International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 5 May 2021, pp: 1363-1371 www.ijaem.net ISSN: 2395-5252

Least Significant Bit Based Image Steganalysis System using Java API

Emmanuel O. Ojei[1], Sylvanus O. Anigbogu[2], Gloria Anigbogu[3]
[1,2,3]Department of Computer Science, Nnamdi Azikiwe University, Awka, Nigeria

Submitted: 15-05-2021 Revised: 26-05-2021 Accepted: 28-05-2021

ABSTRACT: Steganography is the art of hiding confidential information within a digital image. The main goal of steganography is that the very existence of the secret information within the cover media is embedded in such a way that it is hard or even impossible to tell that it is there. Due to this, terrorists and hackers employ digital steganography to facilitate secret intra-group communicationand subsequently it is becoming increasingly important to detect the images that contain steganography such that we can reduce foul-play. Steganalysis is theprocess of detecting the existence of hidden message embedded by LSB steganography method. This paper presents a steganalysis tool which is designed with Object oriented software design methodology and coded with Java programming language. One hundred images are downloaded using image downloader and stored in a specific folder on the computer. The system scans images to test if they have been affected by steganography and the result showed that some of these images contain hidden messages. The outcome of the work is confirmed via performance analysis of the system developed.

KEYWORDS: Steganalysis,Steganography, cryptography, Information Security,Steganographic Algorithm (SA), Least Significant Bit (LSB),Application Programming Interface (API)

I. INTRODUCTION

With the development of computer and expanding its use in different areas of life and work, the issue of information security has become increasingly important. One of the grounds discussed in information security is the exchange of information through the cover media. To this end, different methods such as cryptography, steganography, coding, etc have been used. The method of steganography is among the methods that have received attention in recent years. Steganography is a technique that hides data among the bits of a cover file such as audio, video and image files [1]. In steganography, information is hidden in cover media in such a way that other person will not notice the presence of the information. There is a major distinction between this method and the other methods of covert exchange of information because, for example, in cryptography, the individuals notice the information by seeing the coded information but they will not be able to comprehend the information. However, in steganography, the existence of the information in the sources will not be noticed at all. Most steganography activities have been carried out on images, video clips, texts, music and sounds. Image steganography is defined as the covert embedding of data into digital pictures. Though steganography hides information in any one of the digital medias, digital images are the most popular as carrier due to their frequency usage on the internet [2]. Most researchers use image as cover file with different image formats such as JPEG, BMP, TIFF, PNG or GIF files. A bitmap or BMP format is a simple image file format. Data is easy to manipulate, since it is uncompressed. But the uncompressed data leads to larger file size than the compressed image. JPEG (Joint Photographic Expert Group) is the most commonly used image file format. It uses lossy compression technique; the quality of the image is excellent. The size of the file is also smaller. TIFF format uses lossless compression. The file is reduced without affecting the image quality. GIF (Graphics Interchange format) has colour palette to provide an indexed colours image. It uses lossless compression and it can store only 256 different colours. It is not suitable for representing complex photography with continuous tones. The PNG (Portable Network Graphics) file format provides better colours support, best compression, and gamma correction in brightness control and image transparency. PNG format can be used as an alternative to GIF to represent web images [3]. The counter technique of image steganography is known as image steganalysis, which begins by recognizing the object that exists in the embedded source file. It is the art of discovering and rendering useless covert messages or disabling of hidden message. In this paper, efforts are made to analyse the digital images used to transmit secret information to provide a way of detecting whether or not data is hidden from the embedded content. The system only reveals the existence of a message; it is not concerned with what the message is. This will help to reduce threat of terrorism.

