Research Journal of Applied Sciences, Engineering and Technology 7(22): 4806-4812, 2014
DOI:10.19026/rjaset.7.868
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: February 03, 2014 Accepted: February 10, 2014 Published: June 10, 2014

Research Article

Stego-audio Using Genetic Algorithm Approach

V. Santhi and Logeswari Govindaraju
Department of Computer Science and Engineering, PSG College of Technology,
Coimbatore 641004, India

Abstract: With the rapid development of digital multimedia applications, the secure data transmission becomes the main issue in data communication system. So the multimedia data hiding techniques have been developed to ensure the secured data transfer. Steganography is an art of hiding a secret message within an image/audio/video file in such a way that the secret message cannot be perceived by hacker/intruder. In this study, we use RSA encryption algorithm to encrypt the message and Genetic Algorithm (GA) to encode the message in the audio file. This study presents a method to access the negative audio bytes and includes the negative audio bytes in the message encoding and position embedding process. This increases the capacity of encoding message in the audio file. The use of GA operators in Genetic Algorithm reduces the noise distortions.

Keywords: Encryption, GA operators, genetic algorithm, stegnalysis algorithm

INTRODUCTION

In the vast area of network, information security becomes most challenging part. Several technologies were developed to provide secure transmission over internet. Initially, Cryptography has been widely used to transfer converted data over the internet. In Cryptography, the sender encrypts the plaintext using private/public key to produce cipher text and transfer that in the internet. The receiver extracts the plaintext by decrypting the cipher text using private/public key. But the hacker can easily intercept the cipher text and perceive the information using cryptanalysis attacks. This problem has been eliminated with the help of steganography technique (Zaidoon *et al*., 2010).

Steganography is a technique used to hide information within audio/video/image file, such a way that the hidden message is not known. Steganography must satisfy two basic requirements. The first requirement is perceptual transparency. In this, cover object (object not containing any additional data) and stego object (object containing secret message) must be perceptually indiscernible. The second constraint is high data rate of the embedded data (Samir *et al*., 2008).

Initially, the steganography algorithms were implemented on digital images and video sequences. Some of the algorithms used in image steganography are: simple LSB method, palette based image steganography, steganography on JPEG images using F5 algorithm and outguess embedding algorithm.

In LSB method, the message bits are embedded into the least significant bit of the image plane. This can affect each pixel only in the range of +1 to -1 (Chandramouli and Menon, 2001). In palette based image steganography, color lookup table are used to indicate the colors in the original image. The pixel data is index to the lookup table. After applying the LSB technique to the palette based image, the resulting image has completed with different colors because the index to the color palette is changed. The change is unnoticeable for adjacent palette entries (Morkel *et al*., 2005).

These algorithms have the advantage of less distortions and high robustness. But the main drawbacks of digital images/video sequences steganography algorithms were several malicious attacks which are available to extract the secret message. Some of the attacks are geometrical distortions, spatial scaling, raw image steganalysis, palette image steganalysis etc.

The palette image steganalysis method performs statistical analysis of the palette table and the detection is made when there is an increase in entropy (a measure of variation in palette colors). The raw image steganalysis technique makes use of the property that the number of unique colors for a high quality bit map image is half the number of pixels in the image. The ratio of close color pairs can be calculated between the original color palette and the color palette obtained after performing LSB embedding in original image. Using this information has been revealed. This introduces the concept of audio steganography (Natrajan and Lopamudra, 2010).

Corresponding Author: V. Santhi, Department of Computer Science and Engineering, PSG College of Technology, Coimbatore 641004, India

The advantage of using audio steganography is, it uses the advantages of Human Auditory System (HAS) properties to hide the message and so the available malicious attacks on digital image/video sequences that cannot be implemented against audio steganography technique. The audio file which is used to hide the message is known as cover/host audio file. The audio file which contains the hidden message is called stego-audio file (Gunjan and Puja, 2012).

Most commonly used audio steganography algorithm was substitution technique/LSB encoding. In this technique, each bit from message can be embedded into the Least Significant Bit (LSB) of the cover audio file. This technique has the advantage of hiding high capacity of information by simply modifying the LSB bit of the cover audio file. Normally in 16 KHZ sampled audio, we can embed 16 kbps data (by including all audio samples). But the resulting stego-audio file has two main drawbacks which are commonly known as remained problems of substitution technique in audio steganography (Mazdak and Azizah, 2009). They are:

- Low robustness against attacks which try to reveal or extract the hidden message.
- Low robustness against distortions which provides easy way to detect the availability of information in the audio file and destroy the hidden message.

