International Journal of Advanced Engineering and Technology
www.newengineeringjournal.com
ISSN: 2456-7655
Received: 14-07-2022, Accepted: 01-08-2022, Published: 16-08-2022
Volume 6, Issue 2, 2022, Page No. 21-29

Residue architecture enhanced audio data encryption scheme using the rivest, shamir, adleman algorithm

Abdul-Barik Alhassan[1], Abdul-Hakim Mahama[1], Salamudeen Alhassan[2]
[1] University for Development Studies, Tamale, Ghana
[2] C. K. Tedam University of Technology and Applied Sciences, Navrongo, Ghana

Abstract
This research is relevant in situations where there is the need to speed up the Rivest, Shamir, and Adleman (RSA) encryption process of audio data files that are transmitted or transferred through public channels, as well as when the security protocols involved in the transmission of such data is questionable or when data is being archived on suspicions storage devices locally. The research seeks to aid individuals, organisations or corporate bodies in implementing strategies to ensure a fast and secure way for audio data encryption. The following procedures or methods are used to implement the proposed algorithm. The audio data is obtained and pre-processed. The pre-processed information is then extracted and manipulated. The Residue Number System (RNS), specifically the Chinese Remainder Theorem (CRT)-based approach, is then applied to the RSA cryptographic process, and the reconstruction of data is done. The performance of the proposed scheme was evaluated using different audio files, which was much faster in terms of speed of execution and security than the traditional RSA.

Keywords: audio, residue number system, rivest shamir adleman, chinese remainder theorem, cryptography, security, algorithm

Introduction

Individuals and organisations have private data files that require protection; these data files appear in text, video, audio, graphics, animations, and various file formats. Before the arrival of modern computers, these files or documents were stored in file cabinets, safes or cupboards with locks, but as technologies progressed, these files are now stored on computers and various storage media (the cloud, databases, pen drives, external hard drives, digital versatile disks (DVD)) etc. The first line of defence for most data was through the operating system (OS) with passwords. This has been rendered vulnerable to attacks that bypass the OS authentication procedures or forcefully break it via brute force attacks. Hence, various cryptographic methods boost data security in systems [5].

The research seeks to propose an approach for securing audio data files through Rivest, Shamir and Adleman (RSA) encryption algorithms and Residue Number System (RNS) theorem algorithms to enhance the processing speed of the RSA encryption process.

The process of inventing and applying algorithms to encrypt messages in formats that make it impossible for unwanted or unauthorised parties to process or divulge the contents of such information is known as cryptography [2, 4, 5]. The decryption technique, as discussed below [8, 10], allows the intended or real recipients of the encrypted message to reveal the contents by utilising the approved secret keys agreed upon by the transmitter and receiver.

1. Secret-key (Symmetric) cryptosystems, as presented in [2, 5], are used in instances where a singular key is distributed to the transmitter and a receiver for data encryption and decryption processes.
   The Key has a massive influence on the encrypting and decrypting process of the data files. This is because the cryptographic power of symmetric encryption lies in the dimensions of the Key. There are two forms of symmetric algorithms (the Block and Stream ciphers).
2. (Asymmetric) public-key cryptosystems, as in [2, 5, 8], is an encryption method that generates and uses two separate keys, called private and public keys. Because the two keys are mathematically related, the private key is used for encryption while the common public keys are used for decryption. Rivest, Shamir and Adleman (RSA), Rabin and ElGamal are examples of asymmetric cryptosystems. Encryption techniques for text data are dedicated to cryptography, although other multimedia data formats, such as audio data, have some cryptographic algorithms.

Cryptographic techniques are frequently used to ensure the security of valuable audio data files. The Residue Number System (RNS) is based on an unweighted unique number system that supports limited distribution of carries, parallel computation of data, low power consumption, faster mathematical computations, and secure communications. RNS is applied in conditions requiring arithmetic computations of numbers, including addition, subtraction, and multiplication. It is denoted by a set of modules which is optimal given co-primes [8, 9, 10].

The combination of various systems could be employed to enhance these systems to make the computation of modular sets easier as well as extend the dynamic range [3, 8, 9].

Related Works

The area of audio data encryption regarding the use of Rivest, Shamir and Adleman (RSA) algorithm has been extensively researched. In the work of [6], the RSA technique for voice data encryption and decryption are made more efficient. Initially, five

hundred (500) Bangla (a local Indian language) voice words were recorded from six different speakers and saved as RIFF (.wav) files. Their proposed scheme was then used to extract the data from these words and then encrypted and decrypted. According to [5], a new proposed type of encryption/decryption approach that guarantees the end-to-end secrecy of audio transmissions in real time communication system is essential. It decrypts and encrypts each captured sample at the receiver before processing the sent audio stream. The method is applicable to both complex and simple sound signals such as GSM, VoIP, telephone, and simple radio. A technique was provided in [8], which can be utilised with any sort of message, including audio, video, image, or text data. The study proposed a new encryption/decryption technique that relied on the RSA asymmetric key algorithm to ensure the audio signal's end-to-end privacy and security while preserving the signal's quality whether stored or transferred via any communication channel.

