



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

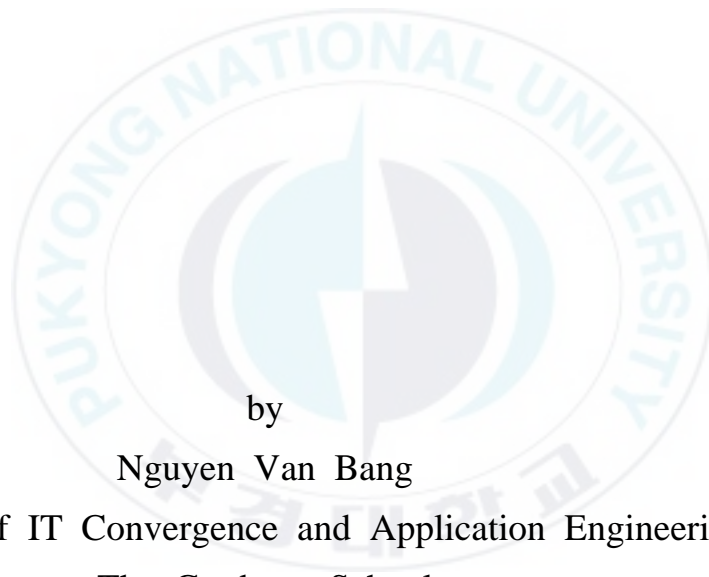
저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

Thesis for the Degree of Master of Engineering

GIS Vector Map Data Encryption
Using Vertex Randomization and
Polyline Simplification Algorithms



by

Nguyen Van Bang

Department of IT Convergence and Application Engineering

The Graduate School

Pukyong National University

August 2016

GIS Vector Map Data Encryption
Using Vertex Randomization and
Polyline Simplification Algorithms

꼭지점 랜덤화 및 폴리라인 단순화 알고리즘을
이용한 GIS 벡터 맵 데이터 암호화

Advisor: Prof. Ki-Ryong Kwon

by

Nguyen Van Bang

A thesis submitted in partial fulfillment of the requirements
for the degree of

Master of Engineering

in Department of IT Convergence and Application Engineering,
The Graduate School,
Pukyong National University

August 2016


GIS Vector Map Data Encryption Using
Vertex Randomization and Polyline Simplification Algorithms

A thesis
by
Nguyen Van Bang

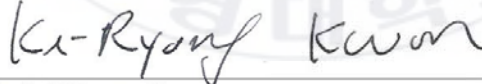
Approved by:



(Chairman) Prof. Sang-Uk Shin



(Member) Prof. Eung-Joo Lee



(Member) Prof. Ki-Ryong Kwon

August 2016

Contents

I. Introduction	1
II. Related Works	4
2.1 GIS vector map	4
2.2 Vector map security	7
2.2.1 Vector map watermarking	7
2.2.2 Vector map full encryption	8
2.2.3 Overview on selective encryption	9
2.3 Line simplification algorithms	10
2.3.1 The Ramer–Douglas–Peucker algorithm (DP)	11
2.3.2 Lang simplification (LA)	12
2.3.3 The sleeve-fitting polyline simplification algorithm (SF)	
.....	13
III. Proposed Method	15
3.1 Overview	15
3.2 Layer selection and backbone of object	16
3.3 Feature point	18
3.4 Vertices Encryption	21
3.4.1 Key values generation	21
3.4.2 Vertices encryption	22
3.5 Decryption process	23

IV. Experimental Results	25
4.1 Simulation system	25
4.2 Visualization	26
4.3 Vertices selection	32
4.4 Distance measure	33
4.5 Decryption error	34
4.6 Security evaluation	35
4.7 Algorithm comparison	37
V. Conclusion	40
References	41
Acknowledge	46

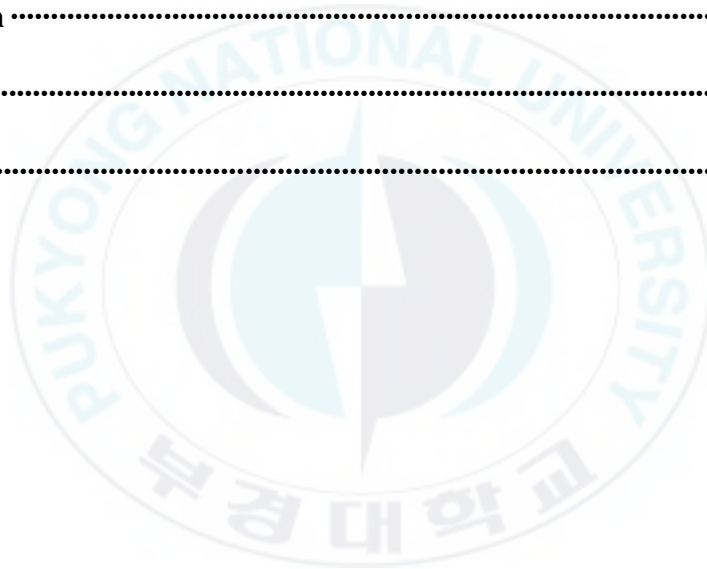


Table of Contents

Table 1: Scaling maps in experiments.	26
Table 2. The decryption error	34
Table 3. The max, min error between original map and decryption map.....	34



Figure of Contents

Figure 1. Australia map; (a) region layer, (b) road layer, (c) water layer, and (d) combine layers.	4
Figure 2. Vector data: (a) point, (b) polyline, and (c) polygon	5
Figure 3. The concept of selective encryption	9
Figure 4. The Ramer–Douglas–Peucker algorithm	12
Figure 5. The Lang simplification algorithm	13
Figure 6. The Sleeve-fitting polyline simplification algorithm	14
Figure 7. The proposed encryption process of vector map data	16
Figure 8. Example of vector map.	17
Figure 9. Object’s backbone definition: (a) object, (b) backbone $n=3$, and (c) backbone $n=6$	18
Figure 10. Breakpoints definition based on simplification using DP, SF, and LA algorithms	18
Figure 11. Process of feature points	20
Figure 12. Section simplification and positional errors	20
Figure 13. Process of key and randomization values generation.	22
Figure 14. An illustration of my proposed method: (a) original object and (b) encrypted object.	23
Figure 15. Decryption process.	24
Figure 16. (a) Block diagram object extracting process and (b) Organization of the main file	25

Figure 17. Illustrations: (a)-(b) GE original/encrypted polyline layer, (c)-(d) MX original/encrypted polygon layer, and (e)-(f) TU original/encrypted polyline and polygon layers	27
Figure 18. Illustrations: (a)-(b) FR original/encrypted layer, (c)-(d) UKR original/encrypted layers, and (e)-(f) AU original/encrypted layers	28
Figure 19. Illustrations: (a) original world map and (b) encrypted map	29
Figure 20. Illustrations: (a) Korea original map and (b) Korea encrypted map	30
Figure 21. Illustrations: (a) Viet Nam original map and (b) Viet Nam encrypted map	31
Figure 22. Ratio of breakpoints, feature points, and encrypted vertices of my method on the scale factor (Calculation with <i>TU</i> map).	32
Figure 23. Distance measure with K_1 and K_1	33
Figure 24. Key sensitivity analysis for encryption process	36
Figure 25. Key sensitivity analysis for decryption process	36
Figure 26. Ratio of encrypted data of my method and the existing methods	37
Figure 27. Computation time according to: (a) Total number of object and (b) Size of maps	38

꼭지점 랜덤화 및 폴리라인 단순화 알고리즘을 이용한 GIS 벡터 맵 데이터 암호화

Nguyen Van Bang

부 경 대 학 교 대 학 원 IT 융 합 응용 공 학 과

요 약

지리 정보 시스템(GIS)은 지표면의 위치에 관한 데이터를 캡처, 저장, 검사, 표시하기 위한 컴퓨터 시스템이다. 약어 GIS는 때때로 지리 정보 시스템을 연구하고 지오인포매틱스의 폭 넓은 학문 내에서 큰 영역인 학문을 참조하는 지리 정보 과학에 사용된다. 지금은, 가치있는 GIS 데이터 셋의 많은 양은 불법복제, 해커, 또는 인증되지 않은 사용자에게 의해 불법으로 배포하고 있다. 따라서 벡터맵 데이터들을 멀티미디어 응용 및 저장, 전송을 위해 어떻게 보호할 것인지에 대하여 집중적인 고민이 필요하다. 바로 이러한 점들 때문에 안전한 네트워킹 및 저작권 보호, 데이터 암호화에 초점을 둔 벡터맵 보안 기술이 연구 되고 있다. GIS 벡터 맵 데이터는 용량이 크고 현재까지 나온 데이터 암호화 방법은 암호화 시 데이터 전체를 암호화 하는 문제점이 있다. 이런 문제점들은 긴 암호화 시간과 높은 계산 복잡도와 데이터의 크기 증가를 야기하기도 한다.

본 논문은 정점 랜덤화 및 단순화 알고리즘 기반 벡터 맵 암호화. 인증된 사용자만이 보안된 부분에 접근할 수 있다. 제안 방법에서, 벡터 맵 내 폴리라인과 폴리곤이 선택적 암호화의 대상이 된다. 각 객체의 특이점에서 변경하기 전에 무질서 맵으로부터 생성된 랜덤 계수별로 무작위로 정의한다.

실험을 통하여 본 제안기법이 효과적이며 안전하다는 것을 확인하였다. 암호화 과정 후의 맵은 전체가 변형되어 허가 받지 못한 사용자들은 변환된 맵을 복사하거나 사용하기 위한 접근이 불가능하다. 또한 암호화된 맵은 원본과 비교하여 파일의 크기가 증가하지 않으며 정밀도의 저하 또한 발생하지 않았다.

GIS Vector Map Data Encryption Using
Vertex Randomization and Polyline Simplification Algorithms

Nguyen Van Bang

Department of IT Convergence and Application Engineering, The Graduate School,
Pukyong National University

Abstract

A geographic information system (GIS) is a large system invented to manipulate, analyze, display, capture, and storing data related to positions on Earth's surface. The acronym GIS is sometimes used for geographic information science to refer to the academic discipline that studies geographic information systems and is a large domain within the broader academic discipline of Geoinformatics. Nowadays, many GIS dataset has been distributed illegally by hackers, pirates, or unauthorized users. Therefore the problem focuses on how to protect the vector map data for storage, multimedia applications, and transmission. As a result, security techniques for vector map which aim to focus on copyright, data protection, and secure network have been researched. However, size of GIS vector map is very large and current data encryption techniques often encrypt all components of data. That means we have encrypted large amount of data lead to the long encrypting time and high complexity computation.

My thesis present selective encryption algorithm based on vertex randomization and simplification in the GIS vector map. The protected part is only accessed by authorized users. In proposed algorithm, vector map is separated to select polyline/polygon layer. I define feature points in each object and encrypt the selected vertices by key values generated from Chaotic map.

In experimental results, I verify the high efficiency visualization by low complexity, high security performance by random processes. The encrypted maps changed absolutely perception of whole maps, and illegal people cannot use them. In addition, experiments also show unique performance, decryption error approximate zero and computation time be very short.

I. Introduction

A geographical information system (GIS) [1-3] is a large system invented to manipulate, analyze, store, capture, and control the geographic information. GIS benefits organizations of all sizes and in almost every industry. There is a growing interest in and awareness of the economic and strategic value of GIS. In a GIS, vectors often express geographical features and illustrate features as geometrical shapes. Many varying phenomena can be represented by vector data. Vector map manages all kinds of the geographic information data as geometric factor, topology and metadata by vector data. Vector data reflect features of the real object in the GIS environment, we only use a small size for storing data; has a high spatial resolution and graphic representation spatial data closely like handed map; easily for making projection and coordinates transformation [4-6]. For those advantages, vector map is used in many domains, and GIS applications use vector map, have provided general users with easy access to services via mobile devices or internet access.

But, detailed vector maps require huge amounts of data, processing, storing, transmitting them poses a challenge and the maintenance of a digital map requires substantial monetary and human resources. However, any company can buy it, make illegal copies from them and distribute or sell them easily many times without taking any permission from the original GIS data provider. Moreover, applications of vector map in military domain require the high security, and must be kept away from unauthorized users. So vector map is necessary to be protected and prevent illegal duplication and distribution of it.

To solve that problem, researchers gave watermarking schemes and encryption methods focus on different domains. Looking for the recent security techniques of vector map, the network security techniques for secure transmission or storage and copyright protection of vector map data have been mainly researched [7-16]. Researchers worked data encryption based on vector map database files or data profile using the cryptography [7-14] and worked the watermarking of vector map for copyright protection [15-16]. In fact, the watermarking is only useful for identifying ownership, copyright and prevent

illegal distribution while providers desiderate unauthorized users or pirate cannot see and attack to extract the content of vector map in the most cases. Thus, the data encryption is necessary to protect vector map.

In vector map security, conventional approaches encrypt whole data, so the cryptography of data files and profiles increase complexity, and these methods take a long time. Furthermore, the decryption data often occurs loss data and it takes a long time to processing time, because authors use complex computation process on large data. It is not also flexible for various data types. Specially, database management system based on security technique is vulnerable by the conversion between data formats. Moreover, current security techniques focus on access of users via internet but the network security technique cannot preserve the security in case of data leakage on off-line or loophole exposure of network administration. So, the security technique for vector map had to preserve the security in various formats of vector map data, reduce complex computation and encrypted data volume.