The robustness against noise and distortion can be improved by increasing the depth of the LSB layer. According to this, several Modified LSB algorithms were introduced. Most commonly fourth LSB layer is used to insert the message bit to maintain perceptual transparency. Reduced distortion LSB coding uses the sixth layer to insert the message bit. This improves the robustness against signal processing manipulation. But this has low robustness against attacks. This is because; the LSB algorithms use the fixed layer to embed the message bit (Nedeljko and Tapio, 2002).

The remaining problem of substitution techniques are solved by using Genetic Algorithm based encoding approach. In this, first problem can be avoided by modifying bits other than LSB to embed the message bit. This introduces randomness in the message encoding that increases the intruder difficulty to extract the secret message. Second problem can be avoided by embedding the message bits into its deeper layers and other bits are altered to decrease the amount of error (Mazdak *et al*., 2009).

The Genetic Algorithm encodes the message in the higher LSB layer of the audio file that includes only the positive bytes to generate collection of chromosomes. Then by using the Genetic Algorithm operators like mutation and crossover, the next generation chromosomes are spawned. Finally, the best chromosome is selected according to the fitness value. Fitness value helps to reduce the distortion by getting a value of LSB position for which, a chromosome with the minimum deviation will be obtained when comparing to the original audio sample. This approach increases the robustness by encoding message in the deeper layers of LSB. But this method still has the noise distortions. Because the positive bytes are used to encode message and overwrite the next audio byte with the position value. This also has less embedding capacity of hidden message (Krishna *et al*., 2010).

In this study, the message is encrypted using RSA encryption algorithm and to encode message by using Genetic Algorithm based encoding approach, in which the Genetic Algorithm and fitness function uses negative audio bytes for message encoding and position embedding.

METHODOLOGY

The proposed algorithm concentrates on reducing noise, increasing robustness and increasing message embedding capacity by including negative audio bytes in the encoding process. In this study, the message can be hidden into the audio file using the following two major steps:

- Encryption using RSA
- Genetic Algorithm based Encoding

In the first step, the message in the text file is encrypted by using RSA algorithm. The sender generates the private key by using provided public key and RSA parameters. This private key is transferred to the receiver using secure connection.

In the second step, encrypted message is encoded using Genetic Algorithm. The proposed algorithm can include the negative byte in the population generation and position embedding. The negative audio byte can be processed as 32-bit sample and the positive audio byte can be processed as 16-bit sample. Initially the message bit is embedded into the audio byte in higher LSB positions (1-8) for both negative and positive audio byte. This generates the initial population. For every audio byte in the initial population, the Genetic Algorithm operators are applied to generate next generation better-chromosomes (Audio bytes). Fitness function selects the best audio byte from the next generation better-chromosomes (Audio bytes) according to the fitness value. The fitness value is the LSB position value, which is the least difference between the original audio byte and the audio bytes in the next generation. If more than one audio byte in the next generation has same difference with original audio

byte then, the audio byte with higher LSB layer can be chosen.

The following steps show that detailed explanation about how to encode the message data into the given audio file:

- The user provides the public key and RSA parameters through GUI.
- Using the public key, encrypt the message and generate private key that key will be transferred to the receiver over secure connection.
- The encrypted message is converted into binary strings array list.
- The given audio file is read by byte-wise. The positive byte can be converted into 16-bit binary string and the negative byte converted to 32-bit binary string.
- For every message bit in the binary string array list do the following:
  - Insert the message bit into audio binary string in n (1-8) positions. This generates initial population with n number of chromosomes.
  - The following Genetic Algorithm can be applied to each of chromosome in the initial population to generate next generation better-chromosomes. For negative audio byte, the Genetic Algorithm is applied only to last 16-bits in the binary string:

```
Generate Genetic Population (audiobyte, msgbit, bitposition)
{
    //bit position is the position in the audio byte where the msgbit
    //is embedded
    If bits position = 1 then
        do nothing
    else if bit position = 2 to 8 then
     if msgbit = 0 and audio bit in bit position = 0 then
         do nothing
      if msgbit = 0 and audio bit in bit position = 1 then
        if audio bit on bitposition-1 to 1 are 0's then
            do crossover operation with
                bit position-1 no's of 1..1
        if audio bit on bitposition+1 is 0
                do mutation operation on
                    bit position+1 and crossover
                     operation on bit position-1 to 1
                     with bit position-1 no's of 0..0
        if msgbit = 1 and audio bit in bit position = 1 then
              do nothing
        if msgbit = 1 and audio bit in bit position = 0 then
           if audio bit on bit position-1 to 1 are 1's then
                    do crossover operation on
                      bit position-1 to 1 with
                        bit position-1 no's of 0…0
                if audio bit on bit position+1 is 1 and
                  bit position-1 to 1 are 0's then
                    do mutation operation on
                      bit position+1 and crossover
                       operation on bit position-1 to 1
                       with bit position-1 no's of 1…1
}
```