Other researchers then leveraged on the efficient properties of Residue Number Systems (RNS) to secure and speed up the encryption and decryption process to make the Rivest, Shamir and Adleman (RSA) algorithm robust. In [3] an efficient RNS implementation of RSA cryptography based on a non-iterative and pure RNS division algorithm by Mansoureh and Mohammed (2012) is proposed. Their algorithm avoids the total loop and supports all numbers in the range as denominators. A new approach on audio encryption using an algorithm base on combining signal samples with RNS through permutation is propose in [4]. The entire method entails permuting signal samples with RNS, resulting in a specific noise. The audio signal is then mixed with the algorithm's noise to produce a quiet signal. The method works by transforming audio signals to silent signals and increasing their security. In [1], a scheme that reduces the computational complexity in overall video encryption is presented. The proposed scheme uses RNS and implemented using the Java programming language, which efficiently protects video data from unauthorised access during transmission and storage.

Materials and Methods

For the success of this research, certain methods and materials are necessary and sufficient. The Residue Number System (RNS) will be applied to the Rivest, Shamir and Adleman (RSA) algorithm and implemented using Python 3.5.

A. Rivest Shamir Adleman (RSA) Algorithm

The RSA algorithm was developed by Ron Rivest, Adi Shamir and Leonard Adleman in the year 1997. This cryptographic method has stood the test of time and has become one of the most popular asymmetric cryptosystem known to man. Their method utilised two prime numbers of distinct magnitudes which are multiplied and passed through various steps to generate the private and public key pairs, which are further used in the encryption and decryption processes. In the use of the RSA algorithm, messages are ciphered by using the public Key at the transmitters end. The public Key used to encrypt the messages can be publicised and can be transmitted through the internet. The RSA algorithmic processes and key generation is presented below.

1. Two secret prime numbers of distinct values (p and q) is selected at random but should be comparable in magnitude. This is done due to security reasons.
2. Then, the value of n which is calculated as the product of the selected prime numbers p and q *is* ($\mathrm{n} = \mathrm{pq}$), which is the modulus of the private and public Key. The length of n (in bits) is the key length.
3. The value of $\lambda(n)$ is calculated, Carmichael's totient function is denoted by (λ), $\lambda(n) = lcm$. The value of n can be represented as ($\lambda(p)$,$\lambda(q)$), so $\lambda(p)$ is equal to $\varphi(p)$, which is also equal to $\mathrm{p} - 1$ and $\lambda(q) = q - 1$. Thus $\lambda(n)$ is equal to $\mathrm{lcm}(\mathrm{p} - 1, \mathrm{q} - 1)$. Through the Euclidean algorithm, the value of the lcm can be obtained by $lcm\ (a,b) = |ab|/\gcd(a,b)$.
4. An integer value e is picked such that $1 < e < \lambda(n)$ and $\gcd\ (e, \lambda(n)) = 1$. This value is a component of the published public Key.
5. An integer value d which is the modular multiplicative inverse of e modulo $\lambda(n)$ is given as $d \equiv e^{-1} \pmod{\lambda(n)}$. This value is a component of the private Key which is kept secret.

From the above procedures, two values (n,e) are generated, representing the modulus and encryption exponent in the public key whiles the value pair (n,d) represents the modulus and the decryption exponent in the private Key. The value of $p, and\ \emptyset(n)$ need to be confidential. This is because are required to obtain d which must be kept secret.

During encryption, the number pair (n, e) is used to encrypt a plain text message (M), which results in the generation of the cipher text message to be transmitted C, using:

$$C = M^e mod\ n \text{----------(1)}$$

In the decryption process, the number pair (n, d) is used to decrypt ciphered text message (C), which results in the generation of the original plain text message (M), using:

$$M = C^d mod\ n \text{----------(2)}$$

It is worth to note that, the encryption and decryption keys correspond to one another. Therefore, the use of a non-corresponding decryption key will generate a wrong plain text.

B. Residue Number System Architecture

A Residue Number System (RNS) constitutes a huge integer value with a series of smaller integers, allowing for more efficient calculation. RNS scheme provides a collection of moduli that are independent of one another. Each modulus's residue represents an integer, and arithmetic operations are performed on the residues separately. The arithmetic operations based on RNS can be done separately on different moduli to eliminate the carry in addition, subtraction, and multiplication, which is generally time

intensive.
The research uses the RNS (Chinese Remainder Theorem-CRT) variant in the implementation of RSA algorithm for cryptographic processes, Pre-computed and kept as part of the private Key per the following values: *p* and *q* the primes from the key generation,

- $d_p = d \pmod{p-1}$
- $d_q = d \pmod{q-1}$ *and*
- $q_{inv} = q^{-1} \pmod{p}$

These values allow the receiver to calculate the exponentiation $m + cd(modpq)$ more quickly, as follows:

1. $m1 = c^{dp} \pmod{p}$
2. $m2 = c^{dq} \pmod{q}$
3. $h = q_{inv}(m1 - m2) \pmod{p}$
4. $m = m2 + hq \pmod{p * q}$