To compensate for limitations in the conventional approaches described above, a new selective encryption algorithm is proposed in my paper for multimedia applications, storage and transmission for vector map security. In proposed algorithm, vector map is separated to select polyline/polygon layer. I use three simplification algorithms to define the feature points in each object. After that, I select randomly vertices of objects based on the ratio of the feature points. For generating key values, I combine SHA-512 bit hashing and 2D-Chaotic map. Finally, I randomize them by random and encrypt them by using DWT transform. In DWT domain, I select some level-k coefficients and encrypt them before IDWT. Main advantages of our algorithm are randomly vertices selection and transforming processes but it still meets requirements of security by random processes, and this algorithm can be implement to many type of vector map formats. In experimental results, I verify the high efficiency visualization by low complexity, high security performance by random processes and cryptography. In addition, experiments also show unique performance, decryption error approximate zero and computation time be very short.

The remaining parts of my thesis are organized as follows: I give a brief description

about vector map security watermarking, full encryption and selective encryption and three simplification algorithms in chapter II. Chapter III explains the proposed selective encryption in the detail. Chapter IV gives experimental results and analytics. Finally, I give a conclusion of the paper in chapter V.



II. Related Works

2.1 GIS vector map

Until now, we can separate GIS data into two categories: vector and raster forms. Vector data is a collection of layers in which the layer includes many geographical objects. These objects reflect geographical features and topography of real objects or location. A layer includes vector data which represent the world using points, polygons, and lines (polylines), as shown Fig. 1.

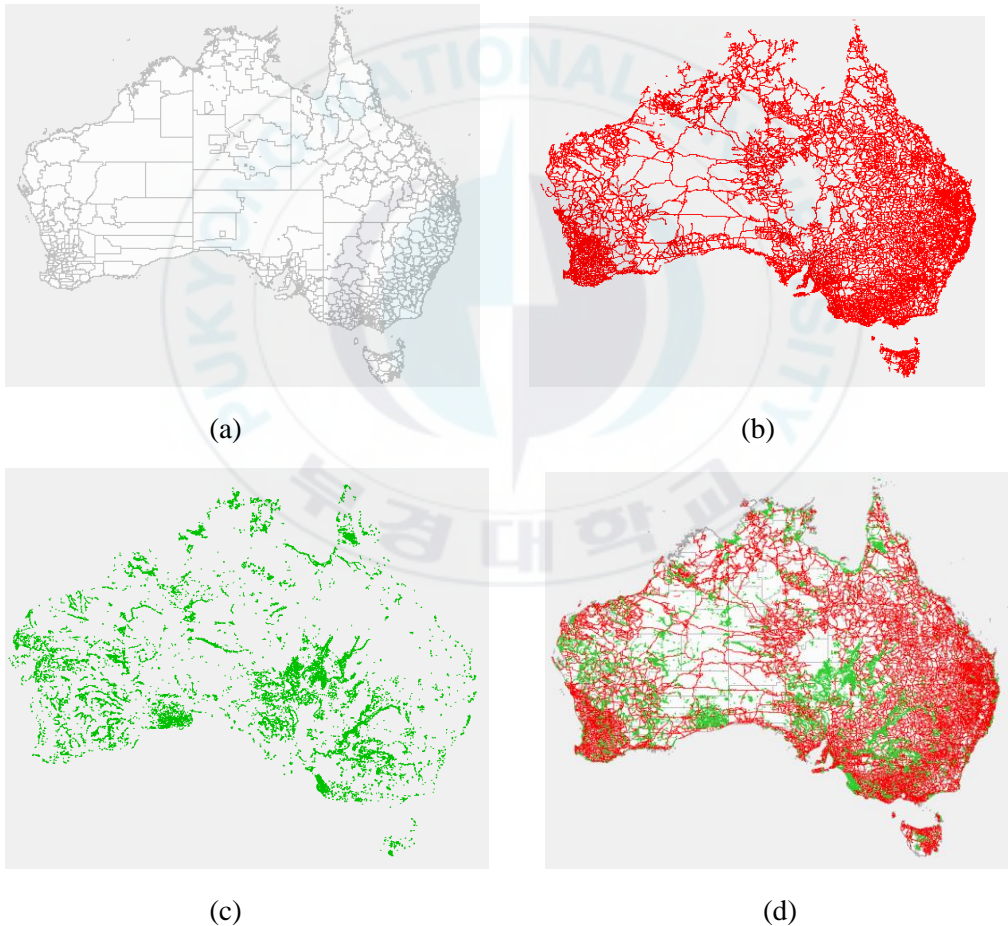


Figure 1. Australia map; (a) region layer, (b) road layer, (c) water layer, and (d) combine layers.

Points: Vector points are pair of simply XY coordinates, as shown in Fig. 2(a). In the fact, they describe peaks, wells, features of interest, and so on. Points convey the least amount of information of these file types. Points can also be used to represent areas when displayed at a small scale. For example, cities on a map of the world might be represented by points rather than polygons.

Lines or polylines: A polyline is a list of points, where line segments are drawn between consecutive points. Polylines are used to represent trails, roads, rivers, topographic lines, as shown in Fig 2(b). Again, as with point features, linear features displayed at a small scale will be represented as linear features rather than as a polygon. Line features can measure distance.

Polygons: Two-dimensional polygons are used for geographical features that cover a particular area of the earth's surface. Such characteristics include park boundaries, lakes, buildings, and so on. Polygons convey the most amount of information of the file types. Polygon features can measure the perimeter and the area. A polygon consists of one or more rings, as shown in Fig. 2(c). Features of ring are non-self-intersecting loop and closed. A polygon may include one or many rings.

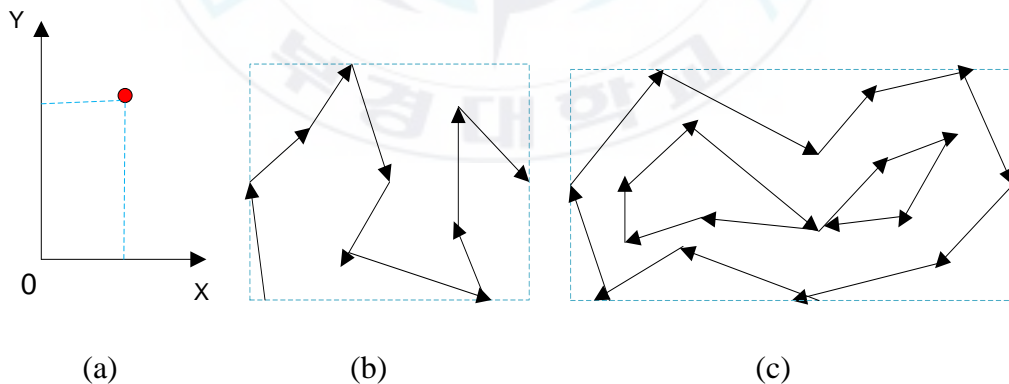


Figure 2. Vector data: (a) point, (b) polyline, and (c) polygon.

Beside geometry characteristic of vertices; vector data can store neighborhood relations or topology. Each of these geometries are linked to a row in a database that describes their attributes. For example, a database that describes lakes may contain a lake's depth, water

quality, pollution level. These information can be used to make a map to describe a particular attribute of the dataset. For example, lakes could be colored depending on the level of pollution. Different geometries can also be compared. For example, the GIS could be used to identify all wells (point geometry) that are within one kilo meter of a lake (polygon geometry) that has a high level of pollution.

There are some important advantages and disadvantages to use vector data model to represent the reality:

Advantages of Vector Data Structures:

- Data can be represented at its original resolution and form without generalization.
- Graphic output is usually more aesthetically pleasing (traditional cartographic representation).
- Since most data, e.g. hard copy maps, is in vector form no data conversion is required.
- Allows for efficient encoding of topology, and as a result more efficient operations that require topological information, e.g. proximity, network analysis.
- Data analysis is usually quick and easy to perform.

Disadvantages of Vector Data Structures:

- Continuous vector data is displayed and poorly stored.
- To get good quality, a lot of manual editing are needed. Many hard boundaries are always introduced.
- Algorithms for manipulative
- Although topology is useful for vector data, it is often processing intensive.
- Any feature edits requires updates on topology. With a lot of features, vector manipulation algorithms are complex.

2.2 Vector map security

According to the recent growth of network digital media, data are needed to protect from distribution and illegal copying. Many approaches are researched for this issues; these include authentication, encryption, and time stamping.

2.2.1 Vector map watermarking

Digital watermarking has been researched to solve the issues in vector map since 2000s. The traditional method embed secret information in some locations of vector map: [18-20] proposes certain rules, they help to select a set of coordinates of vertices, and editing them by using a certain range of precision; [21-22] modify coefficients in the frequency domain to complete watermarking hiding that is one of the important solution of watermarking algorithm, but the disturb to the vector map content is also existing, the resistance performance to data fitting, interpolation, scaling is poor. Due to the vector map high precision requirement, traditional watermarking algorithm can't meet the demand of practical application. Thus, reversible watermarking has become an interesting research topic in recent years. It is also called lossless watermarking, i.e., the original content can be completely restored when decoding. [23-24] improved the traditional difference expansion or shifting reversible watermarking algorithm. Men et al [25] proposed a reversible watermarking scheme for 2D-vector maps based on graph spectral domain, this scheme has preferable invisibility and the capacity to resist simplification attack, thus likes as a pragmatic method for copyright protection of 2D-vector maps. Zhong et al [26] propose the watermarking algorithm that is robust to prove the owner of copy data, this algorithm identifies the different between the legal users and the illegal users.

In summary, the existing methods of geospatial watermarking can be distributed as follows: algorithm uses DCT, DWT, and DFT domain, the algorithms in spatial domain, and algorithms inherited in 3D watermarking. Not similar with general multimedia data types, watermarking in vector map has its distinct features due to the application

environments and special data structures of vector data. However, watermarking isn't a foolproof way to protect them, it only identifies the rightful owner of the work, so it is the final step in security policy.

2.2.2 Vector map full encryption

In traditional digital data (image, video, map...) protection schemes, firstly we compressed the whole data. In the next step, we use standard encryption (IDEA, AES, DES, etc.) for encrypting compressed bitstream. The specific characteristics of this kind of data (high-transmission rate with limited bandwidth) make standard encryption algorithms inadequate. Another limitation of fully layered systems consists of altering the whole bit-stream syntax which may disable some codec functionalities.

The full encryption algorithms usually encrypt all components of original data to change whole data. Wu et al. [12] consider characteristics of the storage, parameters and initial values of chaotic map. After that, he proposed a new compound encryption method, this process is not available to any type formats of data and object indexing. Li et al. [13] selected the vector dataset in external Oracle DBMS for encryption, and he used standard cryptography DES combining with an R-Tree spatial index. This algorithm encrypts the spatial index when the GIS dataset is transmitted to the client and designs the key management of public and private keys on a PKI system. In this process, the key length is short so it can not keep data on the DBMS with high security. Dakroury et al. [14] also described better the encrypting algorithm, AES and RSA cryptography are combined along with watermarking method that used in internet online service. This algorithm encrypts all parts of a shape-file using 256 bit for private key of a block cipher AES. But, this algorithm uses whole shape-files for encryption and they do not consider important features, it is taking a long time to handle.

We also have many proposed methods that relate to access, transmission, management, and storage data. From 2003 to 2007, Bertino et al. [7], Chena et al. [8], and Rybalov et al. [10] presented approaches to create a mechanism for helping identify the protected data on

the Web. Similar, Bertino et al. [9], and Ma et al. [11] also presented approaches to manage GIS datasets, they control on the database or Web in 2008 and 2010. Mostly, authors explained technical challenges raised by the unique requirements of secure geospatial data management such as access control, security and privacy policies. But access control and management on Web or database do not maintain security in the outflow of an authenticated user. Relating storage and transmission of vector map, data is encrypted before storing and transmitting. The aim of encryption is to change data by encrypting algorithms using keys, and make an unauthorized user not to access or unable to read the content of data. It is also called full encryption.

2.2.3 Overview on selective encryption

In digital data protection topic, selective encryption methods have been researched since 2000s. The simple concept of selective encryption is showed in Figure 3. It consists of encrypting only a subset of the data. The aim of selective encryption is to reduce the amount of data to encrypt while preserving a sufficient level of security. This computation saving is very desirable especially in constrained communications (real-time networking, high-definition delivery, and mobile communications with limited computational power devices).

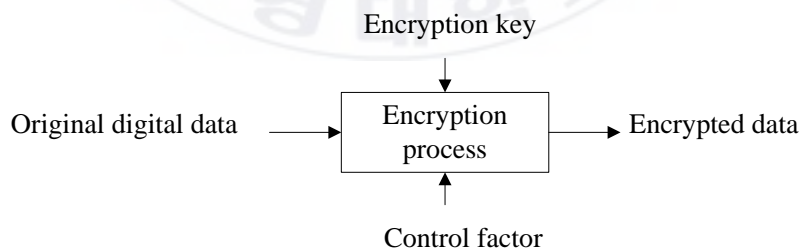


Figure 3. The concept of selective encryption.

In addition, selective encryption allows preserving some codec functionalities such as scalability. The general approach is to separate the content into two parts. The first part is the public part, it is left unencrypted and made accessible to all users. The second part is the protected part; it is encrypted. Only authorized users have access to protected part. One

important feature in selective encryption is to make the protected part as small as possible. The control factor (threshold value) helps to control encrypted data ratio in algorithm, by this way we can make security better. In the main, perceptual process is created based on partial encryption algorithms.

Nowadays, many selective encryption algorithms are proposed to video and image [34]: Shi et al. [35-36] used secret key or DES for selecting DC coefficients and encrypt sign bits of them, it help to reduce encryption bits ratio in image. Podesser et al. [37] propose encryption algorithm using AES standard, this method selects bitplanes and at least 3 bitplanes over 8 of the bitstream have to be encrypted. Pommer and Uhl [38] encrypted the header information of wavelet packet encoding of an image, but statistical properties of wavelet coefficients are preserved by the encryption, then the approximation subband can be reconstructed. However, all algorithms used DES, AES or other symmetric-key algorithms become quite simple algebraic structure and not high security.

