Implementation of RSA Algorithm for Speech Data Encryption and Decryption

Md. Mijanur Rahman[1], Tushar Kanti Saha[2], Md. Al-Amin Bhuiyan[3]

[1,2]Dept. of Computer Science & Engineering, Jatiya Kabi Kazi Nazrul Islam University, Bangladesh.
[3]Dept. of Computer Science & Engineering, Jahangirnagar University, Bangladesh.

ABSTRACT
Today, organizations in both public and private sectors have become increasingly dependent on electronic data processing. This digital data are going through an insecure channel from one place to another and anyone can easily get those important data without the concerns of the sender. So, protecting these important data is crucial task in data communication and Public Key Cryptography is one of the best ways to protect digital data from the unauthorized access. RSA algorithm is extensively used in the popular implementations of Public Key Infrastructures. In this paper, we have done an efficient implementation of RSA algorithm for speech data encryption and decryption. At first, five hundred Bangla speech words were recorded from six different speaker and stored as RIFF (.wav) file format. Then our developed program was used to extract data from these words and this data were stored in a text file as integer data. Finally, we used our implemented program to encrypt and decrypt speech data.

Keywords:
Speech Feature, Cryptography, Encryption, Decryption and RSA Algorithm

1. INTRODUCTION

Data communication is an important aspect of our living. Security of data to maintain its confidentiality, proper access control, integrity and availability has been a major issue in data communication. So, protection of data from misuse is essential. Today in the e-age, the need to protect communications from prying eyes is greater than ever before. Cryptography, the science of encryption plays a central role in mobile phone communication, e-commerce, Pay-TV, sending private e-mails, transmitting financial information and touches on many aspects of daily lives.
A cryptosystem defines a pair of data transformations called encryption and decryption. Encryption is applied to the plain text i.e. the data to be communicated to produce cipher text i.e. encrypted data using encryption key. Decryption uses the decryption key to convert cipher text to plain text i.e. the original data. The symmetric cryptosystem, where the encryption key and the decryption key is the same, can be easily broken if the key used to encrypt or decrypt can be found. To improve the protection mechanism, Public Key Cryptosystem was introduced in 1976 by Whitfield Diffe and Martin Hellman of Stanford University [1]. It uses a pair of related keys one for encryption and other for decryption. One key, which is called the private key, is kept secret and other one known as public key is disclosed [2].
The RSA (named after its authors - Rivest, Shamir and Adleman) is the most popular public key cryptographic algorithm that is used to help ensure data communication security [2]. It is simply based on two main cryptographic processes. First, using a public key it converts an input data called the plaintext into an unrecognizable encrypted output called cipher text (encryption process), such that it is impossible to recover the original plaintext without the encryption password in a reasonable amount of time. Second, using a private key, the RSA then converts the unrecognizable data back to its original form (decryption process) [3]. Today it is used in web browsers, email programs, mobile phones, virtual private networks and secure shells.
This technology is widely expected to be used to conduct billions of dollars in electronic commerce within the next few years. Our work in this paper is focused primarily on the implementation of RSA algorithm for speech data encryption and decryption. For efficient implementation, we have explored the behaviour and feasibility of the algorithm with the change of various input parameters, and finally a user interface is developed to provide an application of our analysis.

2. CRYPTOGRAPHY

Cryptography is the study of Secret (crypto-) and Writing (-graphy). It is the science or art of encompassing the principles and methods of transforming an intelligible message into one that is intelligible and then transforming the message back to its original form. Today's cryptography is more than encryption and decryption. Authentication is as fundamentally a part of our lives as privacy. We use authentication throughout our everyday lives when we sign our name to some document and for instance, as we move to world where our decisions and agreements are communicated

Manuscript received March 5, 2012
Manuscript revised March 20, 2012

electronically, we need to have electronic techniques for providing authentication. Cryptography provides mechanisms for such procedures.
Cryptographic systems are generally classified along three independent dimensions[4]:

1. Type of operations used for transforming plaintext to cipher text. All encryption algorithms are based on two general principles. Those are substitution, in which each element in the plain text is mapped into another element and transposition in which elements in the plaintext are rearranged. The fundamental requirement is that no information be lost. Most systems referred to as product systems, involved multiple stages of substitution and transposition.
2. The number of keys used: If sender and receiver use the same key, the system is referred to as symmetric, single key or secret key conventional encryption. If the sender and the receiver each uses a different key the system is referred to as asymmetric, two key, or public-key encryption.
3. The way in which the plaintext is processed: A block cipher processes the input on block of elements at a time, producing an output block for each input block. A stream cipher processes the input elements continuously, producing output one element at a time, as it goes along.