- The next step is to select the best chromosome from next generation chromosome using fitness function. The fitness function works as follows:
  - For each chromosome in the next generation, find the difference between original audio byte and the chromosome itself.
  - Select the fitness value which represents combination of chromosome position and the chromosome. Here the chromosome has least difference with the original audio byte.
- Embed the fitness value in the next audio byte by converting both the fitness value and the audio byte into binary and then embed the binary fitness into the binary audio (negative: 32-bit and positive: 16-bit).
- Replace the original audio byte with the chosen chromosome and convert both binary audio samples into audio bytes.
- Write the audio bytes into the output audio file (host audio file).
- Mark the end of message encoding.

The fitness value reduces the distortion and increase the robustness by choosing chromosome which has least deviation with the original audio byte. So the host audio file and the original audio file more or less the same.

The decoding at the receiver side is done by extracting the position where the message bit is embedded and then the position value message bits are taken away from the audio byte.

The decoding algorithm works as follows:

- **Read the audio file byte-wise:** Convert audio byte into binary string (negative: 32-bit and positive: 16-bit).
- For getting one message bit, two audio bytes are accessed. Get the position from the second binary audio. Convert the binary into integer p and get the (8+p) Th bit from the first binary audio.
- To get 16-bits of single character 32 audio bytes are used.
- Do the steps 1 to 3 for all audio byte samples until reach the end of encoding marker.
- Convert binary messages into byte messages.
- Using the received private key and necessary parameter decrypt the decoded message byte.
- Write the decrypted original message into the output text file and the audio bytes into the output audio file.

RESULTS AND DISCUSSION

The improvement of proposed work is shown by comparing the previous work with SNR ratio and plotting the resulting stego-audio file in the time domain. Different frequencies of audio files are used to test the performance of proposed work. Each of audio files has the duration of 10 sec, 20 sec and etc. First, the SNR ratio is computed for original audio file, stego-audio file using pervious work and stego-audio file using proposed method. SNR is computed as:

$$SNR = 10.\log_{10} \{\sum_n x^2 (n) / \sum_n [x^2 (n) - y^2 (n)]\} \quad (1)$$

where,
x (n) : Sample of input audio sequence
y (n) : Sample of audio with GA based LSBs

The following table and graphs represent the tested results of first sample audio files with the different frequencies.

In Table 1, four different frequencies of same audio files are taken and the SNR values for each audio file have been calculated using Eq. (1). This is represented in the Fig. 1.

Figure 2 represents the waveform of original file, stego-audio file using proposed work and previous work. The original audio file and the stego-audio file of proposed work are more or less the same when compared with the stego-audio file of previous work. This can be analyzed with the help of higher SNR values of proposed work stego-audio file which is present in Table 1. This increases the intruder difficulty in revealing the hidden message. By including the negative audio bytes for the population generation and embedding fitness value, the capacity of embedded information becomes high. The capacity is based on the duration of audio file.

Table 1: Audio file with 10 sec duration

Audio file sample number and frequency	Previous work SNR (t)	Proposed work SNR (t)
1 (11.025 KHz)	5.4406	19.7745
2 (22.050 KHz)	6.0553	20.4984
3 (44.100 KHz)	5.7855	20.4184
4 (48 KHz)	6.1372	20.8227

Table 2: Audio file with 20 sec duration

Audio file sample number and frequency	Previous work SNR (t)	Proposed work SNR (t)
1 (11.025 KHz)	5.8244	23.3684
2 (22.050 KHz)	8.8788	26.2213
3 (44.100 KHz)	6.6297	24.0865
4 (48 KHz)	8.8788	26.2213

Table 3: Audio file with 30 sec duration

Audio file sample number and frequency	Previous work SNR (t)	Proposed work SNR (t)
1 (11.025 KHz)	7.5708	26.8588
2 (22.050 KHz)	7.7100	31.5019
3 (44.100 KHz)	8.1835	30.0743
4 (48 KHz)	14.1345	36.5839

The SNR values for second sample audio file with different frequencies are shown in Table 2.

