Implementation of ETAS (Embedding Text in Audio Signal) Model to Ensure Secrecy

K. GEETHA [1]
ASSISTANT PROFESSOR

P.VANITHA MUTHU [2]
FINAL YEAR M.TECH

[1, 2] Department of Computer Science and Engineering
Bharathidasan University
Tiruchirappalli- 23
Tamilnadu
India

***Abstract*— Steganography is the art of hiding information that evolves as a new secret communication technology. For a long period time, information hiding was done using plain text, still images, video and IP datagram. Embedding secret messages using audio signal in digital format is now the area of focus. There exists numerous steganography techniques for hiding information in audio medium. In this work we propose a new model ETAS - Embedding Text in Audio Signal that embeds the text like the existing system but with encryption that gains the full advantages of cryptography. Using steganography it is possible to conceal the full existence of the original text and the results obtained from the proposed model is compared with other existing techniques and proved to be efficient for textual messages of size beyond 12 KB as the size of the embedded text is approximately same as that of encrypted text size. This emphasis the fact that we are able to ensure secrecy without an additional cost of extra space consumed for the text to be communicated.**

***Keywords*— Steganography, Audio data hiding, ETAS, cryptography, cover media.**

I. INTRODUCTION

The main purpose of steganography [4] is to hide a message in some cover media, to obtain new data, practically indistinguishable from the original message, by people, in such a way that an eavesdropper cannot detect the presence of original message in new data. With computers and Networks, there are many other ways of hiding information, such as Covert channels, Hidden text within WebPages Hiding files in "Plain sight", Null ciphers.

Today, the internet is filled with tons of programs that use steganography to hide the secret information[7]. There are so many medias are used for digitally embedding message such as plaintext, hypertext, audio/video, still image and network traffic[4]. There exists a large variety of steganographic techniques with varying complexity and possessing some strong and weak aspects[10].

Hiding information in text is the most popular method of Steganography[2]. It is used to hide a secret message in every n^{th} character or altering the amount of white space after lines or between words of a text message. It is used in initial decade of internet era. But it is not used frequently because the text files have a small amount of redundant data. Later on hiding text in images because more popular technique for steganography especially on the internet[2]. But this technique lacks in payload capacity and robustness. To hide data in audio files, the secret message is embedded into digitized audio signal[2]. Audio data hiding method provides the most effective way to protect privacy. Key aspect of embedding text in audio files is that, no extra bytes are generated for embedding. Hence it is more comfortable to transmit huge amount of data using audio signal. Embedding the secret messages in digital sound is usually a very difficult process.

II. PROPOSED ETAS MODEL

The following ETAS Model provides a very basic description of the audio steganographic process in the sender side and receiver side.

