Data Hiding in Audio Signals using Elliptic Curve Cryptography, Huffman Code Algorithm and Low-Bit Encoding

Richard Apau
Department of Computer Science
Kwame Nkrumah University of Science and Technology, Kumasi, Ghana

Samuel Adu Gyamfi
Department of Information Technology Education
University of Education, Winneba, Ghana

ABSTRACT

Information security in this era has become very necessary for many aspects of life. The importance of security and privacy of data in the field of data communication has been discussed for many years. The discussion is as a result of the vulnerabilities associated with the use of the Internet and the significant strive computer technology has made in data communication. In this study, a method of audio file protection is proposed using audio steganography, elliptic curve cryptography (ECC), Huffman Code Algorithm and low-bit encoding. The novelty and the uniqueness of this proposed mechanism aims to improve the security and secrecy of audio data file transmitted over the untrusted internet medium. Audio files to be transmitted are first compressed to achieve reduced file size using Huffman code algorithm. The compressed audio files are then encrypted using elliptic curve cryptography. The compressed encrypted audio file is therefore embedded in the appropriate frame of the converted audio signal based on the histogram values using low-bit encoding scheme. The proposed system recorded low values of Bit Error Rate (BER) and Mean Square Error (MSE), high values of segmented signal-to-noise ratio (SNRseg), high values of payload capacity which far exceed the recommended threshold for audio steganography techniques. From the results obtained it can be concluded that, the study achieved higher level of security, payload capacity and robustness. The study experienced negligible level of distortion with high recorded values of SNRseg and low values of BER and MSE.

General Terms

Audio Signal, Huffman Code, Data Communication, Security

Keywords

Steganography, Cryptography, Low-Bit Encoding, ECC, PSNR, SNRseg, MSE, BER.

1. INTRODUCTION

Information security in modern era is becoming very necessary for many aspects of human life. Information will be highly valued if the information is original and securely received by concerning parties [1]. Therefore, securing information is important. There are several methods in securing information; one of the most popular is the use of Steganography and Cryptography [2, 3]. Steganography and cryptography are increasingly required to provide maximum security in the process of information delivery service. In recent years, the need to label or protect data has taken hold because of the distribution of those files over the Internet. The entertainment industry is exposed to a challenging environment. The production and distribution of digital entertainment media (music, text and video) through the Internet, while beneficial to the industry, has also made digital piracy which is the unauthorized distribution of entertainment media significantly easier. A significant proportion of Internet users are engaged in the illicit sharing of media through peer-to-peer (P2P) networks and file-sharing applications such as µTorrent [4]. A recent example of this problem is the controversy regarding piracy of high-quality music across the Internet in MPEG Layer III, best known as MP3 format [5]. The rapid spread in digital data usage in many real-life applications has resulted in new and effective ways to ensure their security [6]. Currently, any type of data, such as text, images, and audio, can be digitized, stored indefinitely, and transmitted over the Internet at high speeds. Notwithstanding these advantages, digital data also has its own downside. They are easy to access illegally, tamper with, and copy. There is therefore a need to hide secret identification inside certain types of digital data [7, 8].

2. REVIEW OF LITERATURE

In modern times, steganographic discussions have gained prominence and received much attention among researchers and scholars. The discussion has also gained much recognition and exposition among agencies that provide intelligence services, information technology experts, information technology companies as well as print and electronic media. Fundamentally, steganographic applications operate on the principles of imperceptibility. This presupposes that, under no circumstances should a message hidden in a cover object be detected by the human eye. That notwithstanding, this basic feature underpinning steganographic applications have been rendered vulnerable as a result of statistical steganalysis. To this end, cryptography was introduced and envisaged to enhance the provision of data security.

2.1 Concepts of Steganography and Cryptography

According to [9], steganography is the science and art of hiding messages in a way that is not detectable and recoverable by a third party except an authorized recipient. Steganography technique has many methods that can be used to hide the data in a medium. In communicating secret file, steganography uses multimedia carrier files, example of which include; image, video, audio, text and IP/Network protocols [9]. In recent times, steganographic approaches that have been used widely include image steganography, audio steganography, text steganography, video steganography as well as digital watermarking [4]. Cryptography was introduced as a result of the seemingly exploitation of steganography through the use of steganalysis techniques that detect the presence of a message in a hidden object [10]. According to [11], cryptography is the process of sending or transmitting data securely over a communication medium in ensuring that, the intended recipient only process and reads the content of the message. Several fields such as cryptology, cryptosystems, cryptanalysis, symmetric and public key cryptography all exist in cryptography. Essentially, cryptography is categorized into three according to [12], based on the number of keys used for encryption. The reason is that each of these three schemes (categorisation) is optimized for some specific application(s). Steganography and cryptography are closely related. Cryptography scrambles messages so that the message cannot be read and understood. Steganography on the other hand, hides the message so there is no knowledge of the existence of the message as stated by [13]. As such, steganography works towards masking the existence of a secret file while cryptography concerns itself with masking the content of a secret file [7].

2.2 Overview of Audio Steganography

The main goal of steganography is to communicate securely in a completely undetectable manner as stated by [14] and to avoid drawing suspicion to the transmission of a hidden data according to [15]. Steganography is not only meant to prevent others from knowing the secret information, but it also prevents others from thinking that the information even exists. According to [16], if a steganography method causes someone to suspect secret information is being carried, then the method has failed. The basic model of audio steganography consists of *Carrier* (Audio file), *Message* and *Password*. Carrier, also known as a cover-file, conceals the secret information. The message is the data that the sender wishes to make it confidential before it reaches the receiver. The message can be any form of multimedia. It can be in the form of plain text, image, audio or video file. Password is known as a stego-key, which ensures that only the recipient who knows the corresponding decoding key will be able to extract the message from a cover-file. The cover-file with the secret information is known as a stego-file. The information hiding process consists of the following steps according to [17, 18], Identification of redundant bits in a cover-file. Redundant bits are those bits that can be modified without corrupting the quality or destroying the integrity of the cover-file. To embed the secret information in the cover file, the redundant bits in the cover file are replaced by the bits of the secret information. The basic model for audio steganography is shown in fig 1.

