A NOVEL APPROACH OF CRYPTOGRAPHY AND WATERMARKING USING TO PROTECT GIS DATA

G.RAMASWMAY[1] VUDA.SRINIVASARAO[2]
Professor in St Mary's College of Engineering & Technology Hyderabad1.
Assoc. Professor in St Mary's College of Engineering & Technology.2

ABSTRACT

Cryptography is the study and practice of scrambling information in a manner that is difficult to unscramble, and making scrambled information intelligible. It is used as the basis of much computer security, in that it can be used to keep information confidential, and also preserve the integrity if data, particularly when being stored or being transmitted. Most Geographic Information System (GIS) data providers have a problem protecting their data from being copied and accessed by unauthorized users. Most of the researches in this area focus on how to copyright the GIS data using the concept of digital watermarking. However, watermarking has a lot of limitations and if it is used alone, it will not able to achieve the goal. In this paper, we will introduce a new mechanism to protect the GIS data by combining the RC6 and hash functions cryptography algorithms with watermarking techniques. This mechanism can be applied to both raster and vector GIS data.

Keywords: *RC6,GIS,Encryption,DTEDPM*

1. INTRODUCTION:

A cryptographic algorithm, or cipher, is a mathematical function used in the encryption and decryption process. A cryptographic algorithm works in combination with a key a word, number, or phrase to encrypt the plaintext. The same plaintext encrypts to different cipher text with different keys. The security of encrypted data is entirely dependent on two things: the strength of the cryptographic algorithm and the secrecy of the key. A cryptographic algorithm, plus all possible keys and all the protocols that make it work, comprise a cryptosystem. PGP is a cryptosystem. cryptography that will stop your kid sister from reading your files, and cryptography that will stop major governments from reading your files. Cryptography dealt exclusively with the problem of secure communication. Modern cryptography is a much broader field that deals with all adversarial threats facing parties who wish to carry some task in a network. More specifically, the aim of cryptography is to provide secure solutions to a set of parties who wish to carry out a distributed task and either do not trust each other or fear an external adversarial threat. RC6 is an evolutionary improvement of RC5, designed to meet the requirements of the Advanced Encryption Standard (AES). Like RC5, RC6 makes essential use of data-dependent rotations. The use of multiplication greatly increases the diffusion achieved per round, allowing for greater security, fewer rounds, and increased throughput. Like RC5, RC6 is a fully parameterized family of encryption algorithms.

Watermarking was originally used to make an assertion of ownership of art works or books by embedding transparent information. It was used at a paper factory in Italy to prove its production in the 800s. Digitized application of the same watermarking technique is applied onto digital data. It embeds a watermark without affecting the quality of the original data. It doesn't affect the size of the file nor change the original file. The watermarked data file displays the original content; however it will protect ownership when it is violated.

The steganography consists of techniques to allow the communication between two persons, hiding not only the contents but also the very existence of the communication in the eyes of any observer. Watermarks may be perceptible (visible) or imperceptible (invisible) to human vision. Visible watermarks, by nature, are more

ISSN: 1817-3195 / **E-ISSN**: 1992-8615

intrusive to the media and act to deter theft of the media, such as a warning sign announces an alarm system even if one does not exist. Examples of such watermarks can be seen easily on most network television stations by the station's logo in the corner of the viewable screen. These watermarks are typically confined to an area of the image, which is less intrusive to the overall image. Attackers have a visible target and can remove the watermark by cropping the image.

GIS has been used in many organizations like military and commercial applications for many years. The main component of GIS is the data. The GIS data has two important properties. First, the effort it takes to put it in a form suitable for use in the GIS applications. This effort increases its cost. Second, in most cases the GIS Data contains confidential information which must be kept away from unauthorized users. Such confidential information includes GIS layers containing troop locations and additional information like movements and mines places in a tactical environment. Hence, it is very important to protect the GIS data from two threats. First, since the GIS data is too expensive, we have to prevent illegal duplication and distribution of it. Today, it is too easy for a company to buy some GIS layers, make illegal copies from them and distribute or sell them many times without taking any permission from the original GIS data provider. Second, since the GIS data is sensitive and must not be accessed by unauthorized users, we have to enforce some kinds of access control on it. The mechanism we will introduce in this paper will be able to protect the GIS data from these two threats. Moreover, we will try to achieve high fault tolerance and system availability. The remaining sections of the paper are organized as follows: Section 2 explains the related work and its limitations. Section 3 introduces our mechanism called **D**istributed **T**ransparent **E**xtensive **D**ata **P**rotection **M**echanism (DTEDPM) to protect the GIS data from being copied and accessed by unauthorized users. Sections 4 and 5 explain the different modules of the two DTEDPM main components. Section 6 tests DTEDPM using two data sets of Environmental Systems Research Institute (ESRI) shapefiles. Conclusions are summed up in Section 7.

2. RELATED WORK

Today, there are many security holes in the systems we are using [3] and not all of them are documented or even known. Once the attacker finds a security hole in these systems, he is able to access the GIS data, copy and use it any where. GIS data providers have two ways to provide their data to users. The first one is by putting it on their servers and building web applications for users to be able to access this data. Traditional techniques [1] like intrusion detection, access control and firewalls can be used to secure the network infrastructure [2] and protect servers containing GIS data from attackers. However, these techniques alone cannot achieve full protection. The second one is by building standalone applications and providing the application with the data to their users. In this case, some kinds of protection must be applied to the GIS data itself. Most of the work that has been done in this area uses the concept of digital watermarking [4-5]. Watermarking is a process of obtaining a digital watermarked file by embedding hidden information (watermarking pattern or watermark for simplicity) like copyright string in a dataset without producing perceptible changes in the data using a suitable watermarking algorithm. In most cases, this watermark is encrypted. The producer can recover the watermark on request in order to produce evidence of ownership in a court. The typical watermarking system is shown in Fig. 1.