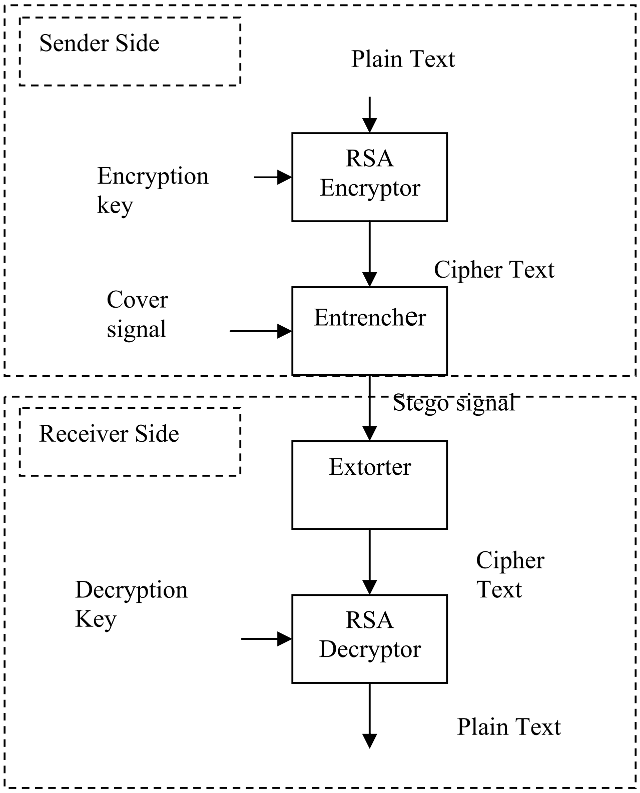
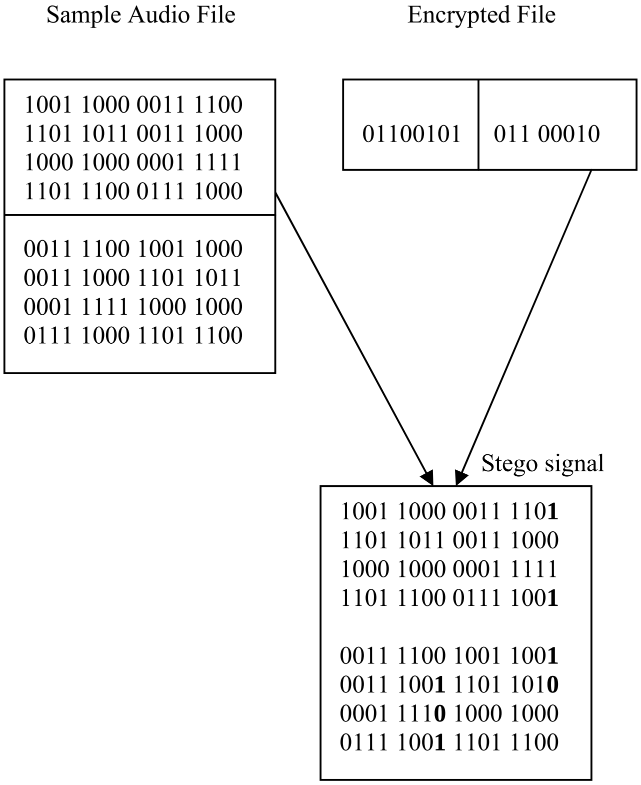


Fig 2.1 ETAS Model

The original text in ASCII format is encrypted by RSA encryptor using an encryption key. The model implements RSA encryption as it proves to be more efficient [11]. The encrypted text is passed on to Entrencher that embeds the encrypted text inside the cover signal which is in audio format *.wav resulting in stego signal. This process happens at sender side. This stego signal is communicated using Network medium. At the receiver side the stego signal is passed on to Extorter module that extracts embedded text from the audio signal that was used a s cover medium,. The resultant cipher text is then decrypted using RSA Decryptor module. The final plain text can then be used for further processing.



Sample Audio File

Encrypted File

```
1001 1000 0011 1100
1101 1011 0011 1000
1000 1000 0001 1111
1101 1100 0111 1000

0011 1100 1001 1000
0011 1000 1101 1011
0001 1111 1000 1000
0111 1000 1101 1100
```

01100101	011 00010

Stego signal

1001 1000 0011 110**1**
1101 1011 0011 1000
1000 1000 0001 1111
1101 1100 0111 100**1**

0011 1100 1001 100**1**
0011 100**1** 1101 101**0**
0001 111**0** 1000 1000
0111 100**1** 1101 1100