2.3 Huffman Code Compression Algorithm

Huffman code is a type of lossless compression scheme that allows reduction in size of file without loss of data [7]. The algorithm belongs to the variable code length family. Huffman’s algorithm is an example of ‘greedy’ algorithms which use small-grained or local minimal or maximal choices in an attempt to result a global minimum or maximum. In the algorithm, each character in the text file is stored as eight bits digits of 0s or 1s that map to the character using ASCII encoding scheme. The algorithm breaks down the 8 rigid structures in that, the characters that are most commonly used are stored using few bits. For instance, the character “a” which is **"01100001"** in ASCII encoding could be **"10" or "1000"** in Huffman encoding. The purpose of Huffman code encoding is to create much smaller file than the original. The process of Huffman code algorithm can be grouped into two parts: Encoding and Decoding, comprising of 10 simple consecutive steps. The steps to follow to encode and decode file in Huffman are as follows: **Step 1:** Count the frequency of each character in the file to be encoded. Assuming we want to encode the text file **"ab ab cab,",** then “a” has a frequency of 3, “b” has 3 frequency, “c” has 1 frequency, “space” has 2 frequency and End Of File (EOF) has 1 frequency. **Step 2:** Store characters as tree nodes and put them into a priority queue. The example **"ab ab cab"** would have a priority queue that looks like: {'c':1, EOF:1, ' ':2, 'a':3, 'b':3}. **Step 3**: Begin to build your tree. Start by de-queuing the two most urgent things from the priority queue. Our priority queue becomes {' ':2, new node:2, 'a':3, 'b':3}. **Step 4:** Finish building your tree**.** When you have finished, the last node in the queue will be the *root* of the encoding tree and the most frequently used characters will be the leaves closest to the top of the tree.

Step 5: Create an encoding map. Start at the root by visiting the left child root and then the left child node. For the example, the map will look like this: {' ':"00", 'a':"10", 'b':"11", 'c':"010", EOF:"011"}. **Step 6**: In the output file, include the encoding map as a header in order for the file to be encoded. **Step 7**: Encode the file**.** For each character in the file to be encoded, write the binary sequence you have stored in the map. For the file **"ab ab cab",** the encoded file will look like this: **"1011001011000101011011". Step 8:** Read in a Huffman-encoded file by first reading the header. **Step 9:** Read in the binary one digit at a time by traversing the trees as the reading goes on, this is because each character has a prefix property. **Step 10**: Repeat until you reach the End Of File (EOF). The basic reasoning and idea behind Huffman code is to build a tree bottom-up with leaves labelled as weights. For example, Let us consider this ASCII fixed length code.

char	ASCII	bit pattern (binary)
h	104	01101000
a	97	01100001
p	112	01110000
y	121	01111001
i	105	01101001
o	111	01101111
space	32	00100000

The string **"happy hip hop"** would be encoded in ASCII as **104 97 112 112 121 32 104 105 112 32 104 111 112** with a stream bits of 104 bits as **01101000 01100001 01110000 01110000 01111001 00100000 01101000 01101001 01110000 00100000 01101000 01101111 01110000**

Now let us consider the variable length code below:

char	bit pattern
h	01
a	000
p	10
y	1111
i	001
o	1110
space	110

The same message string **"happy hip hop"** would be encoded as a 34 bits variable length code as follows: **01 000 10 10 1111 110 01 001 10 110 01 1110 10.** This has therefore saved a total memory space of 67.3% in comparable to the ASCII fixed length encoding.

4

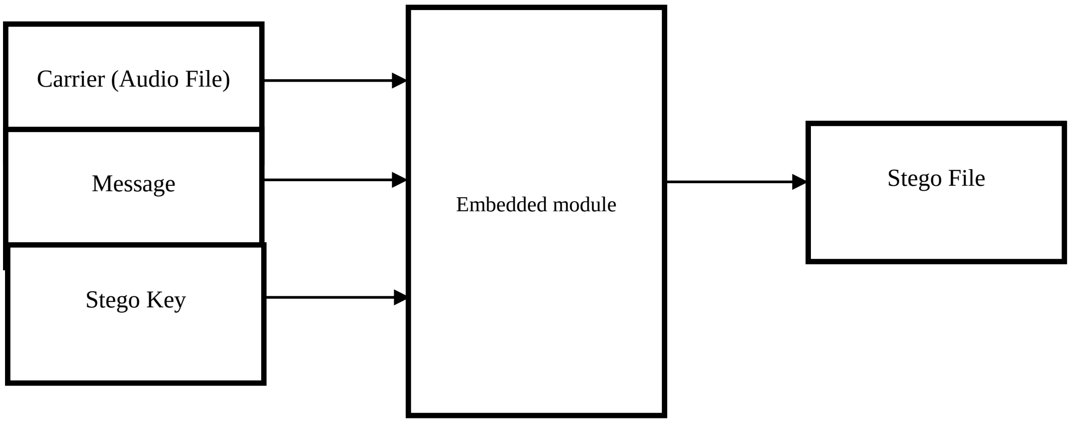

Figure 1: Basic Audio Steganographic Model

Source: Adopted from Shirali, 2008.

2.4 Basics of Elliptic Curve Cryptography

Elliptic Curve Cryptography (ECC) is a public key cryptography. In public key cryptography each user or the device taking part in the communication generally have a pair of keys, a public key and a private key, and a set of operations associated with the keys to do the cryptographic operations. Only the particular user knows the private key whereas the public key is distributed to all users taking part in the communication. Some public key algorithm may require a set of predefined constants to be known by all the devices taking part in the communication. 'Domain parameters' in ECC is an example of such constants. Public key cryptography, unlike private key cryptography, does not require any shared secret between the communicating parties but it is much slower than the private key cryptography. One main advantage of ECC is that it is a small key size. A 160-bit key in ECC is considered to be as secured as 1024-bit key in RSA.

The mathematical operations of ECC is defined over the elliptic curve

$\boldsymbol{y^2 = x^3 + ax + b}$ where $\boldsymbol{4a^3 + 27b^2} \neq 0$. Each value of the 'a' and 'b' gives a different elliptic curve. All points (x, y) which satisfy the above equation plus a point at infinity lies on the elliptic curve. The public key is a point in the curve and the private key is a random number. The public key is obtained by multiplying the private key with the generator point G in the curve. The generator point G, the curve parameters 'a' and 'b', together with few more constants constitute the domain parameter of ECC. Apart from the curve parameters 'a' and 'b', there are other parameters that must be agreed by both parties involved in secured and trusted communication using ECC. There are several standard domain parameters defined by Standard for Efficient Cryptography. Generally, the protocols implementing the ECC specify the domain parameters to be used. Domain parameters for EC over field $F_{p.}$ The domain parameters for Elliptic curve over F_p are p, a, b, G, n and h. p is the prime number defined for finite field F_p . a and b are the parameters defining the curve y^2 mod p= x^3 + ax + b mod p. G is the generator point (x_G, y_G), a point on the elliptic curve chosen for cryptographic operations. n is the order of the elliptic curve. The scalar for point multiplication is chosen as a number between 0 and n – 1. h is the cofactor where h = #E(F_p)/n. #E(F_p) is the number of points on an elliptic curve. Domain parameters for EC over field $F_2{}^m$. The domain parameters for elliptic curve over $F_2{}^m$ are m, f(x), a, b, G, n and h. m is an integer defined for finite field $F_2{}^m$. The elements of the finite field $F_2{}^m$ are integers of length at most m bits. f(x) is the irreducible polynomial of degree m used for elliptic curve operations. a and b are the parameters defining the curve y^2 + xy = x^3 + ax^2 + b. G is the generator point (x_G, y_G), a point on the elliptic curve chosen for cryptographic operations. n is the order of the elliptic curve. The scalar for point multiplication is chosen as a number between 0 and n – 1. h is the cofactor where h = #E($F_2{}^m$)/n. #E($F_2{}^m$) is the number of points on an elliptic curve.

