Vector Map Random Encryption Algorithm Based on Multi-Scale Simplification and Gaussian Distribution

Giao N. Pham 1,2,*, Son T. Ngo 3, Anh N. Bui 3, Dinh V. Tran 4, Suk-Hwan Lee 5 and Ki-Ryong Kwon 6

1 Dept. of Computing Fundamentals, FPT University, Hanoi 10000, Vietnam; giaopn@fe.edu.vn
2 Advance Analytics Center, FPT Software, Hanoi 10000, Vietnam; giaopn@fsoft.com.vn
3 Dept. of Computing Fundamentals, FPT University, Hanoi 10000, Vietnam; sonnt69@fe.edu.vn (S.T.N.); anhbn5@fe.edu.vn (A.N.B.)
4 Dept. of Computer, University of Freiburg, 79098 Freiburg, Germany; Dinh@informatik.uni-freiburg.de
5 Dept. of Information Security, Tongmyong University, Busan 608-737, Korea; skylee@tu.ac.kr
6 Dept. of IT Convergence and Application Engineering, Pukyong National University, Busan 608-711, Korea; krkwon@pknu.ac.kr

* Correspondence: giaopn@fe.edu.vn

Received: 2 October 2019; Accepted: 13 November 2019; Published: 14 November 2019

Abstract: In recent years, GIS (Geographical Information System) vector maps are widely used in everyday life, science, and the military. However, the production process of vector maps is expensive, and a large volume of vector map data is easily stolen and illegally distributed. Therefore, original providers desire an encryption solution to encrypt GIS vector map data before being stored and transmitted in order to prevent pirate attacks and to ensure secure transmission. In this paper, we propose an encryption algorithm for GIS vector map data for preventing illegal copying, and ensuring secured storage and transmission. Polyline/polygon data of GIS vector maps are extracted to compute a backbone object. The backbone object is then selectively simplified by the multi-scale simplification algorithm in order to determine the feature vertices of the backbone object. The feature vertices of the backbone object are encrypted by the advanced encryption standard and the secret key. Finally, all vertices of the backbone object are randomized by the random Gaussian distribution algorithm to obtain the encrypted GIS vector map. Experimental results show that the entire map is altered completely after the encryption process. The proposed method is responsive to the various GIS vector map data formats, and also provides better security than previous methods. The computation time of the proposed method is also significantly shorter than that of previous methods.

Keywords: GIS vector map data; GIS vector map security; selective encryption; simplification method and cryptography

1. Introduction

Currently, GIS vector maps are used in applications in many fields, such as science, navigation, and online or offline digital map services. GIS vector maps are created and developed by the merging of cartography, statistical analysis, and database technology based on vector models [1]. Due to the fact that GIS vector maps may have significant value, GIS vector map data may be stolen or easily purchased and redistributed or resold several times without obtaining any permission from the original data providers. In addition, some applications of vector maps by the military or some personal applications of vector maps must be secured from unauthorized users. Consequently, GIS vector map data protection is necessary to prevent illegal duplication and distribution.
GIS vector map security includes copyright protection [2–4], access control for users, and vector map data encryption in order to prevent attacks or illegal distribution during storage and transmission, and damage in the integration process of geographical information [5–14]. However, access control on the Web is unable to prevent attacks or illegal duplication and distribution, and watermarking techniques are only useful in identifying ownership and for copyright protection, while unauthorized users should not be able to see, attack, or extract GIS vector map content. Therefore, GIS vector map data encryption is necessary and suitable for protection. Moreover, conversion between data formats is vulnerable to attack and security techniques based on database management systems are not responsive to the various data formats of vector maps [10,11]. Thus, the encryption techniques for GIS vector map should be responsive to the various data formats of GIS vector maps and the security requirements.

In order to meet the issues above, we propose an encryption algorithm for GIS vector map data in this paper. Our algorithm solves the current issues of GIS vector map security by encrypting the geometric objects of GIS vector maps. The main intent of the proposed algorithm is to extract geometric objects (polylines and polygons) from GIS vector maps in order to encrypt them. Geometric objects are used to compute the backbone object of polyline/polygon. The backbone object is then selectively simplified by the multi-scale simplification algorithm in order to determine the feature vertices of the backbone object. The feature vertices of the backbone object is encrypted by the AES (Advanced Encryption Standard) algorithm and the secret key. Finally, all vertices of the backbone object are randomized by the random Gaussian distribution in order to generate the encrypted vector map. To clarify the detailed contents of the proposed algorithm, this paper is organized as follows: In Section 2, we discuss vector map data security techniques and explain the relationship between vector map data and the proposed algorithm. In Section 3, we explain the proposed algorithm in detail. We then perform experiments, discuss the experimental results, and evaluate the performance of the proposed algorithm in Section 4. The conclusion is provided in Section 5.

