GIS Vector Map Watermarking using Discrete Fourier Transform

Jun Ho Choi Hedyatmo 13518044
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13518044@std.stei.itb.ac.id

***Abstract*—There are many watermarking methods on GIS Vector map. A lot of those methods uses transformation on the original vertex to make the method robust against various attacks that can happen on the vector map. One such transformation is none other than Discrete Fourier Transform or DFT. Discrete Fourier Transform has an interesting property, which is RST invariance. So, any watermarking method that uses DFT will retain these properties and makes the watermarking method robust against geometric attacks such as translation, rotation, and scaling. In this paper we will be discussing watermarking method on GIS Vector Map that uses Discrete Fourier Transform and improve on their robustness on attacks that will previously work on DFT-based watermarking method. One such attack is vertex deletion attack, where the attacker will delete the vertex of a watermarked GIS Vector Map. This attack will usually make the watermark unrecoverable, but we will be presenting an extra step in the embedding phase to make it possible to recover the watermark after vertex deletion attack. The improved method presented in this paper successfully recover the watermark from an attacked GIS Vector Map, while also retaining It's robustness against geometric attacks such as translation, rotation, and scaling.**

Keywords—watermarking, discrete fourier transform, GIS, vector map, embedding, extracting

I. Introduction

Since ancient times, humans have made many works and publications or other useful things, to protect the work and protect the ownership of the work, copyright was created. Copyright protects works created by someone so that they cannot be misused by others, for example duplicated, falsified, or traded outside the applicable law. The development of computer technology, especially the internet, makes matters relating to copyright very loose, because the methods for copyright used on physical objects cannot be applied to digital works/goods that live on computers. Of course, the issue of copyright on digital goods is also very important to understand and research.

Watermarking is one of the most researched methods to solve the copyright problem in the digital world. In simple terms, watermarking is an activity to insert a marker, called a watermark, into the work/item that needs to be protected. Many types of data can be watermarked, including images, audio, video, text, barcodes, 3D models, CAD data, vector data, and so on. Not all these data types have the same method of watermark insertion, because the differences in these types can make many different things related to their processing, therefore for each data type it is necessary to create their own watermarking method [1].

II. Basic Theory

A. Geographic Information System (GIS)

GIS or Geographic Information System is a framework that provides information about geographic data or other data related to the geographic location, and is represented or displayed as a map in a digital context [2].

GIS Map representation is not universal, there are many formats used to represent maps. Broadly speaking, GIS maps have two types of representation, including raster representation and vector representation. As the name suggests, raster representation uses the concept of rasterization in its GIS information storage method (Wade, T. and Sommer, S). By raster, we mean that the map is stored as many small pixels that represent the actual appearance of the map image. The format of the file is usually similar to an image, i.e. JPEG, TIFF, some of them have BLOB format.

Vector representation means to represent a GIS Map using a decomposition of the map into geometric objects such as points, lines, and polygons, which can be simply represented as vectors. The obvious difference between vector representation and raster representation is that for vectors, the format only cares about the geometric objects, not the general appearance of the map. This is quite useful because unlike raster, vector representation can be enlarged or reduced without damaging its quality as the computer only needs to re-render when reading a GIS Map in vector representation. Usually, this format is stored in a text-based format and the data contains coordinate information and parts of the geometry objects that are displayed on the map.

One example of the vector representation is the GeoJSON format which stores geometry objects in **.json** format.

B. Digital Watermarking

Watermarking is a concept of embedding of watermark into a media, with the aim that what is inserted can show the ownership or authenticity of the media in question. Digital watermarking, as the name implies, is the process of inserting a watermark into a digital media.

The process of digital watermarking on images is a derivative of Steganography, which is the process of inserting data hidden in images for the purpose of secure transmission. The difference between steganography and digital watermarking is the purpose and what is important when performing the process. Steganography only cares about what is inserted, so the medium in which it is inserted is not the main priority of the process. Whereas digital watermarking also cares about what is inserted, namely the watermark itself, and the media where it is inserted as well, whether it is damaged, or whether the content changed after the insertion [3].

- Embedding

The embedding process is done with a kind of method. The input is the original media (host media) like images, audio and other types of data and the watermark itself. And using that kind of method, the new media that has been watermarked is generated. The general flow of the embedding process can be seen on figure I.