In frequency domain: Zeng and Lei [39] proposed combine 8*8 DCT and wavelet transform, they consist of randomly scrambling by using different primitives. Cheng and Li [40] only encrypted “important part” defined compression algorithm by using wavelet transform for SPIHT partitions.

For vector map perceptual encryption, B.-J Jang et al. [17] presented a method, which encrypts parameters mean point and direction of mini coding objects in compression domain by XOR operator. The aim of this method is to select parameters after compression to encrypt by XOR operator. Giao et al. [33] select all vertices in a complex layer and they randomize them before to encrypt in DCT domain, they did not consider the important part in each layer. Bang et al. [43-45] select randomly some vertices and encrypt them, it seems quite easily to attack, because they choose vertices based on simple method leads to the algorithm is so weak. For these reasons, I propose a new algorithm in selective encryption topic for multimedia applications, transmission and storage of vector map data.

2.3 Line simplification algorithms

Line simplification methods have been researched in long time because simplification is necessary to reduce map data redundancy while preserving the geographic features. Researchers have worked on vector map simplification for decades. In my paper, I used some simplification algorithms to select randomly vertices for encryption. So, I only introduce three popular algorithms in this part.

2.3.1 The Ramer–Douglas–Peucker algorithm (DP)

The Ramer–Douglas–Peucker algorithm (RDP) (Fig. 4) is an algorithm for reducing the number of points in a curve that is approximated by a series of points. The initial form of the algorithm was independently suggested in 1972 by Urs Ramer and in 1973 by David Douglas and Thomas Peucker and several others in the following decade [32]. This algorithm is also known under the names Douglas–Peucker algorithm, iterative end-point fit algorithm and split-and-merge algorithm.

The purpose of the algorithm is, give a curve composed of line segments, to find a similar curve with fewer points. The algorithm defines 'dissimilar' based on the maximum distance between the original curve and the simplified curve (i.e., the Hausdorff distance between the curves). The simplified curve consists of a subset of the points that defined the original curve.

The starting curve is an ordered set of points or lines and the distance dimension $\epsilon > 0$. The algorithm recursively divides the line. Initially it is given all the points between the first and last point. It automatically marks the first and last point to be kept. It then finds the point that is furthest from the line segment with the first and last points as end points; this point is obviously the farthest on the curve from the approximating line segment between the end points. If the point is closer than ϵ to the line segment then any point is not currently marked to be kept can be discarded without the simplified curve being worse than ϵ .

If the point farthest from the line segment is greater than ϵ from the approximation then that point must be kept. The algorithm recursively calls itself with the first point and the

worst point and then with the worst point and the last point, which includes marking the worst point being marked as kept. When the recursion is completed a new output curve can be generated consisting of all and only those points that have been marked as kept.

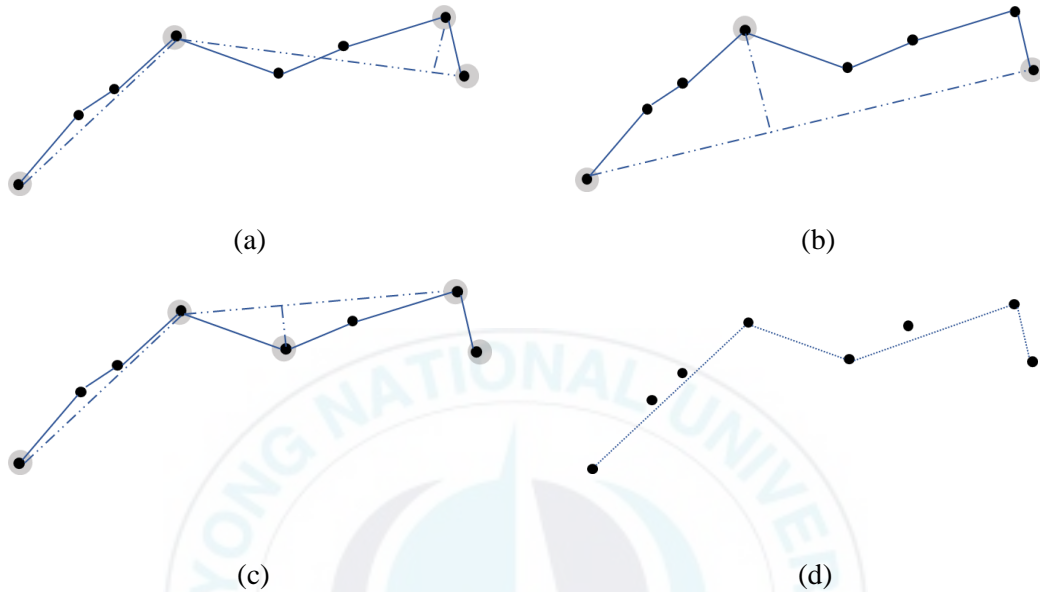


Figure 4. The Ramer–Douglas–Peucker algorithm.

2.3.2 Lang simplification (LA)

Lang algorithm (Fig. 5) is a simplification method which aim to remain the main shape of a polyline or curve by reducing the total of points. In this methods, user defined the search region and the distance threshold from a segment connecting two original points to the original points between them. In this algorithm, we initiate a search area as an area containing a given number of original points. The perpendicular distances from the segment to the intermediate points are calculated and if the calculated distance is larger than the user defined tolerance value, the search region is shrunk by excluding its last point and the distances are calculated again. This process will continue until all the calculated perpendicular distances from intermediate points are below the user defined tolerance value, or until there are no more intermediate points. Once all the intermediate points are

moved, a new search region is defined by stating at last point of the latest (or most recent) search area.

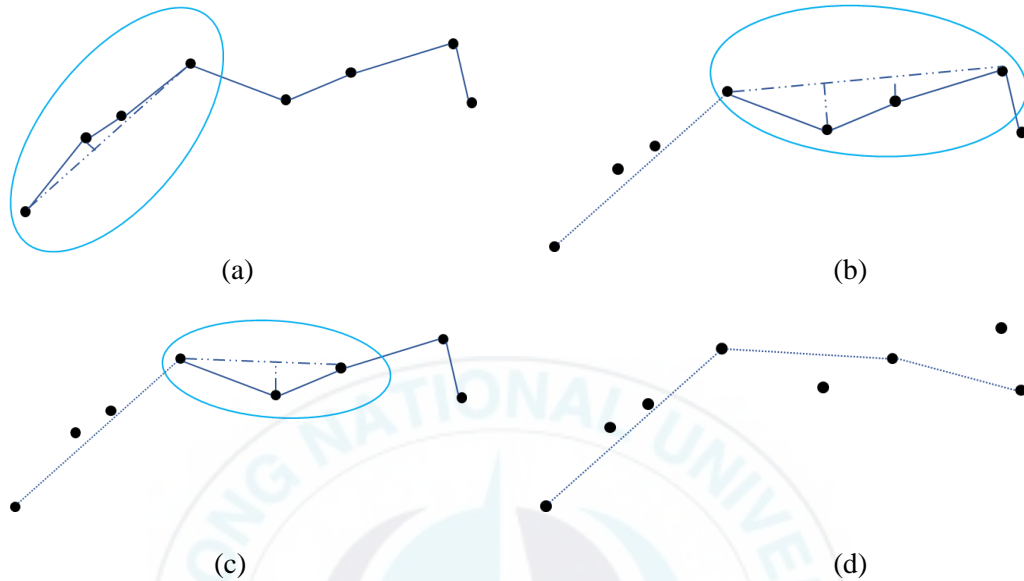


Figure 5. The Lang simplification algorithm.

2.3.3 The sleeve-fitting polyline simplification algorithm (SF)

Proposed by Zhao and Saalfeld (1997) is another unconstrained extended local processing routine evaluated in this paper. This algorithm is similar to the Reumann-Witkam routine because the original line is divided into sections by using a rectangle (or called the sleeve in Zhao and Saalfeld's model). Figure 6 shows sleeves of a user-defined width. Each sleeve starts at an original point P_j and contains consecutive original points $\{P_{j+1}, P_{j+2}, \dots, P_{j+k}\}$, the direction of which is parallel to the line connecting the starting P_j and the last original points P_{j+k} of the sleeve. For the first sleeve, its starting original point is the starting endpoint P_1 of the original line and its last original point is the original point P_s with the largest sub-index such that all original points between P_1 and P_s are inside this sleeve. The second sleeve starts at the last original point P_s of the first sleeve and ends at the original

point P_t with the largest sub-index t such that the sleeve contains all original points between P_s and P_t . The sleeve is moved over the original line into the direction of the line connecting the starting and the last original points of each sleeve. The centerline of each sleeve is a simplified segment representing those original segments within the sleeve.

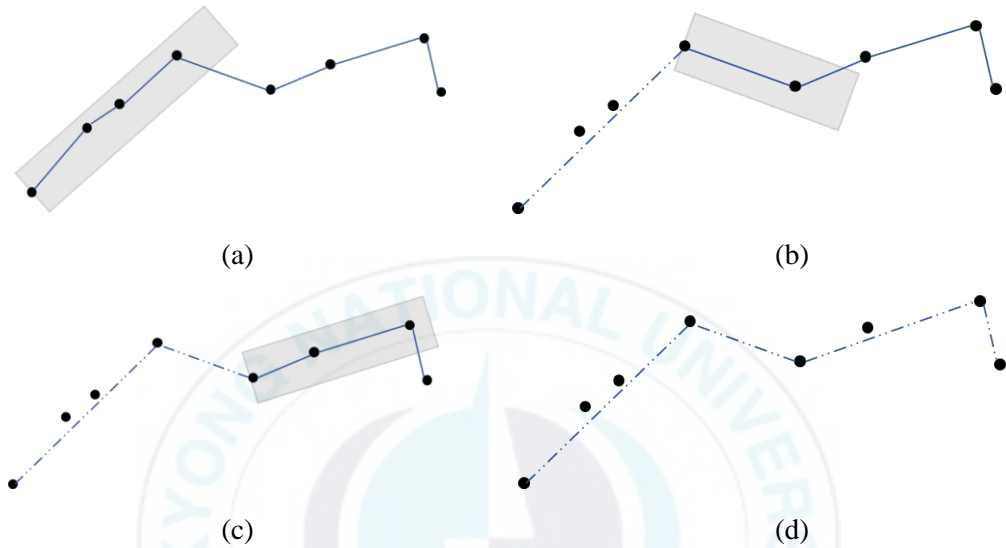


Figure 6. The sleeve-fitting polyline simplification algorithm.

III. Proposed Method

3.1 Overview

In my thesis, the proposed method selects randomly vertices for encryption based on simplification algorithms. In figure 7, I show the schematic of my algorithm, and the step-by-step is explained in detail, as follows:

- A vector map \mathbf{M} is defined as a set of layers: $\mathbf{M} = \{L_i | i \in [1, |\mathbf{M}|]\}$ with $|\mathbf{M}|$: the cardinality in map \mathbf{M} (the cardinality of a set is a measure of the "number of elements of the set" in mathematics).
- A layer L_i include many polyline/polygon objects: $L_i = \{O_{ij} | j \in [1, |L_i|]\}$ with $|L_i|$ is the total number of objects in layer and $|L_i|$ is also the cardinality in a layer L_i .
- An object O_{ij} is a set of vertices $O_{ij} = \{v_{ijk} | k \in [1, |O_{ij}|]\}$ with $|O_{ij}|$ is the number pair coordinates of vertices in object O_{ij} . Then, I define a backbone of the object (This concept will explain the detail in part 3.2).
- My method processes all the backbones in layers using DP, SF, and LA algorithms to define breakpoints of objects. After that, I analyze the positional errors of three exist algorithms to identify feature points.
- Finally, I select vertices based on the ratio of the feature points; generating randomization values and key values using SHA-512 hashing combining with 2D-Chaotic-map; and then I encrypt them in DWT domain.

The detail of steps is illustrated in continuous sections.

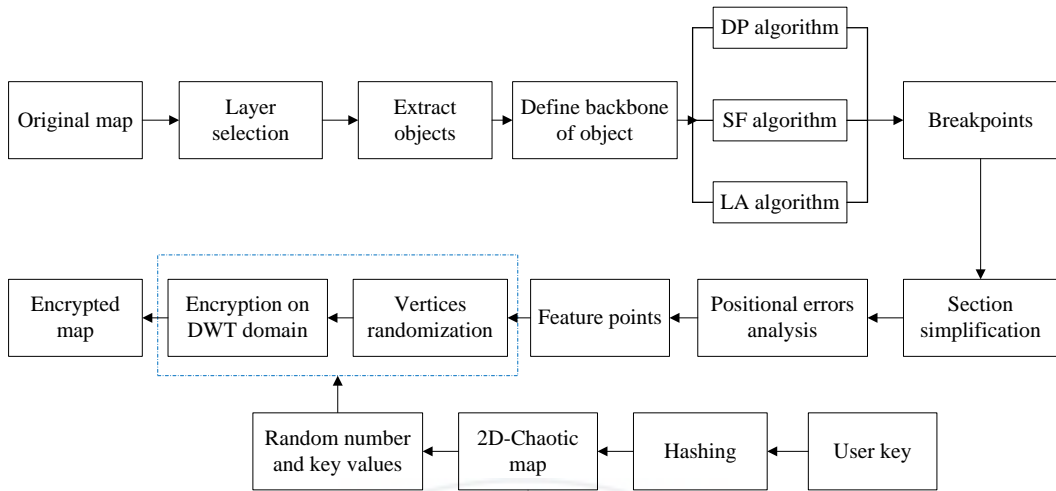


Figure 7. The proposed encryption process of vector map data.