Even though two modular exponentiations must be performed, this is more efficient than calculating the exponentiation by squaring.
The reason for this is that, both of these modular exponentiations utilise a smaller exponent and a smaller modulus. The proposed scheme is then implemented by obtaining the audio data, pre-processing the audio data and then extracting and manipulating the data, RNS is then applied to the RSA algorithm before the encrypted data is reconstructed as shown below in fig 1

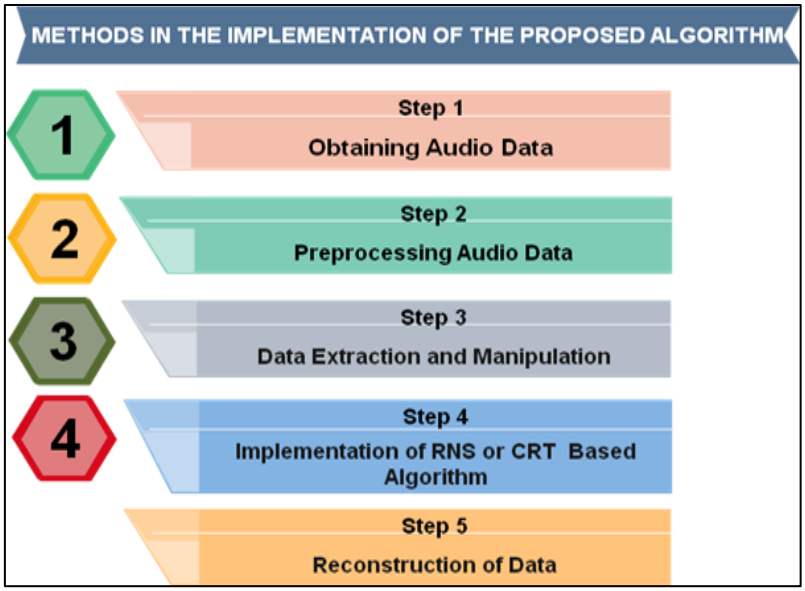

Fig 1: Diagram of proposed algorithm procedure

C. Obtaining Audio Data
The audio files are obtained from recordings made via the microphone of an earpiece using the laptop computer. A custom audio recording program was written in python 3.5 using the Pycharm Integrated Development Environment (IDE). The recordings were basically sentences and words for a few seconds ranging from 3 to 9 seconds. The recorded audio files were then saved in a wave file format (.wav). The audio was recorded at 11025Hz sampling rate and recorded in chunks of 1024 samples.

D. Pre-Processing of Audio Data
To obtain the wave data, we first extract the wave file properties which is contained in the wave file, then split the data into its respective sample frames. This procedure is achieved by accurately identifying the initial start and end periods of each sample audio signal.

E. Data Extraction and Manipulation
The split data which was extracted is stored as numbered dimensional array (nd array). The array data has to be subsequently converted from arrays to integers and stored in a text file (.txt). The text file will therefore serve as the input message for encryption during the cryptographic process. These extractions and manipulations are done to make the integers in a range $0 \leq m < n$, where *m* is the message to encrypt and ***n*** is a large positive integer.

F. Implementation of Residue Number System (Chinese Remainder Theorem) Based Algorithm
The application of RNS to the RSA algorithm as already explained involves Key generation, Encryption and Decryption as shown in the diagram below and in fig 2.

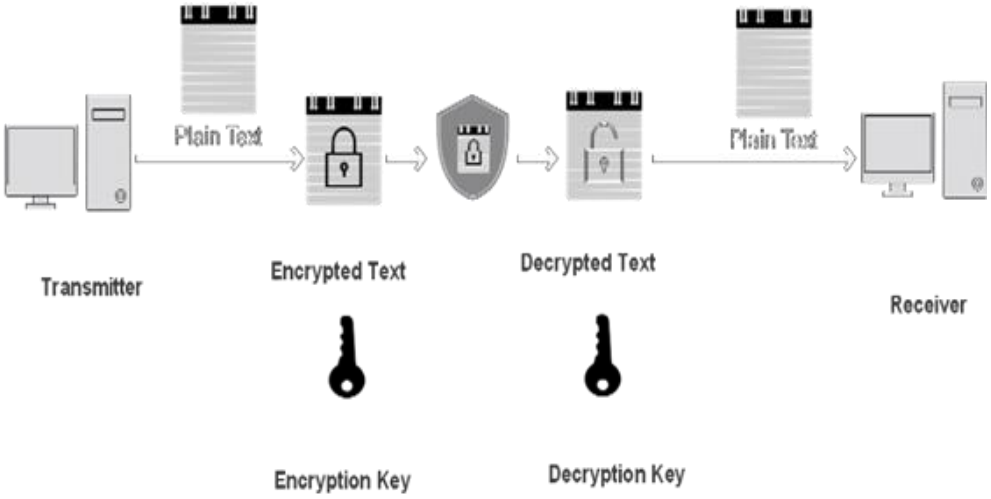

Fig 2: Diagram of the RSA cryptographic process

G. Key Generation

The research process in the implementation of the proposed algorithm, first utilises the traditional key generation processes in RSA as outlined.