II. STATEMENT OF PROBLEM

In recent times, terrorists and hackers use steganography to communicate in secret with their accomplices. Rumours about terrorists using steganography started first in the daily newspaper USA Today on 5 February 2001 in two articles titled "Terrorist instructions hidden online" and "Terror groups hide behind Web encryption". In July the same year, an article was titled even more precisely: "Militant's wire Web with links to jihad". A citation from the article: "Lately, al-Qaeda operatives have been sending hundreds of encrypted messages that have been hidden in files on digital photographs on the auction site eBay.com". Other media worldwide cited these rumours many times, especially after the terrorist attack of 9/11, without ever showing proof [4].This made it necessary to explore image analysis of steganographic content on the internet in order to attempt to solve the problem. This work involved developing a steganalysis system that will be used for passive analysis – to detect whether or not hidden data is present in a digital image. The process of completely removing hidden data, destroying or strategically altering to render it useless was not covered in this work. The system will be easy to use with a graphical user interface and can work on a cross-platform.

III. RELATED WORKS

Steganography is an ancient art of hiding information. The word steganography is derived from the Greek words "stegos" meaning "cover" and "grafia" meaning "writing" [5] defining it as "covered writing". The idea and practice of hiding information has a long history. In Histories the Greek historian Herodotus writes of a nobleman, Histaeus, who needed to communicate with his son-in-law in Greece. He shaved the head of one of his most trusted slaves and tattooed the message onto the slave's scalp. When the slave's hair grew back the slave was dispatched with the hidden message [6]. In the Second World War the Microdot technique was developed by the Germans. Information, especially photographs, was reduced in size until it was the size of a typed period. Extremely difficult to detect, a normal cover message was sent over an insecure channel with one of the periods on the paper containing hidden information [7].

Steganography differs from cryptography in the sense that where cryptography focuses on keeping the contents of a message secret, steganography focuses on keeping the existence of a message secret [8].

Watermarking and fingerprinting are two other technologies that are closely related to steganography. These technologies are mainly concerned with the protection of intellectual property. In watermarking all of the instances of an object are "marked" in the same way. The kind of information hidden in objects when using watermarking is usually a signature to signify origin or ownership for the purpose of copyright protection [10].

With fingerprinting on the other hand, different, unique marks are embedded in distinct copies of the carrier object that are supplied to different customers. This enables the intellectual property owner to identify customers who break their licensing agreement by supplying the property to third parties [9].In watermarking and fingerprinting, the fact that information is hidden inside the files may be public knowledge; sometimes it may even be visible - while in steganography the imperceptibility of the information is crucial [8].

Steganalysis is "the process of detecting steganography by looking at variances between bit patterns and unusually large file sizes". It begins by identifying the artifacts that exist in the suspect file and it has been formed as a result of embedding a message. The goal is not to advocate the removal or disabling of valid hidden information such as copyrights, but to point out approaches that are vulnerable and may be exploited to investigate illicit hidden information [11][12][13][14]. Attacks and analysis of hidden information may get several forms like detecting, extracting, and disabling or destroying hidden information, [15]. An attacker may also embed counter information over the existing hidden information. These approaches vary depending upon the methods used to embed the information in the cover media. Some amount of distortion and degradation may occur in carriers, even though such distortions cannot be detected easily by the human perceptible system. This distortion may be anomalous to the normal carrier and when discovered it may point to the existence of hidden information. Numerous tools exist in performing steganography, and they vary in their approaches of hiding information. The detection of hidden content is quite complex without knowing which tool is to be used and which stego key is to be used. Some of the steganographic approaches have characteristics that act as signatures for the method or tool used

Steganalysis Methods
Based on the way of detecting the presence of hidden message, steganalysis methods are divided as follows [14]:
1. Statistical steganalysis.
a. Spatial domain.
b. Transform domain.
2. Feature based steganalysis

Statistical steganalysis: In order to detect the existence of the hidden message, statistical analysis is done with the pixels. It is further classified as spatial domain steganalysis and transform domain steganalysis. In the spatial domain, the pair of pixels is considered and the difference between them is calculated. The pair may be any twoneighbouring pixels. They may be selected within a block otherwise, across the two blocks. Finally, the histogram is plotted and that shows the existence of the hidden message. In transform domain, frequency counts of coefficients are calculated and then the histogram analysis is performed. With the help of this, the cover and stego images can be differentiated. However, this method does not provide information about the embedding algorithms. To overcome this problem, feature based steganalysis may be chosen [14].