2.5 Comparative Advantages of Elliptic Curve Cryptography

The most commonly used algorithms in public key cryptography are Elliptic Curve Cryptography (ECC), RSA Algorithm, Deffie-Hellman, and Digital Signature Algorithms (DSA). In the public key cryptography, two keys are used for encryption and decryption. The public key which is freely distributed is used for encryption, whereas the private key which is kept secret is used for decryption. In evaluating the strengths and weaknesses of the most commonly used algorithm in public key cryptography, some of the algorithms use factorization, others use Discrete Logarithm Problem (DLP). Some of the algorithms are also based on Digital Envelop (E/D), Digital Signature (D.S) and Key Exchange (KEX), whereas others are not based on all the parameters mentioned. The parameters as shown in table 1 indicate strong security for Elliptic Curve Cryptography and RSA algorithms. In this study, the Elliptic Curve Cryptography is chosen over the RSA algorithms based on its obvious security advantage. In comparing ECC to RSA, ECC is faster, safer and requires less resource. Both RSA and ECC are all leading schemes of Digital signature. Whereas ECC is based on richness of algebraic features of elliptic curve over finite fields, RSA is based on the assumption of factorization of large numbers, with factors being only two large prime numbers. ECC offers security with smaller key sizes, faster computation, lower power consumption as well as memory and band width saving.

Table 1: Key Parameters of Public Cryptographic Algorithms

ALGORITHM	E/D	D.S.	KEX	HARDNESS
RSA	Yes	Yes	Yes	Factorization
DSA	No	Yes	No	Discrete Logarithm Problem (DLP)
Deffie-Hellman	No	No	Yes	Discrete Logarithm Problem (DLP)
Elliptic Curve	Yes	Yes	Yes	Elliptic Curve-DLP

2.6 Review of Related Works

Ahmed et al [19] provided a novel audio steganography techniques to increase the capacity and robustness of low bit audio steganography using noise gate software logic algorithm. The system used the noise gate logic algorithm to obtain the desired signal for embedding the message in the host file. The 8th LSB layer embedding is applied to the desired signal. Parthasarathi and Shreekala [20] proposed secured data hiding in audio files using audio steganography algorithm. The secret message is hidden in the 4th and 5th bits of the audio sample. Earlier, Padmashree and Venugopala [21] had proposed a similar method of audio steganography and cryptography: using LSB algorithm at 4th and 5th LSB layers. This system first encrypt the secret data using RSA algorithm and apply LSB embedding to the 4th and 5th bits of the audio sample frame. Malviya et al [22] proposed an audio steganography by different methods. The method replaces the LSB bits of the audio with the LSB bit of the embedded message to achieve secret communication. Dengre et al [4], proposed Audio Steganography based on LSB insertion with Image Watermarking using AVI video. In this method, a video file separated into audio and frames, the wave file of the extracted audio is selected for embedding using LSB. The secret file to be embedded is encrypted with symmetric key cryptography using DES algorithm, Rijndael algorithm, RC2 algorithm, Triple DES algorithm. Asad et al [23] proposed enhanced least significant bit modification technique for audio steganography. Firstly, the system randomize bit number of host message used for embedding secret message while the second way is to randomize sample number containing next secret message bit. The improvised proposed technique works against steganalysis and decreases the probability of secret message being extracted by an intruder. Advanced Encryption Standard (AES) with 256 bits key length is used to secure secret message in case the steganography technique breaks. Al-Othmani et al [24] conducted a survey of audio steganographic techniques. The survey concluded prioritizing the importance of communication and security characteristics such as data rate, bandwidth, robustness, and noise audibility, must be done before choosing the steganographic technique which should completely fits the nature, environment and requirements of the application. Similarly, Djebbar et al [6] conducted a survey of audio steganographic techniques and concluded that, an SNR below 20 dB, generally denotes a noisy audio signal, while an SNR of 30 dB and above indicates that the audio signal quality is preserved.

3. METHODS

The proposed method in this study is categorized into three (3) phases. The process of compression/decompression is applied as the first phase of the proposed model. The secret file to be encrypted is first compressed while the recipient at the destination also decompressed the file after decryption. The second phase of the proposed model is the encryption/decryption. The process of encryption converts the secret file to be sent to binary data, whereas the recipient undergoes the process of decryption to reveal the content of the encrypted file. In [7], the process of encryption was applied before compression. This study however, applies the process of compression before encryption. The model proposed in [7] does not allow the file to be significantly compressed. This is because encryption turns the file into high entropy data, in the form of random streams. Compression also relies on patterns to get any reduction in the size of any file [25]. Because encryption destroys the patterns, compression algorithm after encryption is not able to give any significant reduction in size. However, compression before encryption increases the practical resistance of the file against differential cryptanalysis since the resulting output is difficult to deduce [25]. The third phase of the proposed model deals with the process of embedding/de-embedding. This process concerns itself with the hiding of the secret file into the cover object whereas the recipient at the destination undergoes the process of de-embedding by extracting the file from the cover object.

3.1 Proposed System

The proposed system has been categorized into three main phases. Phase one deals with the process of compression using a standard lossless compression algorithm, Huffman Code. The second phase comprises of cryptography employing the Elliptic Curve Cryptographic (ECC) algorithm, whereas Steganography is applied as the third and final phase to hide the existence of the file. Audio steganography is used in this proposed system. In this study, the steps proposed by [26] for point adding and point doubling in Elliptic Curve Cryptography (ECC) are adopted.

For instance, in calculating **k*P=Q,** where P and Q are points (set E) on an elliptic curve: $\mathbf{y^2= x^3 +ax +b}$. Operation **"*"** denotes the series of Point doubling and Point adding.