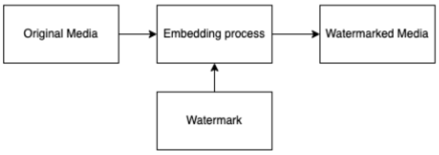


Figure 1 embedding process

- Extracting

The extracting process usually done with the reverse of the embedding method. The input is the watermarked media and the process will try to retrieve the watermark inside the watermarked media. The general flow of the extracting process can be seen on figure II.

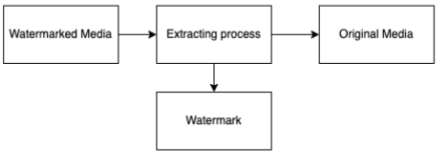


Figure 2 extracting process

C. Watermarking Criteria

The watermarking process needs to fulfill a couple of criteria that shows things that needs to be considered when choosing or creating a watermarking method. Here are the criteria that is widely used.

- Effectiveness

This criterion describes how effective a watermarking process is, where the process can be clearly explained and can fulfill the insertion process and extraction process as described earlier.

- Perceptual Similarity

This criterion is one of the most important ones. The meaning of perceptual similarity is the similarity with the original media. It is no secret that the process of inserting something into another media will certainly change the media in one way or another, this is what is called distortion. This criterion adds another rule when creating a watermarking process, mainly about how the method must still make the watermarked media still recognizable as the original media, in which case the distortion must be minimized, but it is impossible for there to be no distortion at all if a watermark has been inserted on the media.

To measure perceptual similarity there are several methods that can be used. Usually, the method used is to see how much deviation the watermarked media has with the original media. Then the deviation is compared with a certain threshold whether it meets the required standards, or not [4].

- Robustness

Robustness is a criterion that shows how robust the media that has been watermarked is. Has been watermarked. There are so many attacks that can be carried out on the media, in this case the watermarking process must guarantee that the watermarking process can guarantee that the result can face this attack well, so that even though there are changes or alterations that occur due to attacks on the media, the watermark extraction process must still be carried out and the watermark resulting from the extraction process must also be sufficiently well preserved.

- Blindness

Another important criterion is blindness, which will differentiate the process of watermark insertion process into certain schemes. There are blind watermarking, non-blind watermarking, and semi-blind watermarking schemes. In general, blindness refers to how much information is required when performing the extraction process. Although the meaning of this term is sometimes ambiguous or vague, it is generally accepted that for blind watermarking schemes, neither the original media before the watermarking process, nor the watermark itself is needed to extract the watermark from the watermarked media. Non-blind watermarking schemes allow access to the initial media or original media to assist the extraction process to obtain the watermark and the original media. The semi-blind watermarking scheme uses other information or the watermark itself to perform the extraction process, because it requires additional information so this scheme cannot be called a blind watermarking scheme. To determine which scheme applies to a watermarking process, it must be seen from the process or algorithm used to perform the insertion process and the extraction process itself.

- Invertibility

Invertibility is another criterion of watermarking. It indicates whether it is possible to reconstruct the original media after the watermark is extracted, generally the extraction process is done to verify whether the watermark is original. However, in the invertibility criterion, the ability of the extraction process to also reconstruct the watermarked media well is also being considered.

D. Transform Domain

Sometimes, the process of making changes to data within a particular domain makes the process more restrictive and imposes many limitations. This is also true in the context of watermarking. Generally, images, or more specifically maps, are data in the spatial domain. In this case, sometimes performing the data insertion process in other domains that are not spatial can bring things that could not or are difficult to do in the spatial domain, things that will help to fulfill the watermarking criteria in the previous section [5].

Another domain that is commonly used for this is the frequency domain. And some of the techniques used to transform data from the spatial domain are as follows [6]:

- Discrete Fourier Transform
- Discrete Cosine Transform
- Discrete Wavelet Transform

E. Discrete Fourier Transform

Discrete Fourier transform is one of the domain transform methods which is a discrete version of the Fourier transform. The main motivation of the Fourier transform is to determine if a certain frequency is present in a wave. Basically, a wave is the sum of sinusoidal functions in different frequencies and amplitudes. The Fourier transform makes it possible to separate these frequencies from the sample by transforming the domain to the frequency domain. From there it can be seen that the peaks of the frequencies are the frequencies contained in the wave [7].