H. Encryption

Encryption Process: The number pair (n, e) is utilised in the encryption of plain text messages (M), which produces the cipher text message C, by: $C = M^e mod\ n$.

The ciphered message C is then sent from the transmitter to the receiver for decryption.

However, the proposed algorithm uses the RNS or the CRT based approach, the text file (.txt) generated from the data manipulation process, will serve as the input message for the encryption process. RNS is used to improve encryption speed using Modulus of Factors (mod pq using mod p and mod q).

The values ep, eq and qinv that are part of the public Key are calculated as follows:

Pre-computed and kept as part of the private Key are the following values:

p and q the primes from the key generation,

1. $d_p = d\ (mod\ p - 1)$
2. $d_q = d\ (mod\ q - 1)$ and
3. $q_{inv} = q^{-1}\ (mod\ p)$

These values allow the receiver to calculate the exponentiation $\mathrm{m} + \mathrm{cd}(\mathrm{mod pq})$. more quickly, as follows:

4. $m1 = cdp\ (mod\ p)$
5. $m2 = cdq\ (mod\ q)$
6. $h = qinv(m1 - m2)\ (mod\ p)$
7. $m = m2 + hq\ (mod\ p * q)$

After the entire encryption process, an encrypted message (ciphered text) generated from the input text file will be generated at the end of the encryption process. The cipher text file (.txt) will then serve as the input for the decryption process.

I. Decryption

Decryption processes of the RSA cryptographic process begins with the private key component (n), which is used to decipher the ciphered message C generated during the encryption procedure. At the receiver side, the following formula is used to generate the simple message M:

$$M = C^d mod\ n$$

When comparing the traditional RSA with the algorithm proposed in this research, the speed of decryption calculation is improved using the factor module (mod pq using mod p and mod q). The encrypted text file (.txt) that serves as input is decrypted using the following approach. Pre-computed and kept as part of the private Key are the following values:p and q he primes from the key generation,

1. $d_p = d\ (mod\ p - 1)$
2. $d_q = d\ (mod\ q - 1)$ and
3. $q_{inv} = q^{-1}\ (mod\ p)$

These values allow the receiver to calculate the exponentiation m = cd (mod pq) more quickly, as follows:

4. $m1 = c^{dp} \pmod p$
5. $m2 = c^{dq} \pmod q$
6. $h = q_{inv}(m1 - m2) \pmod p$
7. $m = m2 + hq \pmod{p * q}$

(Wikipedia contributors, 2021).

J. Reconstruction of Data

To regain the wave data, we first reconstruct the integer data into the initial start and end periods and then piece the containing audio samples based on the detected starting and end period, then into arrays thus forming the complete sample frames data. The pre-extracted file header details from the beginning of the wave file are attached back to the sample frames data to reconstruct the wave sound file.

K. Algorithm Based on the RNS (CRT) Based Approach for Audio Data Encryption

The algorithm outlined below is based on both the traditional RSA encryption Method and RNS (CRT) based method. The RNS (CRT) based approach enhances the encryption and decryption computational aspects of the cryptographic process by splitting the messages into two parts, calculating them separately and finally merging them into one file at the end of the process.

The outlined algorithm for the audio data encryption using RNS (CRT) is a follows:

1. Start
2. Input "wave audio file"
3. Extract data (audio sample frames)
4. Generate both private and public Key using RSA key generation and RNS (CRT)
5. Encrypt sample frames using RNS (CRT) based process and encryption key
6. Decode using RNS (CRT) based process and decryption key
7. Reconstruct sample frame data and add header file data to construct the wave file
8. Output new wave file.
9. Stop

Below is the schematic diagram depicting the algorithm.