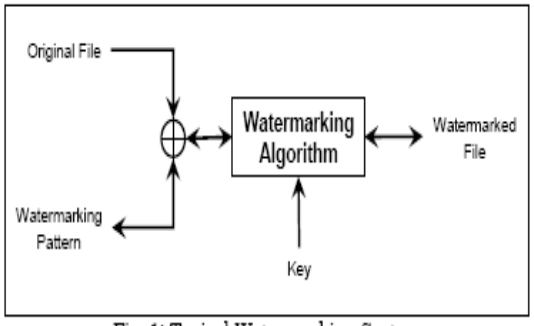


Fig. 1: Typical Watermarking System

Watermarking can be applied to both GIS raster [6-7] and vector [8-9] data. One of the particularly suited watermarking mechanisms for the GIS data is segmenting it into a grid and placing a watermark into different regions of the grid. This mechanism is easier in Watermarking the GIS Raster data than watermarking the GIS vector data since the first is an array of pixels. Applying a symmetric grid (grid with equal size rectangles) to the GIS raster data results in equal pixel density grid regions which are not the case with the GIS

ISSN: 1817-3195 / **E-ISSN**: 1992-8615

vector data. Watermarking has many limitations especially when applied to the GIS data. Some of these limitations are:

(i) An attacker may be interested in only a subset of the watermarked GIS data. In this case, it is crucial that the watermark survives in the subset selected by the attacker and this is very difficult to be achieved.

(ii) In most cases, Watermarking of the GIS vector data is done by displacing a group of vertices from their original places. This affects the data accuracy especially when data is obtained using small scale maps.

(iii) Watermarking cannot protect the GIS data containing confidential information from access by unauthorized users.

(iv) Watermarking depends on the type of the file to be watermarked.

(v) When using watermarking, attackers always seek a method to remove the watermark or change it. They are not interested in knowing the information it includes. Knowing the information the watermark includes is a more difficult task than just removing or changing it especially when the watermark is encrypted. Watermarking is also not surviving against many of the GIS operations like changing the GIS data projection.

The mechanism we introduce in the next sections can be applied to any format of both GIS raster and vector data types. However, through the rest of the paper we will focus on the protection of the ESRI shapefile format [10] as it is one of the most common GIS vector data formats.

3. DTEDPM

The naming convention of the introduced DTEDPM reflects some of its main characteristics:

(i) Distributed: in DTEDPM, the mechanism modules are distributed to many servers and desktops. Shapefiles are also distributed to many servers. Parts of both are replicated. This architecture achieves high fault tolerance and increases the system availability.

(ii) Transparent: in DTEDPM, the shapefiles storage locations are hidden from the users. No user authorized or not authorized is able to know or guess where the shapefiles are being stored.

(iv) Extensive: where DTEDPM is not built for a specific computing environment. So, it can be installed in any computing environment. This makes the reuse or customization of the system very easy. Moreover, the mechanism can also be distributed to different computing environment at the same time.

(v) Data Protection Mechanism: since the main function of DTEDPM is to protect the shapefiles from being copied or accessed by unauthorized users.

3.1 DTEDPM Basic Idea

DTEDPM tries to protect shapefiles against illegal distribution and secure their contents. This is done by encrypting the shapefiles on a stand alone desktop using HASH [11]. The encrypted shapefiles and their encryption password are then distributed to many servers connected to the network. When an authorized user wants to access one of these shapefiles, he has to login first to the DTEDPM from a desktop using his username and password. DTEDPM establishes a secure channel between this desktop and the servers containing the shapefiles using RC6 [11]. DTEDPM chooses the server with the lowest network traffic from the list of servers which are allowed to be accessed by this user and desktop. Then, it displays the selected shapefile located on this server to him. This is done by decrypting the selected shapefile to this desktop memory and displaying it from there.

The shapefiles remain encrypted on the servers. If authorized or unauthorized users try to copy and use them in any GIS application, they will copy the encrypted version of them. This encrypted version cannot be used in any GIS application.

3.2 DTEDPM Components

The first component is the offline Component. The task of this component is to generate the HASH key from the password and encrypt the shapefiles using the generated HASH key. This component is also responsible for generating the RC6 private and public keys. The RC6 private key is used to encrypt the password. The encrypted version of the password is then embedded into the copyright picture. Both keys are used in exchanging messages over the network. Once the encrypted shapefiles and the watermarked copyright picture are ready, they are distributed to the network servers.

All the offline component modules run on a standalone desktop to protect shapefiles from being copied or accessed before being encrypted. Once we get the encrypted shapefiles and the watermarked copyright picture, there is no more need for this component in the DTEDPM operation.

ISSN: 1817-3195 / **E-ISSN**: 1992-8615

The second Component is the online component. This component handles all the DTEDPM operations during the run time phase. It handles the authentication, manages the load balancing of the servers, decrypts the encrypted shapefiles to the authorized users' desktop memories and displays the decrypted shapefiles on their desktops.

