ISSN:0975-9646

Encrypt an Audio file using Combine Approach of Transformation and Cryptography

Vinita Makwana[#], Neha Parmar [*]

[#]*Computer Science Department, Parul Institute of Technology,*
Gujarat Technology University
India

[*]*Asst. Prof. Of Computer Science Department, Parul Institute of Technology,*
Gujarat Technology University
India

***Abstract*— Encryption is a technique which is used to encode a file which can only accessible by authorised person. In this paper, an audio file is encrypted by applying transformation and cryptography. Transformation is used to convert an audio file from time domain to frequency domain. In this we can apply encryption on only lower frequency band. So transformation provides lower frequency band. Cryptography is the study of information hiding. For transformation FFT (Fast Fourier Transformation) is used. For encryption RSA is used. After applying the encryption on different frequency bands, we observe that, the encryption on the lower frequency band is more effective than the higher one. So, we would apply encryption on lower frequencies.**

***Keywords*— FFT, Cryptography, Encryption, Transformation Technique.**

I. INTRODUCTION

In network communications, there is one basic problem of transferring the file securely. There are a lot of encryption techniques available for a text file. If sender wants to send a file to receiver, it must be encrypted. First of all, there must be fast, efficient, easy, reliable, security algorithm for an audio file, which lowers the chances of local security breaks. Main focus resides on the performance of an audio file. If applying cryptographic techniques on the lower frequency of an audio file makes better and effective result. In signal processing, some transformations are used for converting signal from time domain to frequency domain. After applying any algorithm for secure the file makes better performance and more secure.

RSA

RSA stands for Ron Rivest, Adi Shamir and Leonard Adleman, who first publicly described the algorithm in 1977. **RSA** is a cryptosystem, which is known as public-key cryptosystems. In such a cryptosystem, the encryption key is public and decryption key is private which is kept secret. It is also called asymmetric encryption. This asymmetry is based on the practical difficulty of factoring the product of two large prime numbers, the factoring problem. In RSA, User creates two large prime numbers and publishes the product of that numbers and their public key. The prime factor must be kept secret. By using the public key, anyone can encrypt a message. But the private is secret; no one can decrypt that message[1] .

FFT

The basic ideas of this were popularized in 1965, but some FFTs had been previously known as early as 805.The Fast Fourier Transformation is an algorithm to compute the discrete Fourier transform and it's inverse. A Fourier Transform converts time domain to frequency domain and vice versa; an FFT rapidly computes such transformations. There are many applications of FFT in engineering, science, and mathematics. An FFT computes the DFT and produces exactly the same result as evaluating the DFT definition directly; the only difference is that an FFT is much faster[2].

DEFINITION OF FFT[2]

An FFT computes the DFT and produces exactly the same result as evaluating the DFT definition directly; the only difference is that an FFT is much faster. Let $x0$, ..., xN-1 be complex numbers. The DFT is defined by the formula Evaluating this definition directly requires O(N2) operations: there are N outputs Xk, and each output requires a sum of N terms. An FFT is any method to compute the same results in O(N log N) operations. More precisely, all known FFT algorithms require Θ(N log N) operations, although there is no known proof that a lower complexity score is impossible.

To illustrate the savings of an FFT, consider the count of complex multiplications and additions. Evaluating the DFT's sums directly involves N2 complex multiplications and N(N−1) complex. The well-known radix-2 Cooley–Tukey algorithm, for N a power of 2, can compute the same result with only (N/2)log2(N) complex multiplications and Nlog2(N) complex additions. In practice, actual performance on modern computers is usually dominated by factors other than the speed of arithmetic operations and the analysis is a complicated subject, but the overall improvement from O(N2) to O(N log N) remains.

II. SURVEY ON FFT APPLICATION.

In first paper, we show the NUFFT concept. It works with FFT but non-uniform manner. NUFFT is the fast algorithm to calculate the discrete Fourier transform with little error. In audio coding, the time domain is uniform but the

frequency domain is non uniform. NUFFT has a better performance in the numerical precision and audio quality at low bit rates and narrow bandwidth[3].