Standard encryption methods usually have two basic flaws:

(1) A secure channel must be established at some point so that the sender may exchange the decoding key with the receiver; and

(2) There is no guarantee who sent a given message. Public key encryption has rapidly grown in popularity because it offers a very secure encryption method that addresses these concerns.

3. PUBLIC-KEY CRYPTOSYSTEM

The development of public key cryptography is the greatest and perhaps the only true revolution in the entire history of cryptography [4]. With public key techniques, each user has two different keys, one made available to the public and the other kept secret. One of the keys is used to encrypt a message, and the other is used to decrypt the message. If Alice wants to send a secret message to Bob, for example, she looks up Bob's public key and uses it to encrypt the message. Because Bob's public key cannot undo the encryption process, no one who intercepts the message can read it. Only Bob, who possesses the secret key corresponding to his public key, can read the message. Alice never has to meet Bob out of the hearing of others to exchange keys or passwords; this is a substantial improvement over older encryption methods in which an exchange of private keys was necessary.
This system can also be used as a means for Bob to be sure a message comes from Alice. If Alice wants to sign a message, she can encrypt it with her private key. When Bob receives an encrypted message which purports to be from Alice, he can obtain Alice's public key and decrypt the message. If a readable message emerges, Bob can have confidence that the message came from Alice, because Alice's public key would only properly unlock a message which was locked with her private key (known only to Alice). Figure-1 illustrates the public-key encryption process.

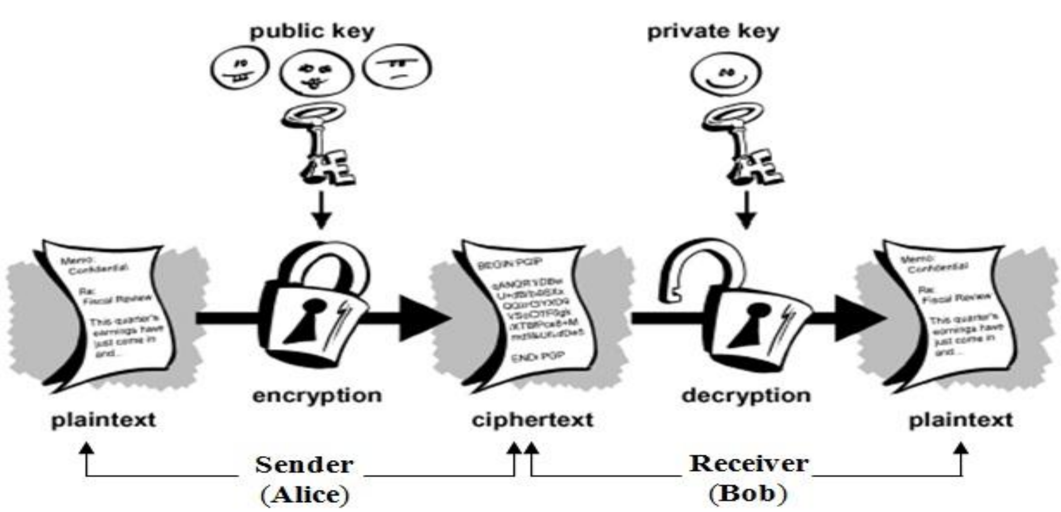


Figure-1. Public-Key Encryption [5].

