GA-based Approach with Reduced RSA Encryption Security for Audio Steganography

Ms. Swati Shishupal, Mrs. Ujwala Gaikwad
ME COMP, Terna College of Engineering
Nerul, Navi Mumbai, India

Mrs. Sowmya Shree, Mrs. Tejal Rachh
ME COMP, Atharva College of Engineering
Malad, Mumbai, India

***Abstract*- To transmit the secrete message by modifying an audio bit samples that is nothing but the Audio Steganography. This is done by using reduced RSA algorithm which will generate the private and public key for the message encryption. Next the encrypted message is then embedded into audio data samples by using Genetic Algorithm (GA) based Least Significant Bit(LSB) approach to increase the robustness against intentional attacks and to ensure secured data transfer between parties.**

Keywords – Audio Steganography; Genetic Algorithm (GA); Least Significant Bit (LSB); robustness, RSA

I. INTRODUCTION

Nowadays security is the main concern in Internet world and internet security is nothing but to prevent data from intentional and unintentional attacks. And for that we need to use some technologies like data hiding, data preventing etc. So Steganography is one the data hiding technique that we are discussing here further.

Steganography is the word, which means "Covered Writing" [1]. As we have deliberated before Steganography is the data hiding technique and briefly which is nothing but the hiding of the primary message in the secondary payload message. Here primary message is any secrete or private message and secondary message or payload is any kind of cover medium, which can be text, audio or video, etc.

In this paper we are using audio as a cover medium or an embedding such as any audio file. However, audio-based steganography exploits the masking effect property of Human Auditory System (HAS) i.e. the main aim of this system as the security concern is humans should not easily detect the difference between the original audio file and the message embedded audio file.

II. EXISTING TECHNOLOGIES

Steganography uses different methods to hide the message into an audio file. Initially, simple LSB encoding is one of the popular method is there to hide the information. However, this technique is characterized by low robustness to noise addition and thus by low security as well since it is very vulnerable even to simple attacks. After that modified LSB [2] method were used by increasing the LSB layers to increase the robustness against attacks. But because of that the hiding capacity gets decrease since some of the samples have to be left unchangeable to preserve the audio perceptual quality of the audio signal. Another method of embedding is Parity coding technique. This method breaks down the signal into separate samples and embeds each bit from the secret message in a sample area's parity bit. If the parity bit of a selected region does not match the secret bit to be encoded, the process tosses the LSB of one of the samples in the area. Thus, the sender has more of a choice in encoding the secret bit, and the signal can be changed in anadditional unobtrusive fashion [3].

III. PROPOSED METHODOLOGY

This approach transmits the hidden information by modifying an audio signal to ensure secured data transfer between parties. We are using reduced RSA [4] encryption algorithm to encrypt the message. And reduce the size of encrypted file. The encrypted message is then encoded into audio data by Genetic Algorithm (GA) based Least Significant Bit approach.

Here by using reduced RSA we can reduce the size of the RSA encrypted file. And then applying proposed LSB algorithm, entrench message bits to the audio bit stream (16 bit sample) in random and higher LSB layer positions (increase the robustness) to get a collection of chromosomes.Now Genetic Algorithm operators are used to get the next generation chromosomes. Now select the best chromosome according to the best fitness value. Fitness value is a value of LSB position in which we get a chromosome with the minimum deviation comparing to the original host audio sample. Here higher LSB layer is given higher preference in case of layer selection. We have original audio sample and inserting message bit in different LSB layer positions we get some new samples. Sometimes it can happen that for more than one LSB layer we get the same difference between original audio sample and new audio samples. In this case, we will choose the higher LSB layer. In this, an intelligent algorithm is used to embed the message bits in the deeper layers of samples and alter other bits to decrease the error rate and if alteration is not possible for any sample it will ignore them, which helps in achieving higher message embedding capacity which refers to the amount of information that a data hiding scheme can successfully embed without introducing perceptual distortion in the marked media against intentional and unintentional attacks.

A. Reduced RSA Algorithm

This algorithm which uses big primes as an input parameter as follows:
Declarations:
a) Ptext.txt: Plain text (source) file to be encrypted.
b) Ctext.txt: Cipher text file to be decrypted.
c) Array L: Used to store the matters of Ptext.txt for encryption.
d) Array S: Used to store the matters of Ctext.txt for decryption.

Algorithm:

1. Choose two large prime numbers p, q.
2. Generate two very large mersenne prime numbers [4] as: m = 2p -1 and n = 2q -1.
3. Calculate c= m* n.
4. Calculate the value of Φ using the formula: Φ(c) = (m-1) * (n-1).
5. Generate the public key 'e' such that it is co-prime with Φ(c).
6. Find the value of private key'd' such that (d *e) ≡ 1 mod Φ(c)
7. Read plaintext in the form of binary data from the file Ptext.txt, store it in an array(L**).**
8. Perform Le mod c (on each element of an array L) to get cipher text and store it in the file Ctext.txt.
9. For decryption, read the file Ctext.txt, store it in an array (S).
10. Perform Sd mod c (on each element of an array S) to get a plain text.