2. Related Works

2.1. Vector Map Data Security

The main idea of the watermarking schemes for GIS vector maps is to embed the watermark into the vector map by modifying the coordinates of vertices of geometric objects in the spatial domain [2,3], or embedding the watermark into the spectrum coefficients of a sequence of vertices or topologies in the frequency domain [4]. Thus, vector map watermarking is not suitable for preventing attacks and illegal copying.

The security techniques for GIS vector map data have been proposed in recent years [5–9]. Authors have mostly explained various methods to define an access control system for spatial data on the Web, or mentioned the security requirements for geospatial database management systems and privacy policies. However, access control on the Web and the management of databases do not prevent attacks, illegal duplication, and distribution. Wu et al. [10] proposed a compound chaos-based encryption algorithm for vector data by considering the storage characteristics and the parameters of a chaos-based system; however, this method is not responsive to the various data formats of vector maps. Li et al. [11] encrypted the spatial index of a set of vector data in an external Oracle database management system when it is transmitted to the client; however, this algorithm does not ensure the security of vector maps because the key length is very short. Yasser et al. [12] also described an encryption algorithm that combined AES and RSA (Rivest–Shamir–Adleman) cryptography with a simple watermarking technique in order to protect the copyright protection of vector maps in online and offline services. This method encrypts all parts of a shape-file using an AES block cipher operator of 256 bits. This approach is typical because it encrypts the data stream of a shape-file using the AES cipher, and the computation time is very long. Jang et al [13] proposed a perceptual encryption method that is combined with the compression process for vector map data. This method only encrypted the direction and position of data units in the compression process. This method has low security. Bang et al. [14] proposed a selective encryption method for vector map data
based on a chaotic map in the frequency of discrete wavelet transform (DWT); however, this method only selects some objects and encrypts DC (Direct Current) values in the DWT domain by a common secret key. This method is very simple and weak because it does not encrypt all contents of vector maps and uses a common key. Moreover, this research did not present performance or security evaluations. In summary, the weakness of previous methods is solved by the proposed algorithm, which is presented in Section 3.

2.2. Vector Map Data-Based Encryption

GIS vector map data is stored in layers. Each layer contains a number of geometric objects, such as points, polylines, and polygons, and annotations, such as symbols. This notation is shown in Figure 1a. Annotation is used to display notes on vector maps, while geometric objects are used to represent geographical objects on vector maps. Points are used to represent simple objects, such as position, while polygons and polylines are used to represent complex objects. A polyline is a set of ordered vertices used to represent objects such as roads, contour lines, rivers, and railways. A polygon is a set of connected polylines used to represent objects such as buildings, areas, lakes, and boundaries. Thus, polylines and polygons are considered the important components of a vector map.

In addition to annotation and geometric objects, vector maps also contain the storage information as header and text. Because points, polylines, and polygons determine the content of vector maps, we consider these geometry data, while the annotation, header, and text are considered attribute data. Figure 1b shows the data components of a GIS vector map. The attribute data of a GIS vector map does not contain geographical information or determines the shape of the GIS vector map; thus, it is only required to extract polylines and polygons to perform the random encryption process for GIS vector maps.

![Figure 1](image)

(a)

![Figure 1](image)

(b)

Figure 1. (a) Vector map data model, and (b) data components of a vector map.

3. The Proposed Method

3.1. Overview
The proposed algorithm is shown in detail in Figure 2. To begin, each geometric object (polyline/polygon) is extracted from the GIS vector map to compute a backbone object. The backbone object is then selectively simplified by the multi-scale simplification algorithm with differential scale factors in order to obtain new geometric objects. If the backbone object cannot be simplified, it is then encrypted by a key value and randomized by a set of random Gaussian numbers to generate an encrypted object. The key value is generated by a hashing function with a user’s key input. New geometric objects, which are simplified from the backbone object, are then used to compute the feature vertices and the remaining vertices. The feature vertices of the backbone object will be continuously encrypted by a key value in order to generate the encrypted feature vertices. Finally, the remaining vertices and encrypted feature vertices of the backbone object will be randomly encrypted by a set of random Gaussian numbers in order to obtain the encrypted object. The encrypted GIS vector map is a set of the encrypted geometric objects.