This type of encryption has a number of advantages over traditional symmetric Ciphers. It means that the recipient can make their public key widely available- anyone wanting to send them a message uses the algorithm and the recipient's public key to do so. An eavesdropper may have both the algorithm and the public key, but will still not be able to decrypt the message. Only the recipient, with the private key can decrypt the message.

This makes it possible for Alice and Bob to simply send their public keys to one another, even if the channel they are using to do so is insecure. It is no problem that another person Eve now gets a copy of the public keys. If Alice wants to send a secret message to Bob, she encrypts the message using Bob's public key. Bob then takes his private key to decrypt the message. Since Eve does not have a copy of Bob's private key, she cannot decrypt the message. Of course this means that Bob has to carefully guard his private key. With public key cryptography it is thus possible for two people who have never met to securely exchange messages.

A disadvantage of public-key algorithm is that they are more computationally intensive than symmetric algorithms, and therefore encryption and decryption take longer. This may not be significant for a short text message, but certainly is for bulk data encryption.

4. RSA ALGORITHM

The RSA Algorithm was named after Ronald Rivest, Adi Shamir and Leonard Adelman, who first published the algorithm in April, 1977 **[6]**. Since that time, the algorithm has been employed in the most widely-used Internet electronic communications encryption program. It is also employed in both the Netscape Navigator and Microsoft Explorer web browsing programs in their implementations of the Secure Sockets Layer (SSL), and by Mastercard and VISA in the Secure Electronic Transactions (SET) protocol for credit card transactions.

The RSA Algorithm is only one implementation of the more general concept of public key cryptography. Typical encryption techniques use mathematical operations to transform a message (represented as a number or a series of numbers) into a *ciphertext*. Mathematical operations called *one way functions* are particularly suited to this task. A one way function is one which is comparatively easy to do in one direction but much harder to do in reverse.

The RSA system uses one way functions of a more complex nature **[7]**. Specifically, the system uses *modular arithmetic* to transform a message into unreadable ciphertext. Modular arithmetic is often called "clock" arithmetic, because addition, subtraction, and the like, work like telling time. In a **12**-hour system, six hours after **10:00** is not **16:00** (**10** + **6** is not equal to **16**); it is **4:00**. This is because we subtract out **12** after doing the addition. In modular arithmetic notation, the operation is as follows:

$\mathbf{4 = (10 + 6) \bmod 12}$

$\mathbf{4 = 16 \bmod 12}$

One can do multiplication in modular arithmetic much the same way addition is done in the above example:

$\mathbf{4 = (8*2) \bmod 12}$

$\mathbf{4 = 16 \bmod 12}$

This process is sometimes called *modular reduction*. Because the number **16** is "reduced" to the number **4** in the above example, one can say that "**16** is reduced modulo **12**."

The RSA system uses multiplication in modular arithmetic. The RSA system multiplies one number (called the *base*) by itself a number of times and the product is then divided by a modulus. The number of times a base is multiplied by itself is called the *exponent* and the process is called *modular exponent*.

$\mathbf{4 = (2*2*2*2) \bmod 12}$

$\mathbf{4 = 2^4 \bmod 12}$

In this example, the number **2** is the base, and is multiplied by itself four times, making the exponent the number **4** and the number **12** is the modulus.

In the RSA encryption formula, the message M is multiplied by itself e times and the product is then divided by a modulus n, leaving the remainder as a ciphertext C:

$C = M^e \bmod n$

In the decryption operation, a different exponent, d is used to convert the ciphertext back into the plain text:

$M = C^d \bmod n$

The modulus n is a composite number, constructed by multiplying two prime numbers, p and q, together:

$n = p * q$

Also, $\varphi(n)$ is known as *Euler's Phi-Function* [12] and can be calculated by using the following equation:

$\varphi(n) = (p-1)(q-1)$

The encryption exponent e is chosen such that:

$gcd(e, \varphi(n)) = 1$, where $1 < e < \varphi(n)$

The decryption exponent d is calculated by solving the following equation:

$e.d = 1 \bmod \varphi(n)$ or $d = e^{-1} \bmod \varphi(n)$, where $0 \le d \le n$.

Thus, the public encryption key is $\{e, n\}$ and the private decryption key is $\{d, n\}$.