DTEDPM has two main Components as shown in Fig. 2.

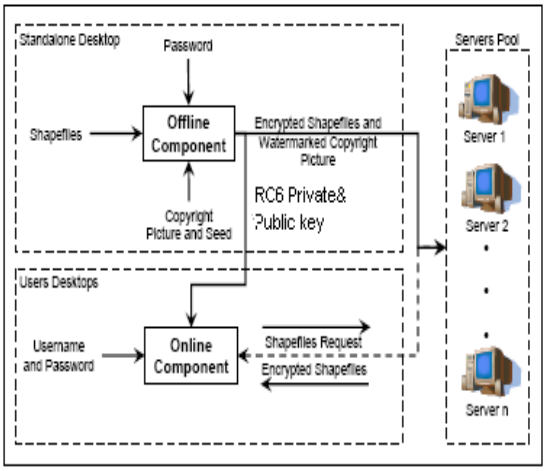


Fig. 2: DTEDPM Main Components

4. OFFLINE COMPONENT:

The offline component has four modules as shown in Fig. 3. The task of these modules is to generate the encrypted shapefiles and the watermarked copyright picture. The offline component takes the following four inputs:

(i) Password: it can be any set of alphanumeric characters. It can also contain the copyright string of the company which produces the shapefiles. The password is used as an input to two modules. The first module is the HASH key generator module. This module uses the password to generate a binary HASH key of the specified length. The second module is the watermark creator module which watermarks the copyright picture with the password.

(ii) Shapefiles: these are the GIS vector data files that are encrypted and used in our DTEDPM. There is no limitation on the size of the shapefiles or the number of records they contain. Also, the shapefiles can be of any spatial feature type (point, plotline, polygon) and with any number of attributes.

(iii) Copyright picture: it is mainly used as storage for the password. Password must be supplied to the online component to be able to decrypt the shapefiles. If we store it as plaintext, attackers can steal it by logging to the servers using any of the existing weak points. Also, the encryption of it is not sufficient. DTEDPM encrypts it using the generated RC6 private key to get a watermark and then hide this watermark into the copyright picture. The copyright picture can be any picture like the logo of the company which produces the shapefiles.

(iv) Seed: it is used to select some of random pixels from the copyright picture. These random pixels are used to hide the watermark in the copyright picture.

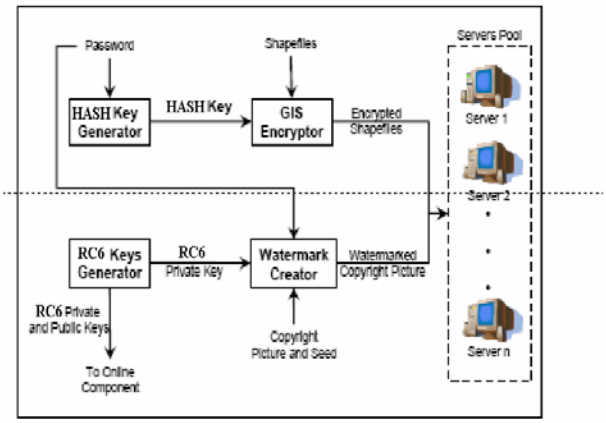


Fig. 3: DTEDPM Offline Component

The offline component has three outputs:

(i) Encrypted shapefiles: these are the result of encrypting the input shapefiles using the generated HASH key. The encrypted shapefiles are distributed to the servers once they are created.

(ii) Watermarked copyright picture: it is the copyright picture containing the watermark. The watermarked copyright picture is also distributed to the servers once it is created.

(iii) RC6 private and public keys: The RC6 keys generator module generates two keys. The private key is used to encrypt the password to generate the watermark. Both keys are used by the DTEDPM online component to secure the communication between it and the servers. Also, they are used to secure the communication between the different modules of the online component.

4.1 HASH Key Generator

The HASH key generator module converts the password to a 256-bit HASH key. The module is also used in the online component to be able to

ISSN: 1817-3195 / E-ISSN: 1992-8615

decrypt these shapefiles when they are accessed by authorized users. Passwords are easier to remember than keys which are streams of zeros and ones. However, passwords are always chosen from a relatively small space. This makes them not suitable for cryptography.

A good encryption implementation does not only use the password, but mixes it with a random number, known as a salt, to create the key. A salt produces a large set of keys (2 to the power of the number of salt bits) corresponding to the given password, among which one is selected at random. Salt has two benefits. First, having a salt makes the same password encrypts the same file in many different ways. Encrypting the same file more than once with the same password and different salt, results in different encrypted files. This eliminates the content analysis attack. Second, each bit added to the salt doubles the numbers of generated keys which complicates the dictionary attack. Assume a user uses one of 100 English words as his password and the system uses a 16 bit salt. Because of this salt, the attacker must calculate 65536 (216) keys for each word until a match is found. The total number of keys can be obtained by multiplying the number of words in the dictionary with the number of possible salts. This results in 6553600 keys as opposed to 100 keys in case the salt is not used.

The process of generating the key from the password is done by using a Key Derivation Function (KDF). KDF uses password, salt and an iteration count to achieve its job. An iteration count has traditionally served the purpose of increasing the cost of producing keys from a password and thereby also increasing the difficulty of attack. A typical application of KDF is explained in [12]. The salt, iteration count and the key length are selected. Then KDF is applied to them to get the derived key with a specified length.