A GIS vector map contains a number of data layers. Each layer \( L \) contains a number of objects (polyline/polygon) \( L = \{ P_i \mid i \in [1, |L|] \} \) and each object contains a number of vertices \( P_i = \{ v_{i,j} \mid j \in [1, |P_i|] \} \). We briefly define the main notations as the following: \( P_i \) indicates a polyline/polygon object, and \(| L |\) and \(| P_i |\) are the cardinalities of a layer \( L \) and an object \( P_i \) respectively. Thus, \( v_{i,j} \) indicates the \( j^{th} \) vertex in the \( i^{th} \) object of a layer \( L \) and is defined by two coordinates \( v_{i,j} = (x_{i,j}, y_{i,j}) \). Next, \( B_i \) is the backbone object of \( P_i \) and \( B_i^p \) is the simplified object of \( B_i \) after the multi-scale simplification process with the differential scales. \( B_i' \) is the changed backbone object after the feature vertices encryption process or the backbone object encryption process. \( F_i \) represents the feature vertices of the backbone object \( B_i \), \( G_n \) is a set of random Gaussian numbers, \( K \) is the secret key, and \( E_i \) is the encrypted object of \( P_i \). Finally, \( C_p(\cdot) \), \( G_p(\cdot) \), and \( R_p(\cdot) \) are the cipher function, the random Gaussian number function, and the randomization function, respectively.

![Figure 2. The proposed algorithm.](image)

3.2. Backbone Object Simplification and Feature Vertices Computation

The backbone object is a set of vertices, in which each vertex is the average point between two continuous vertices in that object. Thus, the backbone object of \( P_i \) is \( B_i = \{ b_{i,j} \mid j \in [1, |B_i|] \} \) where \(| P_i | = |B_i |\) and the value of \( b_{i,j} \) is computed as shown in Equation (1). Figure 3a shows a backbone object of a polyline object.
Our purpose is to simplify the backbone object with the scale factor \( s \) in order to compute the feature vertices of the backbone object. Thus, we must check the condition before simplifying. The condition of a backbone object \( B_{ij} \) can be simplified as:

\[
|B_{ij}| > \frac{1}{s} \tag{2}
\]

If the value of \( |B_{ij}| \leq \frac{1}{s} \), that backbone object is encrypted by the cipher function \( C_p(.) \) as shown in Equation (3).

\[
B'_{ij} = C_p(K, B_{ij}) = \{b'_{ij} \mid j \in [1, |B_{ij}|]\} \tag{3}
\]

The backbone object \( B_{ij} \) is simplified by the multi-scale simplification algorithm in order to obtain the simplified backbone object \( B_{ij}^{s} \) as shown in Equation (4):

\[
B_{ij}^{s} = \{b_{ij}^{s} \mid j \in [1, |B_{ij}| \times s]\} \tag{4}
\]

\[
F_i = B_{ij}^{1/2} \cap B_{ij}^{1/3} \cap B_{ij}^{1/4} \tag{5}
\]

Here, we obtain three simplified objects \( B_{ij}^{1/2}, B_{ij}^{1/3}, B_{ij}^{1/4} \) with differential scales \( s = \frac{1}{2}, s = \frac{1}{3}, \) and \( s = \frac{1}{4} \) (see Figure 3b). We then calculate the feature vertices \( F_i \), which is a set of the common vertices of three simplified backbone \( B_{ij}^{1/2}, B_{ij}^{1/3}, \) and \( B_{ij}^{1/4} \) as shown in Equation (5). From Equations (4) and (5), we can see that \( F_i \) will be determined if \( |B_{ij}| > 4 \). Figure 3b shows three simplified backbone objects of a polyline object with differential scales 1/2, 1/3, and 1/4, and the feature vertices of the backbone object.
3.3. Random Encryption