Fig. 2.2 ETAS encoding format

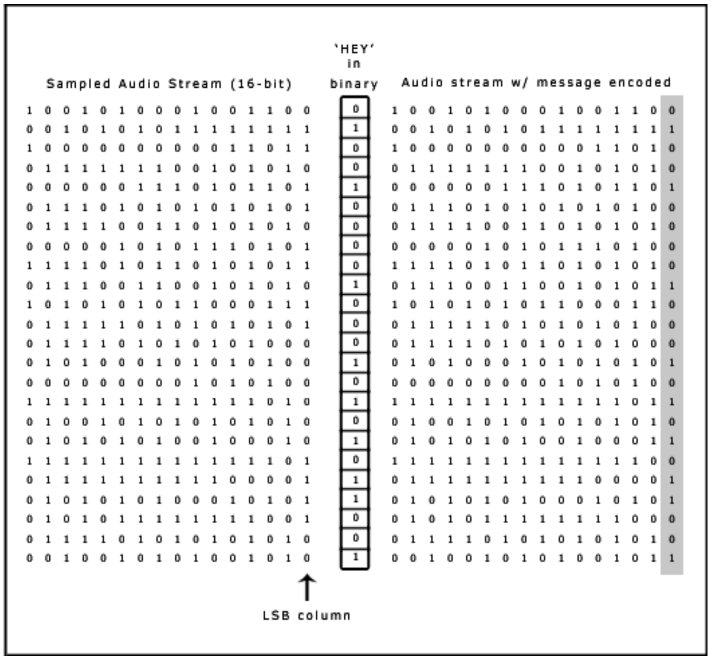
To hide a letter A & B to an digitized audio file where each sample is represented with 16 bits then the LSB bit of sample audio file is replaced with each bit of binary equivalent of the letter A & B.

III. RELATED WORK

In audio steganography, secret message is embedded into digitized audio signal which result slight altering of binary sequence of the corresponding audio file. There are several methods are available for audio steganography. Some of them are as follows: -

LSB Coding[5]:

Least significant bit (LSB) coding is the simplest way to embed information in a digital audio file. By substituting the least significant bit of each sampling point with a binary message, LSB coding allows for a large amount of data to be encoded. The following diagram illustrates how the message 'HEY' is encoded in a 16-bit CD quality sample using the LSB method:



```
                                        'HEY'
                                         in
 Sampled Audio Stream (16-bit)         binary   Audio stream w/ message encoded
1 0 0 1 0 1 0 0 0 1 0 0 1 1 0 0          0      1 0 0 1 0 1 0 0 0 1 0 0 1 1 0 0
0 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1          1      0 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1
1 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1          0      1 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0
0 1 1 1 1 1 1 1 0 0 1 0 1 0 1 0          0      0 1 1 1 1 1 1 1 0 0 1 0 1 0 1 0
0 0 0 0 0 0 1 1 1 0 1 0 1 1 0 1          1      0 0 0 0 0 0 1 1 1 0 1 0 1 1 0 1
0 1 1 1 0 1 0 1 0 1 0 1 0 1 0 1          0      0 1 1 1 0 1 0 1 0 1 0 1 0 1 0 0
0 1 1 1 1 0 0 1 1 0 1 0 1 0 1 0          0      0 1 1 1 1 0 0 1 1 0 1 0 1 0 1 0
0 0 0 0 0 1 0 1 0 1 1 1 0 1 0 1          0      0 0 0 0 0 1 0 1 0 1 1 1 0 1 0 0
1 1 1 1 0 1 0 1 1 0 1 0 1 0 1 1          0      1 1 1 1 0 1 0 1 1 0 1 0 1 0 1 0
0 1 1 1 0 0 1 1 0 0 1 0 1 0 1 0          1      0 1 1 1 0 0 1 1 0 0 1 0 1 0 1 1
1 0 1 0 1 0 1 0 1 1 0 0 0 1 1 1          0      1 0 1 0 1 0 1 0 1 1 0 0 0 1 1 0
0 1 1 1 1 1 0 1 0 1 0 1 0 1 0 1          0      0 1 1 1 1 1 0 1 0 1 0 1 0 1 0 0
0 1 1 1 1 0 1 0 1 0 1 0 1 0 0 0          0      0 1 1 1 1 0 1 0 1 0 1 0 1 0 0 0
0 1 0 1 0 0 0 1 0 1 0 1 0 1 0 0          1      0 1 0 1 0 0 0 1 0 1 0 1 0 1 0 1
0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0          0      0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0          1      1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1
0 1 0 0 1 0 1 0 1 0 1 0 1 0 1 0          0      0 1 0 0 1 0 1 0 1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1 0 0 1 0 0 0 1 0          1      0 1 0 1 0 1 0 1 0 0 1 0 0 0 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1          0      1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1          1      0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1
0 1 0 1 0 1 0 1 0 0 0 1 0 1 0 1          1      0 1 0 1 0 1 0 1 0 0 0 1 0 1 0 1
0 1 0 1 0 1 1 1 1 1 1 1 1 0 0 1          0      0 1 0 1 0 1 1 1 1 1 1 1 1 0 0 0
0 1 1 1 1 0 1 0 1 0 1 0 1 0 1 0          0      0 1 1 1 1 0 1 0 1 0 1 0 1 0 1 0
0 0 1 0 0 1 0 1 0 1 0 0 1 0 1 0          1      0 0 1 0 0 1 0 1 0 1 0 0 1 0 1 1
                              ↑
                          LSB column
```