Point adding

For given two points $\mathbf{P(x_p, y_p), Q(x_q, y_q)}$ $\mathbf{(P\neq\pm Q)}$ in the set E, the group operator will allow us to calculate a third point $\mathbf{R(x_r, y_r)}$, also in the set E, such that **P + Q = R**. Not difficult to find the coordinates of point **R:** $\mathbf{x_r = s^2 - x_p - x_q}$ where $\mathbf{s^2 = 2x_p + x_q + x_r - x_p}$. As point R belongs to the straight line (PQ) then $\mathbf{s= y_r - y_p / x_r - x_p}$ and we find: $\mathbf{y_r = y_p + s(x_r - x_p)}$

Point doubling

For given point $\mathbf{P(x_p,y_p)}$ in the set E, the group operator will allow us to calculate a third point $\mathbf{R(x_r,y_r)}$, also in the set E, such that **P + P = 2P= R.** $\mathbf{x_r = s^2 - 2x_p}$ where $\mathbf{s= 3x^2_p - a/2y_p}$ and $\mathbf{y_r= y_p + s\ (x_r - x_p)}$

Point multiplication

One of the most important operations for all applications of elliptic curves is scalar multiplication. Scalar multiplication consists of computing the value of a large integer multiplied by a point by doing a series of point doublings and additions until the product point is reached. In this study, the approach for computing **k*P** introduced by [27] is used

Algorithm: Binary method

INPUT: An integer k>0 and a point P.

```
OUTPUT: $Q=k*P$
1. Set k $\leftarrow (k_{l-1}\ldots k_1k_0)_2$
2. Set $P_1 \leftarrow P$, $P_2=2P$.
3. for I from l-2 down to 0 do
    If $k_i=1$ then
        Set $P_1 \leftarrow P_1 + P_2$, $P_2 \leftarrow 2P$.
    Else
        Set $P_2 \leftarrow P_2+P_1$, $P_1 \leftarrow 2P_1$.
4. RETURN ($Q=P_1$)
```

3.2 Proposed Model

The model proposed in this study consists of twelve (12) consecutive steps. The first step has to do with the input of the original file or data into the developed application. In the second step, compression is applied to reduce the size significantly for optimize bandwidth consumption. In this process, a standard compression algorithm is applied to achieve a reduced file size. Encryption is applied as a third step. This involves the conversion of the data or file in the form of binary data using a standard encryption algorithm. In the fourth step, the audio signal is converted into frames to begin the process of embedding. In the fifth step, an appropriate frame is selected whereby the compressed encrypted data is hidden using a standard embedding algorithm. The converted frame is reconstructed to obtain the stego audio as the sixth step. The seventh step involves the transmitting of the secret file to the recipient over a communication channel, for example, the Internet. At the destination, the recipient undergoes the process of converting the stego audio into frames again in order to get the frame holding the compressed encrypted file or data. This constitutes the eighth step. In the ninth step, the secret data is extracted from the holding frame. The tenth step decrypts the data using the recipient's private key in a process called decryption. At the eleventh step, the data is decompressed to obtain the original data and the original input file is obtained as the last step, the twelfth step. Fig. 2 demonstrates the proposed model in this study.

3.3 Parameters for Evaluation

The proposed system is evaluated on four key parameters. The parameters adopted for the evaluation of the proposed system include; imperceptibility, robustness, payload capacity and real-time communication. The fig 3 illustrates the criteria used in evaluating the proposed system in this study.

3.3.1 Imperceptibility (Im)

The most important requirement for all steganographic applications is the imperceptibility, as it demonstrates the strength of the system and its ability to be unnoticed by human eyes. If the cover object is detected to contain a message either visually or acoustically, the security of the system is broken, and the application is rendered needless. A steganographic applications is described as imperceptible, if it is highly impossible to distinguish the cover object from the hidden secret data. In evaluating the imperceptibility of the steganographic application, Segmented–Signal-to-Noise Ratio (SNRseg), Bit Error Rate (BER) and Mean Square Error (MSE) were used. The SNRseg is the average representation of all Signal to Noise Ratios (SNRs) of all modified audio signal frames. The value obtained in the calculation of SNRseg indicates the amount of distortion induced by the embedded data in the cover audio signal. The value of SNRseg is obtained in decibels (dB). The Segmented Signal-To-Noise-Ratio (SNR*seg*) is calculated using the formula:

$$SN\,Rseg = \frac{10}{M}\sum_{m=0}^{M-1}\log_{10}\sum_{i=N_m}^{N_m+N-1}\left(\frac{\sum_{i=1}^{N} x^2(i)}{\sum_{i=1}^{N}\left(x(x)-y(i)\right)^2}\right)$$

Where **N** and **M** are the segment length and the number of segments respectively**, x(i)** and **y(i)** are the original and processed audio signal samples indexed **i .**

The MSE represents the average of the squares of the "errors" between the original audio and the stego/embedded audio. The error is the difference in value between the original audio and the stego audio. The Mean Square Error (MSE) is calculated using

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j)-K(i,j)]^2$$

where **M** and **N** are the Height and the Width of the audio respectively.

The Bit Error Rate 'BER' is the number of erroneous bit received over the total number of the transmitted bits. The higher the BER value the poorer the performance of the system. BER is calculated using **BER= 1/PSNR**. PSNR is an audio quality measure by comparing the original audio to the stego audio. The unit of measurement of PSNR is decibels (dB). The higher the PSNR value the quality the audio. PSNR is calculated using:

$$PSNR = 10.\log_{10}\left(\frac{MAX_I^2}{MSE}\right)$$

where MAX_I is the maximum possible audio sample. $MAX_I = 2^B-1$, B is the bits per sample.

3.3.2 Robustness (Rb)

Robustness of the steganographic techniques illustrates how strong the application is against changes. Robustness also ensures the capabilities of the embedded data to withstand intentional and unintentional manipulations. The data hidden in the cover object should be very hard to alter, eliminate and modify without making any changes to the cover object. It is an important criterion for all steganographic applications to be robust against manipulations and statistical analysis. Steganographic applications should withstand data manipulations such as compression or rotation. The quality of the system against such manipulations is dependent on the techniques of methods used in embedding the data in the audio signal.

3.3.3 Payload Capacity (PC)

Payload capacity is the size of embedded data that can be hidden into a particular innocent cover medium relative to the size of this medium. The challenge and difficulty associated with steganographic methods is how to embed as many larger files as possible without losing the quality of the hidden medium to compromise on the imperceptibility requirement. The payload capacity is measured in bits per pixels (bps) whereas the maximum capacity for which a steganograpic system can endure is measure in percentage terms. The Payload Capacity is calculated using the following formula:

3.3.4 Real-Time Suitability (RTS)

In real time communication, steganography using audio signals must involve some other requirements including system complexity, throughput, bandwidth, delay, absence of duplications, failure recovery, and service setup time. The requirements have effect on the communication processes in real time and may have significant influence on the real time steganographic processes.