The secret key $K$ is generated by the SHA-512 hashing algorithm [15] with user’s key input. We use the secret key $K$ to compute a set of random numbers $G_n$ by the Gaussian distribution [16,17] as shown in Equation (6):

$$G_n = G_p(K) = \{g_{ij} | j \in [1, |G_n|]\}$$  \hspace{1cm} (6)

Therein, $|G_n| = |P_l|$, $g_{ij}$ is calculated by the Gaussian function as shown in Equation (7), and $x$ is the value of $K$:

$$g_{ij} = i \times j \times \frac{1}{|P_l|} \times \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$  \hspace{1cm} (7)

As mentioned above, after the feature vertices computation process, we have a set of feature vertices $F_i$ and the remaining vertices of the backbone object $(B_i - F_i)$. We then encrypt the feature vertices $F_i$ by the cipher function $C_p(\cdot)$ using the secret key $K$, as shown in Equation (8), in order to obtain the encrypted feature vertices $F'_i$. The cipher function $C_p(\cdot)$ can be the AES cipher function, the DES cipher function, or others.

$$F'_i = C_p(F_i, K)$$  \hspace{1cm} (8)

The encrypted feature vertices $F'_i$ and the remaining vertices $(B_i - F_i)$ are used for the random encryption process in order to obtain the encrypted object $E_i$. From Section 3.2, the changed backbone object $B'_i$ is computed by Equation (3) when $|B_i| \leq \frac{1}{s}$. Here, $B'_i$ includes the encrypted feature vertices $F'_i$ and the remaining vertices $(B_i - F_i)$ as shown in Equation (9).

$$B'_i = (B_i - F_i) \cup F'_i$$  \hspace{1cm} (9)
The random encryption process is performed by a randomization process using the random Gaussian numbers $G_i^n$:

$$E_i = R_p(B_i', G_i^n) = \{e_{i,j} | j \in [1, |E_i|]\}$$  \hspace{1cm} (10)

where $|E_i| = |P|$ and $e_{i,j}$ is computed as follows:

$$e_{i,j} = b'_{i,j} \times g_{i,j} = b'_{i,j} \times j \times \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \prod_{j \in [1, |P|]}$$  \hspace{1cm} (11)

Figure 4 shows the random encryption process for the backbone object $B_i$ when the value of $|B_i| > \frac{1}{5}$ (Figure 4a with $|B_i| = 17$) and when $|B_i| \leq \frac{1}{5}$ (Figure 4b with $|B_i| = 4$). When the value of $|B_i| > 4$, the feature vertices of $B_i$ will be encrypted by the cipher function $C_p(\cdot)$. Then, the randomization process is performed as shown in Figure 4a. In case $|B_i| = 4$, the backbone object will be directly encrypted by the cipher function $C_p(\cdot)$ and then randomized by a set of Gaussian numbers, as shown in Figure 4b.
3.4. Decryption Process

The decryption process is the inverse of the encryption process. Firstly, the key value $K$ is generated from the user’s key by the SHA-512 hashing algorithm. The key value $K$ is then used to compute a set of random Gaussian numbers as described by Equations (6) and (7) in Section 3.3. The encrypted objects are then extracted from the encrypted vector map and vertices are re-randomized by the random Gaussian numbers before decryption using the key value $K$. After the vertex re-randomization process, if the encrypted object cannot be simplified, it is decrypted by the key value $K$ to generate the backbone object. If the encrypted object can be simplified, it is simplified to compute the encrypted feature vertices. These encrypted feature vertices are then decrypted by the key value $K$ to generate the backbone object. From the backbone object, we can calculate the decrypted objects based on Equation (1). The decrypted GIS vector map is a set of the decrypted objects.

4. Experimental Results and Analysis

We used the GIS vector maps of the country of Scotland [18] with differential scales in visualization experiments, and evaluation of security and computation time. The detailed information of the GIS vector maps is shown in Table 1. The data format of the GIS vector map was the shape-file (SHP) format [19], which is a popular geographical vector data format. The proposed algorithm was applied to the polylines and polygons of the vector maps. The backbone encryption process and the feature vertices encryption process were performed by the AES algorithm. We selected the AES algorithm because its security is higher than others. Compared with conventional approaches, the proposed algorithm is more original than previous methods because it encrypts objects based on encrypting the value of feature vertices in polylines/polygons. Consequently, it does not alter or expand the size of the encrypted file, thus preventing data loss. The GIS vector maps are completely altered after the random encryption process (see Figure 5 to Figure 9).