Fig.3.1. Message 'HEY' is encoded in a 16-bit CD quality sample using the LSB method

In LSB coding, the ideal data transmission rate is 1 kbps per 1 kHZ. In some implementations of LSB coding, however, the two least significant bits of a sample are replaced with two message bits. This increases the amount of data that can be encoded but also increases the amount of resulting noise in the audio file as well. Thus, one should consider the signal content before deciding on the LSB operation to use. For example, a sound file that was recorded in a bustling subway station would mask low-bit encoding noise. On the other hand, the same noise would be audible in a sound file containing a piano solo.

To extract a secret message from an LSB encoded sound file, the receiver needs access to the sequence of sample indices used in the embedding process. Normally, the length of the secret message to be encoded is smaller than the total number of samples in a sound file. One must decide then on how to choose the subset of samples that will contain the secret message and communicate that decision to the receiver. One trivial technique is to start at the beginning of the sound file and perform LSB coding until the message has been completely embedded, leaving the remaining samples unchanged. This creates a security problem, however in that the first part of the sound file will have different statistical properties than the second part of the sound file that was not modified. One solution to this problem is to pad the secret message with random bits so that the length of the message is equal to the total number of samples. Yet now the embedding process ends up changing far more samples than the transmission of the secret required. This increases the probability that a would-be attacker will suspect secret communication.

A more sophisticated approach is to use a pseudorandom number generator to spread the message over the sound file in a random manner. One popular approach is to use the random interval method, in which a secret key possessed by the sender is used as a seed in a pseudorandom number generator to create a random sequence of sample indices. The receiver also has access to the secret key and knowledge of the pseudorandom number generator, allowing the random sequence of sample indices to be reconstructed. Checks must be put in place, however, to prevent the pseudorandom number generator from generating the same sample index twice. If this happened, a collision would occur where a sample already modified with part of the message is modified again. The problem of collisions can be overcome by keeping track of all the samples that have already been used. Another approach is to calculate the subset of samples via a pseudorandom permutation of the entire set through the use of a secure hash function. This technique insures that the same index is never generated more than once.

Parity Coding[5]

Instead of breaking a signal down into individual samples, the parity coding method breaks a signal down into separate regions of samples and encodes each bit from the secret message in a sample region's parity bit. If the parity bit of a selected region does not match the secret bit to be encoded, the process flips the LSB of one of the samples in the region. Thus, the sender has more of a choice in encoding the secret bit, and the signal can be changed in a more unobtrusive fashion.

Using the parity coding method, the first three bits of the message 'HEY' are encoded in the following figure. Even parity is desired.