Thus, the RSA Algorithm can be divided into three steps:

(1) ***Key generation*:** in which the factors of the modulus n (the prime numbers p and q) are chosen and multiplied together to form n and

φ(n), an encryption exponent ***e*** is chosen, and the decryption exponent ***d*** is calculated using ***e*** and ***φ(n)***. The public encryption key is {***e, n***} and the private decryption key is {***d,n***}.

(2) ***Encryption*:** in which the message ***M*** is raised to the power ***e***, and then reduced modulo ***n***, so the ciphertext *C* can be calculated as $M^e \bmod n$.

(3) ***Decryption*:** in which the ciphertext ***C*** is raised to the power ***d***, and then reduced modulo ***n***. So the plaintext ***M*** is regenerated using the formula, $C^d \bmod n$

4.1. Security of RSA [8]

The security of RSA algorithm depends on the ability of the hacker to factorize numbers. New, faster and better methods for factoring numbers are constantly being devised. Obviously the longer a number is the harder is to factor, and so the better the security of RSA. As theory and computers improve, large and large keys will have to be used. The advantage in using extremely long keys is the computational overhead involved in encryption/decryption. This will only become a problem if a new factoring technique emerges that requires keys of such lengths to be used that necessary key length increases much faster than the increasing average speed of computers utilizing the RSA algorithm. RSA's future security relies solely on advances in factoring techniques.

5. METHODOLOGICAL STEPS

The block diagram of the overall system is shown in Figure-2. The individual steps are discussed in the following sub-sections.

5.1. Speech Acquisition

The recording of Bangla speech words was completed in a sound proof laboratory environment with the help of close-talking microphone, high quality sound card and sound recorder software. The 500 (five hundred) Bangla words originated from six speakers were recorded as *wav* file to make a sample database. The utterances were recorded at a sampling rate of 8.00 KHz and coded in 8 bits PCM **[9]**.

5.2. Pre-processing and Data Extraction

To extract wave data, we first discard 58 bytes (file header) from the beginning of the wave file and then read wave data as character. The data extraction process extracts require voiced data from the input speech signal, which may contain silence, unvoice and voice. This data are stored in a text file as integer data. This is usually done by detecting the proper start and end points of the speech events (voicing and unvoicing) and then separated into different pieces containing the audio signals on the basis of the detected start and end points **[10]**, as shown in Figure-3.

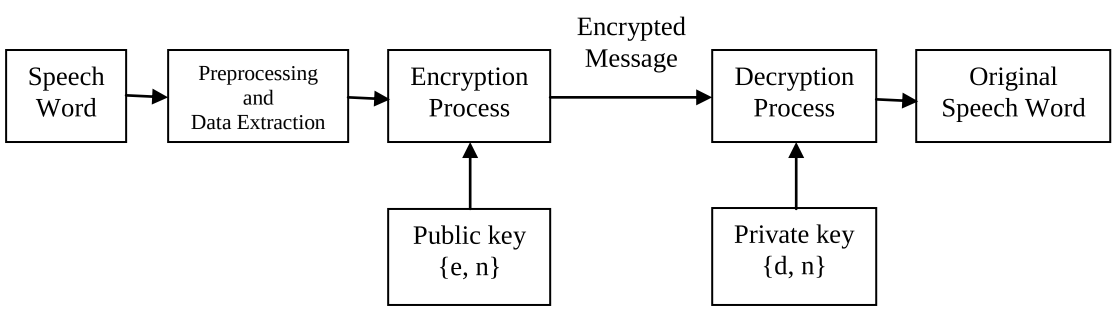


Figure-2. The developed System.