4.2 GIS Encryptor

The GIS encryptor module encrypts all the used shapefiles using the 256-bit HASH in counter mode [11]. HASH block cipher operates on fixed length blocks of 128 bits. This can lead to many problems when encrypting messages longer or shorter than this fixed length. Longer messages must be divided into blocks of this fixed length. Also shorter messages must be padded to block of this fixed length taking into consideration that the original length of the plaintext must be recovered. Sometimes padding is also required for the last block of longer messages. Another problem is that encrypting the same plaintext block under the same key always produces the same output which affects confidentiality. Several modes of operation have been invented to override these problems. One of these common modes is the counter mode. The HASH counter mode fixes all the mentioned problems since it uses a fixed length counter regardless of the plaintext block size. HASH counter mode has many advantages over other invented modes of operation [11]. One of the most important advantages is that any block of plaintext or cipher text can be processed in random access fashion. Obtaining any plaintext or cipher text block does not depend on the preceding or successor block. This makes it possible to be implemented in a parallel processing environment.

4.3 RC6 Keys Generator

The RC6 keys generator is responsible for generating both the RC6 private and public keys. These keys are used in many modules of the DTEDPM offline and online components. The length of both keys used in DTEDPM can be 512, 1024 or 4096 bits. The more number of bits result in a stronger encryption but a more time is needed for encryption and decryption processes. Both the private and public keys can be used in encryption and decryption. Messages encrypted using the private key must be decrypted using the public key and vice versa.

4.4 Watermark Creator

The watermark creator module encrypts the password using the RC6 private key to get a watermark. Then it watermarks the copyright picture with this watermark using a predefined seed. We use a 24-bit bitmap format as a copyright picture. A 24-bit bitmap is one of the most common graphic formats used by computers. It is a standard file format containing a header and data sections. The header section contains the picture meta-data such as its size, width and height. The data section contains the actual data used when displaying the picture to the screen. Bitmap can be considered as a two dimensional array of picture elements called pixels. Each pixel is represented by four bytes. Three bytes of them are used to store the pixel color (24-bit color depth). This allows each pixel to have 16777216 different colors. The pixel color has three color components (red, green,

ISSN: 1817-3195 / **E-ISSN**: 1992-8615

blue), each component is stored in one byte. If we consider just the red color component, there will be 255 different values of red. The fourth byte is reserved and should be zero.

We select some of the bitmap pixels to hide the watermark based on Random Number Generators (RNG) [13]. RNG are algorithms for generating a sequence of random numbers. It can be started from an arbitrary starting state called seed. If the same random seed is deliberately shared, it becomes a secret key. Two or more systems use the same RNG algorithm and seed, generate a matching sequence of non repeating numbers. The watermark creator module consists of five sub modules. These sub modules are connected together as shown in Fig. 4.

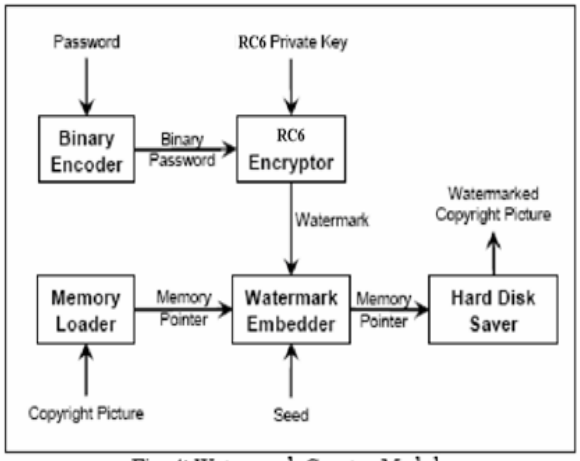


Fig. 4: Watermark Creator Module

The functions of these sub modules are as follows:

(i) Binary encoder: here the password is converted to a binary password. Each password character is represented by eight bits to support passwords with both English and Arabic characters.

(ii) RC6 encryptor: this sub module encrypts the binary password using the RC6 private key. This is done by converting it to a decimal value. This decimal value is then encrypted. The resulting decimal is converted

again to binary. There are more two bytes added to the starting of the binary encrypted password to construct the watermark. These two bytes store the length of the binary encrypted password.

(iii) Memory loader: the function of this sub module is to read the copyright picture from the standalone desktop hard disk and load it into its memory as a two dimensional array of pixels. A memory pointer containing the starting address of the copyright picture in memory is passed to the watermark embedder sub module.

(iv) Watermark embedder: this sub module is the core of the watermark creator module. It hides the watermark generated by the RC6 encryptor sub module into the copyright picture located in memory. This sub module uses one seed to generate three integer random numbers for each watermark bit. The first and second one are used to select the X and Y coordinates of the pixel to be changed. They take values greater than or equal to zero and less than the picture width and height respectively. The third one is used to choose the selected pixel color component which its Least Significant Bit (LSB) is changed. It takes only three values (0, 1, 2) representing the red, green and blue color components. For each selected pixel, the chosen color component value is read and changed according to the corresponding value of the current watermark bit. The new value of the chosen color component is written to the memory replacing the old one.

(v) Hard disk saver: this is the last sub module. Its function is to read the obtained watermarked copyright picture from the standalone desktop memory and save it to its hard disk. At this point, both the encrypted shapefiles and the watermarked copyright picture are ready on the hard disk of the standalone desktop. The system administrator can start distributing them to the different servers. Once this task is completed, the authorized users can start accessing the encrypted shapefiles using the DTEDPM online component.