In the Fig. 3, the SNR values of stego-audio file are higher than previous work. This represents proposed work has less or no noise. This SNR graph is plotted using data present in Table 2.

In the Fig. 4, the waveform of proposed work stego-audio file is more or less equal to the waveform of original file. This is because of improvement of SNR values of proposed work, which is referred in Table 2. This represents the reduction of noise distortion in the stego-audio file. By including both positive and negative bytes the capacity of stego-audio file is increased.

The SNR values for third sample audio file with different frequencies are shown in Table 3.

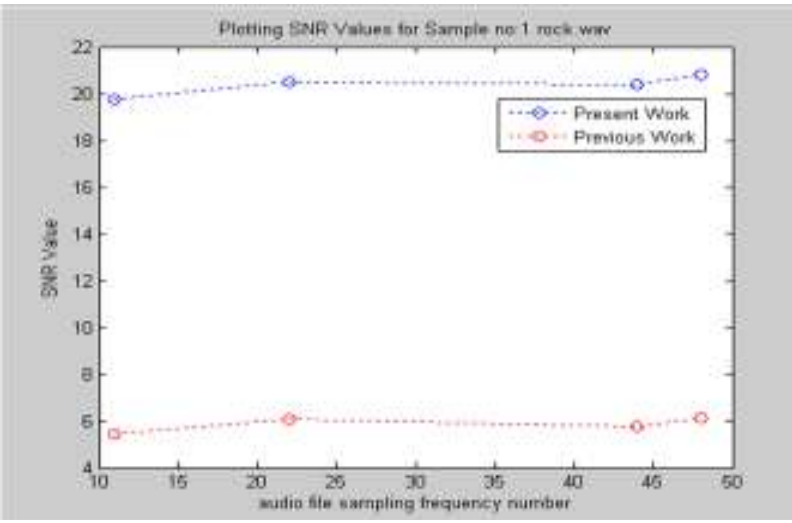

Fig. 1: SNR comparison (10 sec duration)

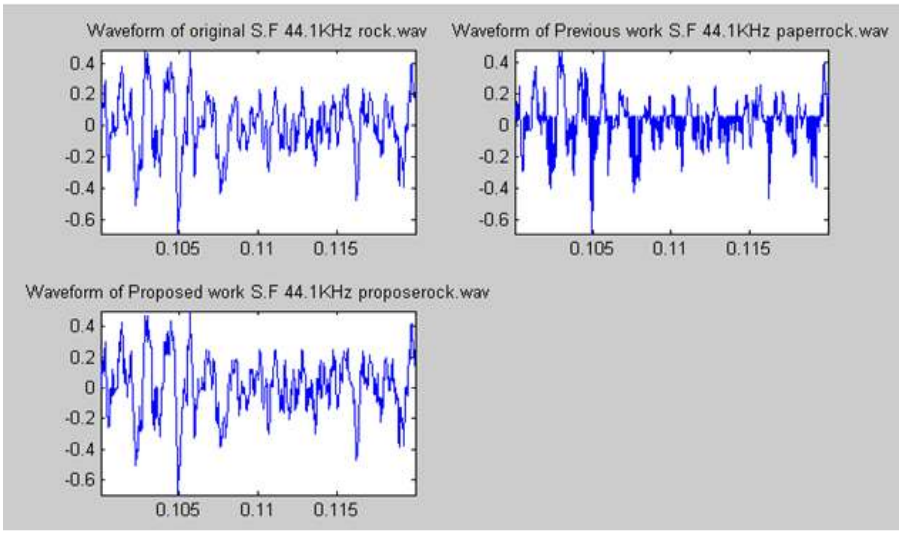

Fig. 2: Wave file in time domain (10 sec duration)