3.2 Layer selection and backbone of object

The vector map data includes layers and each layer includes geometry feature which is illustrated by geometric objects, and attribute features, as shown in Fig. 8. Text and annotation stored, display information and attributes. The geographic information used a set of coordinates to provide position and shapes of objects as point, polyline and polygon. Thus, I consider the content of layers in a map consists of two parts. The first part includes insignificant information as text and annotation. The second part is the geometric data needed to protect, includes geometric objects as point, polyline and polygon. Because, text and annotation do not contain geometric features and do not determine the shape of map, so they are not targets for vector map security. Moreover, the point only uses a pair of coordinates to represent simple, small and zero-dimensional objects in the real on the map while polylines and polygons use a set of coordinate to represent complex structure and huge objects. Therefore, polylines and polygons are targets to select for vector map security, and in my algorithm I perform selective encryption for polylines and polygons.

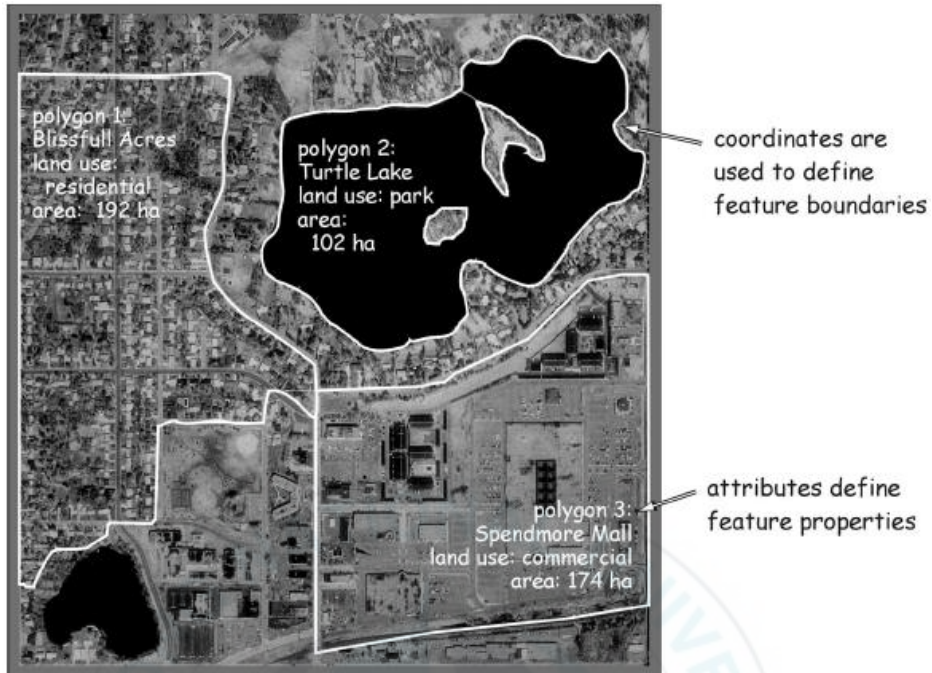
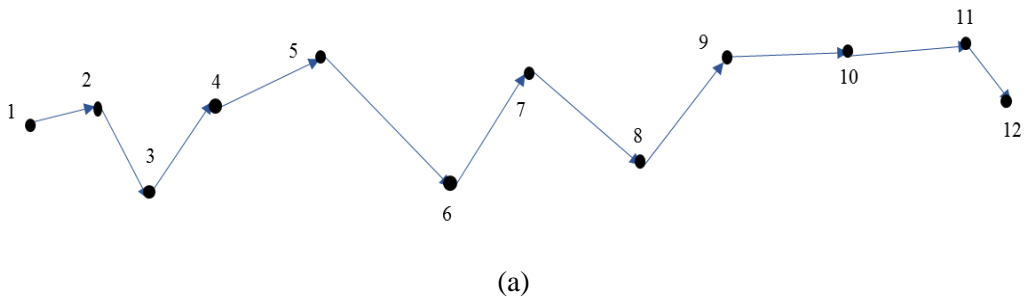
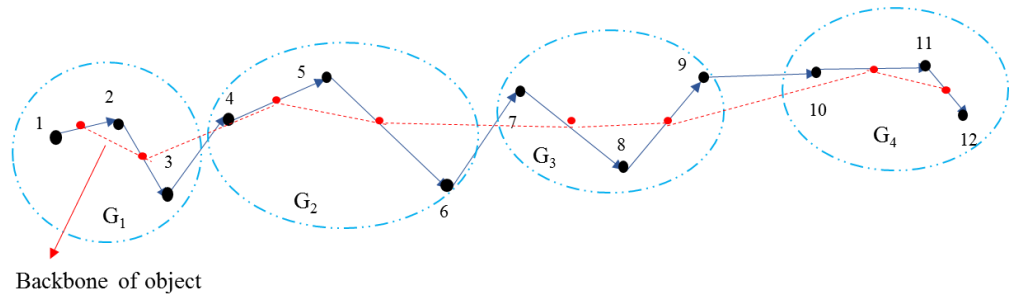


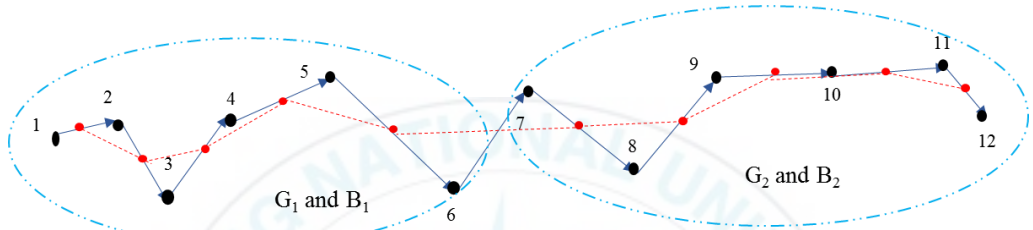
Figure 8. Example of vector map.

As you know, a layer is a set of objects and an object \mathbf{O}_{ij} is a set of vertices $\mathbf{O}_{ij} = \{v_{ijk} | k \in [1, |\mathbf{O}_{ij}|]\}$ with $|\mathbf{O}_{ij}|$ is the cardinality of the object. I separate object to m groups $\mathbf{G} = \{G_k | k \in [1, m]\}$, $m = |\mathbf{O}_{ij}|/n$ with n : the total of vertices in one group. Object's backbone is set of vertices with a vertex is the average point between two continuous vertices in a group, as shown in Fig. 9. Therefore, backbone $\mathbf{B} = \{B_k | k \in [1, m]\}$ with B_k is a backbone section in group G_k .





(b)



(c)

Figure 9. Object's backbone definition: (a) object, (b) backbone $n=3$, and (c) backbone $n=6$.

3.3 Feature point

I use the scale factor α for controlling the simplification quantity and for generating breakpoints with different ratios. Thus, I set parameters for the three algorithms using the scale factor α and initial parameters:

$$\delta = \delta_0 * (1 + \alpha) \text{ for DP} \quad (1)$$

$$\varepsilon = \varepsilon_0 * (1 + \alpha) \text{ for SF} \quad (2)$$

$$\gamma = \gamma_0 * (1 + \alpha) \text{ for LA} \quad (3)$$

where δ_0 , ε_0 , and γ_0 are the initial distance threshold for DP, SF, and LA. The scale factor α controls the number of breakpoints. If $\alpha = -1$, a backbone is not simplified. If α increases from -1, the total number of breakpoints also increase.

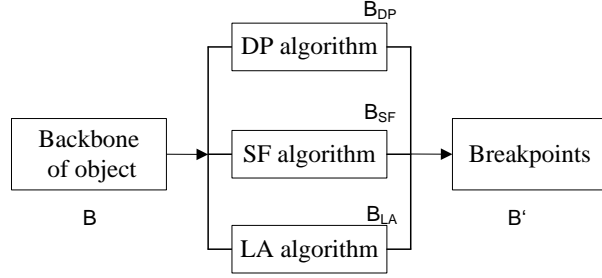


Figure 10. Breakpoints definition based on simplification using DP, SF, and LA algorithms.

Given a backbone \mathbf{B} of object, let me consider that \mathbf{B} consists of N vertices, $\mathbf{B} = \{v_{b1}, v_{b2}, \dots, v_{bN}\}$, and a vertex has two coordinate values $v_{bi} = \{x_{bi}, y_{bi}\}$. We generate three simplified backbone of backbone \mathbf{B} using the DP, SF, and LA algorithms.

$$\mathbf{B}_{DP} = DP(\mathbf{B}, \alpha) = \{v_{bk}^D \in \mathbf{B} | k \in [1, N_{DP}]\} \quad (4)$$

$$\mathbf{B}_{SF} = SF(\mathbf{B}, \alpha) = \{v_{bk}^S \in \mathbf{B} | k \in [1, N_{SF}]\} \quad (5)$$

$$\mathbf{B}_{LA} = LA(\mathbf{B}, \alpha) = \{v_{bk}^L \in \mathbf{B} | k \in [1, N_{LA}]\} \quad (6)$$

where N_{DP} , N_{SF} , and N_{LA} are the number of vertices in simplified backbone \mathbf{B}_{DP} , \mathbf{B}_{SF} , and \mathbf{B}_{LA} respectively. Their start vertices are the same as the start of \mathbf{B} : $v_{b1}^D = v_{b1}^S = v_{b1}^L = v_{b1}$ and their end are also the same: $v_{bN_{DP}}^D = v_{bN_{SF}}^S = v_{bN_{LA}}^L = v_{bN}$

As shown in Fig. 10, I define breakpoints \mathbf{B}' for a backbone \mathbf{B} as vertices that exist in three simplified backbone \mathbf{B}_{DP} , \mathbf{B}_{SF} , and \mathbf{B}_{LA} .

$$\mathbf{B}' = \mathbf{B}_{DP} \cap \mathbf{B}_{SF} \cap \mathbf{B}_{LA} = \{v'_{bk} | k \in [1, N_B]\} \quad (7)$$

Thus, breakpoints are common vertices in three simplified backbone of a backbone. This means that they represent the main shape of the polyline and should be kept simplified. N_B is the number of vertices in the breakpoints.

After that, I use the breakpoints to define feature points in the backbone, as shown in Fig. 11.

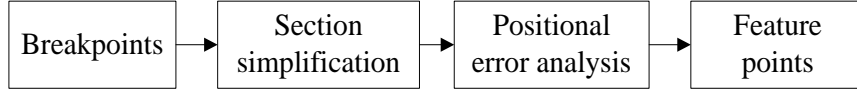


Figure 11. Process of feature points.

A backbone \mathbf{B} is segmented to N_B-1 sections $\mathbf{S} = \{\mathbf{S}_k | k \in [1, N_B - 1]\}$ using N_B breakpoints. A section \mathbf{S}_k consists of the k th breakpoint v'_{bk} , $k+1$ th breakpoint $v'_{b_{k+1}}$, and middle vertices between two breakpoints.

$$\mathbf{S}_k = \{v'_{bk}, v_{bi+1}, \dots, v_{b(i+N_k-1)}, v'_{b_{k+1}} | N_k \geq 1\} \quad (8)$$

where $v'_{bk} = v_{bi}$ and $v_{b_{k+1}} = v_{bi+N_k}$

The three existing algorithms are applied to each section to find the simplified sections of backbone, as shown in Fig. 12. And then, I calculate the positional errors that produces by three algorithms at each vertex (For each original point, the positional error is calculated as the perpendicular difference between that point and the corresponding line segment of the simplification. *Better performing simplification algorithms consistently produce lower positional errors*). From the above error values, I choose the best simplification algorithm for a section based on comparing the total of the positional errors among three algorithms.

For this reason, my method simplified a backbone \mathbf{B} as a unit of a segmented section using the DP, SF, and LA algorithms.

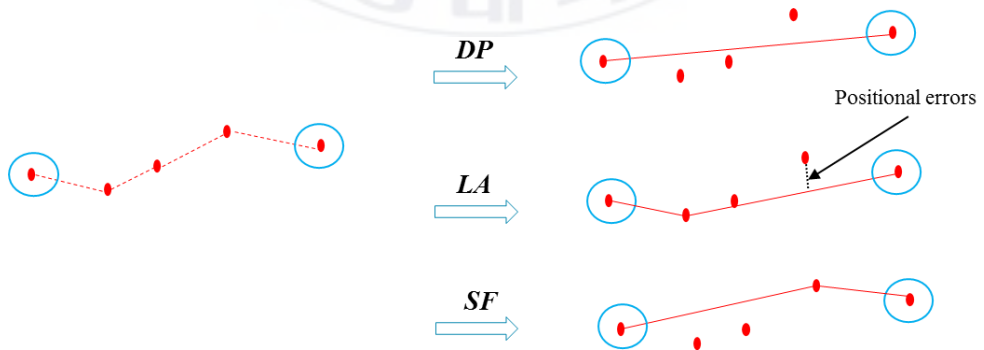


Figure 12. Section simplification and positional errors.

Finally, my method generate feature points of object's backbone by simplifying all segmented section (that is created from breakpoints) of backbone $\mathbf{B} = \{v_{b1}, v_{b2}, \dots, v_{bN}\}$ through the simplification algorithm that is assigned to each segmented section.

$$\mathbf{F} = \{v_{f1}, v_{f2}, \dots, v_{fM}\}$$

where $v_{b1} = v_{f1}, v_{bN} = v_{fM}$, M is a total of feature points and $M < N$. \mathbf{F} is a subset of \mathbf{B} ($\mathbf{F} \subseteq \mathbf{B}$). One exception with no feature points can exist through the occurrence probabilities of them are very low. That happen when all vertices except for the start and end points are deleted by *DP*, *SF*, and *LA* algorithms when the scale factor is very high or the backbone is very flat.