5. ONLINE COMPONENT

The online component has seven modules as shown in Fig. 5. Six modules of them are installed on each user desktop while the seventh module is installed on each server. Also, each server has its database which stores all the information of the users who are authorized to access the shapefiles located on this server. The main task of these modules is to give the authorized users the ability to access and display the shapefiles and prevent unauthorized users from doing that.

ISSN: 1817-3195 / E-ISSN: 1992-8615

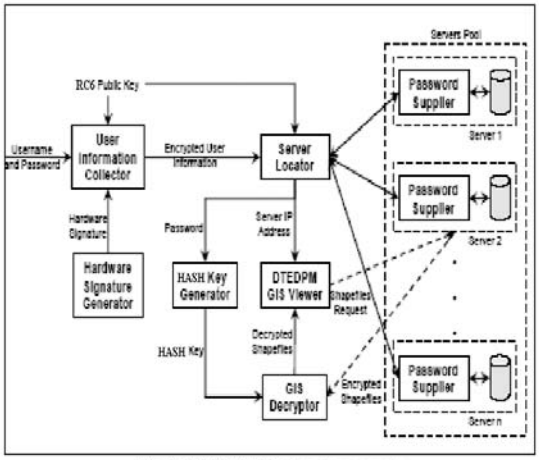


Fig. 5: DTEDPM Online Component

The online component modules run on each user desktop take the following three inputs:

(i) RC6 public key: this key is generated by the offline component RC6 keys generator module. It must be known by the server locator module to be able to decrypt the information received from the password supplier module. Also it is used to encrypt the user information before being sent to the password supplier module. In other words, both the RC6 private and public keys are used to establish a secure channel between the server locator and the password supplier modules.

(ii) Username and password: These are the credentials of the user wants to access the shapefiles located on the servers. The user is not able to access the shapefiles located on a specific server unless his information is stored in this server database.

(iii) Encrypted shapefiles: these are the shapefiles requested to be accessed by the authorized user. They are transferred from the server selected by the server locator module to the GIS decryptor module located on the authorized user desktop.

The online component has only one output. This output is the list of all the shapefiles, the authorized user wants to access. This list is sent to the server selected by the server locator module.

5.1 Password Supplier

The password supplier module is installed on each server and it works in conjunction with the server database. The server database contains all the information of the users who are authorized to access the shapefiles located on this server. This information includes the users' usernames, passwords and the hardware signatures of the desktops they must login from to be able to access the server shapefiles. The database can also be expanded to contain more information according to the nature of the GIS application. The more information the database includes, the more restrictions can be applied to the authorized users.

The password supplier module takes the following five inputs:

(i) Watermarked copyright picture: It is the one generated by the offline component watermark creator module.

(ii) RC6 private key: This is the private key generated by the offline component RC6 key generators module.

(iii) Seed: it is used to retrieve the watermark from the watermarked copyright picture. It must have the same value as the one used with the offline component watermark creator module to get the same sequence of random numbers used in watermarking the copyright picture.

(iv) Server Internet Protocol (IP) address: It is the IP address of the server running the password supplier module.

(v) Shapefiles request: This is the request for accessing the shapefiles located on the server. It is issued by the users' desktops DTEDPM GIS viewer modules. The module has two tasks. The first task is executed when the module is started up. The role of this job is to encrypt the server IP address using the RC6 private key and get the RC6 private key encrypted password from the watermarked copyright picture. Once these two strings are obtained, they are concatenated into one string. This string is loaded into the server memory.

The module has three sub modules which are responsible for doing this job. These sub modules are connected together as shown in Fig. 6

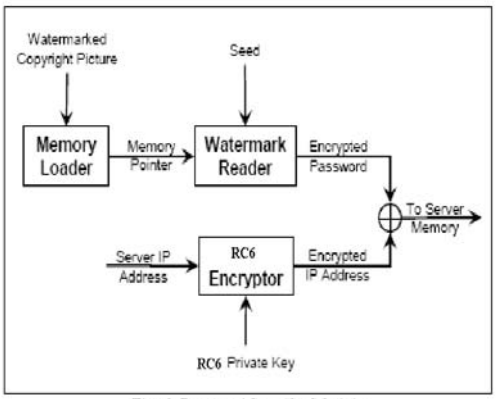


Fig. 6: Password Supplier Module

ISSN: 1817-3195 / E-ISSN: 1992-8615

The functions of these sub modules are as follows:

(i) Memory loader: the function of this sub module is to read the watermarked copyright picture from the server hard disk and load it into its memory as a two dimensional array of pixels. A memory pointer containing the starting address of the watermarked copyright picture in memory is passed to the watermark reader sub module.

(ii) Watermark reader: this sub module uses a seed to get the RC6 private key encrypted password from the watermark stored in the watermarked copyright picture. The seed is used to generate the same sequence of random pixels and color components as the one generated when watermarking the copyright picture to get the correct watermark. First, the module gets the length of the binary encrypted password which is stored in the watermark first 16 bits. This is done by generating 48 integer random numbers. Using each three of them, the chosen color component LSB (represented by the third number) of the selected pixel

(represented by the first and second numbers) is read. The resulting 16 bits are converted to decimal to get the required length. Then, according to this length, the module keeps generating a sequence of integer random numbers until all the bits of the watermark are read. These bits are concatenated (without the first 16 bits storing the length of the binary encrypted password) and converted to a decimal value. This decimal value is the RC6 private key encrypted password.