PAYLOAD CAPACITY = NUMBER OF BITS USED TO HIDE DATA
TOTAL NUMBER OF BITS IN AUDIO

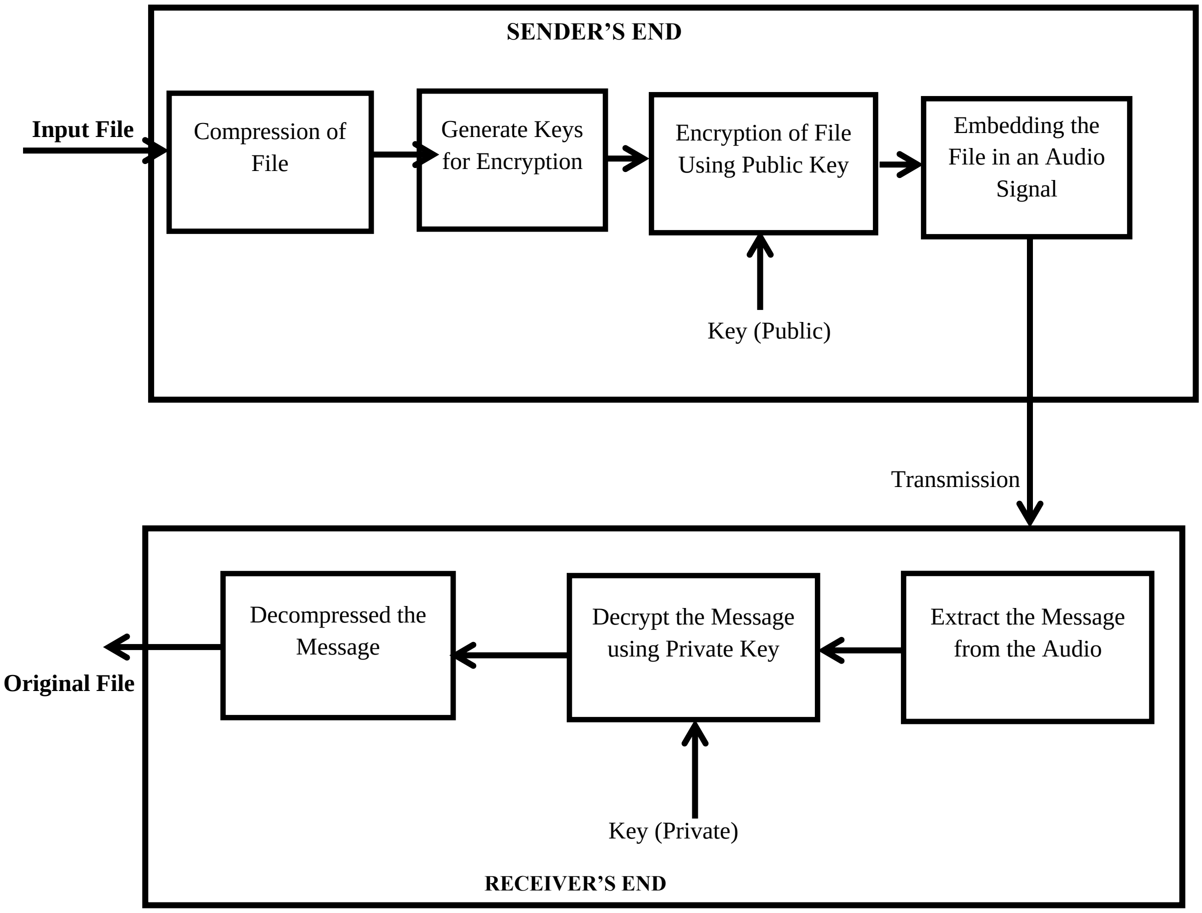
SENDER'S END
Input File
Compression of File
Generate Keys for Encryption
Encryption of File Using Public Key
Embedding the File in an Audio Signal
Key (Public)
Transmission
Decompressed the Message
Decrypt the Message using Private Key
Extract the Message from the Audio
Original File
Key (Private)
RECEIVER'S END

Figure 2: The Proposed Model

4. RESULTS AND DISCUSSIONS

The study used a total of ten (10) 16 bits .wav format audio signals for testing the system proposed in this study. The selected audio signals consist of both speech signals and music signals. The audio sample rates of each signal file are between the ranges of 16kHz and 44.2kHz. The duration of the audio files varies between 10 seconds to 100 seconds length. The characteristics of the audio files used for the study are presented in table 2

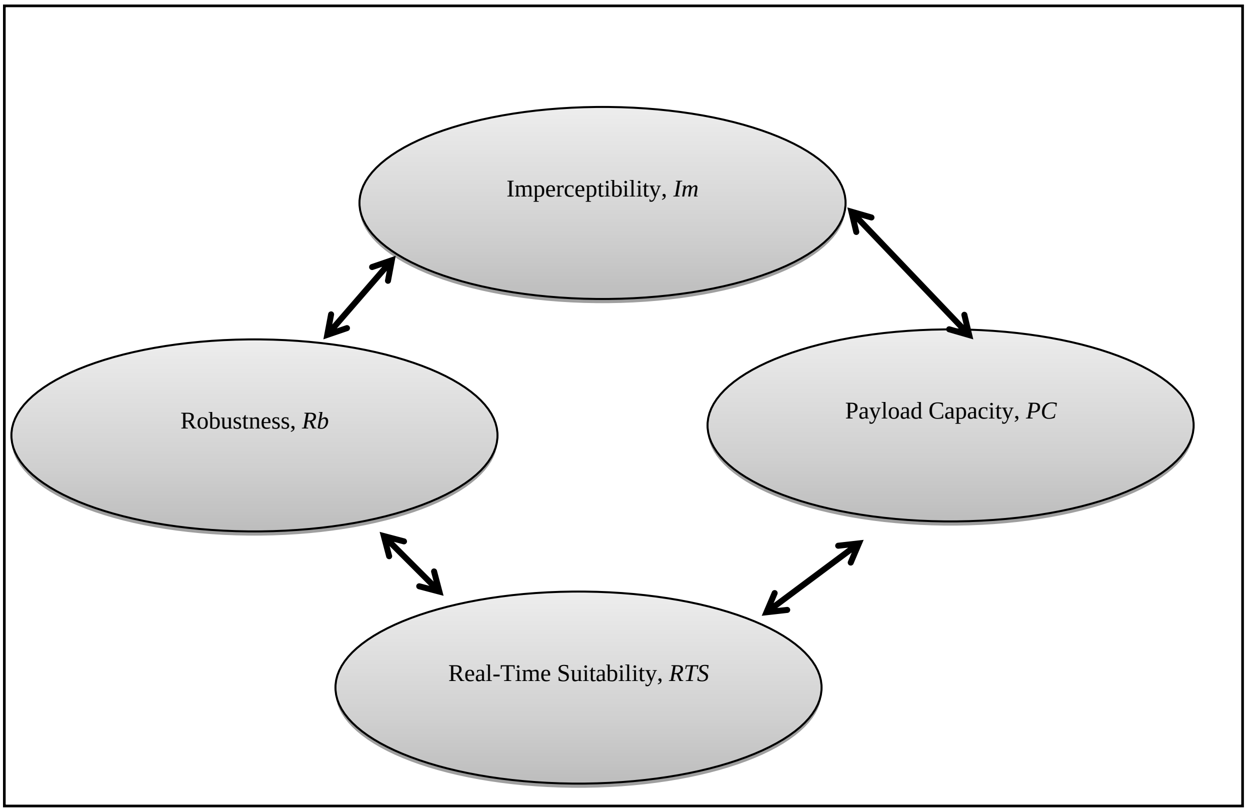

Figure 3: Audio Steganographic Evaluation Parameters

Table 2: Characteristics of Sampled Audio Files for Testing

Audio File	Bit Rate (kbps)	Audio Sample Rate (kHz)	Length (seconds)	File Category
Audio File1	20	24	91	Speech File
Audio File2	117	48	20	Speech File
Audio File3	26	16	86	Speech File
Audio File4	96	44	21	Music File
Audio File5	64	44	65	Music File
Audio File6	19	16	10	Music File
Audio File7	64	44	100	Speech File
Audio File8	92	44.1	48	Speech File
Audio File9	50	44.2	89	Music File
Audio File10	53	24	26	Music File