The Discrete Fourier Transform does the same thing but with discrete (finite) data. Mathematically, here is the formula used to perform the Discrete Fourier Transform (DFT).

$$X_k = \sum_{n=0}^{N-1} x_n . e^{-\frac{2\pi i}{N}kn} \qquad (1)$$

x_n here indicates the initial (spatial) domain data, so x_n must be a number that can be multiplied by a complex number. So if x_n has not been represented in complex numbers, it must be converted first. This change is quite easy in general, in the case of a GIS Map, the data contained in it is a vector or coordinates directly from the map. So if the coordinate/vector has coordinates (x, y) then the equivalent complex number is

$$x_i = x + yi \qquad (2)$$

Looking at the formula for DFT above, one would think that the computation process will run quadratically because the formula above will be done N times for each X_k. So, the time complexity is $O(N^2)$.

However, there is an algorithm that can compute the Discrete Fourier Transform (DFT) faster than $O(N^2)$. The algorithm is known as Fast Fourier Transform (FFT) which can perform DFT computation in $O(N \log N)$ time complexity. This algorithm utilizes the concept of divide-and-conquer to break its computation part into 2 equal parts and work on them separately to get faster complexity. This FFT algorithm is commonly used to perform DFT computation. In the context of a GIS Map where one map can contain up to 1 million vectors or points, of course the naive way will not be fast enough to be used on such samples. So, the FFT is needed to handle it.

Other than DFT, there also exist Its counterpart formula to return the data to the original domain, which is called the Inverse Discrete Fourier Transform (IDFT). Here is the formula for it:

$$x_k = \frac{1}{N} \sum_{n=0}^{N-1} X_n . e^{-\frac{2\pi i}{N}kn} \qquad (3)$$

The formula is quite like the DFT, but the positions of x_k and X_n are swapped, and the values are multiplied by $\frac{1}{N}$. Note that the FFT algorithm can also be applied to the computation of the Inverse Discrete Fourier Transform (IDFT) so the computation time complexity is also $O(N \log N)$.

In general, multiplication by $\frac{1}{N}$ is referred to as the normalization factor along with 1. Regarding their position, it is not too important which normalization factor is used in the DFT and which is used in the IDFT. What is important is that if the DFT uses 1 then the IDFT must use $\frac{1}{N}$. And if the DFT uses $\frac{1}{N}$ then the IDFT must use 1.

III. Previous Research

Related to digital watermarking on vector maps, the methods used are quite diverse. One of the most popular methods is utilizing the transform domain to change the coordinates of the vector map to another domain. Here we will be discussing a method for digital watermarking on vector maps with a blind type of scheme, and robust to certain attacks. The method is based on changing the domain of the vector map from spatial to DFT domain, using the Discrete Fourier Transform algorithm discussed in the previous section. The method that we will be discussing is according to [8].

In general, the method developed by [8] utilizes the DFT domain as a medium to insert a watermark that must be represented as binary data. Binary data in the form of 0 and 1 will distinguish the distortion of the vertex in the vector map that has been converted to the DFT domain.

Then the numbers that already exist in the DFT domain with certain criteria can be divided into two types, namely the type that represents 0, and the type that represents 1. The insertion process starts by pairing the first vertex with the binary data of the first watermark, and the second vertex with the binary data of the first watermark. check whether the vertex representation in the DFT corresponds to the binary data in the watermark, the correspondence is seen from the criteria for dividing the angle into several regions based on the step size. The division can be seen in Figure 3.

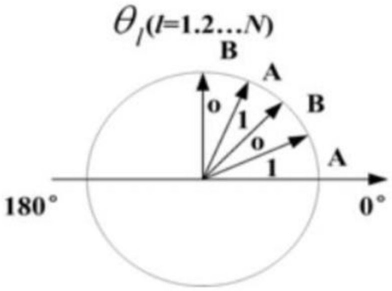


Figure 3 Angle division

Suppose the angle to be quantized is 30 degrees, and the step size is 20 degrees, then 30 degrees will be part of region 0. If the watermark bit matches then it does not change, if it turns out that the watermark bit is different than the 30 degrees angle will be added with the step size to 50 degrees. This is the new angle of the number in the DFT.

The process continues until all binary data has been inserted. This creates another requirement for the [8] method, namely that the size of the binary data watermark must be smaller or equal to the size of the vector map. Once everything is inserted, an IDFT (Inverse Discrete Fourier Transform) is performed to return the map to the spatial domain, and the map is ready for reuse.

For the extraction process, it is quite intuitive, because this method has defined criteria to divide the vertex into two types, namely the type that represents 0, and the type that represents 1, so the map vertex that is already in the DFT domain (after the previous DFT) can be categorized using the criteria of this method, whether the vertex is part of group 0 or part of group 1. also means that this method requires the size of the original watermark, as it must know where to stop for the extraction process. Once the sequence of vertex groups is obtained, the result can be recreated into a watermark, according to the original specification. Some results of watermark insertion using this method are shown in Figure 4.