Table 1. Experimental results of Scotland maps.

| Scotland Maps | Size (Kb) | # Objects | Computation time (ms) | Entropy (dB) |
|---------------|-----------|-----------|----------------------|--------------|
| Railway       | 1951      | 9724      | 77                   | 128,817      |
| Land-use      | 12,452    | 41,242    | 263                  | 632,315      |
| Waterway      | 27,954    | 53,389    | 668                  | 838,434      |
| Nature        | 51,417    | 99,835    | 1367                 | 1,657,985    |
| Road          | 78,136    | 372,138   | 1896                 | 6,886,591    |

4.1. Visualization Experiments

Experimental results are shown in Figures 5–9, which show the original map and a part of the original map beside the encrypted map for comparison. Figure 5a shows the original railway map of Scotland and a part of the original railway map. The content of the railway map is presented by polylines. After encryption, polylines are altered, broken into segments, and positioned in a disorderly manner (Figure 5b). In the experiment with the land-use map of Scotland, the content of the map includes polygons (Figure 6a). After encryption, the shape of all polygons is changed to smaller polygons on the map, and the shape of the entire map is altered completely (Figure 6b). Experiments on the waterway map (Figure 7a), the nature map (Figure 8a), and the road map (Figure 9a) of Scotland also yield similar results. Waterway lines and roads on the maps are broken into shorter polylines and positioned in a disorderly manner (Figures 7b and 9b), and the shape of the original polygons are altered and moved to other positions (Figure 8b). Consequently, the content of the GIS vector maps is altered completely.
Figure 5. (a) Original railway map, and (b) encrypted railway map.

Figure 6. (a) Original land-use map, and (b) encrypted land-use map.
Figure 7. (a) Original waterway map, and (b) encrypted waterway map.

Figure 8. (a) Original nature map, and (b) encrypted nature map.
4.2. Security Evaluation

In order to evaluate the security of the proposed method, in this section we evaluate the randomness of the encrypted map. If the randomness of the encrypted map is high, it will also be difficult to attack. The randomness of the encrypted map is measured by its entropy. Thus, we calculate the entropy of the encrypted map to evaluate the security of the proposed method. From Section 3, we can see that the entropy of the encrypted object is dependent on the secret key $K$ and the value of $|P_i|$. Both $K$ and $|P_i|$ are discrete random variables. Thus, the entropy $H_{P_i}$ of the encrypted object $E_i$ is the sum of the entropies of the random variables:

$$H_{P_i} = H(K) + H(|P_i|) = |K| \log_2 \left( \frac{1}{|K|} \right) + |P_i| \log_2 \left( \frac{1}{|P_i|} \right)$$  \hspace{1cm} (12)

Furthermore, the entropy $H_L$ of the encrypted map from the original map layer $L$ will be the sum of the entropies of the encrypted objects:

$$H_L = \sum_{i=1}^{||L||} H_{P_i}$$  \hspace{1cm} (13)

Clearly, $H_L$ is dependent on the values of $||L||$ and $K$; the key value $K$ is a random variable dependent on a user’s key input. If the value of $K$ is fixed, $H_L$ is only dependent on the value of $||L||$. As a result, if $||L||$ is high, the entropy is high. For example, the railway map of Scotland has $||L|| = 9724$ and $H_L = 128,817 dB$, but the natural map of Scotland has $||L|| = 99,835$ and $H_L = 1.66 \times 10^6 dB$. In the vector maps of Scotland in Table 1, the entropy of the proposed method ranges from $1.3 \times 10^5 dB$ to $6.88 \times 10^6 dB$ with $||L|| \in [9724, 372,138]$.

The method of Yasser [10] only uses the AES-256 cipher operator to encrypt the data stream of a shape-file. Thus, the entropy of this method is dependent on the length of the secret key and the length of the data stream. In the work of Bang [12], about 70% of polylines/polygons are selected for the encryption process. Bang used a common secret key to encrypt all DC values in a vector map in

![Figure 9. (a) Original road map, and (b) encrypted road map.](image)
the DWT domain. The length of the secret key was 512 bits. Thus, the entropy of this method is dependent on the length of the secret key. We applied the methods of both Yasser and Bang for the GIS vector maps of Scotland for comparison. The entropy of our method was higher than that of Yasser (1.2×10^5 vs. 6.6×10^6 dB), while the entropy of Bang’s method was fixed at 4608 dB for every GIS vector map (Table 2). In conclusion, our method offers more security than previous methods. Figure 10 shows the difference between the entropy of our method and the entropy of the methods of Yasser and Bang according to the number of objects. The entropy of our method is much higher than that of Yasser or Bang.