(iii) RC6 encryptor: this module gets the IP address of the server and encrypts it using the RC6 private key to get the encrypted server IP address.

The second task of the module is to wait for shapefiles requests. Once the module receives a request, it decrypts the request contents using the RC6 private key to get the information of the user wants to access the server shapefiles. It checks if this user is authorized to access the shapefiles located on this server or not. This is done by comparing the decrypted user information with the information stored in the server database. If the user is authorized to access this server shapefiles, the module answers the server locator module with its server memory string. Otherwise it remains silent.

5.2 Hardware Signature Generator

The hardware signature generator module is responsible for generating the user desktop hardware signature. The hardware signature can include many of the hardware component identifiers of the user desktop. These identifiers can include processor identifier, network card Media Access Control (MAC) address… etc.

When the user opens the online component installed on his desktop, the hardware signature is generated and sent to the user information collector module. Concatenating the hardware signature with the user information that has been sent to the servers plays an important role in making the servers' shapefiles accessible only by the authorized users when they are using specific desktops.

5.3 User Information Collector

The user information collector module constructs the encrypted user information network packet and sends it to the server locator module. When the user opens the online component installed on his desktop, he has to enter a valid username and password. This module takes the username and the password entered by the user and concatenates them with the hardware signature generated by the hardware signature module to construct the user information string. The user information string can contain more information fields about the user desktop like its IP address, the operating system username and password the user uses to login to his desktop. These added information fields must have corresponding fields in the server database. The user information string is encrypted using the RC6 public key. The encrypted user information is then sent to the server locator module.

5.4 Server Locator

The server locator module is responsible for finding the lowest network traffic server from the list of servers which are allowed to be accessed by the authorized user and the desktop he is login to the DTEDPM online component using it. Once this server is found, the sever locator module sends the server IP address to the DTEDPM GIS viewer module and the password to the HASH key generator module.

The server locator module gets the encrypted user information from the user information collector module, broadcasts it over the network and waits to get the response from the servers running the password supplier module. When the module gets the first server reply, it chooses this

ISSN: 1817-3195 / **E-ISSN**: 1992-8615

server as the one with the lowest network traffic. It divides the reply string into two parts. Both parts are decrypted using the RC6 public key to get the server IP address and the shapefiles decryption password. The server IP address is sent to the DTEDPM GIS viewer module while the password is sent to the HASH key generator module.

5.5 DTEDPM GIS Viewer

The DTEDPM GIS viewer module is used by the authorized users to access and display the shapefiles. It contains all the GIS functions like zoom in, zoom out, measure distances … etc.
Once the module receives the selected server IP address from the server locator module, the authorized user on request gets a list of all the shapefiles located in a specific folder on this server. The authorized user can select one or more shapefiles to be displayed. This selection is sent to the selected server which collects all the required shapefiles and sends them back to the GIS decryptor module. The GIS decryptor module decrypts all the user selected shapefiles and sends them to the DTEDPM GIS viewer module to display them.

5.6 HASH Key Generator

This module is the same as the offline component HASH key generator module. It converts the password sent by the server locator module to a 256-bit HASH key. This key is used as an input to the GIS decryptor module to decrypt the shapefiles. This module must use the same parameters values of the offline component HASH key generator module to get the same HASH key as the one used in encrypting the shapefiles. These parameters include the salt, iteration count, pseudorandom function … etc.

5.7 GIS Decryptor

The GIS decryptor module decrypts all the authorized user selected shapefiles using the 256-bit HASH in counter mode. The decrypted shapefiles are loaded into the authorized user desktop memory. Memory pointers of the decrypted shapefiles are passed to the DTEDPM GIS viewer module to display them.

6. TESTING AND RESULTS

DTEDPM is tested and analyzed using two data sets of shapefiles. The first set contains eight shapefiles of Africa while the second set contains six shapefiles of Egypt. The two data sets [14], [15] are derived from maps created for the open street map project [16] provided by Frederik Ramm of GeoFabrik [17]. The analysis of DTEDPM involves studying the effect of using cryptography on the time required to display the shapefiles. DTEDPM uses cryptography in five tasks. These tasks are: (i) Encrypt shapefiles using HASH.
(ii) Encrypt password using RC6 private key to generate the watermark.
(iii) Encrypt the server IP address using RC6 private key to construct the server memory string.
(iv) Decrypt the selected server memory string using RC6 public key to get the password and server IP address.
(v) Decrypt shapefiles using HASH.

The two data sets are shown in Fig. 7.