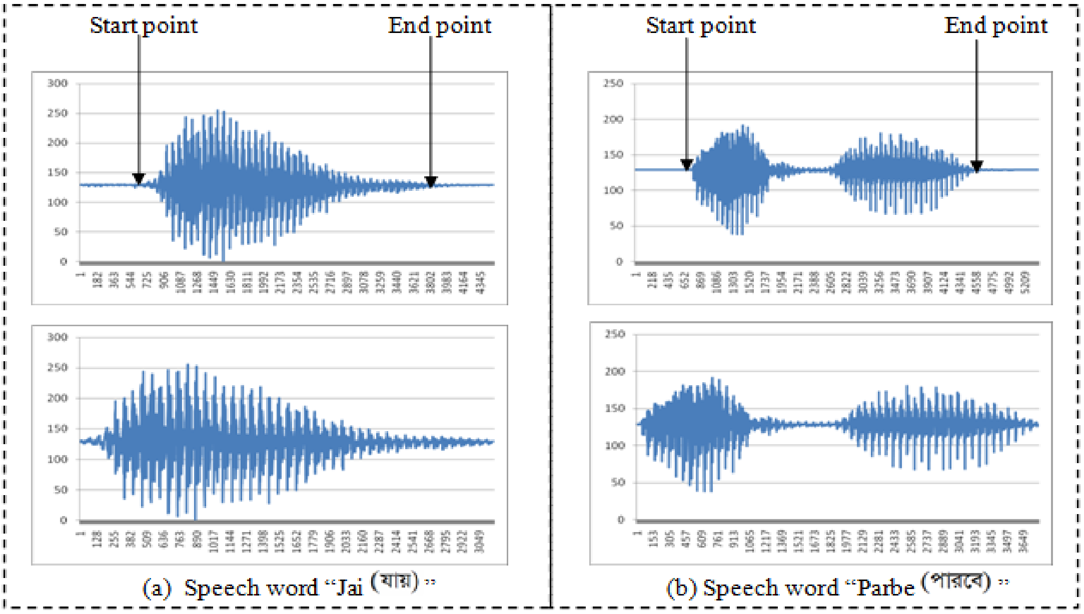


Figure-3. Detection of start and end points of Bangla speech words [11].

The RSA algorithm

Key Generation

(1) Choosing two very large prime numbers *p* and *q*.

(2) Compute their system modulus, $n = p*q$ and the ‘totient’ function $\varphi(n) = (p-1)(q-1)$. Note that the factors *p* and *q* remain secret and *n* is public.

(3) Select the encryption key *e* at random, so that $\gcd(e, \varphi(n)) = 1$, where $1 < e < \varphi(n)$.

(4) Solve the following equation to find the decryption key *d*: $e.d = 1 \bmod \varphi(n)$, where $0 \le d \le n$.

(5) Publish the public encryption key: PU = {*e*, *n*}, which is known to everyone.

(6) Keep secret or private the decryption key: PR = {*d*, *n*}, which is known only to the person who has to decrypt or sign the message.

Data Encryption

(1) Input the plaintext or message *M*, where $0 \le M \le n$.

(2) Obtain the public key of recipient, PU = {*e*, *n*}.

(3) Compute the cipher C, using the following equation: $C = M^e \bmod n$

Data Decryption

(1) Input the cipher text *C*.

(2) Use their private key, PR = {*d*, *n*}.

(3) Compute the message *M*, using the following equation: $M = C^d \bmod n$

Figure-4. The RSA Algorithm.

5.3. Implementation of RSA algorithm

As discuss earlier, the RSA Algorithm can be divided into three parts: key generation, encryption and decryption. The summary of the RSA algorithm is shown in the Figure-4.

5.3.1. Key Generation

The system architecture for key generation is shown in Figure-5. A random number generator generates 512-bit pseudo random numbers and stores them in the rand FIFO. Once the FIFO is full, the random number generator stops working. The primality tester takes a random number as input and tests if it is a prime. When new key pair is required, the down stream component pulls out two primes from the prime FIFO, and calculates ***n*** and ***φ(n)***. ***n*** is stored in a register. ***φ(n)*** is then sent to the Greatest Common Divider (GCD), where encryption key exponent ***e*** is selected such that ***gcd(φ(n), e) = 1***, and decryption key exponent ***d*** is obtained by inverting ***e modulo φ(n)***.

