J.; Turton, T.L.; Samsel, F.; Schoonover, J.; Hamann, B. An Image-Based Framework for Ocean Feature Detection and Analysis. *J. Geovis. Spat. Anal.* **2021**, *5*, 1–21. [CrossRef]
27. Lingfeng, Q. Security analysis of multiple permutation encryption adopt in reversible data hiding. *Multimed. Tools Appl.* **2020**, *79*, 29451–29471. [CrossRef]
28. Zhu, H.; Lu, X.; Zhang, X.; Tang, Q. A novel image encryption scheme with 2D-logistic map and quadratic residue. *J. Northeast Univ.* **2014**, *35*, 20–23.
29. Seyedzadeh, S.M.; Mirzakuchaki, S. A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map. *Signal Process.* **2012**, *92*, 1202–1215. [CrossRef]
30. Yan, H.; Zhang, L.; Yang, W. A normalization-based watermarking scheme for 2D vector map data. *Earth Sci. Inform.* **2017**, *10*, 471–481. [CrossRef]
31. Li, Y.; Zhang, L.; Wang, H.; Wang, X. Multiple Security Protection Algorithm for GF-2 Images Based on Commutative Encryption and Watermarking. In Proceedings of the International Conference on Spatial Data and Intelligence, Hangzhou, China, 22–24 April 2021; Springer: Cham, Switzerland, 2021; pp. 141–147.