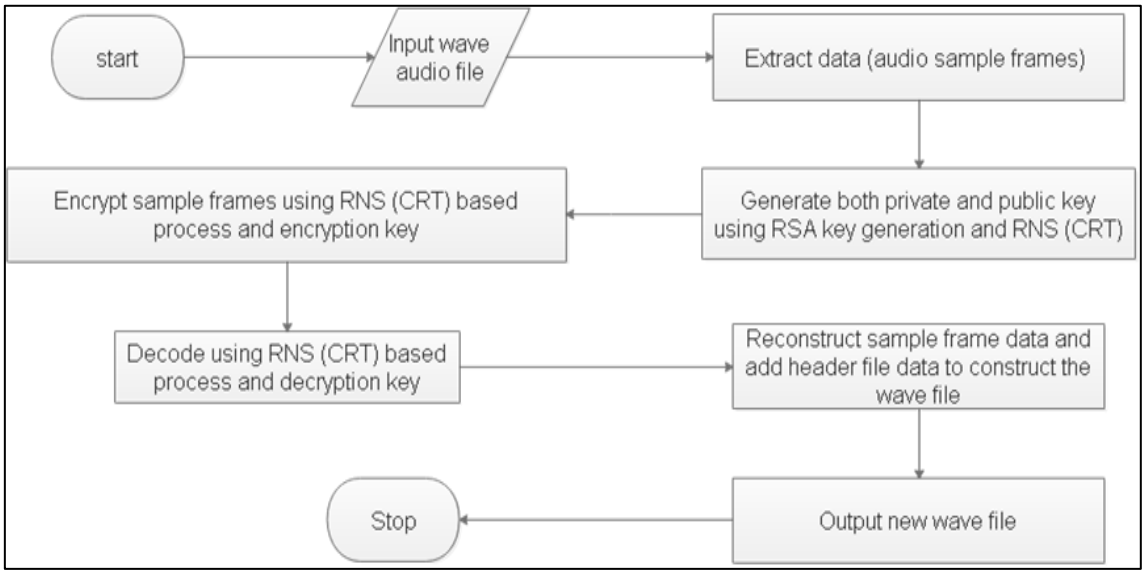

Fig 3: Flow chart for Audio data Encryption and Decryption

Results and Discussions

In this research work, the proposed algorithm uses RNS and RSA and implemented using python 3.5.2 in Pycharm Integrated Development Environment (IDE). The audio data were wave files generated from a personally written python audio recording program. The recordings were made in conjunction with a wired microphone from an earpiece, the generated audio files were saved as .wav file format.

The first part which is the key generation was done with traditional RSA key generation processes and in conjunction with the CRT key generation method. The result of the key generation process was to obtain the public key values (n, e), private key values (n, d), where n is the product of p and q values ($n = pq$).

To encrypt these audio files, the wave data was obtained, and the wave file properties which is contained in the header file embedded at the beginning of the wave file was extracted, then the data was split into its respective sample frames, this procedure is achieved by detecting the correct start and end points of each sample audio signal.

The split data which was extracted is stored as numbered dimensional array (nd array). The array data has to be subsequently converted from arrays to positive integer data or Unicode values and stored in a text file (.txt). The text file will therefore serve

as the input message for encryption after which a ciphered text file will be generated at the end of the encryption process. The cipher text file (.txt) will then serve as the input for the decryption process.
To decrypt, the ciphered text file which serves as the input message for decryption is passed through the decryption algorithm, after the computation processes of the decryption algorithm, the decrypted information is reconstructed back to a wave file.

A. Results
To generate the results, different values for p and q were tested using the traditional RSA cryptographic method and RNS (CRT) based method. Excellent encryption and decryption results were achieved in most of the presented cases where the prime numbers p and q were used in the encryption and decryption processes in terms of the time it takes for the encryption and decryption processes to complete.
During the testing, 3 different p and q values and the resulting private and public key combinations used for test were generated respectively as shown below.

Table 1: RSA Cryptographic Process

Input Audio Filename	Audio file Duration (Seconds)	Sampling Rate (Hz)	Samples	P and Q Value	Encryption Key time (Seconds)	Decryption Key time (Seconds)	Output Audio Filename
Input1.wav	3	11025	44032	23, 29	22.9310	14.0624	Output1.wav
Input2.wav	6	11025	44032	53, 59	53.9508	240.5331	Output2.wav
Input3.wav	9	11025	44032	73,79	115.62733	118.7743	Output3.wav

Table 2: RSA and RNS Proposed Scheme

Input Audio Filename	Audio File Duration (Seconds)	Sampling Rate (Hz)	Samples	P and Q Value	Encryption Key Time (Seconds)	Decryption Key Time (Seconds)	Output Audio Filename
Input1.wav	3	11025	44032	23, 29	8.1093	5.5468	Output11.wav
Input2.wav	6	11025	44032	53, 59	14.6718	12.3281	Output21.wav
Input3.wav	9	11025	44032	73, 79	25.4843	31.3317	Output31.wav

Table 3: Comparison of Encryption Time of RSA Scheme and Proposed Scheme

P and Q Values	RSA Encryption Time (Seconds)	RNS(CRT) Encryption Time (Seconds)	Time Difference of both Methods (seconds)	Time Increase of two Methods (%)
23, 29	22.9310	8.1093	14.2217	62.0195
53, 59	53.9508	14.6718	39.2790	72.8052
73, 79	115.6273	25.4843	90.1430	77.9600

Table 4: Comparison of Decryption Time of RSA and Proposed Scheme

P and Q Values	RSA Decryption Time (Seconds)	RNS(CRT) Decryption Time (Seconds)	Time Difference of both Methods (Seconds)	Time Increase of two Methods (%)
23, 29	14.0624	5.5468	8.5156	60.5558
53, 59	240.5331	12.3281	228.2050	94.8746
73, 79	118.7743	31.3317	87.4426	73.6208

B. Discussion
From the first test conducted, for example, an audio note saved as “input1.wav” was recorded for 3 seconds. The note contained the words “hello testing mic one two hello”. The sampling rate was at 11025Hz and the samples was 44032. The P and Q values were 23 and 29 respectively, the results for the other audio files for the RSA cryptographic scheme and proposed scheme are shown in the above tables.
From the analysis, it can be observed that the RNS (CRT) method is much faster and clearly enhances the speed of audio data encryption and decryption when compared to RSA method.
It can also be observed from the graphs below of the input and output audio files that there is no loss in the audio sound quality as well as playback of both file
sound quality as well as playback of both file