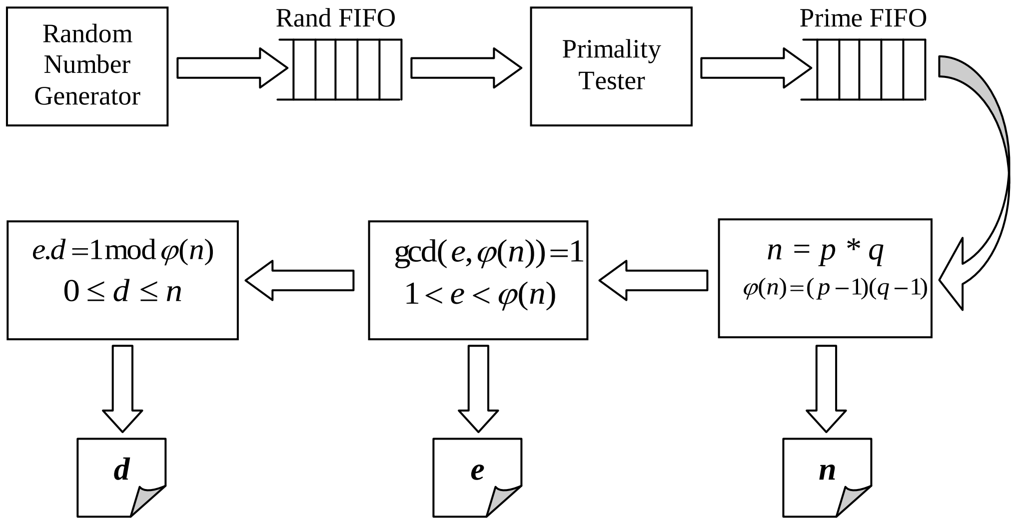
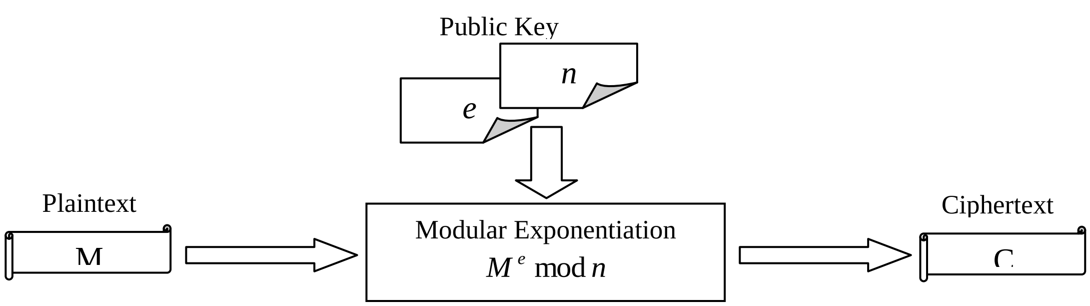
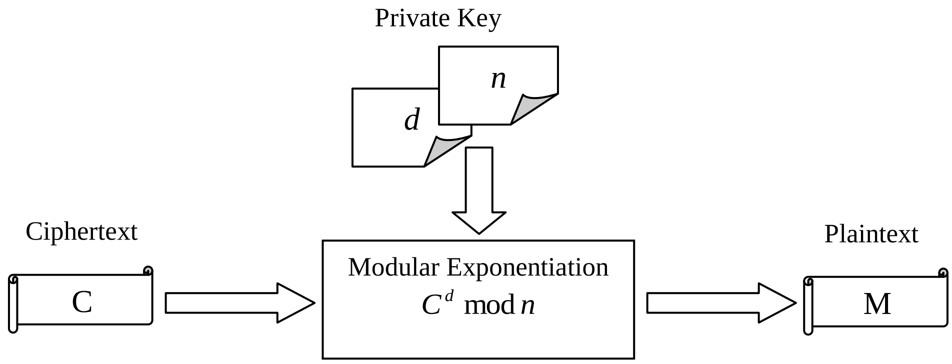


Figure-5. The system architecture for RSA key generation.



(a) RSA Encryption



(b) RSA Decryption

Figure-6. The RSA encryption/decryption structure.