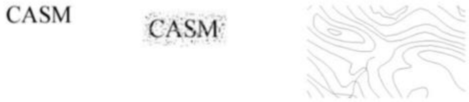


Figure 4 watermark and embedded map

This method is robust against any geometric attacks such as rotation, scaling, and translation. But it is not robust against vertex deletion attacks.

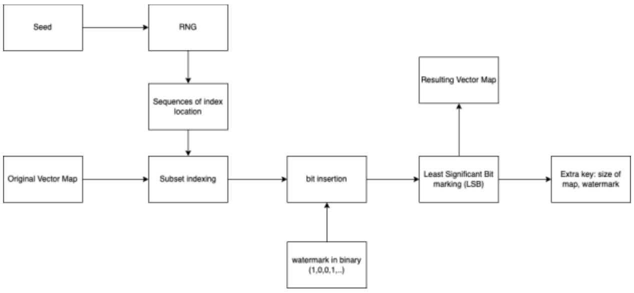
IV. MODIFICATION ON EXISTING METHOD

Related for the vertex deletion attack, usually the deletions are random and scattered. This is a problem for the method proposed by [8] because DFT takes all vertices of the map to be embedded, so if even a single vertex is deleted, DFT will no longer be uniform. Considering the DFT formula, when viewed in vector terms, DFT takes the weighted sum of all vectors of vertices and uses it to create a new vertex in a different domain, and the weight of each vertex is different because it is a combination of the old vertex index and the new vertex index, if deletion occurs, the order of vertices will be shifted so that the value will change considerably. So even if only one vertex is deleted, the result will be immediately incomprehensible. This is a direct drawback of methods that embed DFT, utilizing DFT is a good thing because by its nature DFT has properties that are directly robust to some geometric attacks such as translation, but for non-geometric attacks such as vertices deletion, methods that require the contribution of all vertices' as weighted sum will not be robust, if the measure to handle the attack is only handled from the DFT domain.

The first observation is that we do not need to perform DFT on all vertices of the GIS Map, if we do it only on a subset of the GIS Map vertices, the robustness will still be maintained. And this leads us to another observation, since the DFT is only done on a subset of the total vertices, in case of vertex deletion, as long as the deleted vertex does not overlap with the subset selected for DFT, there will be no problem when attacked with vertex deletion. Now the problem arises as to how we can know which vertex is part of the subset when it has been attacked with vertex deletion.

In this modification, we consider the implementation in JavaScript. JavaScript stores decimal numbers using 64-bit double precision format, changing the least significant bit (LSB) of the number can be done to minimize the shift. If it is determined that vertices that are part of the DFT subset have an LSB of 1 and vertices other than that have an LSB of 0, then it can be easily categorized as a subset of vertices that have been attacked by vertex deletion. Then for the selection of the subset, a random number generator will be used with a certain seed which is considered as the key.

We can use a pseudorandom number generator to generate a random sequence using a certain seed, where this seed is the input or parameter of this pseudorandom number generator algorithm [9]. After that, this sequence is used to select the vertex that becomes the subset that will be operated DFT and inserted bits of the watermark. This seed is the key to the generated sequence, this random placement makes the collision between the random number generator used by the attacker to delete vertices with the same place smaller. The flow of insertion with additional steps is depicted in Figure 5.



Seed
RNG
Sequences of index location
Resulting Vector Map
Original Vector Map
Subset indexing
bit insertion
Least Significant Bit marking (LSB)
Extra key: size of map, watermark
watermark in binary (1,0,0,1...)

Figure 5 new pipeline for embedding process

Assuming the seed used is secret and this is an important key to the algorithm, because by utilizing the same pseudorandom number generator with the same seed, the owner of the watermarked vector map can also know the selected subset. Since the seed can generate a sequence that exactly shows the location of the bit, the extraction process can still be done. There is some additional information required for extraction, including the initial map size, watermark size and initial seed. This information are useful for classifying the extraction, whether the map has been attacked by vertex removal, and also the watermark size is useful for reconstructing the watermark dimensions. The extraction process is depicted in Figure 6.