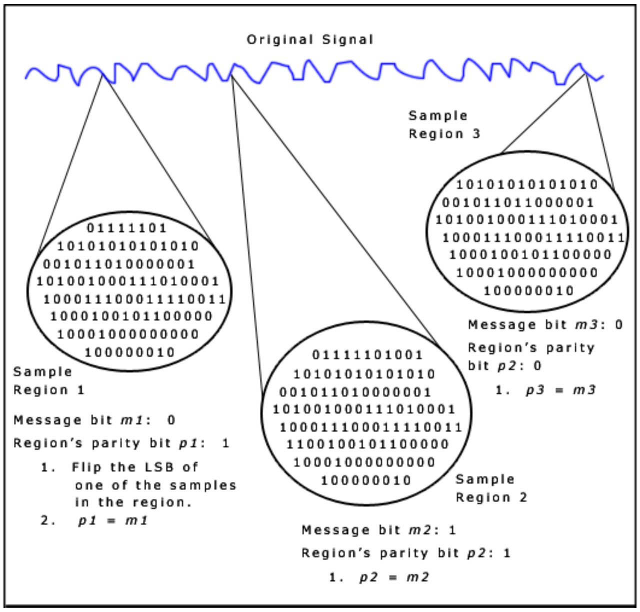


Fig.3.2. First three bits of the message 'HEY' are encoded using Parity coding method

The decoding process extracts the secret message by calculating and lining up the parity bits of the regions used in the encoding process. Once again, the sender and receiver can use a shared secret key as a seed in a pseudorandom number generator to produce the same set of sample regions.

There are two main disadvantages associated with the use of methods like LSB coding or parity coding. The human ear is very sensitive and can often detect even the slightest bit of noise introduced into a sound file, although the parity coding method does come much closer to making the introduced noise inaudible. Both methods share a second disadvantage however, in that they are not robust. If a sound file embedded with a secret message using either LSB coding or parity coding was resampled, the embedded information would be lost. Robustness can be improved somewhat by using a redundancy technique while encoding the secret message. However, redundancy techniques reduce data transmission rate significantly.

Phase Coding[6]**:**

Phase coding addresses the disadvantages of the noise-inducing methods of audio steganography. Phase coding relies on the fact that the phase components of sound are not as perceptible to the human ear as noise is. Rather than introducing perturbations, the technique encodes the message bits as phase shifts in the phase spectrum of a digital signal, achieving an inaudible encoding in terms of signal-to-perceived noise ratio.

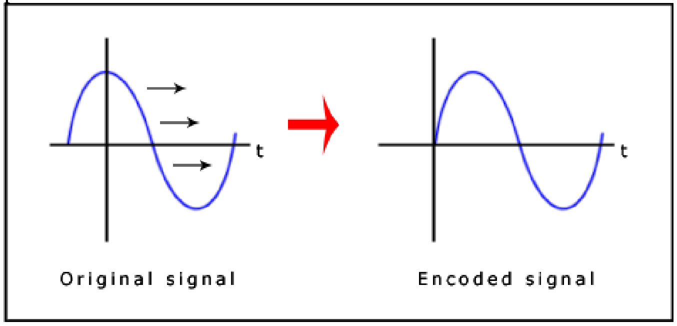


Fig.3.3. Phase Coding

Phase coding is explained in the following procedure:

1. The original sound signal is broken up into smaller segments whose lengths equal the size of the message to be encoded.
2. A Discrete Fourier Transform (DFT) is applied to each segment to create a matrix of the phases and Fourier transform magnitudes.
3. Phase differences between adjacent segments are calculated.
4. Phase shifts between consecutive segments are easily detected. In other words, the absolute phases of the segments can be changed but the relative phase differences between adjacent segments must be preserved. Therefore the secret message is only inserted in the phase vector of the first signal segment as follows:

$$phase\_new = \begin{cases} \pi/2 & if \quad message \quad bit = 0 \\ -\pi/2 & if \quad message \quad bit = 1 \end{cases}$$

5. A new phase matrix is created using the new phase of the first segment and the original phase differences.
6. Using the new phase matrix and original magnitude matrix, the sound signal is reconstructed by applying the inverse DFT and then concatenating the sound segments back together.

To extract the secret message from the sound file, the receiver must know the segment length. The receiver can then use the DFT to get the phases and extract the information.