5.3.2. RSA Encryption and Decryption

Once the public encryption key {*e*, *n*} and private decryption key {*d*, *n*} are generated, RSA encryption/decryption is simply a modular exponentiation operation. Figure-6 shows the RSA encryption/decryption structure that was implemented in our research work.

5.3.3. Modular Exponentiation

The modular exponentiation operation is simply an exponentiation operation where modular multiplication is intensively performed. We implemented the 512-bit and 1024-bit modular exponentiation components using LR binary method, where LR stands for the left-to-right scanning direction of the exponent. The pseudo code of the LR binary algorithm is given in the Figure-7.

In our research work, the RSA algorithm was implemented using C language and the implemented programs were compiled and run by using Turbo C++ IDE compiler.

```
Input: A, B, n
Output: E = A^B mod n

E <= 1;
for i = k-1 to 0
    if B_i = 1
        E <= A*E mod n;
    end if
    if i ≠ 0
        E <= E*E mod n;
    end if
end for
return E;
```

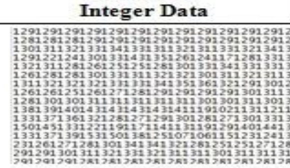
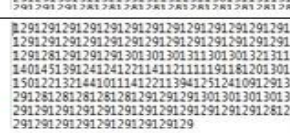
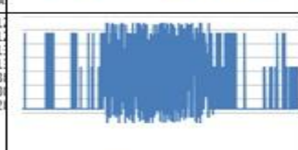
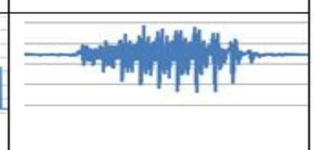
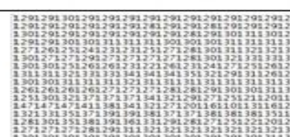
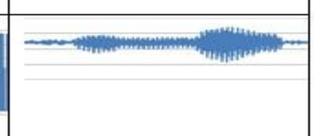
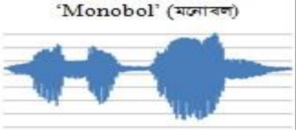
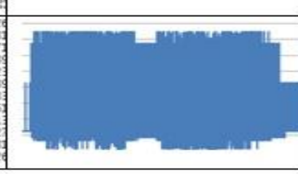
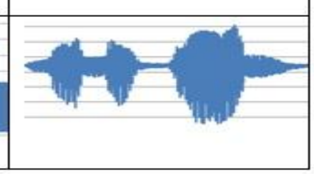
Figure-7. The LR binary algorithm for calculating modular exponentiation.