3.4 Vertices Encryption

3.4.1 Key values generation

In mathematics, the two-dimensional coupled Chaotic (Logistic) map [29] is a map that exhibits some sort of chaotic behavior, by following:

$$x_{n+1} = \mu_1 x_n (1 - x_n) + \gamma_1 y_n^2 \quad (9)$$

$$y_{n+1} = \mu_2 y_n (1 - y_n) + \gamma_2 (x_n^2 + x_n y_n) \quad (10)$$

Three quadratic coupling terms are introduced to strengthen the complexity of 2D Logistic map. System is chaotic when $2.75 < \mu_1 \leq 3.4, 2.7 < \mu_2 \leq 3.45, 0.15 < \gamma_1 \leq 0.21, 0.13 < \gamma_2 \leq 0.15, k=0, 1, 2, \dots$ and all the values of $\{x_i, y_i\}$ appear in the range $[0,1]$ for the initial value $x_0, y_0 \in [0,1]$.

I use the key \mathbf{H}_i to create randomization values and key values in a layer \mathbf{L}_i . These values are created randomly the first values by SHA-512 process using a key given by user [27], 512 bit key size. And other values are generated by using 2D- Chaotic map. The Fig. 13 shows a key process:

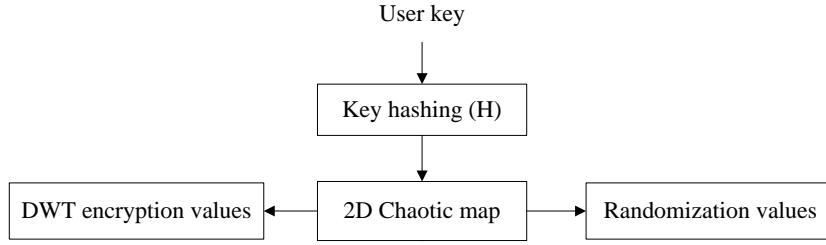


Figure 13. Process of key and randomization values generation.

The DWT encryption values $\mathbf{E}_i = \{e_{ik} | k \in [1, n]\}$ and the randomization values $\mathbf{R}_i = \{r_{ik} | k \in [1, M_i^{max}]\}$, M_i^{max} is the maximum number of group of an object among objects in the layer L_i .

3.4.2 Vertices encryption

At last, I use the feature points and the results in part 3.4.1 to select randomly vertices of object and encrypt them in DWT domain. The step-by-step procedure is explained hereafter.

- Original object \mathbf{O} is set of vertices: $\mathbf{O} = \{v_i | i \in [1, |\mathbf{O}|]\}$ with $|\mathbf{O}|$ is the cardinality of the object \mathbf{O} , $|\mathbf{O}|$ also is the total number of vertices of object \mathbf{O} .
- I separate \mathbf{O} to m groups $\mathbf{G} = \{G_k | k \in [1, m]\}$ (detail in part 3.2) with $G_k = \{v_{n*k+1}, v_{n*k+2}, \dots, v_{n*(k+1)}\}$, n (user defined) is the number vertices in group.
- Backbone \mathbf{B} of object \mathbf{O} is set of vertices: $\mathbf{B} = \{v_{bi} | i \in [1, |\mathbf{B}|]\}$ and $|\mathbf{B}| = m * (n - 1)$, because I have $(n-1)$ vertices belong backbone \mathbf{B} in each group.
- Feature points \mathbf{F} of \mathbf{O} is set of vertices: $\mathbf{F} = \{v_{fi} | i \in [1, |\mathbf{F}|]\}$, $\mathbf{F} \subseteq \mathbf{B}$, and \mathbf{F} is identified by using three simplification algorithms (detail in part 3.3).8
- In group $G_k = \{v_{1G_k}, \dots, v_{kG_k}\}$, if I have more than a half vertices of backbone section \mathbf{B}_k defined as feature points, that means $|F_k| > |B_k|/2$ ($|F_k|$ is the number of feature points in group G_k), I select all vertices in group G_k to change their position, as follow:
 - First, I select all the start vertex in the selected groups: $F = \{v_{1G_k}, \dots, v_{1G_i}\}$.

- Second, I randomize these vertices by multiplying them using random coefficients R_i , as equation:

$$F^R = \{v_{1G_k} * r_k | k \in [1, |F|]\} \quad (11)$$

- Third, the randomized values are applied DWT-k level to get DWT coefficients, k is determined by using the floor function ($\lfloor x \rfloor$, also called the greatest integer value or integer function, gives the largest integer equal or less than x) and $|F|$, as follows: $k = \lfloor \log_2 \sqrt{|F|} \rfloor$. In DWT domain, I encrypt level k coefficients by multiplying with encryption values and inverse DWT-k level. And so, I have encrypted points from set F : $F' = \{v'_{1G_k}, \dots, v'_{1G_i}\}$
- Lastly, other vertices in the selected group G_k are changed their position by equation:

$$v'_{iG_k} = v_{iG_k} + (-1)^{i-1} * (v'_{1G_k} - v_{1G_k}) \quad (12)$$

where $i \in [2, n]$

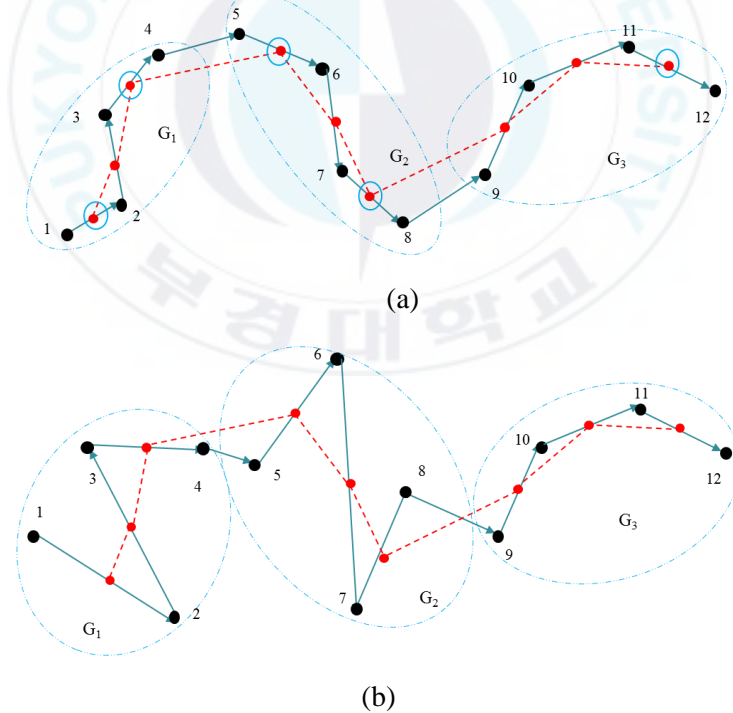


Figure 14. An illustration of my proposed method: (a) original object and (b) encrypted object.

3.5 Decryption scheme

To perform decryption, after I define object's backbone and feature points, I use the decryption block to decrypt the selected vertices of object, as shown in Fig. 15.

If the user has the correct key when he decrypts the encrypted map, and the decoded data should be like a transcript of the input map. The output map is very various if the user provides an incorrect key. I will verify the high efficiency visualization by low complexity, high security performance in the next section. In addition, experiments also show a unique performance, decryption error approximate zero and computation time faster than the existing algorithms.

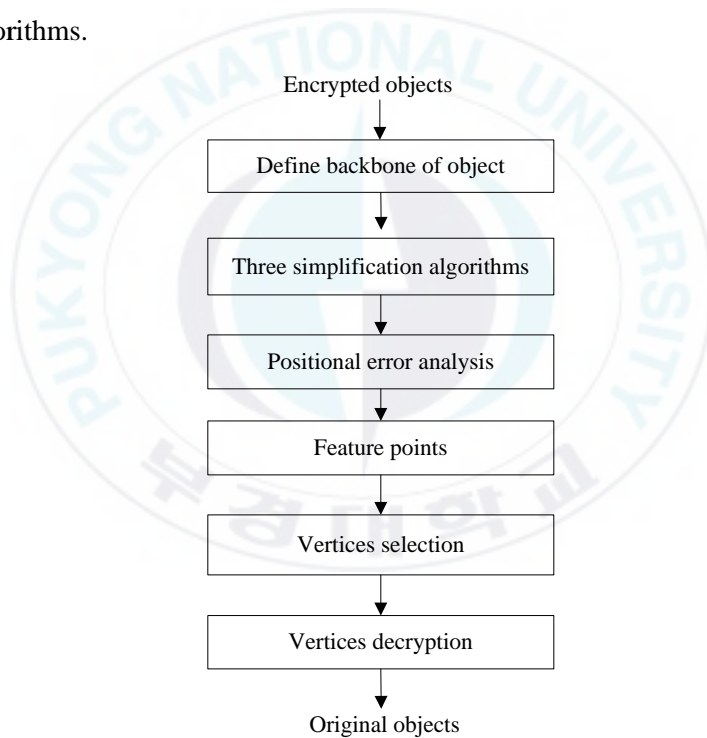


Figure 15. Decryption process.

IV. Experimental Results

4.1 Simulation system

System configuration: System manufacturer Dell Inc, Processor Intel i7-540 @ 3.4 GHz, 8 CPU, Memory 8192 MB Ram, Graphic card NIVIDA GeForce GT 640, Graphic memory 4045 MB, Window 7 professional 64 bit.

Coding environment: Microsoft Visual studio 2010 C#, Additional Library: Easy GIS .NET (Support edit, create and load shapefile).

System implementation: I used shapefile [30] consists of a dBase data, an index data, and a main file to simulate our algorithm. The main file contains a fixed-length file header followed by variable-length records. As shown in Fig. 16(a)-(b), firstly I separated record contents from main file (.shp) and extract polyline/polygon objects. Then, objects are encrypted step by step.

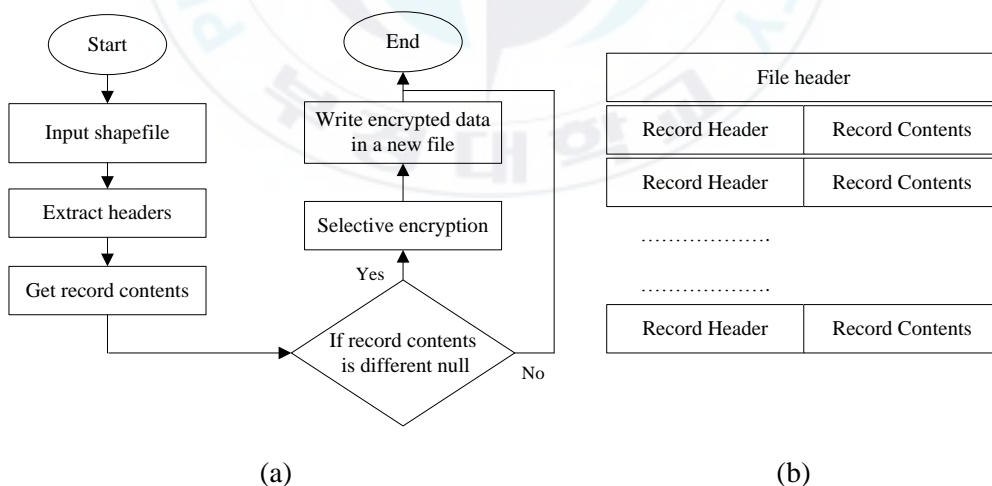


Figure 16. (a) Block diagram object extracting process and (b) Organization of the main file.

4.2 Visualization

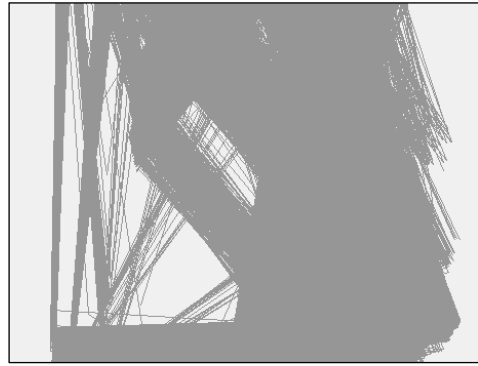
For performance evaluation of my encryption algorithm, I used many different scaled maps that contain layers, including rivers, roads, lakes, and countries in the WorldSimulation results prove that the encrypted map change absolutely perception of whole maps, as shown in Fig. 17-21. The proposed method changes the whole visual image of map because I encrypted randomly some vertices lead to scrambling shape of all objects. In addition, my method changes only vertices position of objects in the layer. The size of input and output map are the same, so it is not a lose data. I used some scaling map in visualization, the details in table 1.

Table 1: Vector maps in experiments.