Feature based steganalysis: In this method, the features of the image will be extracted for selecting and retaining relevant information. These extracted features are used to detect hidden message in an image. They can also be used to train classifiers.

Classification of Steganalysis
The steganalysis algorithm may or may not depend on the Steganographic Algorithm (SA). And based on this, steganalysis is classified as follows [14]:
1. Specific / Target steganalysis.
2. Generic / Blind / Universal steganalysis.

Specific steganalysis: The SA is known and the designing of detector (steganalysis algorithm) is based on SA. The steganalysis algorithm depends on the SA. This type of steganalysis is based on analysing the statistical properties of an image that change after embedding. The advantage of using specific steganalysis is that the results are very accurate. The disadvantage of using this method is that it is very limited to particular embedding algorithm as well as the imageformat.

Blind / Universal steganalysis: In universal steganalysis, the SA is not known by everyone. Hence, anyone can design a detector to detect the presence of secret message and that will not depend on SA. Comparing to specific steganalysis, universal is common and less efficient. Again, universal steganalysis is widely used than the specific one because it is independent of the SA. It includes the following two phases such as
(a). Feature Extraction and (b). Classification

Feature Extraction is a process of creating a set of distinct statistical attributes of an image. These attributes are known as a feature. Feature extraction is nothing but a dimensionality reduction. The extracted features must be sensitive to the embedding artifacts. Image quality metrics, wavelet decompositions, moment of image statistic histograms, Markov empirical transition matrix, moment of image statistic from spatial / frequency domain, and co-occurrence matrix are some of the feature extraction methods. While classification is a way of categorizing the images into classes depending on their feature values. Supervised learning is one of the primary classifications in steganalysis. Supervised learning allows learning under some supervision. In this learning, training input set, that includes input features, is given as input to train the classifier. After the training, class label is predicted based on the features that are given [14].

Applications of Steganalysis
a. Medical safety: Current image formats such as Digital Imaging and Communications in Medicine (DICOM) separate image data from the text (such as the patient's name, date and physician) and with the result of that the link between image and patient occasionally gets distorted by protocol converters. Hence, embedding the patient's name within the image could be a useful safety measure.
b. Terrorism: According to government officials, terrorists use to hide maps and photographs of terrorist targets and give instructions for terrorist targets.
c. Hacking: The hacker working on servershides secret message on image or audio or text file; shares it with mail or chat which will get installed and executed.It will help the hacker to do anything with the workstation.

IV. METHODOLOGY

The methodology engaged in this paper is the Object-Oriented Analysis and Design (OOAD). This is because object is the core concept involved in object-oriented language and it sees a system from the viewpoint of the objects themselves as they function and interact.

Figure 1 presents a flowchart of the steganalysis system which consist of two main parts namely:

- Image Download Module
- Image Detection Module

Image Download Module

The image download module represents all the actions taken by the user to interact with the image downloader. The image downloader software application is used to help to type URL and download the images from webpage. The user must specify the location on the computer to which the downloaded images can be saved. The user can select the type of images which he/she wants to download. The downloaded images are saved into a zip file that will be automatically downloaded after completion of process in the active tab and stored to a specific folder on the computer. These will be the images used within the steganalysis system to detect steganography.

Image Detection Module

Image detection module represents all actions taken by the user to interact with the system to detect steganography. The steganalysis system is based on the least significant bit steganography technique. The system can detect steganography in images by checking if the least significant bits of the image have been modified. Within this system, the user will load an image into the system. Once the user clicks on the image detect button, the image will be scanned using the least significant bit detection method. The system can then run the appropriate scan on the image. The result of the scan, whether the image is steganographic or not will be displayed to the user.