4.1 BER, MSE and PSNR Calculations

Bit Error Rate (BER) analysis was performed to ensure that the embedded audio reaches its intended destination without any noticeable distortion. The results were compared to the previous works of Kaur and Singh [28] and the proposed system proved far better with very low BER values. The figure 4 shows the BER results for different file sizes. Kaur and Singh [28] proposed a method of data hiding using video steganography, elliptic curve cryptography and LSB (Low-Bit Encoding) and Huffman code algorithm.

The study further computed the Mean Square Error (MSE), Peak –Signal-To-Noise-Ratio (PSNR), as well as the Bit Error Rate (BER) for the 10 sampled audio signals used for the study. The proposed system recorded low values of MSE and BER with high values for PSNR. This is an indication that the bit error rate between the original audio carrier file and the received file is minimal. Kaur and Singh [28] achieved average PSNR value of 52dB and lowest BER of 0.0185. This system however achieved average PSNR value of 58.42dB, MSE value of 0.0954 and BER value of 0.0171. Table 3 shows the computed results.

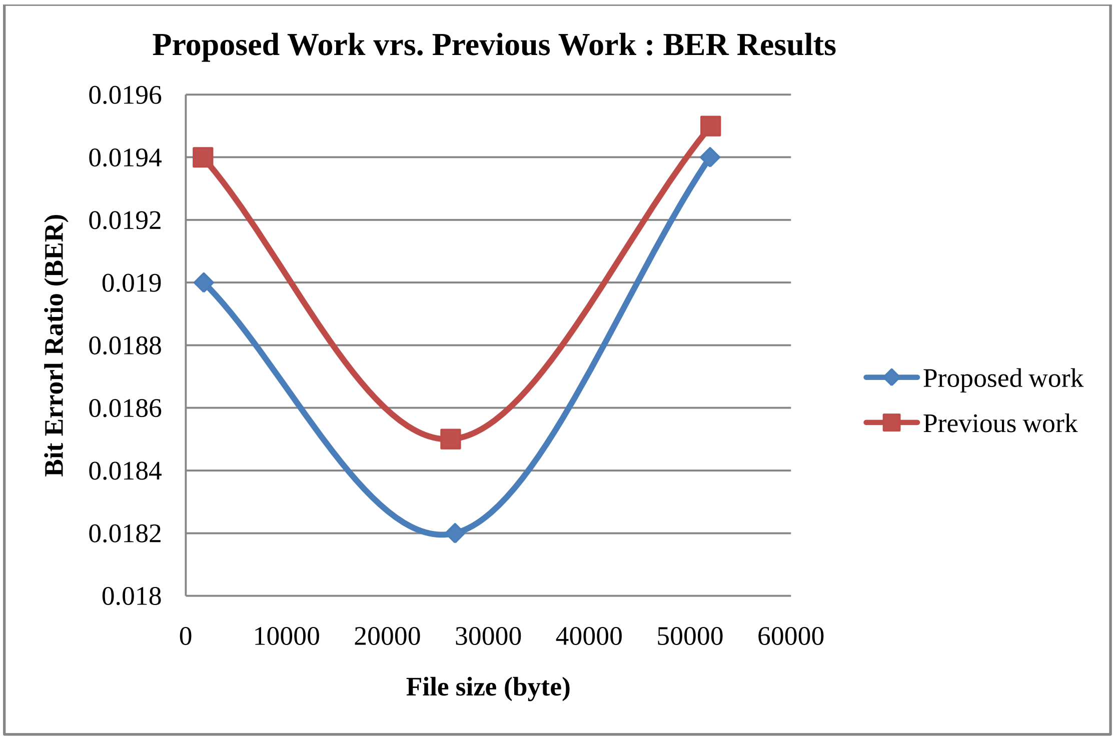

Figure 4: BER results of previous work compared to proposed work

Table 3: Results of computed PSNR, MSE and BER

Audio File	**Size of Embedded File (bits)**	PSNR**(dB)**	BER	MSE
Audio File1	19,289	57.30	0.0175	0.1211
Audio File2	19,289	58.09	0.0172	0.1009
Audio File3	19,289	58.08	0.0172	0.1012
Audio File4	19,289	58.59	0.0171	0.0899
Audio File5	19,289	59.69	0.0167	0.0698
Audio File6	19,289	57.63	0.0174	0.1122
Audio File7	19,289	58.14	0.0172	0.0997
Audio File8	19,289	58.13	0.0172	0.0999
Audio File9	19,289	60.41	0.0166	0.0591
Audio File10	19,289	58.12	0.0172	0.1002

4.2 Segmented Signal-to-Noise-Ratio (SNRseg) Calculations

From the results, 46.78dB was the highest SNRseg obtained by the proposed system with 33.20dB as the lowest SNRseg both obtained at audio sampled rate of 44.4kHzand 16kHz respectively. The file size embedded was 19,289 bits, an equivalent of 2.4KB of data size. The system achieved the lowest SNRseg value of 33.20dB with an audio file of 16 kHz audio sample rate. The highest SNRseg value as illustrated in Table 4.2 and fig 4.6 is 46.78dB which was recorded with audio file of 44.2 kHz. The value of SNRseg is generated and calculated to determine the level of security of the proposed system. From the analysis conducted on 10 selected audio files, the system produced an average Segmented Signal to Noise Ratio (SNRseg) value of 42dB. This figure is very high and demonstrates the high security of the system. The value of the SNRseg primarily indicates the distortion amount induced by the embedded data on the cover audio signal. The SNRseg value of 42dB recorded is far higher than the recommended threshold of 30dB for audio steganography application by Djebbar *et al*.[6]. Table 4 shows the calculated results .