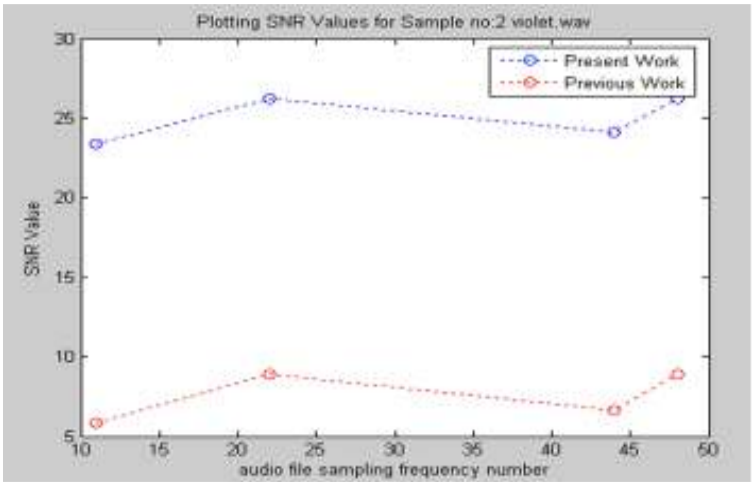

Fig. 3: SNR comparison (20 sec duration)

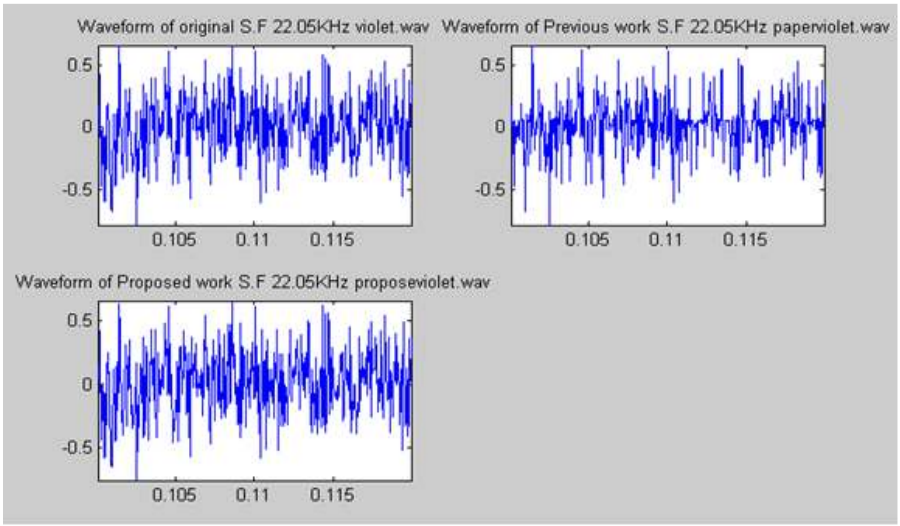

Fig. 4: Wave file in time domain (20 sec duration)

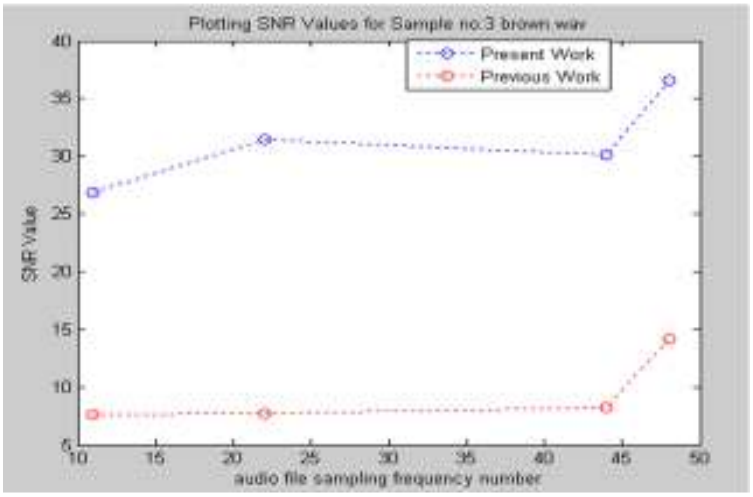

Fig. 5: SNR comparison (30 sec duration)