Java Classes and Interfaces

Graphical User Interface:GUI contains a number of buttons to interact with the system; one button to load an image and one to run the detection class. This GUI is created using a JFrame. The class used to create the JFrame is called MainMenu. The MainMenu class initializes all the variables required for this class and builds the GUI within its constructor. This class includes two methods; displayImage () to load an image into an imageicon which is then displayed on a JPanel and openImageFile () which opens a file chooser to allow the user to select an image to process. The JFileChooser class calls the imageFilter class which allows only gif, jpeg, jpg & png files to be selected.

ImageFilter Class:This class sets up the types of files which the user can select using the JFileChooser. The directory and name of the file are returned from the openImageFile method to the actionPerformed method. The JFileChooser also calls the imagePreviewPanel class to give a small preview of the images.If the close icon is pressed, JFileChooser closes and returns to the main GUI.

ImagePreviewPanel Class: This class also works with JFileChooser. It sets up a preview panel which allows the user to view all images contained within a file before selecting on to load into the system.

The user then clicks on the detect Image button which can run the detection algorithm by calling the extract class.

Extract Class: This class calls steganalysisLsbImage method which returns the hidden message as a string. It takes in the 3D array representing the image. It loops through the image extracting the least significant bitin each pixeland appends that bit to a string representing the bits.

MainMenu Class: This class contains an inner classButtonHandlerclass which contains an actionPerformed method to perform each of the actions associated with clicking on each of the buttons.

Algorithm to detect secret message using LSB:

Step1: Read the image.
Step2: Determine the height and weight of image
Step3: Calculate LSBs of each pixel of image.
Step4: Retrieve bits & convert each bit into character.
Step 5: Repeat Step 3 and 4 until the binary representation of the target character is retrieved.
Step6: The retrieved characters are placed sequentially to get back the original secret message.
Step 7:If the secret message is empty a message is displayed which states “No message was found in the chosen image.
Step 8:If the secret message does contain data, a message is displayed stating “This is a steganographic image

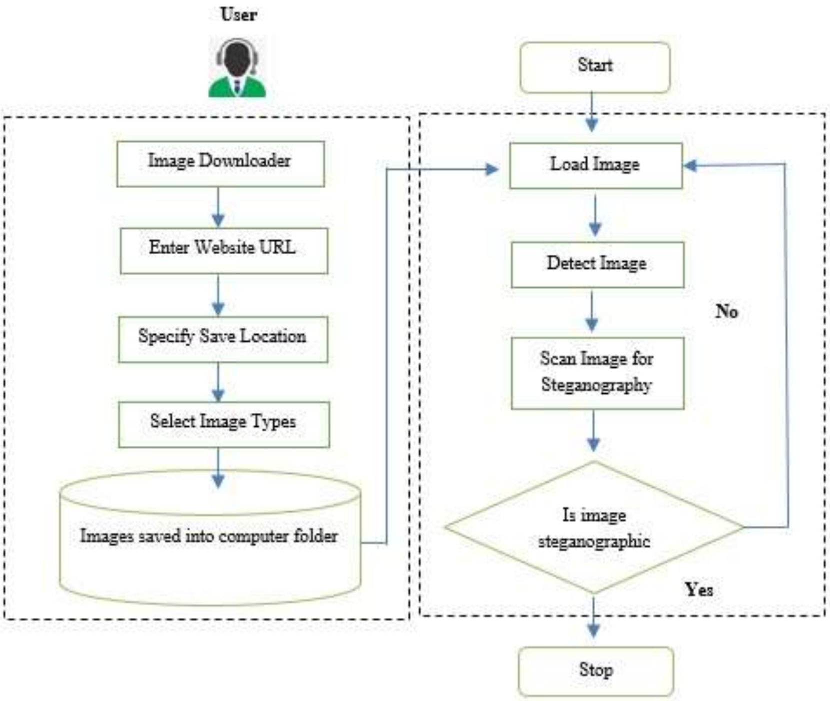


Figure 1: System Flowchart

V. SYSTEM IMPLEMENTATION

The steganalysis system contains a number of buttons to interact with the system; one button to load an image and one to run the detection class.