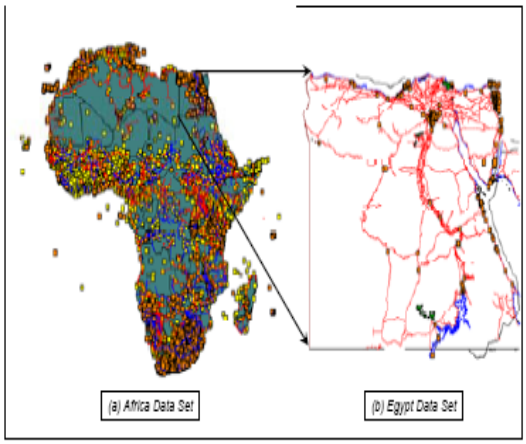


Fig. 7: Two Data Sets used in DTEDPM Testing

The first and second tasks are parts of the offline component and do not have any effect on displaying the shapefiles. The third and fourth tasks are parts of the online component but also they do not have any effect on displaying the shapefiles because they are executed once at the startup of the online component. The only task affects the displaying of shapefiles are the fifth one because it has to be executed every time a shapefile is opened. The only difference between displaying the encrypted and non encrypted shapefiles is the way by which they are loaded into memory. The encrypted shapefiles are decrypted to memory while the non encrypted shapefiles are loaded directly into memory. Once the shapefile becomes in memory, both the encrypted and non encrypted versions of it take the same amount of time to be displayed.

ISSN: 1817-3195 / E-ISSN: 1992-8615

So, in our analysis we compare between the loading time required to decrypt each encrypted shapefile to memory and the time required to load the non encrypted shapefile into memory. The shapefiles of the previous two data sets are sorted ascending according to their sizes and we analyze the DTEDPM on two phases. In the first phase, we study the effect of decryption on the displaying of shapefiles. In the second phase, we study the effect of the changing the iteration count on the displaying of shapefiles.

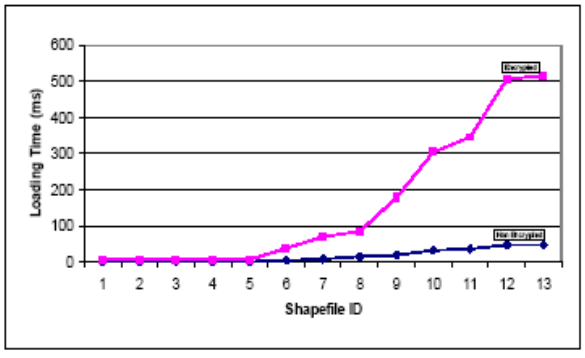
6.1 Decryption Effect

In this phase we encrypt all the shapefiles using an iteration count equal to zero. The loading times of both the encrypted and non encrypted shapefiles are recorded and plotted as shown in Table 1 and Fig. 8.

Table 1: Non Encrypted and Encrypted Shapefiles Loading Times

ID	Shapefile Size	Loading Time (ms)	
		Non Encrypted	Encrypted
1	0.05	0	6
2	.08	0	6
3	010	0	6
4	023	0	6
5	025	0	6
6	076	4	37
7	1.41	7	69
8	1.57	15	85
9	2.98	18	178
10	4.96	31	303
11	6.10	35	345
12	8.07	46	506
13	8.35	47	516
14	69.20	453	4272

Fig. 8. Non Encrypted and Encrypted Shapefiles Loading Times



(a) Africa Data Set (b) Egypt Data Set

The results show that increasing the size of both the non encrypted and encrypted shapefiles results in increasing their loading times. However, the increasing is higher in case of the encrypted shapefiles than the non encrypted shapefiles. Increasing the size of the encrypted shapefile, results in increasing the number of its HASH encrypted blocks. This increases the time required to decrypt the shapefile and so, increases the time the GIS decryptor module should wait before start loading the decrypted shapefile into memory.

The high increasing in the encrypted shapefiles loading times does not cause a problem since in most cases the size of the used shapefiles are small and do not exceed several megabytes. The largest shapefile we use is the Africa roads shapefile. The size of this shapefile is 69.20 megabyte and it contains the roads of all the countries of the Africa continent. DTEDPM takes an extra 3819 ms (4272 - 453) to load it into memory when it is encrypted. This time can be reduced by implementing the HASH decryption using parallel processing since this is one of the key features which makes us choose AES in counter mode.

The loading time is consumed by the online component at the first time the authorized user open a shapefile. Further GIS functions applied to the displayed shapefile, like zoom in, zoom out …etc, use the shapefile loaded in memory to redisplay and manipulate it and therefore, they take the same amount of time in case they are applied to both the non encrypted and encrypted shapefiles.

6.2 Iteration Count Effect

One of the most important parameters that affects the HASH decryption of the shapefiles and can be tuned is the iteration count. A large number of iteration count results in a strong key but it increases the time required to generate the key at the decryption time and so, increases the time required to decrypt the shapefiles. In this phase we encrypt all the shapefiles using different iteration counts (IC) 10, 500 and 25000. For each iteration count, the loading times of the encrypted shapefiles are recorded and plotted as shown in Table 2 and figure 9.

ISSN: 1817-3195 / **E-ISSN**: 1992-8615

Table 2: Encrypted Shapefiles Loading Times at Different Iteration Counts

ID	Loading Time (ms)		
	Encrypted (IC = 10)	Encrypted (IC = 500)	Encrypted (IC = 25000)
1	15	109	1437
2	15	109	1437
3	15	109	1437
4	15	109	1437
5	15	109	1437
6	46	140	1468
7	78	172	1500
8	94	188	1516
9	187	281	1609
10	312	406	1734
11	375	469	1797
12	515	609	1937
13	525	619	1947
14	4281	4375	5703