Enhancement:

We perform enhancement in the above algorithm to reduce the size of encrypted file. So the reduction logic is given as follows:

For Encryption
1. Each element of array L is reduced modulo 'e' and the quotients are stored is an array(QUE) and store the remaindersin array (REM).
2. Now array(REM) is a cipher text, supply it in the file ctext.txt.

For decryption

1. Read the file ctext.txt, store it in an array (S).
2. Retrieve an array(QUE) and multiply each element of array (QUE) by e.
3. Now add each element of array (QUE) into array (S) respectively.
4. Perform Sd mod c to get a plain text.

In this enhancement, the public key e is used again for reduction. Dividing each array element of the ciphertext again toget quotient and take mod e of each array element of the ciphertext to get the remainder. The array of remainder isnow reduced with great extent. Hence the size of each character(in digits) to be encoded get reduced which results thereduced file size. But at the phase of decryption, the values of quotients in array(QUE) are needed to retrieve to get theciphertext again.

B. Genetic Algorithm:

In this approach all results and achievements that we expect are depending on this algorithm. In this stage we are trying to decrease the error rate and to improve the transparency by using the Genetic Algorithm, chromosome (a gene) is a set of parameters which defines a proposed solution to the problem that the genetic algorithm is trying to solve. The simplest form of genetic algorithm involves three types of operators: selection, crossover (single point) and mutation.

*Selection***:** This operator selects chromosomes from the population for reproduction of bit positions. The fitter the chromosome, the more times it is likely to be selected to reproduce.

*Crossover***:** This operator randomly chooses a locus and exchanges the subsequences before and after that locus between two chromosomes to create two offspring. For example, the bit strings 10000100 and 11111111 could be crossed over after the third locus in each to produce the two offspring 10011111 and 11100100.

*Mutation***:** This operator randomly flips some of the bits in a chromosome. For example, the string 00000100might be mutated in its second position to yield 0100010**.**

Since transparency is simply the difference between original sample and modified sample. So, if we can decrease the difference of them, transparency will be improved. There is an example of adjusting for expected intelligent algorithm below.

Sample bits are: 001**0**1111 = 47
Target layer is 5, and message bit is 1
Without adjusting bit string: 001**1**1111 = 63 (difference is 16)
After adjusting bit string: 001**1**0000 = 48 (difference will be 1 for1 bit embedding)

A. GA-based Approach with reduced RSA Encryption Security for Audio SteganographyAlgorithm:

Embedding of message streams in an audio file:

Steps:

1. Convert Message data to byte streams.
2. Encryption of message is done by using reduced RSA algorithm. The sender signs a message with public key exchange that is two sides cooperates to exchange message.
3. Read an audio file byte-wise and Convert to an integer array (16 bits sample).
4. Generate n (2-16) number of chromosomes of 16 genes by inserting a Message bit into 16 bits audio sample in n (2-16) random positions.
5. To generate next generation population, following GA operators based algorithm is used:

Let 'position' is the insertion position of the audio data,
If position = 1, then no action is to be taken
If position = 2 to 16
If 0 bit is to be embedded
If data bit is for position i
If data bit on i-1 position to 1 position are 0's, performcrossover operation with 1……11 (i-1 no's) bit string
If data bit on i+1 position is 0, perform mutation operation oni+1 position and crossover operation for i-1 to 1 with0…..00(i-1 no's) bit string
If data bit on i+1 and i-1 to 0 are 1 no action is taken
If 0 bit is to be embedded
If data bit is 0 then no action is taken
If 1 bit is to be embedded
If data bit is 1 then no action is taken
If 1 bit is to be embedded
If data bit is 0 then
If layer is 1, then no action is taken
If layer = 2 to 16
If i-1 to 0 positions are holding 1 crossover operation for i-1 to1 with 0…..00(i-1 no's) bit string
If i+1 position holding 1 and i-1 to 1 position holding 0,perform mutation operation on i+1 position and crossover operation for i-1 to 1 with 1…..11(i-1 no's) bit string

6. Now from above GA steps, select the best chromosome, which has the minimum difference with the original 16 bit audio sample.
7. Here fitness value is the position/location number for which we get the best chromosome. Again, the position number, best chromosome and distortion are closely related, because whenever we will choose the best chromosome, it will reducethe distortion.
8. Fitness values are representing two things here
9. Position number which is very important at the receivingend to extract the message.
10. Distortion which is again very important regarding security. So, multi-objectiveGA is used here.
11. Embedding the fitness value to the next audio data sample.
12. Writing stego-audio samples: Convert 16 bit stego audiosample to 8 bit bytes. And write stego audio byte stream to audio file

Extracting hidden message:

At the time of inserting message data bits into an audio sample, we optimize the discrepancy between cover data and stego-data,so need to focus more over the original stego- audio. So, to extract the hidden message data we need to know only theposition number of hidden message bits.