The user is presented with the screen shown in Figure 2:

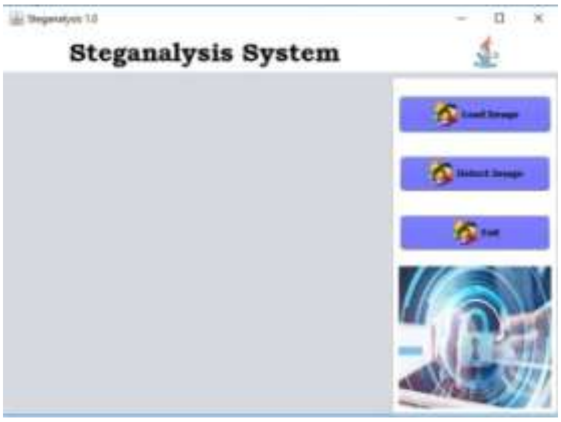
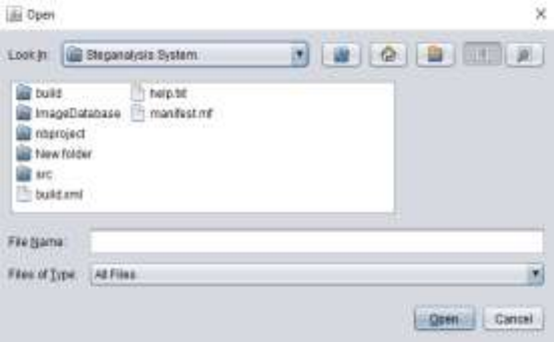


Figure 2: Initial Screen

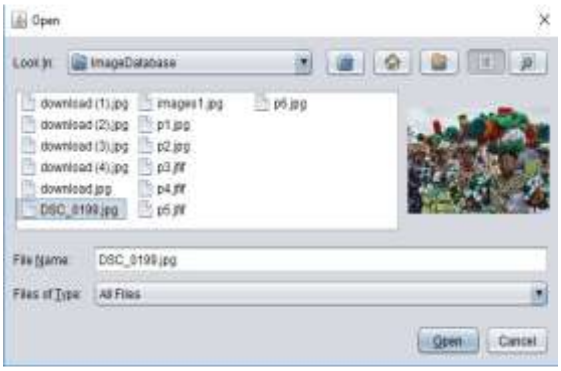
The user clicks on the “Load Image” command button. This causes the Steganalysis System to load a JFileChooser which will allow the user to select an image. This JFileChooser is shown in Figure 3.



File Name:
Files of Type: All Files

Figure 3: JFileChooser

Here, a test will also be performed to test the imagePreviewPanel class. When the user selects an image, a preview of this image should be shown on the JFileChooser with only the image files of type jpg, jpeg, gif and png should be displayed. This is illustrated in Figure 4.



File Name: DSC_0199.jpg
Files of Type: All Files

Figure 4: JFileChooser with image preview

Next, the user clicks on the "Open" button command. This will prompt the Steganalysis System to load the image within the JPanel on the system GUI. This is shown in Figure 5.

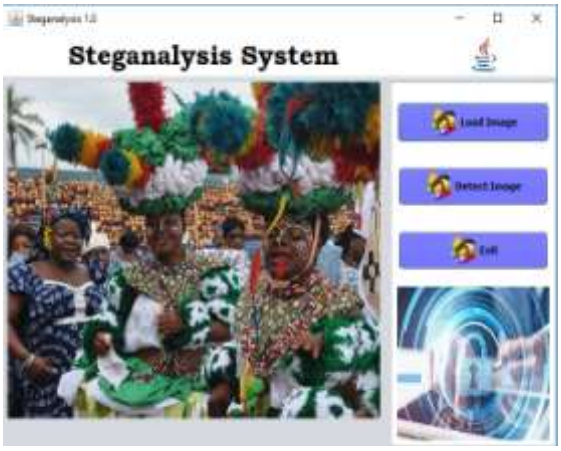
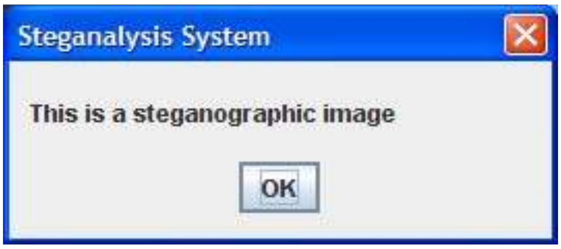


Figure 5: Selected image displayed on GUI.