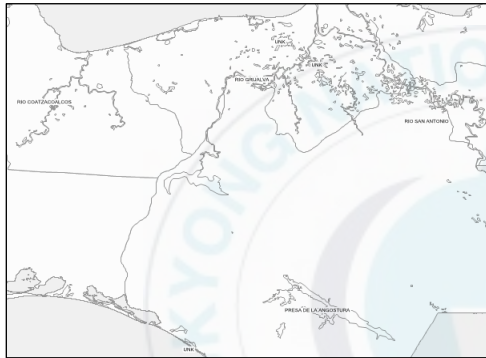
Test map	Layers	Total of objects	Size (kb)	Scale
Germany (<i>GE</i>)	Roads	35	257	1:5000
Mexico (<i>MX</i>)	Areas and lakes	47	152	1:5000
Turkey (<i>TU</i>)	Water lines	65	328	1:5000
France (<i>FR</i>)	Roads	16,915	3,202	1:10000
Ukraine (<i>UKR</i>)	Island water and areas	8,125	4,402	1:10000
Australia (<i>AU</i>)	Areas, roads, rivers, and railroads	45,256	10,658	1:15000
World map (<i>W</i>)	Countries	5,652	30,678	1:50000
Korea map (<i>KR</i>)	Full map	85,628	45,523	1:8000
Viet Nam (<i>VN</i>)	Full map	73,628	40,254	1:8000



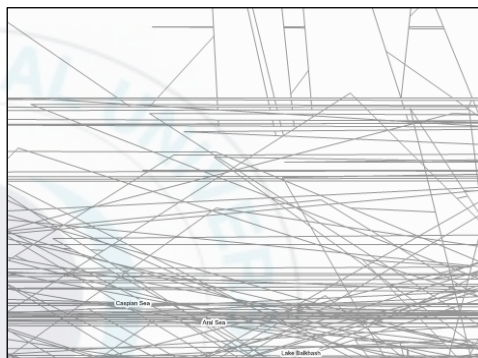
(a)



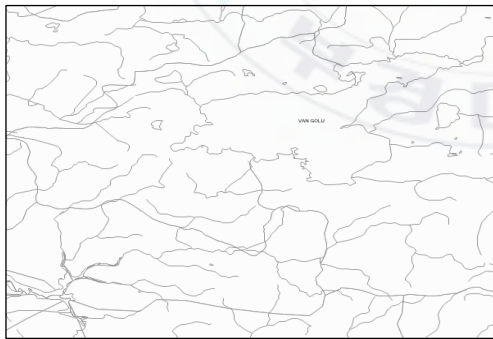
(b)



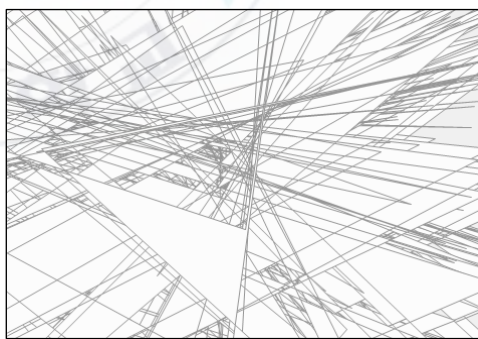
(c)



(d)

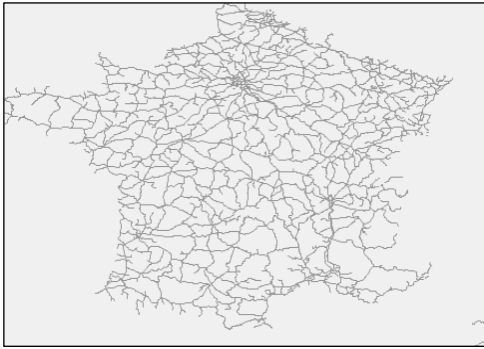


(e)

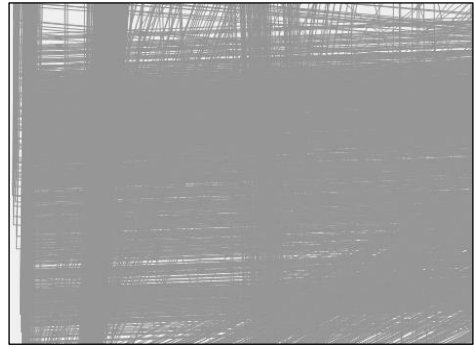


(f)

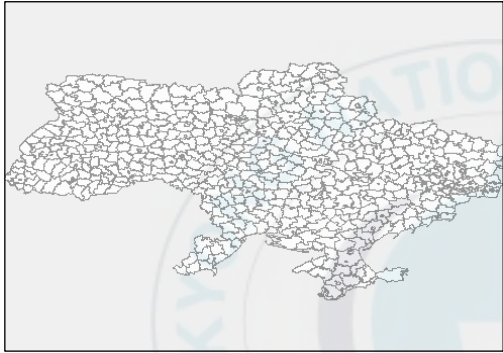
Figure 17. Illustrations: (a)-(b) **GE** original/encrypted polyline layer, (c)-(d) **MX** original/encrypted polygon layer, and (e)-(f) **TU** original/encrypted polyline and polygon layers.



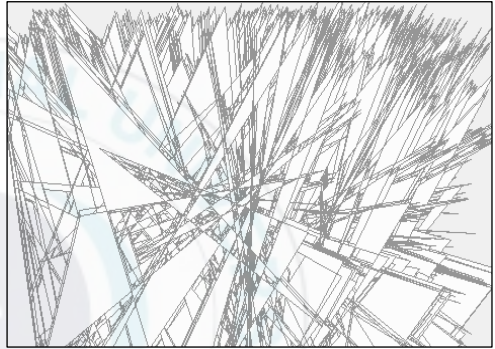
(a)



(b)



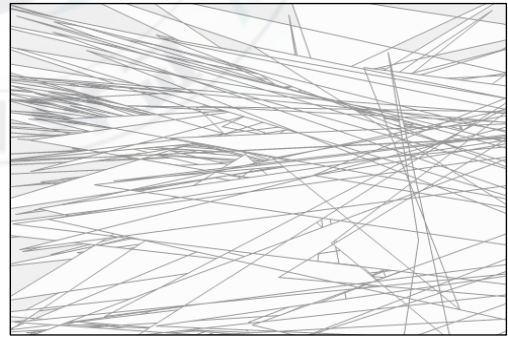
(c)



(d)



(e)

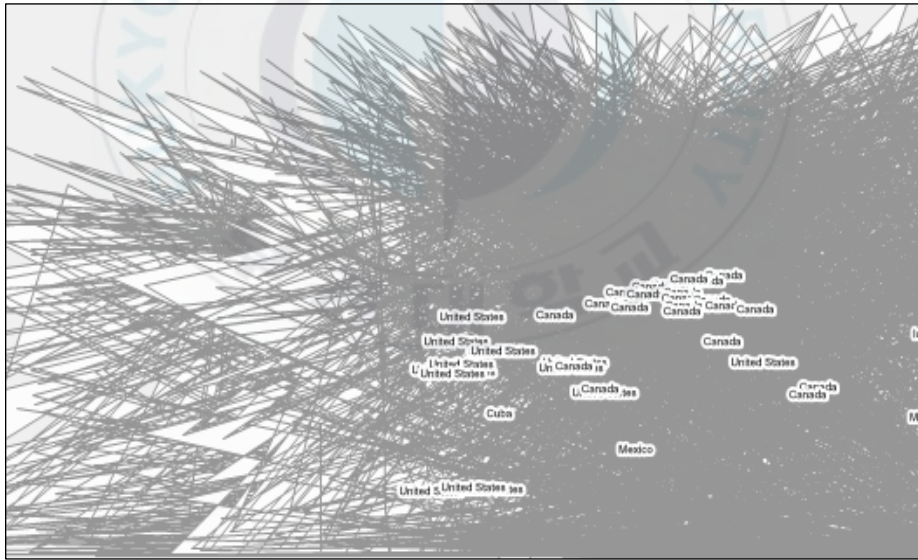


(f)

Figure 18. Illustrations: (a)-(b) **FR** original/encrypted layer, (c)-(d) **UKR** original/encrypted layers, and (e)-(f) **AU** original/encrypted layers.

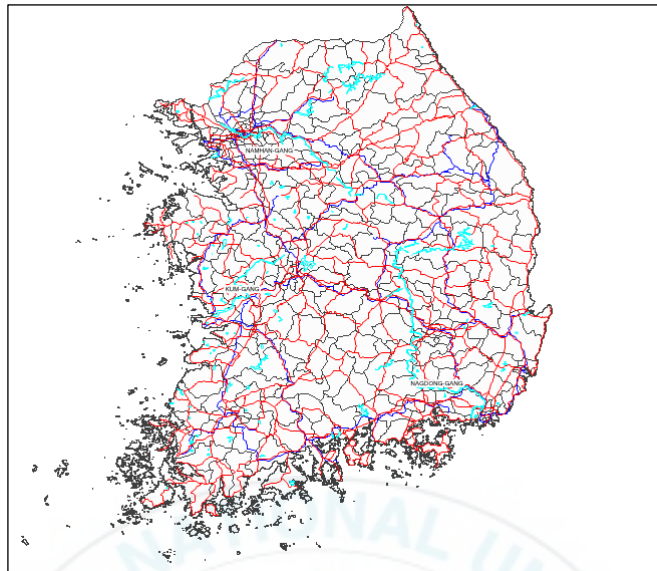


(a)

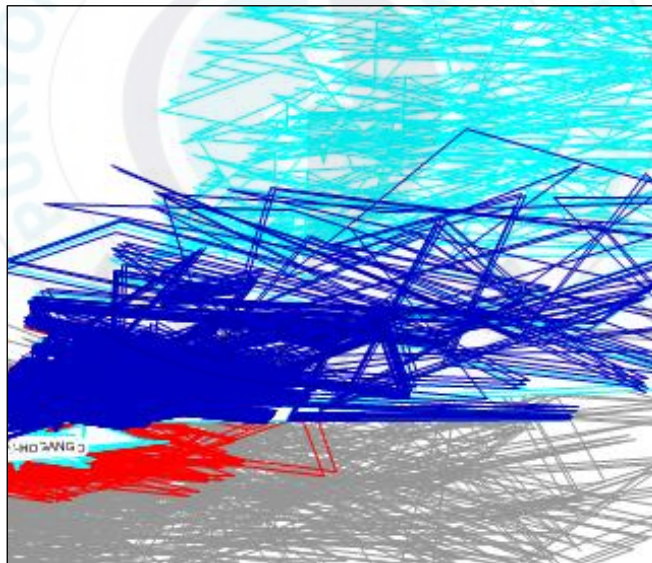


(b)

Figure 19. Illustrations: (a) original world map and (b) encrypted map.

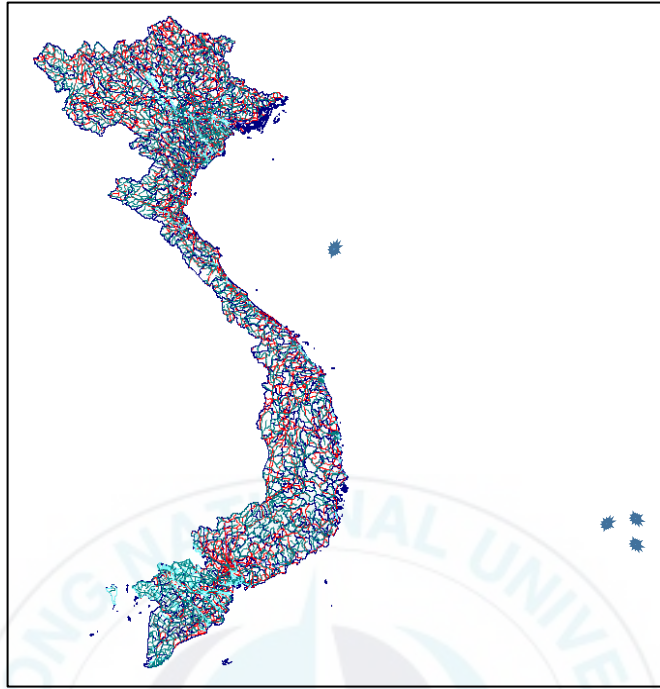


(a)

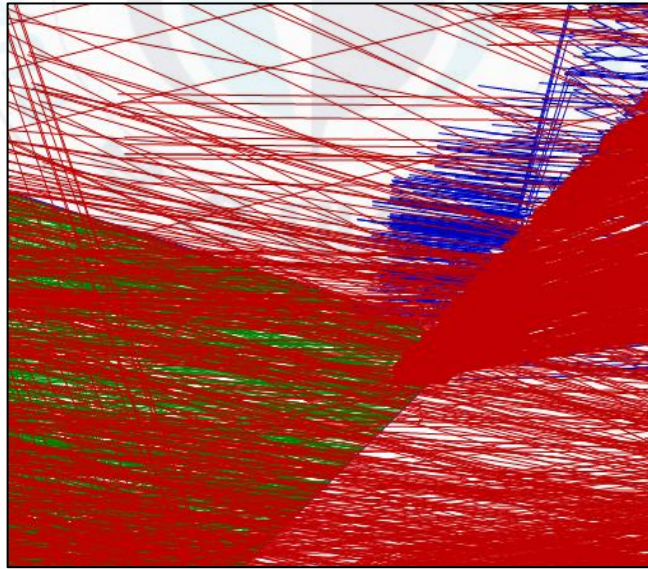


(b)

Figure 20. Illustrations: (a) Korea original map and (b) Korea encrypted map.



(a)



(b)

Figure 21. Illustrations : (a) Viet Nam original map and (b) Viet Nam encrypted map.

4.3 Vertices selection

My method selects randomly vertices based on different precisions by adjusting the scale factor α in Eq. (1)-(3) (that determines the quantity of simplification). $\alpha = -1$ means that my method bypasses the simplification. $\alpha = 0$ means that my method simplifies a map using the initial parameters $(\delta_0, \varepsilon_0, \gamma_0)$ of the DP, SF, and LA algorithms.

The breakpoints in my method are defined based on the scale factor α that controls the threshold parameters of three simplification algorithms. The feature points are defined by using the breakpoints. After that, I select vertices based on ratio of feature points in object for encryption. Figure 22 shows the ratio of breakpoint number, feature point number and encrypted vertices calculated by formula: number of breakpoints/ number of vertices of object's backbone, number of feature points/ number of vertices of object's backbone, number of encrypted vertices/ number of vertices of object.