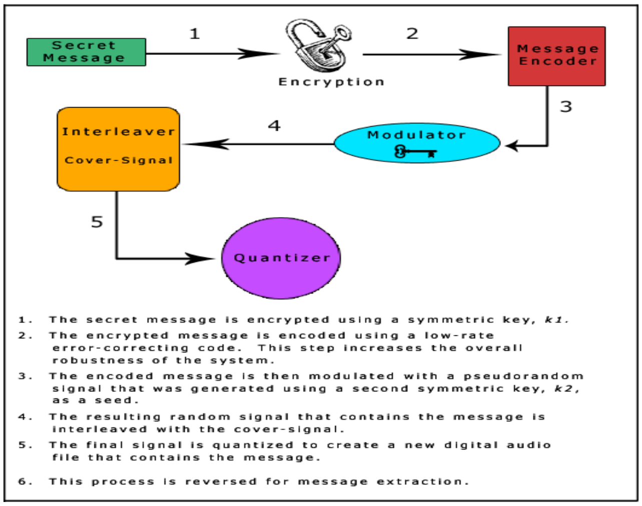
One disadvantage associated with phase coding is a low data transmission rate due to the fact that the secret message is encoded in the first signal segment only. This might be addressed by increasing the length of the signal segment. However, this would change phase relations between each frequency component of the segment more drastically, making the encoding easier to detect. As a result, the phase coding method is used when only a small amount of data, such as a watermark, needs to be concealed.

Spread Spectrum[6]

In the context of audio steganography, the basic spread spectrum (SS) method attempts to spread secret information across the audio signal's frequency spectrum as much as possible. This is analogous to a system using an implementation of the LSB coding that randomly spreads the message bits over the entire sound file. However, unlike LSB coding, the SS method spreads the secret message over the sound file's frequency spectrum, using a code that is independent of the actual signal. As a result, the final signal occupies a bandwidth in excess of what is actually required for transmission.

Two versions of SS can be used in audio steganography: the direct-sequence and frequency-hopping schemes. In direct-sequence SS, the secret message is spread out by a constant called the chip rate and then modulated with a pseudorandom signal. It is then interleaved with the cover-signal. In frequency-hopping SS, the audio file's frequency spectrum is altered so that it hops rapidly between frequencies.

The following procedural diagram illustrates the design of that system when applied to our specific topic of audio steganography.



1. The secret message is encrypted using a symmetric key, *k1*.
2. The encrypted message is encoded using a low-rate error-correcting code. This step increases the overall robustness of the system.
3. The encoded message is then modulated with a pseudorandom signal that was generated using a second symmetric key, *k2*, as a seed.
4. The resulting random signal that contains the message is interleaved with the cover-signal.
5. The final signal is quantized to create a new digital audio file that contains the message.
6. This process is reversed for message extraction.

Fig.3.4. The design of system when applied audio steganography.

Echo Hiding[8]**:**

In echo hiding, information is embedded in a sound file by introducing an echo into the discrete signal. Like the spread spectrum method, it too provides advantages in that it allows for a high data transmission rate and provides superior robustness when compared to the noise inducing methods.

To hide the data successfully, three parameters of the echo are varied:

Amplitude, decay rate, and offset (delay time) from the original signal. All three parameters are set below the human hearing threshold so the echo is not easily resolved. In addition, offset is varied to represent the binary message to be encoded. One offset value represents a binary one, and a second offset value represents a binary zero.

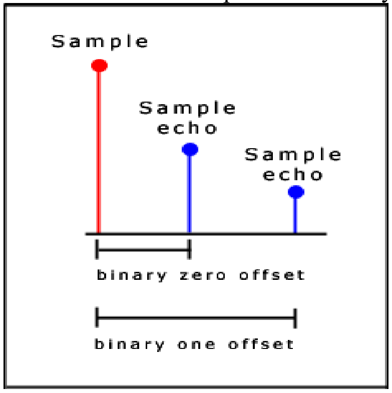
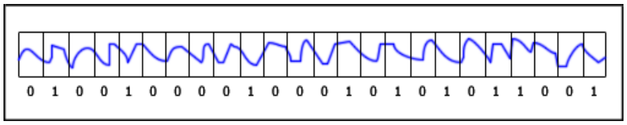


Fig.3.5. Echo Hiding

If only one echo was produced from the original signal, only one bit of information could be encoded. Therefore, the original signal is broken down into blocks before the encoding process begins. Once the encoding process is completed, the blocks are concatenated back together to create the final signal.