VI. RESULTS AND DISCUSSION

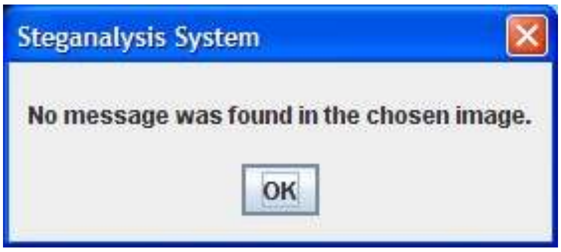
A number of images were downloaded which were known to be steganographic images. One of these images will be used within this test. The user must load one of these images in the same way described in Figure 1. Once the image has been loaded, the user will click on the detect button. This causes the Steganalysis System to run the necessary processing classes. As it is known that this image has been affected by steganography, the system should respond with a message to say it has detected steganography. The result is shown in Figure 6.


Steganalysis System

This is a steganographic image

Figure 6: Detected Steganography Message

The user must load one of these images in the same way described in test scenario 1. Once the image has been loaded, the user will click on the detect button. This causes the Steganalysis System to run the necessary processing classes. A message will be displayed to the user stating whether or not the image has been affected by steganography. Here, the result should be negative. The result is shown in Figure 7.


Steganalysis System

No message was found in the chosen image.

Figure 7: Not detect steganography message

Performance Evaluation

The system will scan sample twenty (20) images to check how many of these images test positive for hidden content in the least significant bit.

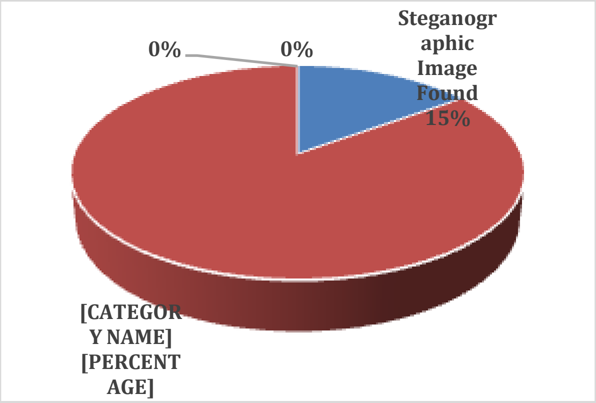


Figure 8: Performance Evaluation Results

The pie chart (Figure 8) shows that 17 of the images were found not to contain steganography whereas 3 of the images tested positive. The results demonstrated that the system can detect least significant bit steganography from known steganographic images but may be susceptible to false positives.

VII. CONCLUSION

This work was about the designing and implementing a steganalysis system. The system developed functionality was tested and evaluated to ensure that it ran without errors. The performance evaluation dealt with the actual scanning of the images from a specific website and compiled results found. Overall, it demonstrated that the system can detect least significant bit steganography from known steganographic images. At present, the system can only reveal the existence of a message. It did not cover what the hidden message is. Future work can be done to include extracting or destroying the hidden message. Also, we can expand the detection to other steganography techniques such as Discrete Cosine Transformation (DCT) and Masking & Filtering techniques.

REFERENCES

[1]. Artz, Donovan. "Digital steganography: hiding data within data." IEEE Internet computing 5.3 (2001): 75-80. program (NCEP) expert panel on detection, evaluation, and treatment of high blood cholesterol in adults (adult treatment panel III) final report. Circulation. 2002;106(25, article 3143).