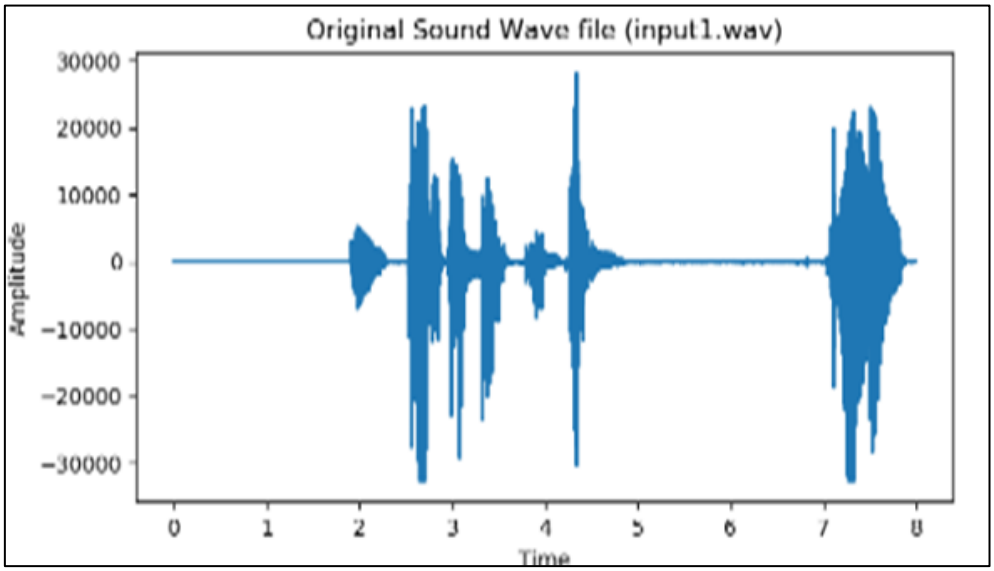

Fig 6: input1.wav sound graph

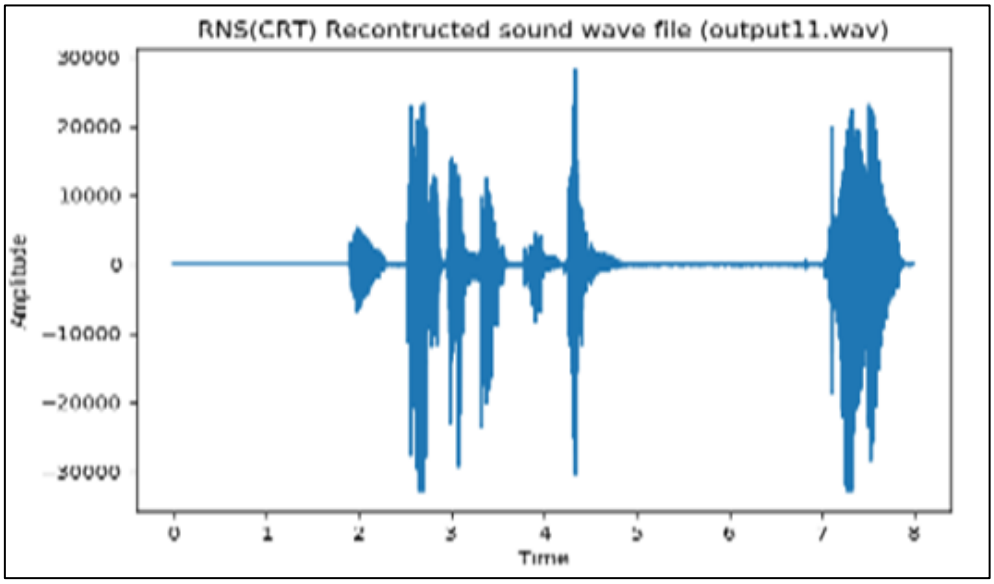

Fig 7: output1.wav sound graph

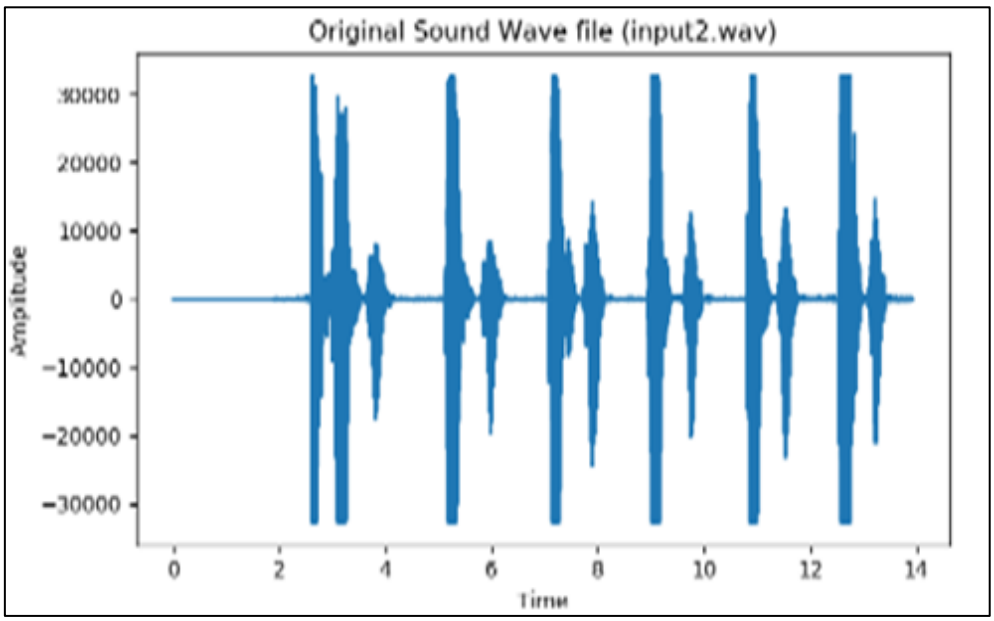

Fig 8: input2.wav sound graph

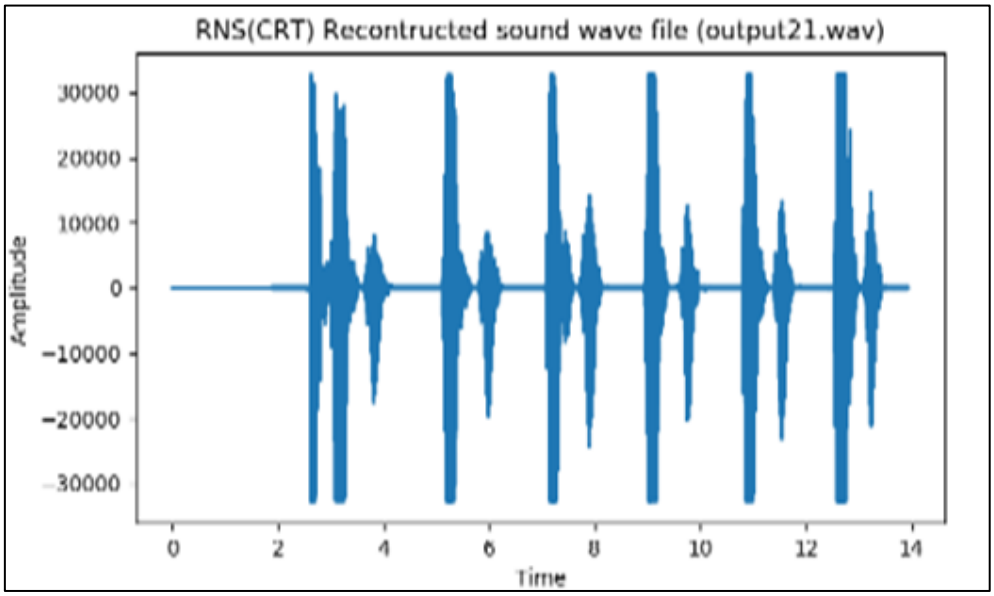

Fig 9: output2.wav sound graph

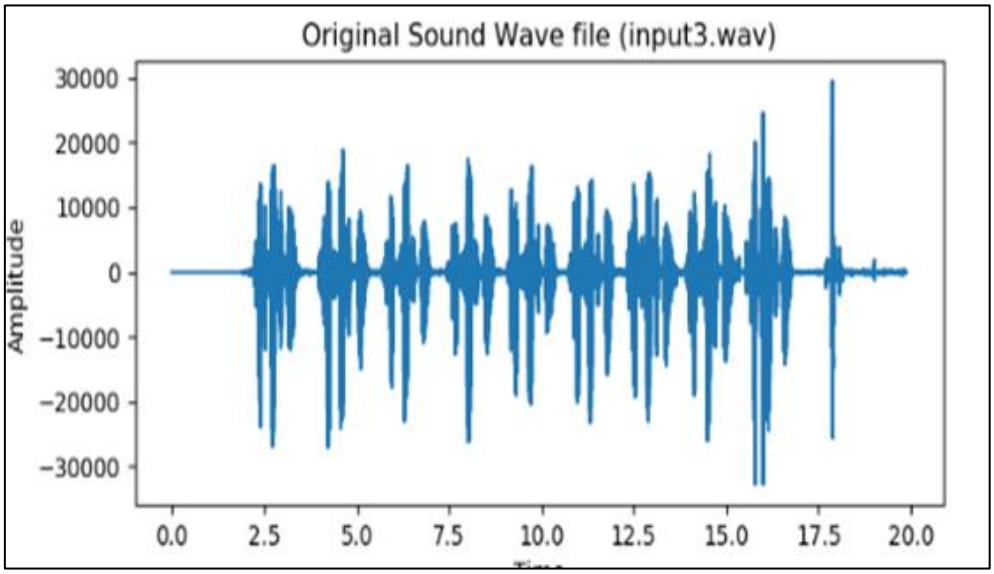

Fig 10: input3.wav sound graph

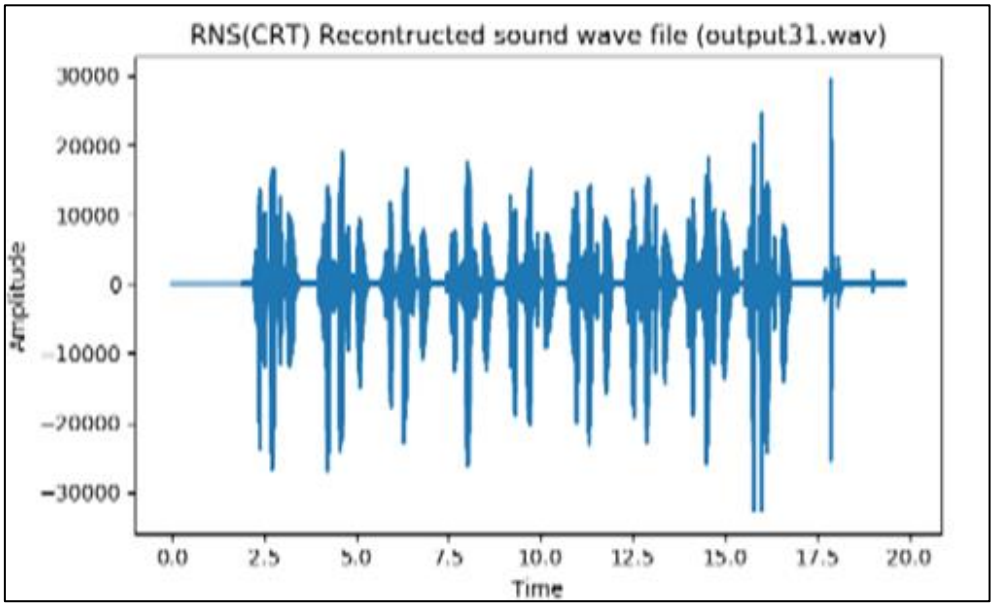

Fig 11: output3.wav sound graph

Conclusion and Extension For Future Work

RSA is a well-established encryption technology that has withstood the test of time. RSA is a public-key cryptographic technology that enables secure communications and digital signatures. The difficulty of factoring large integers contributes to its security. However, with the development of the RNS (CRT) approach, which is a way to improve the performance of decryption and encryption while also reducing mathematical calculations to a significant level. It has also reduced the computer power consumption, and enhanced key generation.