We'll now go through a simple form of the echo hiding process using the message 'HEY'. For brevity, we'll divide the signal completely up into blocks, although under normal circumstances a random number of samples between each pair of blocks should remain unused to reduce the probability of detection. First the signal is divided up into blocks, and each block is assigned a one or a zero based on the secret message. In this case, the message is the binary equivalent of 'HEY'.



IV. METHODOLOGY

ETAS Algorithm - at the sender side

Input: Audio file, Key and Original message
Output: Mixed Data.

```
Step 1: Load the audio file (AF) of size 12 K.
Step 2: Input key for encryption
Step 3: Convert the audio files in the form of bytes and this
        byte values are represented in to bit patterns.
Step 4: Using the key, the original message is encrypted using
        RSA algorithm.
Step 5: Split the audio file bit patterns horizontally into two
        halves.
Step 6: Split the Encrypted message bit patterns vertically into
        two halves.
Step 7: Insert the LSB bit of the vertically splitted encrypted
        text file (TF) into the LSB bit of the horizontally
        splitted audio file.
Step 8: Repeat Step 7 for the remaining bits of encrypted text
        file.
Step 9: If size (AF) ≥ size (TF) then
        embedding can be done as explained above
        else
        The next higher order bit prior to previous bit
        position can be used
        Until it is exhausted.
```

ETAS Algorithm - at the Receiver side:

Input: Mixed data, Key
Output: Original message, audio file.

```
Step 1: Load the Stego signal
Step 2: Input key for decryption (as used in encryption)
Step 3: Extract the hidden data and audio files bit patterns
        from mixed data.
        // Reverse process of step 7 of ETAS algorithm at
        sender side.
Step 4: Combine the two halves of audio files bit patterns.
Step 5: Combine the two halves of encrypted messages bit
        pattern.
Step 6: Using Key, decrypt the original message.
```

V. EXPERIMENTAL RESULTS

Table 5.1 Evaluation of Steganography requirements associated with Cover Medium

	Plain Text	Image	Audio	Video
Invisibility	Medium	High	High	High
Payload Capacity	Low	Low	High	High
Robustness against Statistical Attacks	Low	Medium	High	High
Robustness against Text Manipulation	Low	Medium	High	High
Variation in file size	Medium	Medium	High	Medium

Table 5.1 shows the different levels of satisfaction for few parameters like payload capacity, Robustness and file size for different cover medium like Text, Image, Audio and video. A high level means that the algorithm completely satisfies the requirement, while a low level indicates that the algorithm has weakness in this requirement. A medium level indicates that the requirement depends on outside influences.

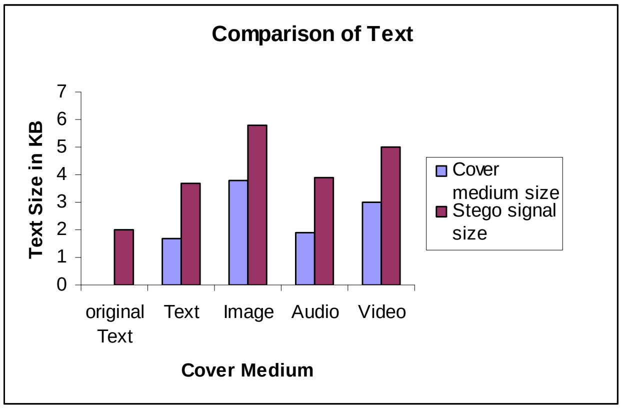


Fig 5.1 Comparison of text merged with in terms of size.

Fig. 5.1 depicts the original text size against the stego text embedded in different cover medium.