[2]. Cheddad, Abbas, et al. "Digital image steganography: Survey and analysis of current methods." Signal processing 90.3 (2010): 727-752.

[3]. Rathika, B. Loganathan “Approaches and Methods for Steganalysis – A Survey”. International Journal of Advanced Research in Computer and Communication Engineering. Vol. 6, Issue 6, June 2017.

[4]. USA Today. Kelley, J. 2001. Terrorist Instructions Hidden Online. http://www.usatoday.com/tech/news/2001-02-05-binladen-side.htm

[5]. Moerland, T., “Steganography and Steganalysis”, Leiden Institute of Advanced Computing Science, www.liacs.nl/home/ tmoerl/privtech.pdf

[6]. Silman, J., “Steganography and Steganalysis: An Overview”, SANS Institute, 2001

[7]. Jamil, T., “Steganography: The art of hiding information is plain sight”, IEEE Potentials, 18:01, 1999

[8]. Wang, H & Wang, S, “Cyber warfare: Steganography vs. Steganalysis”, Communications of the ACM, 47:10, October 2004

[9]. Anderson, R.J. & Petitcolas, F.A.P., “On the limits of steganography”, IEEE Journal of selected Areas in Communications, May 1998

[10]. Marvel, L.M., Boncelet Jr., C.G. & Retter, C., “Spread Spectrum Steganography”, IEEE Transactions on image processing, 8:08, 1999

[11]. Anderson & Petitcolas 1998, 'On the limits of steganalysis', IEEE Journal on Selected Areas in Communications, vol. 16, no. 4, pp. 474-484.

[12]. Johnson & Jajodia, S 1998, 'Steganalysis: The investigation of hidden information', IEEE Information Technology Conference, pp. 113-116.

[13]. N. Johnson and S. Jajodia. “Exploring Steganography: Seeing the Unseen”, IEEE Computer, vol. 31, no. 2, pp. 26-34, 1998. foundation. Diabetes Care. 2008;31(4):811–822

[14]. Neil Provos & Peter Honeyman 2003, 'Hide and seek: An introduction to steganalysis', IEEE Security and Privacy, vol. 1, no. 3, pp. 32-44.

[15]. Colhoun HM, Betteridge DJ, Durrington PN, et al. Primary prevention of cardiovascular disease with atorvastatin in type 2 diabetes in the collaborative atorvastatin diabetes study (CARDS): multi-centre trial. The Lancet. 2004; 364(9435) :685–696.

[16]. Chandramouli, R, Kharrazi, M & Memon, NN 2004, 'Image steganography and steganalysis: Concepts and practice', International Workshop on Digital Watermarking, ed. KalkerCoxRo, Springer, Lecture Notes in Computer Science, Berlin, Heidelberg, pp. 35-49.

[17]. N. Provos. “Defending Against Statistical Steganalysis”, Proceedings of the 10th USENIX Security Symposium, vol. 10, pp. 323-335, 2001.

[18]. Westfeld, A & Pfitzmann, A 1999, 'Attacks on steganographic systems', International Workshop on Information Hiding, ed. Pfitzmann, Springer, Berlin, Heidelberg, pp. 61-76.

[19]. X. Quan, H. Zhang, and H. Dou. “Steganalysis for JPEG Images Based on Statistical Features of Stego and Cover Images”, Lecture Notes in Computer Science, vol. 4681, pp. 970-977, 2007.

[20]. N. F. Johnson and S. Jajodia, “Steganalysis: The investigation of Hidden Information”, IEEE Information Technology Conference, 1998.

[21]. Bin Li Junhui and He Jiwu Huang, “A Survey on Image Steganography and

Steganalysis", Journal of Information Hiding and Multimedia Signal Processing, ISSN: 2073-4212, 2011.

[22]. Li, Bin, et al. "A survey on image steganography and steganalysis." Journal of Information Hiding and Multimedia Signal Processing 2.2 (2011): 142-172