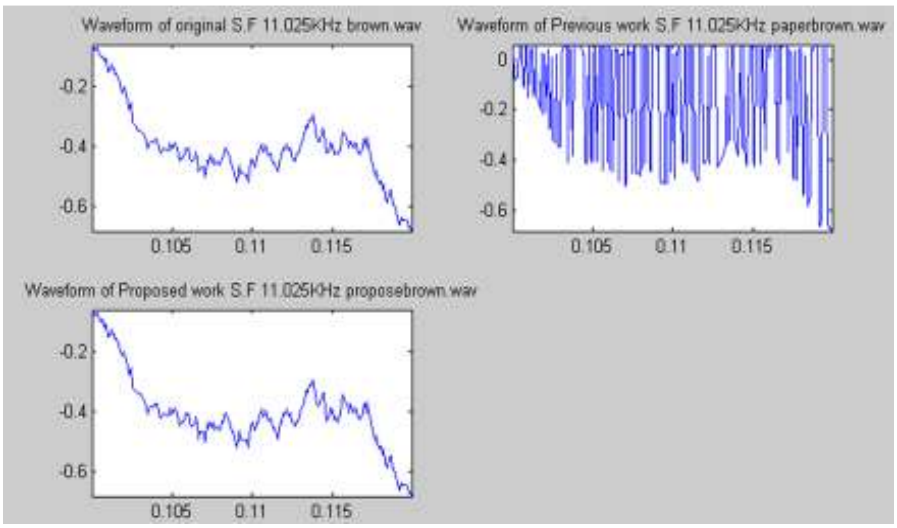

Fig. 6: Wave file in time domain (30 sec duration)

In the Fig. 5, the previous work of SNR value is varying than the proposed work. Because the stego-audio file of the previous work has more noise when compared to proposed work. This SNR graph is plotted using data present in Table 3.

In the Fig. 6, the waveform of the previous work has more noise because of lesser SNR values, which is referred in Table 3. The stego-audio file of proposed work is more or less equal to the original audio file. In the sample, the message bits are embedded in the higher LSB layer of both positive and negative audio bytes.

CONCLUSION

The proposed technique ensures that large capacity of message can be embedded and drastically reduce the noise distortion. This can be done by including the negative audio bytes in the message encoding and position embedding. Steganalysis attack on proposed technique is very difficult, because more than one bit in the cover audio file has been changed to produce stego-audio file. The resulting stego-audio file has higher SNR ratio. The higher SNR ratio represents that the stego-audio file has less or no noise distortion. Therefore the resulting stego-audio file is more or less equal to the cover audio file.

REFERENCES

Chandramouli, R. and N. Menon, 2001. Analysis of LSB based image steganography techniques. IEEE-2001, pp: 1019-1021.

Gunjan, N. and D. Puja, 2012. A detailed look of audio steganography techniques using LSB and genetic algorithm approach. Int. J. Comput. Sci. Issues, 9(1/2): 403-405.

Krishna, B., J.P. Anindya, S. Geetam, P.P. Tomar and Sarkar, 2010. Audio steganography using GA. Proceeding of the International Conference on Computational Intelligence and Communication Networks, pp: 449-453.

Mazdak, Z. and A.M. Azizah, 2009. Knots of substitution techniques of audio steganography. Proceeding of the International Conference on Computer Engineering and Applications, 2: 371-374.

Mazdak, Z., A.M. Azizah, B.A. Rabiah and M.Z. Akram and M. Zeki, 2009. A genetic-algorithm-based approach for audio steganography. World Acad. Sci. Eng. Technol., 30: 360-363.

Morkel, T., J.H.P. Eloff and M.S. Olivier, 2005. An overview of image steganography. Proceeding of the 5th Annual Information Security South Africa Conference (ISSA, 2005). Sandton, South Africa.

Natrajan, M. and N. Lopamudra, 2010. Steganalysis algorithms for detecting the hidden information in image, audio and video cover media. Int. J. Netw. Secur. Appl., 2(1): 43-46.

Nedeljko, C. and S. Tapio, 2002. Increasing robustness of LSB audio steganography by reduced distortion LSB coding. Proceeding of the 5th IEEE International Workshop on Multimedia Signal Processing. St. Thomas, pp: 336-338.

Samir, K.B., B. Debnath, G. Debashis, M. Swarnendu and D. Poulami, 2008. A tutorial review on steganography. Proceeding of the International Conference on Contemporary Computing (IC3-2008). Noida, India, August 7-9, pp: 105-114.

Zaidoon, K.A.A., A.A. Zaidan, B.B. Zaidan and O.A. Hamdan, 2010. Overview: Main fundamentals for steganography. J. Comput., 2(3): 158-159.