Another paper is based on MORPHING. Speech morphing aims to preserve the shared characteristics of the starting and final signals, while generating a smooth transition between them. Voice morphing technology has numerous applications such as text-to-speech adaptation, where the voice morphing system can be trained on relatively small amounts of data and allows new voices to be created at a much lower cost than the currently existing systems. FFT is used roe extracting features[4].

Another paper is based on transformation. The Gabor transformation with a Gaussian window has several advantages over classical short time transformations such as the windowed FFT and Wavelets. The FFT approach outperforms the DFT approach for many window lengths. The finer the window length is chooses, the smaller are the speed-up factors. Also the longer the window becomes, the better the FFT performs even on very fine window length raster [5].

III. METHOD FOR ENCRYPTION AND DECRYPTION

In our approach, we are using Modified FFT. We have done some modification in FFT.

Ordinary FFT can only work with complex numbers. It can give final output in complex numbers. Encryption is hard to deal with complex number. So we are modifying FFT that gives us real or arithmetic numbers.

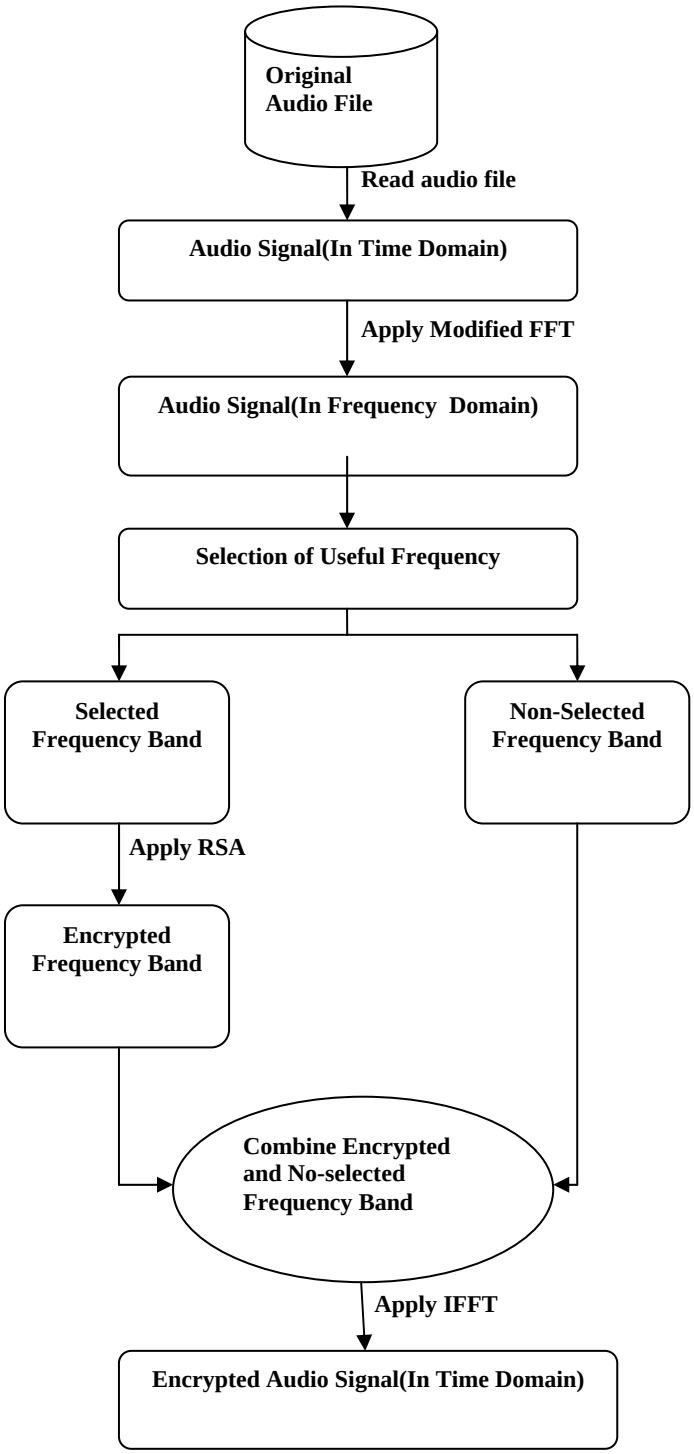
Steps for Encryption:

1. Read Original Audio file.
2. Apply Modified FFT on that audio file.
3. Select lower frequency band.
4. Apply encryption on selected lower frequency band. i.e. RSA technique.
5. Combine Encrypted signal and Non-selected signal.
6. Apply IFFT on signal.