**Table 2.** The entropy of the proposed method compared with previous methods.

| Scotland Maps | # Objects | Entropy (dB) |
|---------------|-----------|--------------|
|               |           | Our method   | Yasser’s method | Bang’s method |
| Railway       | 9724      | 128,817      | 9216           | 4608          |
| Land-use      | 41,242    | 632,315      | 55,296         | 4608          |
| Waterway      | 53,389    | 838,434      | 124,416        | 4608          |
| Nature        | 99,835    | 1,657,985    | 235,008        | 4608          |
| Road          | 372,138   | 6,886,591    | 299,520        | 4608          |

**Figure 10.** Entropy according to the number of objects.

### 4.3. Computation Time

We implemented the proposed method using the C# language in Visual Studio 2013. We then conducted our experiments on a PC with Intel Core i7 Quad 3.5 GHz, 8 GB of RAM, and Windows 7 64-bit. Section 3 indicates that the computation time of the proposed method is dependent on the number of objects in the vector maps. With the GIS vector maps of Scotland, the computation time of our method ranged from 77 ms to 1896 ms, with the size of maps ranging from 1951 Kbs to 78,136 Kbs. We also implemented the methods of Yasser and Bang using the C# language in Visual Studio 2013, in a similar environment, to measure and compare the computation time between methods. Yasser performed full encryption for GIS vector maps, which means he encrypted notation, header, text, and all geometric objects. Thus, the computation time of Yasser’s method is dependent on the size of the GIS vector maps. Compared with Yasser’s method, the computation time of our method was less (40 ms vs. 2670 ms; Table 3). In the method of Bang, the computation time is dependent on...
the computation time of the selection process, the number of selected objects, and the computation time of the DWT and inverse DWT processes. The computation time of the DWT and inverse DWT processes is dependent on the number of vertices in each object. If an object has many vertices, the computation time of the DWT process is long. Conversely, if the number of selected objects is small, the computation time is short (Table 3). Figure 11 shows the computation times of the proposed method compared to the methods of Yasser and Bang according to the size of the vector maps. The time of the proposed method is significantly shorter than that of previous methods.

![Figure 11. Computation time according to the size of map.](image)

Table 3. Computation time of the proposed method compared with previous methods.

| Scotland Maps | Size (Kb) | Computation Time (ms) |
|---------------|-----------|-----------------------|
|               | Our method | Yasser’s method | Bang’s method |
| Railway       | 1951       | 77                 | 117             | 416             |
| Land-use      | 12,452     | 263                | 759             | 1916            |
| Waterway      | 27,954     | 668                | 1705            | 5564            |
| Nature        | 51,417     | 1367               | 3136            | 9036            |
| Road          | 78,136     | 1896               | 4766            | 23,563          |

5. Conclusions

In this paper, we proposed a random encryption algorithm based on multi-scale simplification and the Gaussian distribution for GIS vector maps. Experimental results showed that the proposed method is very effective with GIS vector maps that contain many geometric objects. The presented method provides higher security than previous methods. The computation time of the method is significantly shorter than that of previous methods, and it could be used to replace previous methods for secured storage and transmission. By encrypting only geometric objects, the proposed method can be responsive to the various formats of GIS vector maps. In addition, to apply the method presented in this paper, developers only need to extract geometric objects before performing the encryption process. Furthermore, the proposed method can be applied to the security of online and off-line map services [20,21].

Author Contributions: Idea and Validation, S.H.L.; Supervision and Evaluation, K.R.K.; Research and Methodology, G.N. P.; Implementation and Software, S.T.N.; Visualization and Verification, A.N.B.; Data and Analysis, D.V.T.

Funding: This research received no external funding.
Acknowledgments: This research was supported by the FPT University; and the FPT Software, Hanoi, Vietnam; and also supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (No. 2016R1D1A3B03931003, No. 2017R1A2B2012456).