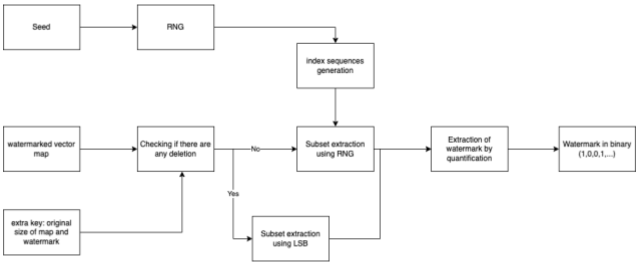

Seed
RNG
index sequences generation
watermarked vector map
Checking if there are any deletion
No
Subset extraction using RNG
Extraction of watermark by quantification
Watermark in binary (1,0,0,1,...)
Yes
extra key: original size of map and watermark
Subset extraction using LSB

Figure 6 new pipeline for extracting process

V. Experiments

We will be conducting some experiments on the new and improved method of watermarking using DFT based on the method by [8]. The first set of experiment will test on the new method's robustness against geometric attacks such as rotation, translation, and scaling.

This experiment is conducted to ensure that the modified embedding method based on the algorithm of the watermark embedding method developed by [8] remains robust to rotation and translation attacks. Because from the paper made by [8] it is stated that the developed method is robust to rotation and translation attacks. In addition, it is also added to test against scaling attacks because scaling includes geometric attacks such as rotation and translation. However, in the paper of [8] it is stated that the method is not robust to scaling attacks, so it will be tried again to the modified embedding method whether it is now robust to scaling attacks or not.

The experiment is declared successful if the modified embedding method based on the algorithm of the watermark embedding method developed by [8] remains robust to rotation and translation attacks by looking at the results of the watermark extracting process and compared to the initial watermark. And compare the results of extracting the watermark from scaling attacks to see if the new embedding method is robust to scaling attacks, for the metric used is bit error. Some limitations and test scenarios need to be set so that the test environment is limited, and accurate conclusions can be drawn. For this experiment the map used will be the map of Indonesia, banten and shanghai respectively depicted in Figure 7, Figure 8, and Figure 9.

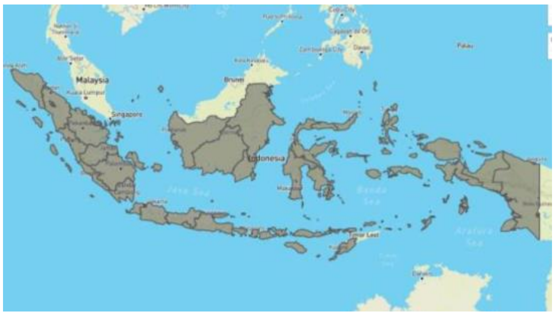


Figure 7 Map of Indonesia

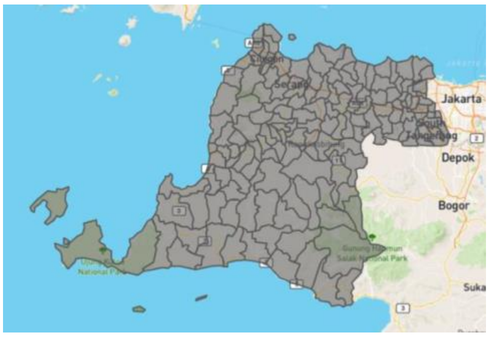


Figure 8 Map of Banten

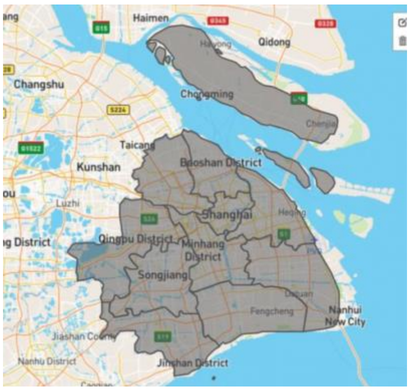


Figure 9 Map of Shanghai

And then the watermark used for this experiment is the logo of ganesha ITB which is depicted in Figure 10.

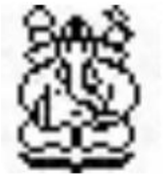


Figure 10 Ganesha ITB

The result of the first set of experiment is that the new modification on the watermarking method by [8] doesn't change its robustness against geometric attacks such as rotation, scaling, and translation. So the method is still robust against such attacks.

The next experiment is to test the method's robustness against vertex deletion. The map that will be used for this experiment is the map of Europe depicted in Figure 11.