Table 4: Results Obtained from SNRseg Computation

Audio File	**Audio Sample**	**Size of Embedded File**	**SNRseg(dB)**
Audio File1	24	19,289	39.12
Audio File2	48	19,289	42.65
Audio File3	16	19,289	33.20
Audio File4	44	19,289	45.62
Audio File5	44	19,289	45.78
Audio File6	16	19,289	38.14
Audio File7	44	19,289	42.98
Audio File8	44.1	19,289	42.02
Audio File9	44.2	19,289	46.78
Audio File10	24	19,289	40.70

The proposed system compared the results of the segmented signal to noise ratio (SNRseg) of the speech files and music files. This comparison is necessary to throw more light on the two types of audio file that present the best secured medium for hiding data in audio steganographic environment. The results show that, at each given audio sample rate, the file category of music is higher in SNRseg values than speech file. The results therefore show that music audio signals are comparatively better in terms of hiding secret data as it produces high values of SNRseg values as against audio speech signals. The result presupposes that, music signals are better hosts to hide data in terms of imperceptibility and payload capacity. Figure 5 shows the comparison of music file and audio file.

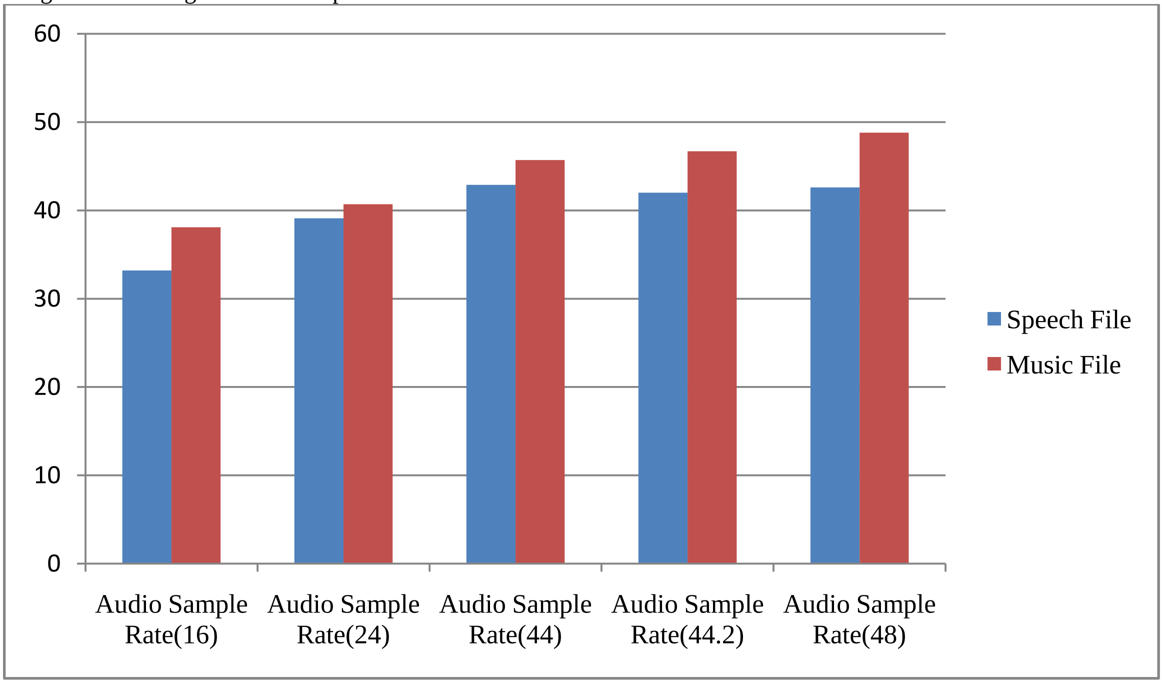

Figure 6: SNRseg Values of Speech File vrs. Music File

In order to obtain a variation in SNRseg values of embedded file of different sizes, the three best performing cover audio files were used for the test and computation. The files are AudioFile4, Audio File5 and Audio File9. Table 5 shows the results of the computation. The results of the study show that, as the file size increases significantly, the calculated SNRseg decreases. It must also be noted that, though the SNRseg values decreases as the embedded file size increases, none of the audio files recorded SNRseg values below 30dB or the value 20dB for which reason, the conclusion could be that the system is not secured as indicated by Djebbar et al., [6].

Table 5: SNRseg Values of Audio Files with Different Embedded File Sizes

File Size (bits)	SNRseg Values		
	Audio File4	Audio File5	Audio File9
10,021	51.89	53.89	57.23
14,856	48.63	49.11	51.79
19,289	45.62	45.78	46.78
20,125	43.58	44.12	45.44
20,564	40.12	41.06	43.85

4.3 Payload Capacity

The payload capacity determines the maximum allowable number of bits the proposed system can contain without distortion. It is calculated by the number of bits embedded in the proposed system by the total number of bits in the cover objects in this case the audio carrier file. Table 6 gives details of the results obtained for the calculation of payload capacity for the sample audio files used for the analysis. The minimum payload capacity as recorded was 4012 bps. The highest payload capacity recorded was also 8859 bps. From the table 5 the average embedding capacity or payload capacity as computed is 6867 bps. This value though, significant with respect to audio steganographic applications could be higher in other methodology. It is important to note that, Elliptic Curve Cryptography though has high security advantage, one of its comparative disadvantages is low embedding capacity. However, an average payload capacity of 85847.79 bytes for the proposed system means that, the embedded capacity of the proposed system is high.

Table 6: Results Obtained for Payload Capacity

Audio File	Audio Sample Rate (kHz)	Size of Embedded File (bits)	Payload Capacity (bps)
Audio File1	24	19,289	8756
Audio File2	48	19,289	5428
Audio File3	16	19,289	6235
Audio File4	44	19,289	8796
Audio File5	44	19,289	8500
Audio File6	16	19,289	5623
Audio File7	44	19,289	4012
Audio File8	44.1	19,289	7895
Audio File9	44.2	19,289	8859
Audio File10	24	19,289	4568

5. CONCLUSIONS

Audio steganography can be used anytime one wants to hide data for secret communication. Several reasons can be eluded for the importance of hiding data in audio signal but key among such reasons is to prevent unauthorized person from becoming aware of the existence of the message. In the world of business, audio steganography can be used to hide secret chemical formula or an anticipated plan for new inventions. In the non-commercial sector, audio steganography can as well be used to hide information that a person wants to keep private and confidential. Data hiding in audio is of great interest for the protection of copyrighted digital media. Hiding data in audio is also of significant importance to government for information system security and covert communication. In forensic applications, audio steganography can be used for inserting hidden data in audio files for the authentication of spoken words and sounds. More importantly also, audio steganography applicability is relevant in the music industry to monitor the songs over broadcast radio. This study brings to the fore the techniques of efficiently combining audio steganography with elliptic curve cryptography and Huffman code compression algorithm using low-bit encoding embedding. The preference of ECC over any other cryptographic algorithm is that ECC offers security with smaller key sizes, faster computation, lower power consumption as well as memory and bandwidth saving. From the results obtained in this study, it can be concluded that, the study achieved higher level of security, payload capacity, robustness and efficient real time suitability (RTS). The study experienced negligible level of distortion with high recorded values of SNRseg. The audio data hiding proposed in this study is suitable for any audio type and can be applied for other purposes. Although some data hiding techniques have been proposed by various researchers, the specific requirements of each data hiding technique vary from one application to another; with each of these techniques having some advantages and disadvantages. The flexible nature of audio formats, signals and files, is what makes them good and practical medium for steganography. Another aspect of audio steganography that makes it so attractive and promising is the ability to combine steganography techniques with existing cryptography technologies. We do not have to depend on one technique only. Secret data not only can be encrypted, they can be hidden and encrypted at the same time.