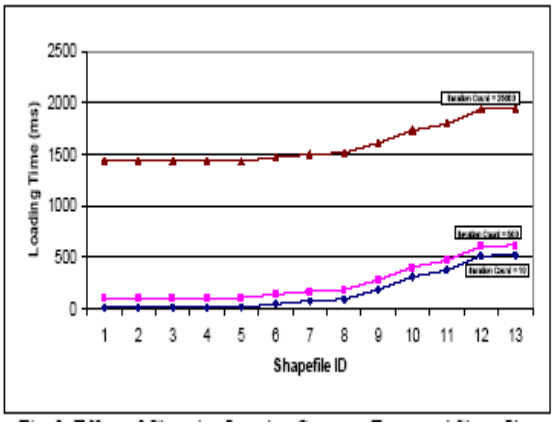


Fig. 9: Effect of Changing Iteration Count on Encrypted Shapefiles Loading Times

The results show that increasing the iteration count results in increasing the encrypted shapefiles loading time. For the same iteration count, all the encrypted shapefiles loading time is increased by the same amount of time regardless of their sizes. Table 3 shows the amount of increase in the loading time according to the selected iteration count.

Table3: Increase in Loading Time for Different Iteration Counts

Iteration Count	Increasing in Loading Time (ms)
10	9
500	103
25000	1431

7. CONCLUSION

GIS data is expensive and sometimes contains confidential information. For this reason, it is very important to protect it against illegal distribution and access by unauthorized users. Watermarking alone cannot protect shapefiles from being distributed or accessed by unauthorized users. Watermarking is exactly like the copyright message written on the welcome screen of the executable programs.
Welcome screen does not eliminate the needs of protecting these programs against illegal distribution. In this paper, we introduced a complete mechanism named DTEDPM for protecting the GIS data from these two threats. It is built using a combination of the encryption and digital watermarking techniques. DTEDPM is tested and analyzed using two data sets of shapefiles and the effect of decryption and iteration count on the time required to display the shapefiles is studied.

REFERENCES:

[1] Joseph Migga Kizza, A Guide to Computer Network Security,USA: Springer, February 2009.

[2] Angus Wong and Alan Yeung, Network InfrastructureSecurity, USA: Springer, April 2009.

[3] Stuart Mcclure, Joel Scambray and George Kurtz, HackingExposed 6: Network Security Secrets & Solutions, USA:McGraw Hill, Sixth Edition, January 2009.

[4] Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich and Ton Kalker, Digital Watermarking and Steganography, USA: Morgan Kaufmann, Second Edition,November 2007.

[5] Peter Wayner, Disappearing Cryptography: InformationHiding: Steganography & Watermarking, USA: Morgan Kaufmann, Third Edition, December 2008.

[6] M.A.Dorairangaswamy, "A Novel Invisible and Blind Watermarking Scheme For Copyright Protection of Digital Images," IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.4, April 2009.

[7] Manjunatha Prasad.R and Shivaprakash Koliwad, "A Comprehensive Survey of

ISSN: 1817-3195 / E-ISSN: 1992-8615

Contemporary Researches in Watermarking for Copyright Protection of Digital Images,"

IJCSNS International Journal of Computer Science and

Network Security, Vol. 9 No. 4, April 2009.

[8] Chang-qing Zhu, Cheng-song Yang and Qi-sheng Wang, "AWatermarking Algorithm for Vector Spatial Geo-Data Based on Integer Wavelet Transform," The International Archives

of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXVII Part B4, Beijing, July2008.

[9] Anbo Li, Bingxian Lin, Ying Chen and Guonian Lv, "Studyon Copyright Authentication of GIS Vector Data Based onZero-Watermarking," The International Archives of the

Photogrammetry, Remote Sensing and Spatial InformationSciences, Vol. XXXVII Part B4, Beijing, July 2008.

[10] Environmental Systems Research Institute, An ESRI WhitePaper: ESRI Shapefile Technical Description, USA: ESRI,July 1998.

[11] William Stallings, Cryptography and Network SecurityPrinciples and Practices, USA: Prentice Hall, Fourth Edition,November 2005

.

[12] RSA Laboratories, PKCS #5 v2.1: Password-BasedCryptography Standard, October 2006.

ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-5v2/pkcs5v2_1.pdf

[13] Averill Law and W. David Kelton, Simulation Modeling andAnalysis, USA: McGraw Hill, Third Edition, December

1999.

[14] Download Free African Continent ArcGIS Shapefile MapLayers, October 2009.

http://www.mapcruzin.com/free-africa-arcgis-maps-shapefil

es.htm

[15] CloudMade Downloads, October 2009.

http://downloads.cloudmade.com/africa/egypt

[16] OpenStreetMap, October 2009.

http://www.openstreetmap.org/

[17] GEOFABRIK // Home, October 2009.

http://www.geofabrik.de/

ISSN: 1817-3195 / **E-ISSN**: 1992-8615

AUTHOR PROFILES:

Prof.G.Ramaswamy received the M.Tech degree in Information Technology from the Punjabi University, in 2003. He received the Ph.D degree in Computer Science & Engg from Magadh University from the 2007. Currently he is professor in St.Mary's College of Engg & Tech. His research interests include Network Security, Cryptography.

V.Srinivasarao received the M.Tech degree in Computer Science& Engg from the Satyabama University, in 2007. . He is research student of Annamalia University. Currently he is Assoc.Professor in St.Mary's College of Engg & Tech. His research interests include Network Security, Cryptography.

Vol. 16 No.2 June , 2010 pp [116 – 128]

ISSN: 1817-3195 / **E-ISSN**: 1992-8615

PAGE INTENTIONALLY LEFT BLANK!