The RNS (CRT) based algorithm is a really efficient algorithm which can be seen from its application in the proposed scheme as applied to archived audio files. This can in future be applied to continuous audio signals transmission.

References

1. Akinbowale NB, Rasheed GJ, Kazeem AG. An Algorithm for a Residue Based Video Encryption System. *Annals. Computer Science Series*,2016:14(2):137-145.
2. Cobb M. *RSA algorithm (Rivest-Shamir-Adleman)*. SearchSecurity., 2018. https://searchsecurity.techtarget.com/definition/RSA
3. Fadulilahi I, Bankas E, Ansuura J. Efficient Algorithm for RNS Implementation of RSA. International Journal of Computer Applications,2015:127(5):14-19. https://doi.org/10.5120/ijca2015906381
4. Fereshteh Ghanbari Adivi, Mohammad Mernia. Audio Signal Encryption Based on Permutation Relations and Residue Number System. Journal of Advances in Computer Research Quarterly,2016:7(3):67-76.
5. Khalil M. Real-Time Encryption/Decryption of Audio Signal. International Journal of Computer Network and Information Security,2016:8(2):25-31. https://doi.org/10.5815/ijcnis.2016.02.03
6. Mijanur Rahman M, Tushar Kanti Saha, Al-Amin Bhuiyan3 M. Implementation of RSA Algorithm for Speech Data Encryption and Decryption. International Journal of Computer Science and Network Security,2012:12(3):74-82.
7. Wikipedia contributors. *RSA (cryptosystem)*. Wikipedia., 2021. https://en.wikipedia.org/wiki/RSA_(cryptosystem)#Using_the_Chinese_remainder_algorithm
8. Salifu A-M, Alhassan A-B. An Enhanced RSA Encryption Scheme based on Mixed Radix Conversion with Data Encryption. International Journal of Computer Applications.,2019:177(17):26-29.
9. Alhassan A-B. A Partial Residue to Decimal Converter for the Moduli Set. International Journal of Innovative Research in Computer and Communication Engineering.,2018:6(11):8928-8933.
10. Alhassan A-B, Gbolagade KA, Bankas EK. A Novel and Efficient LZW-RNS Scheme for Enhanced Information Compression and Security. International Journal of Advanced Research in Computer Engineering and Technology.,2015:4(11):1450-4019.
11. Alhassan A-B, Ibrahim S, Agbedemnab P. The Huffman's Method of Secured Data Encoding and Error Correction using Residue Number System (RNS). Communications on Applied Electronics.,2015:2(9):14-18.