The future scope of this work will employ the use of Genetic Algorithm (GA) in addition to the ECC to further strengthen the security of the proposed system. Also, since the insertion algorithm used in this study that is, low-bit encoding or Least Significant Bits (LSB) is susceptible to noise, future studies will combine the LSB with Discrete Cosine Transform (DCT) to achieve high embedding security and capacity.

6. REFERENCES

[1] Apau, R., and Adomako, C. (2017). Design of Image Steganography based on RSA Algorithm and LSB Insertion for Android Smartphones. *International Journal of Computer Applications (IJCA),* 164(1), 13-22.

[2] John, C. (2004). VIII Steganography - Hiding Data in Wave Audio Files. The Code Project.[Online] http://www.codeproject.com/Articles/6960/Steganograph y VIII-Hiding-Data-in- Wave-Audio-Files . Accessed on: March 1, 2018.

[3] Wajgade, V. M., & Kumar, D. S. (2013). Enhancing Data Security Using Video Steganography. *International Journal of Emerging Technology and Advanced Engineering*, *3*(4), 549-552.

[4] Dengre , A. R., Gawande, A. D. Deshmukh, , A. B. (June, 2013). Effect of Audio Steganography based on LSB insertion with Image Watermarking using AVI video . International Journal of Application or Innovation in Engineering & Management (IJAIEM), 2(6), 2319 – 4847.

[5] Wilson, C. (2003). The Canonical WAVE file format. Wave PCM soundfile format. [Online] 2003. Available at https://ccrma.stanford.edu/courses/422/projects/ , Accessed on April 25, 2017.

[6] Djebbar, F., Ayad, B., Meraim, K. A., & Hamam, H. (2012). Comparative study of digital audio steganography techniques. EURASIP Journal on Audio, Speech, and Music Processing, 2012(1),1-16.

[7] Apau, R., Hayfron-Acquah, J.B., & Twum, F. (2016). Enhancing Data Security using Video Steganography, RSA and Huffman Code Algorithms with LSB Insertion. *International Journal of Computer Applications (IJCA)*, 143(4), 28-36.

[8] Arnold, M. (2000) “Audio watermarking: feature, applications and algorithms,” Proc. of IEEE ICME, 2, 1013-1016.

[9] Prabakaran, G., & Bhavani, R. A.(2012). High Capacity Video Steganography Based on Integer WaveletTransform. http://scholar.google.com/scholar?hl=en&q=A+High+Capacity+Video+Steganogr

Based+on+Integer+Wavelet+Transform.&btnG=&as_sdt =1%2C5&as_sdtp=(accessed 2017 May 8).

[10] Das, S., Das, S., Bandyopadhyay, B., & Sanyal, S. (2011). Steganography and Steganalysis: different approaches. arXiv preprint arXiv:1111.3758.

[11] Ramalingam, M. (2011). Stego Machine–Video Steganography using Modified LSB Algorithm. World Academy of Science, Engineering and Technology, 74, 502-505.

[12] Kessler, G. C. (2015). An overview of cryptography. http://www.garykessler.net/library/crypto.html#purpose (accessed 2017 May 11)

[13] Kumar, A. and Pooja, K.M. (2010) Steganography- A Data Hiding Technique *International Journal of Computer Applications. 9(7)*

[14] Cvejic, N., and T. Seppanen. 2002. Increasing the capacity of LSB-based audio steganography. In *2002 IEEE workshop on multimedia signal processing*. IEEE.

[15] Shirali-Shahreza, S and Manzuri-Shalmani, M.T (2008) "High capacity error free wavelet domain speech steganography" ICASSP .

[16] Johnson, N. F., Duric, Z. and Jajodia, S (2001). Information Hiding Steganography and Watermarking- Attacks and Countermeasures",Kluwer Academic Publishers, 2001.

[17] Taraghi-Delgarm, N (2006). "*Speech Watermarking", M.Sc. Thesis, Comptuer Engineering Department,* Sharif University of Technology, Tehran, IRAN.

[18] Pooyan, M and Delforouzi, A (2007) "LSB-based Audio Steganography Method Based on Lifting Wavelet Transform", in Proc. 7th IEEE International Symposium on Signal Processing and Information Technology (ISSPIT'07), Egypt.

[19] Ahmed, M. A., Kiah, M. L. M., Zaidan, B. B., & Zaidan, A. A. (2010). A novel embedding method to increase capacity and robustness of low-bit encoding audio steganography technique using noise gate software logic algorithm. *Journal of Applied Sciences*, *10*(1), 59-64.

[20] Parthasarathi, M., & Shreekala, T. (2017) Secured Data Hiding in Audio Files Using Audio Steganography Algorithm. International Journal of Pure and Applied Mathematics. 114(7), 743-753

[21] Padmashree, G., & Venugopala, P. S. (2012). Audio Stegnography and Cryptography: Using LSB algorithm at 4th and 5th LSB layers. *International Journal of Engineering and Innovative Technology*, *2*(4).

[22] Malviya, S., Saxena, M., & Khare, D. A. (2012). Audio steganography by different methods. International Journal of Emerging Technology and Advanced Engineering Website: www. ijetae. com (ISSN 2250-2459, Volume 2, Issue 7.

[23] Asad, M., Gilani, J., & Khalid, A. (2011, July). An enhanced least significant bit modification technique for audio steganography. In *Computer Networks and Information Technology (ICCNIT), 2011 International Conference on* (pp. 143-147). IEEE.

[24] Al-Othmani, A. Z., Manaf, A. A., & Zeki, A. M. (2012). A survey on steganography techniques in real time audio signals and evaluation. *International Journal of Computer Science Issues (IJCSI)*, *9*(1).

[25] Ahmad, A. (2012). Encryption and compression of Data. Available at https://security.stackexchange.com/questions/19969/encryption-and-compression- accessed: 3rd May, 2017.

[26] Vidakovic, D., & Parezanovic, D. (2013). Generating keys in elliptic curve cryptosystems. International Journal of Computer Science and Business Informatics. 4(1), 1-9.

[27] López, J., & Dahab, R. (1999). Fast multiplication on elliptic curves over GF (2 m) without precomputation. In *Cryptographic Hardware and Embedded Systems* (pp. 724-724). Springer Berlin/Heidelberg.

[28] Kaur, R., & Singh, T. (2015). Hiding Data in Video Sequences using LSB with Elliptic Curve Cryptography. *International Journal of Computer Applications*, *117*(18).

[29]