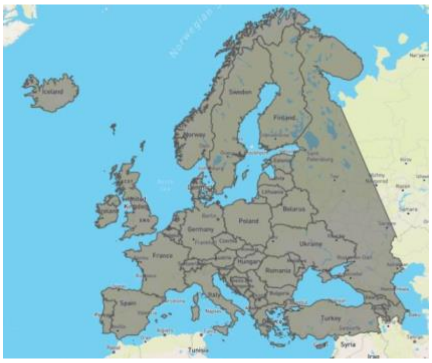


Figure 11 Map of Europe

And the watermark used will be the image of "eropa" which means Europe in Indonesian, depicted in Figure 12.

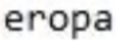


Figure 12 text watermark

The experiment to test the robustness of the method against vertex deletion will run on 10 different seed, and the attack will also run on 10 different seed, so there are 100 cases, and the method of the test is to find the maximum number of vertices that the attacker can delete but still make the watermark recoverable from the attack. Here is the result of the second experiment.

Table 1 Experiment Result

Information	Value
Number of cases	100
Average	31.1
Standard Deviation	34.8
Minimum Value	0
Maximum Value	194

VI. Conclusion

The development of a watermarking method for GIS vector maps using Discrete Fourier Transform was developed on the basis of the method already made by [8] with some changes to the watermarking method. The proposed modification to the watermarking method by[8] is robust to vertex deletion, so the development of the method to address the weaknesses of the previous method was successful.

However, its performance cannot be accurately assessed because of course vertex removal attacks vary and may not be uniformly removed. The modification made assumes that the attack occurs in a distributed manner and is not focused on one part only, therefore the distribution of the vertex where the watermark bit data is embedded is also uniformly distributed to minimize the collision between the vertex carrying the watermark information and the attacked vertex.

The modifications made to the watermarking method using Discrete Fourier Transform developed by [8] do not interfere with the ability of the basic method to be robust to rotation and translation. Mathematically this is quite obvious, because the Discrete Fourier Transform application is only moved from the whole vertex to a subset of the vertex and there is only a slight change in embedding the data bits into the vertex, so the robustness against rotation and translation is within expectations.

The scaling attack test on the modified method was found to be robust. This is quite strange because in the method of embedding the data bits into the vertex, nothing has changed, so the robustness should not change against scaling attacks from the previous method with the modified method as well. This is probably because the scaling attack carried out in the [8] paper performs scaling by multiplying by two different numbers on the abscissa and ordinate, while for testing in this task the scaling test is carried out by multiplying the abscissa and ordinate by the same number so that the results are robust. If multiplied by two different numbers it will still not be robust, as concluded in the paper [8].

Acknowledgment *(Heading 5)*

The completion of this paper could not have been possible without the help from the Almighty God. The author thanks Dr. Ir. Rinaldi Munir, M. T. as supervisor of this whole research and project, as all of this would not be possible without his guidance. The author also thanks his family, friends, and many more people for always supporting him. The author also thanks Institut Teknologi Bandung for giving him the opportunity to learn and grow much more.

References

[1] Niu, XiaMu., Shao, C. Y., Wang, X. T. (2006). A Survey of Digital Vector Map Watermarking

[2] Clarke, K. C. (1986). Advances in geographic information systems, computers, environment and urban systems, Vol. 10, pp. 175–184.

[3] Kumar Saini, Lalit ., Shrivastava, Vishal. (2014). A Survey of Digital Watermarking Techniques and its Applications. Arya College of Engineering & Information Technology, Jaipur, India.

[4] Nyeem, Hussain,. Boles, Wageeh,. Boyd, Colin. (2014). Digital image watermarking: its formal model, fundamental properties, and possible attacks. Eurasip Journal on Advancees in Signal Processing, 2014, Article number: 135 1-33.

[5] Kleiman, Dave. (2007). The Official CHFI Study Guide (Exam-312-49): For Computer Hacking Forensic Investigator.

[6] Kumar Saini, Lalit ., Shrivastava, Vishal. (2014). A Survey of Digital Watermarking Techniques and its Applications. Arya College of Engineering & Information Technology, Jaipur, India.

[7] Rabiner, Lawrence R.; Gold, Bernard. (1975). Theory and Application of Digital Signal Processing.

[8] Tao, S., Dehe, Xu., Chengming, Li., Jianguo, Sun. (2009). Watermarking GIS Data for Digital Map Copyright Protection.

[9] Barker, Elaine,. Kelsey, John. (2012). Recommendation for Random Number Generation Using Deterministic Random Bit Generators.