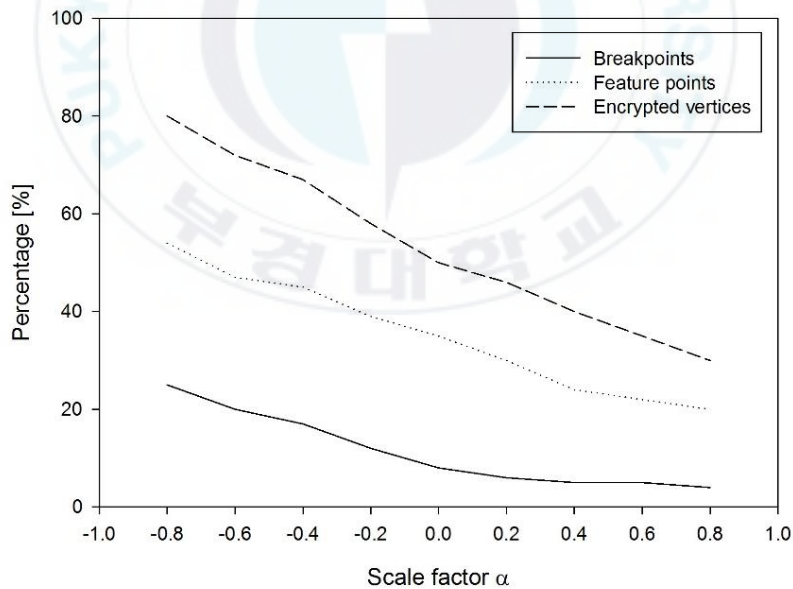


Figure 22. Ratio of breakpoints, feature points, and encrypted vertices of my method on the scale factor (Calculation with **TU** map).

4.4 Distance measure

I compare the difference between encrypted and original map by equation (13):

$$D(E', L) = \sum_{i=1}^N d(P_{ij}) \quad (13)$$

where L is an original map and $E'(L)$ is the corresponding encrypted map and N is the total object in the original map. $d(P_{ij})$ is the distance between the corresponding objects in $E'(L)$ and L , is computed by equation (14):

$$d(P_{ij}) = \sum_{j=1}^{N_{ij}} \sqrt{(|x'_j - x_j|^2 + |y'_j - y_j|^2)} \quad (14)$$

where N_{ij} is the total number of points in object P_{ij}

FR map is used to simulate with $K_1 \neq K_2$ (two user's password) and I find $D(E', L)$, as shown in Fig. 23.

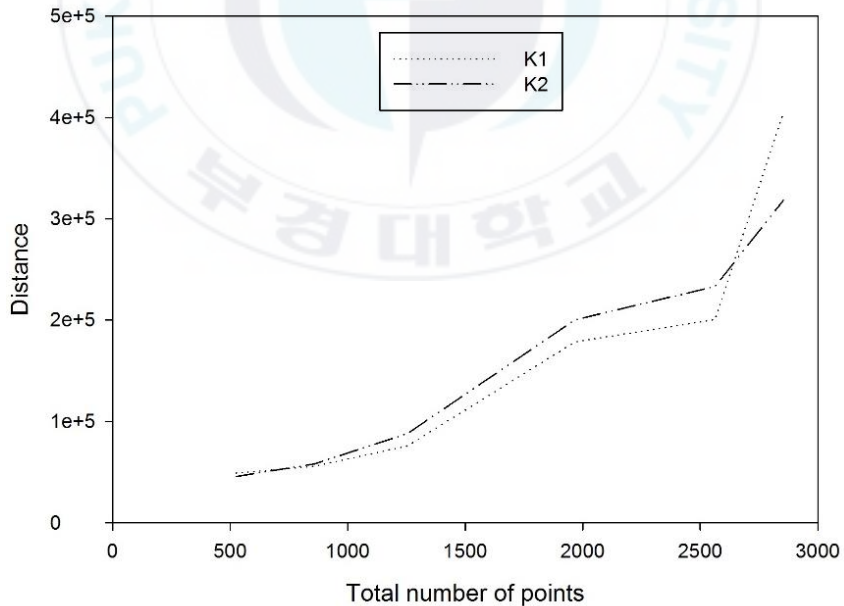


Figure 23. Distance measure with K_1 and K_2 .

4.5 Decryption error

In this algorithm, I only change coordinates of points (vertices) in objects. The encrypted still have the same size as the original map. However, I use 2D-Chaotic map to generate random number and key values from user's key hashing SHA-512 bits, it make these values not absolutely similar in encryption and decryption step. Meaning of this issues come from the problem of system calculation when it stored real numbers in memory. With storing vertices in double type, it seems be no problem when the decryption errors values are approximately zero, as given by Table 2. Then, I tested with many maps to find the maximum error and calculate the average error, as shown in Table 3.

Table 2. The decryption error

Original coordinates	Decryption coordinates	Error
144.13473	144.134729999995	5.00222E-12
-37.32319	-37.3231899999772	2.27942E-11
.....
-37.45185	-37.4518500000175	1.75006675817713E-11
145.31026	145.3102600009	9.00001850823173E-10

Table 3. The max, min error between original map and decryption map

Size (kb)	Total Object	Total Point	Max error	Min error	Average error
45	104	3900	4.58763E-07	0	1.75211E-08
449	147	28162	3.70411E-07	0	1.72524E-08
965	7011	37209	1.13562E-07	0	2.92078E-10
1246	375	79499	6.41549E-07	0	6.00142E-08
1730	13960	61798	7.83001E-08	0	1.76301E-09
2246	575	88948	6.41549E-07	0	6.00142E-08

4.6 Security evaluation

Any pirate should recover all of the encrypted parts by the statistical attacks in perceptual encrypted map without the knowledge of keys. If the randomness of perceptual encryption is very high, it will be very difficult to recover the encrypted map.

Cryptographic security: In my algorithm, by using simplification algorithms and control factor α to select randomly vertices in a layer, I encrypt 40%- 70% of data with the key values and random coefficients created by using Chaotic map and secret key. It would be very difficult to break the encryption algorithm or predict the encrypted part.

Key sensitivity analysis: A highly key sensitive encryption algorithm protects the encrypted data against various cryptanalytic attacks. While developing a cryptosystem, it is assumed that an intruder knows the encryption structure and a-priori probability of used key $k \in K$. As per the Kerckhoff's principle [28], only the secrecy of the used key is required. Even a strong or well designed cryptosystem can be broken easily if the key is poorly chosen or the key space is too small. This makes the encryption or decryption key as the most important part of any cryptosystem. Thus, a good cryptosystem should satisfy the following two conditions to verify the key sensitivity and key space:

- The key space should be discretized in such a way that two ciphertexts encrypted by two slightly different.
- With the generated keys, the ciphertext should be responding to slight changes, signals, or influences.

I test to encrypt the original layer with a slightly different encryption key, and evaluates the difference between the obtained encrypted layers. The difference layers are evaluated to verify the condition that, "layer encrypted with slightly different keys should be completely different".

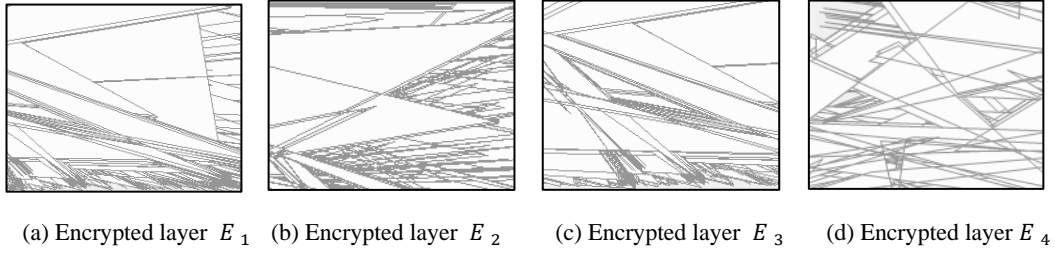


Figure 24. Key sensitivity analysis for encryption process.

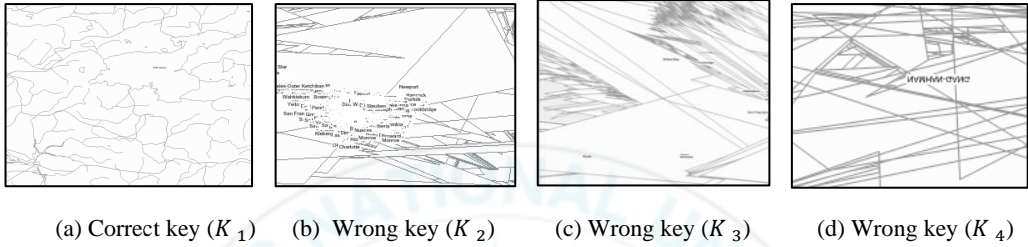


Figure 25. Key sensitivity analysis for decryption process.

For security evaluation, I generate slightly different keys by modifying the first key x_1 , parameter μ_1 used to generate the key values in section 3.4.1 and scale factor α . And then, I change one of them when keeping other values in the modified key. So, when test the key sensitivity, I use the original key with three components $K: (x_1, \mu_1, \alpha)$. I generate the key $K_1: (0.62, 2.9, -0.6)$, the modified keys are expressed as $K_2: (0.75, 2.9, -0.6)$, $K_3: (0.62, 3.1, -0.6)$, $K_4: (0.62, 2.9, 0.1)$. I use the key values that are created from K_1 to encrypt the original layer (*TU* map) and generate the first encrypted map. The encrypted layer E_1 for this case is shown in Fig. 24(a). The original layer is encrypted with key sets are generated from slightly modified key K_2 , K_3 and K_4 . Fig. 24(b)-(d) perform the corresponding encrypted layers. It is observed that layers encrypted with slightly different keys are completely incomprehensible. This means, “ciphertexts generated using slightly different keys are completely different from each other”.

With the second provision of key sensitivity, encrypted layer is decoded with key K_2 , K_3 , K_4 instead of the correct key K_1 , as shown in Fig. 25. The layer is decoded by using incorrect keys, it is completely incomprehensible, and do not leak any information about

the original layer. In contrast, decryption using actual key retrieves the layer correctly. This depicts the high key sensitivity of the proposed cryptosystem, and also verifies the second condition of Kerckhoff's principle, "decryption using a wrong decryption key should not reveal any information".

4.7 Algorithm comparison

The proposed method is compared with two existing algorithms Giao [33] and Bang [41-42]. In these papers, first author encrypted all data in original file and he did not consider the important part in each layer. The second author selected all vertices of the significant objects (defined based on threshold value) in a layer for encryption, it seems quite easily to attack, because he defined the threshold value very simple leads to the algorithm is so weak. My method selects randomly some vertices (that are defined by owner based on scale factor α and three initial parameters) for encryption, attackers will be very difficult to determine the encrypted data and predict key. Figure 26 shows the ratio of encryption data in comparison with two existing methods.

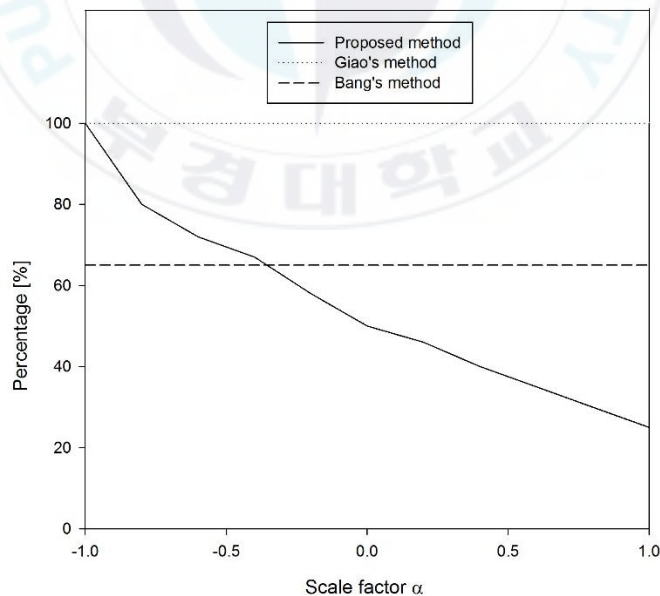
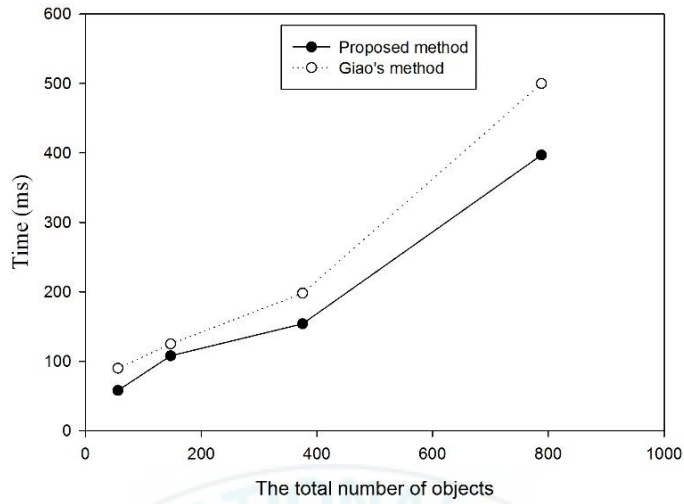
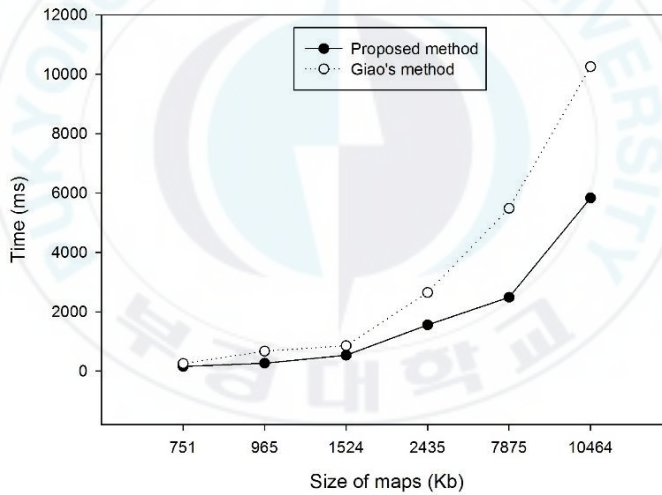


Figure 26. Ratio of encrypted data of my method and the existing methods.