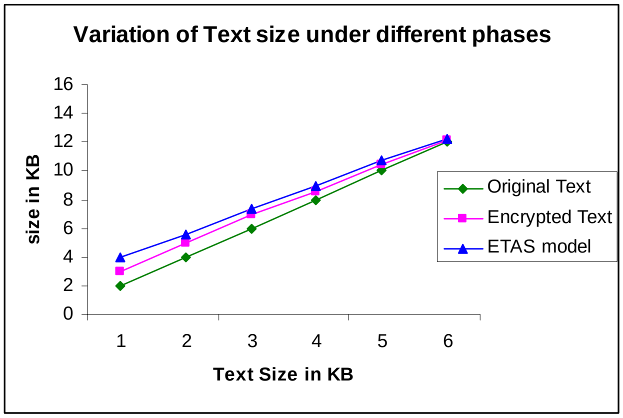


Fig.5.2 Assessments of Original Text, Encrypted Text and Data Hiding in Audio

Fig. 5.2 shows that if the original text size is beyond 12K ETAS model is more suitable for selecting as a cover medium as it is equivalent to that as of normal encryption but with higher level of secrecy as it involves cryptography and steganography. This efficiency is achievable as the audio signal pattern has more provision to embed plain text at different slots by default.

VI. CONCLUSION

This work proposes a new model for data transmission at higher degree of secrecy by using audio signal as cover medium. This proposed system provides an efficient method for hiding the data from the eavesdropper. LSB data hiding technique is the simplest method for inserting data into audio signals. ETAS model is able to ensure secrecy with less complexity at the cost of same memory space as that of encrypted text and the user is able to enjoy the benefits of cryptography and steganography combined together without any additional overhead. This work is more suitable for automatic control of robotic systems used in military and defence applications that can listen to a radio signal and then act accordingly as per the instructions received. By embedding the secret password in the audio signal the robot can be activated only if the predefined password matches with the incoming password that reaches the robot through audio signal. It can then start functioning as per the instructions received in the form of audio signal. More such sort of applications can be explored but confined to audio medium usage.

VII. REFERENCES

[1] Poluami dutta, Debnath Bhattacharyya and Tai-hoon Kim, "Data Hiding in audio signal : A Review", International Journal of Database theory and application, vol.2,No.2,June 2009.

[2] T.Morkel, J.H.P.Eloff and M.S. Oliver, "An overview of Image Steganography", Information and computer society Architecture (ICSA) Research Group.

[3] Robert Krenn, "Steganography and Steganalysis," An article, January 2004.
http://www.krenn.nl/univ/cry/steg/article.pdf

[4] Soum,y endu Das,Subhendu Das , Bijoy Bandyopadhyay and sugata sanyal, " Steganography and Steganalysis: Different Approaches", An Article.

[5] Methods of Audio Steganography, Internet publication on www. Snotmonkey.com

[6] W.Bender, w.Butera, D.Gruhl, R.Hwang, F.J.Paiz, S.Pogreb, " Techniques for data hiding", IBM systems Journal, volume 39, Isuue 3-4, July 2000, pp. 547 – 568.

[7] Mehdi kharrazi, Husrev T.Sencar, and Nasir Memon, "Image Steganography: Concepts and Practice ", WSPC/Lecture notes series: 9inX6in, April 22, 2004.

[8] Johnson, N.F and Jajodia S.," Exploring Steganography: Seeing the unseen",Computer Journal February 1998.

[9] Petitcolas, F.A.P., Anderson, R.J. & Kuhn, M.G., "Information Hiding – A Survey", Proceedings of the IEEE, 87:07, July 1999.

[10] Silman J., "Steganography and Steganalysis: An Overview", SANS Institute 2001.

[11] Chandra M. Kota and Cherif Aissi , "Implementation of the RSA algorithm and its cryptanalysis" Proceedings of the 2002 ASEE Gulf-Southwest Annual Conference,The University of Louisiana at Lafayette, March 20 – 22, 2002.