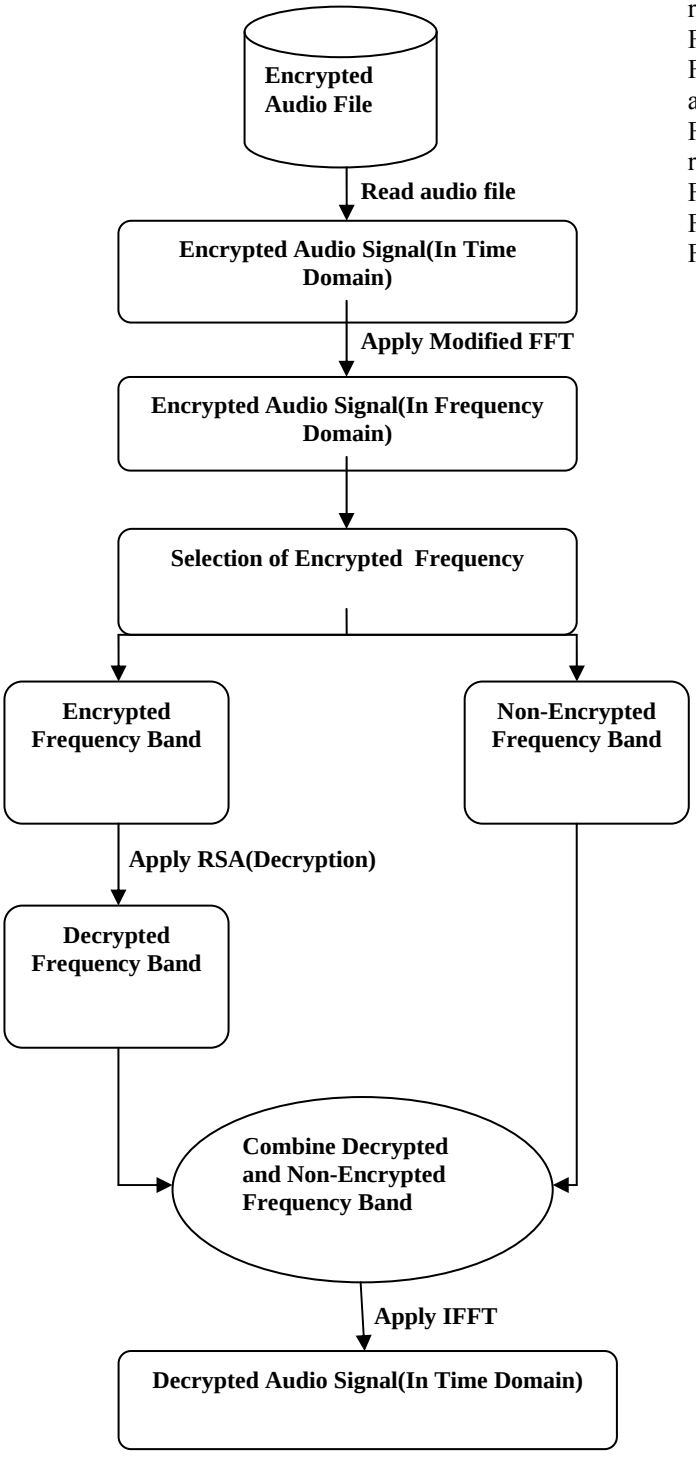
Steps for Decryption:

1. Take encrypted audio file.
2. Apply Modified FFT on that audio file.
3. Select encrypted audio signal.
4. Apply Decryption technique on that signal. i.e. RSA.
5. Combine Decrypted and Non-selected signal.
6. Apply IFFT on that signal.

Flow Diagram of Encryption:



Flow Diagram of Decryption:



IV. EXPERIMENTAL RESULTS

We conduct experiments on many audio file, we get the results.

Figure 1 indicate original sound wave in time domain.