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

References

1. What is Geographic Information Systems (GIS)? Available online: https://gisgeography.com/what-gis-geographic-information-systems/ (accessed on 30 October 2019).
2. Wang, C.; Peng, Z.; Peng, Y.; Yu, L.; Wang, J.; Zhao, Q. Watermarking geographical data on spatial topological relations. *Multimed. Tools Appl.* **2012**, *57*, 67–69.
3. Lee, S.H.; Kwon, K.R. Vector watermarking scheme for GIS vector map management. *Multimed. Tools Appl.* **2013**, *63*, 757–790.
4. Ohbuchi, R.; Ueda, H.; Endoh, S. Robust watermarking of vector digital maps. In Proceedings of the IEEE International Conference on Multimedia and Expo, Lausanne, Switzerland, 26–29 August 2002; pp. 577–580.
5. Chena, S.C.; Wang, X.; Rishea, N.; Weiss, M.A. A web-based spatial data access system using semantic R-trees. *J. Inf. Sci.*** **2003**, *167*, 41–61.
6. Bertino, E.; Damiani, M.L. A Controlled Access to Spatial Data on Web. In Proceedings of the 7th Conference on Geographic Information Science, Heraklion, Greece, 12–13 November 2004; pp. 369–377.
7. Rybalov, N.B.; Zhukovsky, O.I. Access to the Spatial Data in the Web-Oriented GIS. In Proceedings of the Siberian Conference on Control and Communications, Tomsk, Russia, 20–21 April 2007; pp. 104–107.
8. Bertino, E.; Thuraisingham, B.; Gertz, M.; Damiani, M.L. Security and privacy for geospatial data: Concepts and research directions. In Proceedings of the International Workshop on Security and Privacy in GIS and LBS, Irvine, CA, USA, 4 November 2008; pp. 6–19.
9. Fuguang, M.; Yong, G.; Menglong, Y.; Fuchun, X.; Ding, L. The fine-grained security access control of spatial data. In Proceedings of the 18th International Conference on Geo-informatics, Beijing, China, 18–20 June 2010; pp. 1–4.
10. Wu, F.; Cui, W.; Chen, H. A Compound Chaos-Based Encryption Algorithm for Vector Geographic Data under Network Circumstance. In Proceedings of the Cardholder Information Security Program, Hainan, China, 27–30 May 2008; pp. 254–258.
11. Li, G. Research of Key Technologies on Encrypting Vector Spatial Data in Oracle Spatial. In Proceedings of the International Conference on Industrial Electronics and Computer Science, Wuhan, China, 25–26 December 2010; pp. 1–4.
12. Yasser, D.; Elghafar, I.A.; Tamam, A. Protecting GIS Data Using Cryptography and Digital Watermarking. *Int. J. Comput. Sci. Netw. Secur.* **2010**, *10*, 75–84.
13. Jang, B.J.; Lee, S.H.; Kwon, K.R. Perceptual Encryption with Compression for Secure Vector Map Data Processing. *J. Digit. Signal Process.* **2014**, *25*, 224–243.
14. Bang, N.V.; Lim, S.H.; Moon, K.S.; Lee, S.H.; Kwon, K.R. Selective Encryption Scheme for Vector Map Data using Chaotic Map. *J. Korea Multimed. Soc.* **2015**, *18*, 818–826.
15. RSA Laboratories. *Password-Based Cryptography Standard*; RSA Security LLC, 2006.
16. Maria, I.R. Gaussian Probability Density Functions: Properties and Error Characterization; Institute for Systems and Robotics: Lisbon, Portugal, 2004.
17. Gaussian Distribution. Available online: http://hyperphysics.phy-astr.gsu.edu/hbase/Math/gaufcn.html (accessed on 30 October 2019).
18. United Kingdom Shapefile. Available online: https://www.igismap.com/united-kingdom-shapefile-download/ (accessed on 30 October 2019).
19. White paper: ESRI Shape-File Technical Description; Environmental Systems Research Institute, USA, 1998.
20. Open GIS Geography Markup Language (GML) Encoding Standard. Available online: https://www.opengeospatial.org/standards/security (accessed on 30 October 2019).
21. Mapinför. Available online: http://www.pbinsight.com/welcome/mapinfo/ (accessed on 30 October 2019).