(a)



(b)

Figure 27. Computation time according to: (a) Total number of object and (b) Size of maps.

The computation time of proposed methods depends on many factors. Moreover, it also depends on random processes, size of the map. Therefore, to measure the computation time we need to use specific mathematical model. In this section, I just show the dependence of

computation time and I compared results with Giao's algorithm under the same total number of objects and size of maps, as shown in Fig. 27. Analyzing the detail results, I verify that the computation time of my method is lower than those of Giao's method.



V. Conclusion

In my thesis, I proposed a new method which aim to reduce the ratio of encrypted data in GIS vector map but still assure the performance and the high security. This considers how to select randomly vertices of object in a layer by using simplification algorithms. After that, the selected vertices are encrypted with random numbers and key values generating from 2D-Chaotic map. I confirm that: Human perception do not see any information in encrypted map, poor error in decryption step, computation time is less than it in the existing methods, high security and a large amount of GIS vector map data can be protected by this algorithm.

The algorithm can be used in many kind of application or standard vector map because I proposed to encrypt randomly vertices of important/complex objects (polygons and polylines), so it can be applied for any vector map data and GIS database on on/off-lines server.

References

- [1] Geographic Information Systems (GIS), https://en.wikipedia.org/wiki/Geographic_information_system, accessed Mar. 2016.
- [2] M.F. Goodchild, "Twenty years of progress: GIS science in 2010," *Journal of Spatial Information Science*, no. 1, pp. 3-20, July 2010.
- [3] GIS spatial data, <http://gis.washington.edu/phurvitz/professional/SSI/datatype.html>, accessed to Mar. 2016.
- [4] GIS vector map, <http://www.mapmart.com/Products/DigitalVectorMapping.aspx>, accessed to Mar. 2016.
- [5] Vector data, https://www.qgis.org/en/docs/gentle_gis_introduction/vector_data.html, accessed to Mar. 2016.
- [6] Advantage of vector map, http://bgis.sanbi.org/gis-primer/page_19.htm, accessed to Mar. 2016.
- [7] E. Bertino and M. L. Damiani, "A Controlled Access to Spatial Data on Web," *Proc. Conference on Geographic Information Science*, pp. 369-377, April 2004.
- [8] S.-C. Chena, X. Wang, N. Rishia, and M. A. Weiss, "A Web-based Spatial Data Access System using Semantic R-trees," *Journal of Information Sciences*, vol. 167, issues 1-4, pp. 41-61, October 2003.
- [9] E. Bertino, B. Thuraisingham, M. Gertz, and M. L. Damiani, "Security and Privacy for Geospatial Data: Concepts and Research Directions," *Proc. of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS*, pp. 6-19, 2008.
- [10] N.B. Rybalov and O.I. Zhukovsky, "Access to the Spatial Data in the Web-Oriented GIS," *Proc. Siberian Conference on Control and Communications*, pp. 104-107, April 2007.
- [11] M. Fuguang, G. Yong, Y. Menglong, X. Fuchun, and L. Ding, "The Fine-grained Security Access Control of Spatial Data," *Proc. 18th International Conference on Geoinformatics*, pp. 1-4, June 2010.

- [12] F. Wu, W. Cui, and H. Chen, "A Compound Chaos-Based Encryption Algorithm for Vector Geographic Data under Network Circumstance," *Proc. of Cardholder Information Security Program*, vol.1, pp.254-258, May 2008.
- [13] G. Li "Research of Key Technologies on Encrypting Vector Spatial Data in Oracle Spatial," *Proc. of International Conference on Industrial Electronics and Computer Science*, pp.1-4, December 2010.
- [14] Y. Dakroury, I. A. El-ghafar, and A Tammam, "Protecting GIS Data Using Cryptography and Digital Watermarking," *International Journal of Computer Science and Network Security*, vol.10, no.1, pp.75-84, January 2010.
- [15] Ch. Zhu, Ch. Yang and Q. Wang, "A Watermarking Algorithm for Vector Spatial Geo-Data Based on Integer Wavelet Transform," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 37 Part B4, pp. 15-18, July 2008.
- [16] A. Li, B. Lin, Y. Chen and G. Lu, "Study on Copyright Authentication of GIS Vector Data Based on Zero-Watermarking," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 37 Part B4, pp.1783-1786, July 2008.
- [17] B. Jang, S. Lee and K. Kwon, "Perceptual Encryption with Compression for Secure Vector Map Data Processing," *Journal Digital Signal Processing*, vol. 25, pp.224-243, February 2014.
- [18] R. Ohbuchi, H. Ueda and S. Endoh, "Robust Watermarking of Vector Digital Maps," *Proc. of IEEE International Conference on Multimedia and Expo*, vol. 1, pp. 577-580, August 2002.
- [19] M. Vogit and C. Busch, "Feature-based watermarking of 2D-vector data," *Proc. of the SPIE, Security and Watermarking of multimedia Content*, pp. 359-366, June 2003.
- [20] G. Schulz and M. Vogit, "A High Capacity Watermarking System for Digital Maps," *Proc. of the Multimedia and Security Workshop on Multimedia and Security*, pp. 180-186, September 2004.

- [21] C. Wang, Z. Peng, Y. Peng, L. Yu, J. Wang, and Q. Zhao, "Watermarking Geographical Data on Spatial Topological Relations," *Proc. of Multimedia Tools and Applications*, vol. 57 issue 1, pp. 67-69, March 2012.
- [22] H. Chang, B. Jang, S.-H. Lee, S.-S. Park and K.-R. Kwon, "3D GIS Vector Map Watermarking Using Geometric Distribution," *Proc. of IEEE International Conference on Multimedia and Expo*, pp. 1014-1017, June 2009.
- [23] S.-H. Lee and K.-R. Kwon, "Vector Watermarking Scheme for GIS Vector Map Management," *Multimedia Tools and Applications*, vol. 63, issue 3, pp. 757-790, April 2013.
- [24] R. Ohbuchi, H. Ueda, and S. Endoh, "Watermarking 2D Vector Maps in the Mesh-spectral Domain," *Proc. of International Conference on Shape Modeling and Applications*, pp. 216-225, May 2003.
- [25] V. Solachidis and I. Pitas, "Watermarking Polygonal Lines using Fourier Descriptors," *IEEE Computer Graphics and Applications*, vol. 24, no. 3, pp. 44-51, June 2004.
- [26] V.R. Doncel, N. Nikolaidis, and I. Pitas, "An Optimal Detector Structure for the Fourier Descriptors Domain Watermarking of 2D Vector Graphics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 5, pp. 851-863, September 2007.
- [27] RSA Laboratories, PKCS #5 v2.1: Password-Based Cryptography Standard, October 2006.
- [28] Kerckhoffs's principle, http://en.wikipedia.org/wiki/Kerckhoffs's_principle, accessed to Mar. 2016.
- [29] C. Pellicer-Lostao , R. López-Ruiz, "Pseudo-Random Bit Generation Based on 2D Chaotic Maps of Logistic Type and Its Applications in Chaotic Cryptography", *Computational Science and Its Applications – ICCSA 2008*, pp 784-796, 2008.
- [30] ESRI shapefile description: <https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>, accessed Mar, 25, 2016.

- [31] S.-L. Chen, T. Hwang, W.-W. Lin, "Randomness enhancement using digitalized modified logistic map", *IEEE Transactions on Circuits and SystemsII: Express Briefs* 57, vol. 57, issue 12, pp 996–1000, 2010.
- [32] Heckbert, Paul S.; Garland, Michael (1997). "Survey of polygonal simplification algorithms".
- [33] Giao Pham Ngoc, Gi-Chang Kwon, S.-H. Lee, and K.-R. Kwon, "Selective Encryption Algorithm Based on DCT for GIS Vector Map," *Journal of Korea Multimedia Society*, vol. 17, no. 7, pp.769-777, July 2014.
- [34] A. Massoudi, F. Lefebvre, C. De Vleeschouwer, B. Macq, and J.-J. Quisquater, "Overview on Selective Encryption of Image and Video: Challenges and Perspectives," *Hindawi Publishing Corporation EURASIP Journal on Information Security*, no. 5, ArticleID: 179290, Dec. 2008.
- [35] C. Shi and B. Bhargava, "A fast MPEG video encryption algorithm," *Proc. of the 6th ACM International Conference on Multimedia*, pp. 81–88, Bristol, UK, September, 1998.
- [36] C. Shi, S. Y. Wang, and B. Bhargava, "MPEG video encryption in real-time using secret key cryptography," *Proc. of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '99)*, pp. 191–201, Las Vegas, Nev, USA, June-July 1999.
- [37] M. Podesser, H. P. Schmidt, and A. Uhl, "Selective bitplane encryption for secure transmission of image data in mobile environments," *Proc. of the 5th Nordic Signal Processing Symposium (NORSIG '02)*, Tromsø, Norway, October 2002.
- [38] A. Pommer and A. Uhl, "Selective encryption of waveletpacket encoded image data: efficiency and security," *Multimedia Systems*, vol. 9, no. 3, pp. 279–287, 2003.
- [39] W. Zeng and S. Lei, "Efficient frequency domain selective scrambling of digital video," *IEEE Transactions on Multimedia*, vol. 5, no. 1, pp. 118–129, 2003.
- [40] H. Cheng and X. Li, "Partial encryption of compressed images and videos," *IEEE Transactions on Signal Processing*, vol. 48, no. 8, pp. 2439–2451, 2000.

- [41] N.V. Bang, Kwang-Seok Moon, Suk-Hwan Lee and Ki-Ryong Kwon, "Selective Encryption Scheme Based on DFT, DWT Domain for GIS Vector Map Data," *The 11th International Conference on Multimedia Information Technology and Applications (MITA2015)*, July 2015.
- [42] N.V. Bang, Sang-Hun Lim, Suk-Hwan Lee, Ki-Ryong Kwon, "Selective encryption scheme for vector map data using Chaotic map," *Journal of Korea Multimedia Society (KCI index)*, Vol.18, No.7, July 2015.
- [43] N.V. Bang, Kwang-Seok Moon, Suk-Hwan Lee and Ki-Ryong Kwon, "Vector Map Selective Encryption using Lang Simplification Algorithm," *International Workshop on Advanced Image Technology 2016 (IWAIT 2016)*, Jan 2016.
- [44] Bang Nguyen Van, Kwang-Seok Moon, Chong-Ho Woo, Suk-Hwan Lee, and Ki-Ryong Kwon, "Encryption Algorithm for GIS Vector Map Based on Vertex Randomization and Hybrid Transform," *2016 International Joint Conference on Convergence (IJCC'16)*, Jan 2016.
- [45] N.V. Bang, Kwang-Seok Moon, Suk-Hwan Lee and Ki-Ryong Kwon, "Encryption for Random Vertices Selection in Vector Map Data," *Korea Multimedia Society Conference*, Nov 2015.

Acknowledgement

Finally, I have finished my 2 years study at PKNU. Also, this thesis would not have been completed without guidance and supporting comments from people around me, who also brace me to keep studying and make my life in Busan pleasant.

I would like to express my most sincere gratitude to my advisor, **Prof. Ki-Ryong Kwon** who gave me the scholarship opportunity and fully support this research. He didn't only give me a lots of knowledge in GIS security and Digital image processing, but also abundance of experiences in order to encourage me improving my soft and hard skills.

And also I would like to thank my co-advisor, **Prof. Suk-Hwan Lee**, for his endless dedication in helping me through my toughest time. I am also thankful to him for encouraging the use of correct grammar and consistent notation in my writings and for carefully reading and commenting on countless revisions of this manuscript.

Thank you so much for all of your encouragement, your generous support, patience, and unlimited supplies with Korean food. I am truly grateful for the knowledge, experience and everything I have learned from you.

I would also thank my Lab seniors: **Mr. Teak-Young Seung**, who is always ready to help me in any condition and any problems, **Mr. Jin-Hyeok Park**, **Mr. Liu Yang**, and **Mr. Pham Ngoc Giao**, they are always great seniors and big brothers; also my Lab mates: **Mr. Sang-Hyeon Park**, **Mr. Nguyen Viet Hoan**, and **Mr. Caleb**, they fought together with me and helped me in difficult situations, especially for Korean guys who taught me about the good and bad things about life in Korea.

Thanks to all Vietnamese students in PKNU who encourage and always give me a helping hand. You guys are stress-reliever and great helper in any situations, both inside and outside of campus. You will never be forgotten! Also to any person or institution whose name I could not mention here, thank you for your kind support.

Most importantly, none of this would have been possible without the love and patience of my family. My immediate family to whom this dissertation is dedicated to, has been a constant source of love, concern, support and strength all these years. I would like to

express my heart-felt gratitude to my family. I love you guys! And to all my families and relatives who care for me wherever you are, thank you.