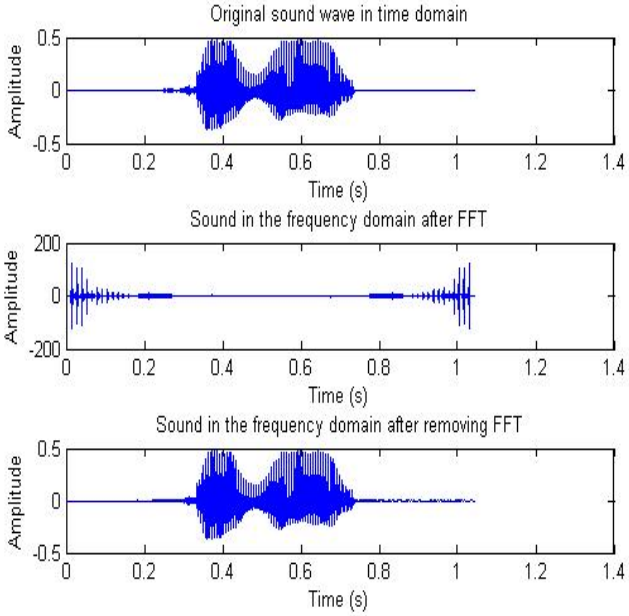
Figure 2 indicate sound wave in the frequency domain after applying modified FFT.

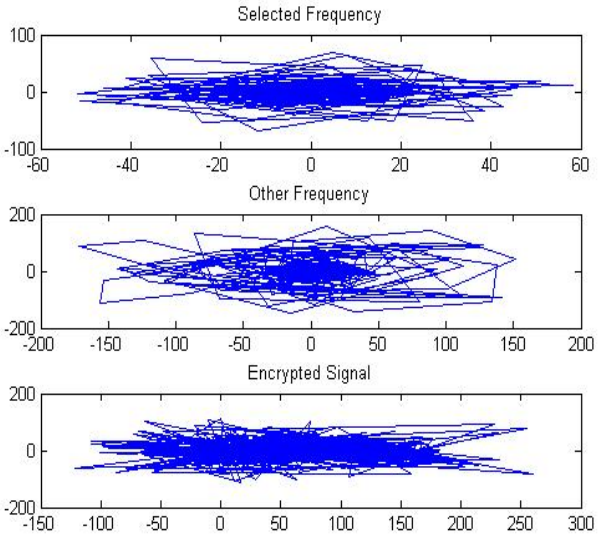
Figure 3 indicate sound wave in frequency domain after removing FFT

Figure 4 indicate selected frequency band

Figure 5 indicate Non-Selected frequency band

Figure 6 indicate encrypted signal





V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a encryption technique with transformation technique. This approach first transforms and then encrypt only selected portion of an audio file. Here fft modification has done which can help us deal with real or arithmetic numbers, which can also help us to encrypt frequency band easily. The important portion is encrypted so that the audio security is protected against interceptors or eavesdroppers in the network. To improve the security or performance, we can use other encryption technique aes, des or transformation techniques in future.

REFERENCES

[1] http://en.wikipedia.org/wiki/RSA_%28cryptosystem%29
[2] http://en.wikipedia.org/wiki/Fast_Fourier_transform
[3] Zheng Deng , Jing Lu, "*The Application of Nonuniform Fast Fourier Transform in Audio Coding*", 978-1-4244-1724-7/08/$25.00 ©2008 IEEE, ICALIP2008.
[4] Palak Chawda, Prof. Jaikaran Singh, Prof. Mukesh Tiwari, "*High Speed FFT Based Audio MORPHING Processor Using VHDL*", International Journal of Engineering Sciences and Research Technology, Vol. 2(1), Jan 2013, ISSN: 2277-9655.
[5] *C. G. v. d. Boogaart, R. Lienhart,* "*FAST GABOR TRANSFORMATION FOR PROCESSING HIGH QUALITY AUDIO",* Multimedia Computing Lab, University of Augsburg, 86159 Augsburg, Germany.
[6] Sheetal Sharma,Lucknesh Kumar , Himanshu Sharma, "*Encryption of an Audio File on Lower Frequency Band for Secure Communication*", International Journal of Advanced Research in Computer Science and Software Engineering Vol. 3,Issue 7, July 2013, ISSN: 2277 1278X