Steps:

1. Read the stego audio file byte-wise. And convert it into an integer array (16 bits sample).
2. From the fitness value or location number of thehidden message data bit's into the stego audio sample, extract message data bits from the stego audio file.
3. To get 8 bits of the message data and random location numberfrom audio data, choose 16 (16 bits) stegoaudio data.
4. Get the message byte streams for all the random positions.
5. Convert message byte stream to a data file.
6. Then write the extracted message data to a message file
7. Decryption of message is done by using reduced RSA algorithm. The receiver signs a message with private key and decrypts the message file.
8. **8.** Write the audio byte stream to an audio file by converting 16 bit sample to 8 bits.

IV. PERFORMANCE EVALUATION PARAMETERS

Latest audio Steganography techniques and their performance analysis are presented in these Sections. To analyze the performance of Steganography techniques three parameters are used[5].

A. *Embedding Capacity*

The embedding capacity (EC) indicates the maximum data size that is possible to hide in the cover object. It is defined as follows:

$$EC = \frac{\text{Secrete message size}}{\text{Cover object size}} \qquad (1)$$

B. *PSNR*

The Peak to Noise ratio provides the similarity between the cover object, the original file, and the stego object, the file where the private or secret message has been hidden. It is defined by the Mean Square Error (MSE):

$$MSE = \left(\frac{1}{MN}\right) \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (I(i,j) - I^{i}(i,j))^{2}$$

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \quad (2)$$

C. *Embedding Efficiency (EE)*

From the Embedding Efficiency(EE), it is indicated the average number of bits inserted for each variation. It is defined as follows:

$$EC = \frac{\text{Secrete message size}}{\text{Number of variations}} \quad (3)$$

V. CONCLUSIONS

By using this concept ofdata hiding the viewer will not be able to dubious that thedata is there at all. Again, if somebody knows that data is inthe audio, it is very difficult to extract the data from the hostaudio. By using the reduced RSA we can decrease the complexity of message file. And using the proposed genetic algorithm, message bits could beinserted into multiple, imprecise and deeper layers to achievehigher capacity and robustness.The basic idea behind this is to maintained unpredictability in message bit insertioninto audio data. This will be helpful for hiding the data from hackers and to provide a good, well-organized method for hiding the data from hackers and sent to the endpoint in a safer manner.

REFERENCES

[1] Fridrich, J. et al. (2000) ‘Steganalysis of LSB encoding in color images’, Proceedings of the IEEE International Conference on Multimedia and Expo, IEEE Press, New York, pp.1279–1282

[2] Krishna Bhowal, AnindyaJyoti Pal, Geetam S. Tomar, P. P. Sarkar, “Audio Steganography using GA”, IEEE Proceedings, 2010.

[3] FatihaDjebbar, BeghdadAyady, Habib Hamamzand Karim Abed-Meraim, “A view on latest audio stegnography”, International Conference on Innovations in Information Technology, 2011, pages 409-414.

[4] Shilpa M Pund, Chitra G Desai, “Enhancing RSA algorithm using Mersenne Primes with reduced size of encrypted file”, International Journal Of Computers & Technology Vol 7, No 1, 2013.

[5] Raffaele Pinardi1, Fabio Garzia1,2, Roberto Cusani1 ,”Peak-Shaped-Based Steganography Technique for MP3 Audio”,Journal of Information Security, 2013, 4, 12-18

[6] M. Nutzinger,”Real-time Attacks on Audio Steganography”, Journal of Information Hiding and Multimedia Signal Processing, Ubiquitous International Volume 3, 2012 ISSN 2073-4212

[7] Prof. Samir Kumar, BandyopadhyayBarnali, Gupta Banik, “Lsb modification and phase encoding technique of audio steganography revisited”, International Journal of Advanced Research in Computer and Communication Engineering Vol. 1, Issue 4, June 2012

[8] JuhiSaurabh, Asha Ambhaikar , Audio Steganography using RPrime RSA and GA Based LSB Algorithm to Enhance Security, International Journal of Science and Research (IJSR), India Online ISSN: 2319-7064

[9] MazdakZamani, HamedTaherdoost, Azizah A. Manaf, Rabiah B. Ahmad, and Akram M. Zeki, “Robust Audio Steganography via Genetic Algorithm”, IEEE, 2009.