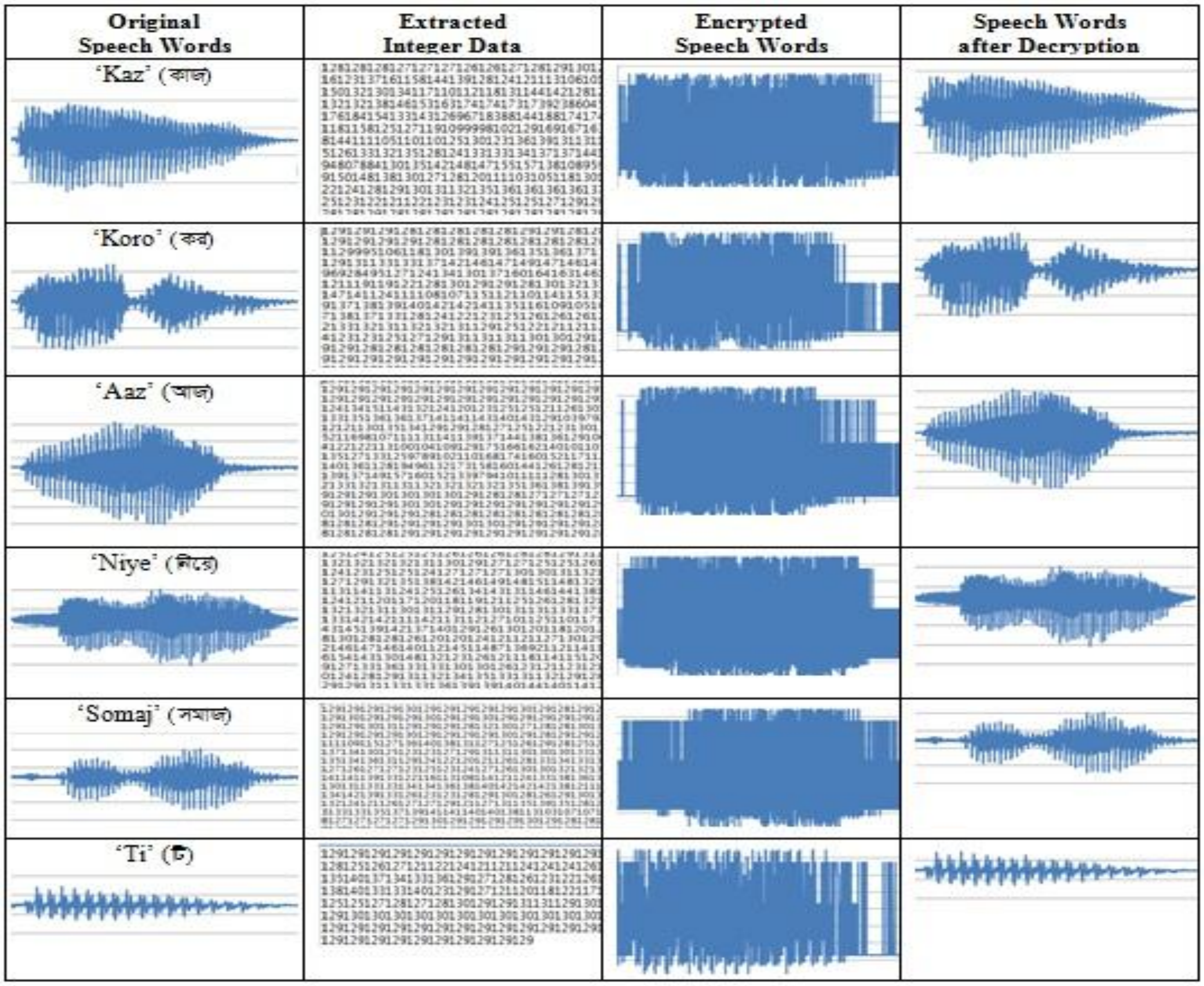
6. EXPERIMENTAL RESULTS AND DISCUSSION

The main objective of this work was to encrypt and decrypt speech data using public key RSA algorithm. In this research work, 500 (five hundred) speech words were recorded from different speakers and were saved as .wav file format. To encrypt these speech words, at first we extracted integer data from the speech .wav file and saved into another file as .txt format. The encryption program taken the txt file as input and performed encryption operation on the file to produce an unreadable encrypted message and saved another txt file.

On the other hand, the decryption program taken the encrypted message file as input and performed decryption operation on the file to produce the original text file. To generate the key pair, different values for ***p*** and ***q*** were tested. In this work, ***p =13*** and ***q = 29*** were used to calculate the encryption and decryption keys and these values produced better encryption result. We selected ***5*** as the value of the encryption key ***e***. Thus, the public and private keys were {***5, 377***} and {***269, 377***}, respectively. Table-1 shows the encrypted and decrypted results of some speech words.

Table-1. Encryption and Decryption Result of Some Speech Words.

Original Speech Words	Extracted Integer Data	Encrypted Speech Words	Speech Words after Decryption
'Aamra' (আমরা) 			
'Ek' (এক) 			
'Sundar' (সুন্দর) 			
'Monobol' (মনোবল) 			



7. CONCLUSION

RSA is a strong encryption algorithm that has stood a partial test of time. RSA implements a public-key cryptosystem that allows secure communications and digital signatures, and its security rests in part on the difficulty of factoring large numbers. In this paper, an efficient implementation of RSA algorithm is used to encrypt and decrypt the speech data. It must always be kept in mind that the integer representation of the message to be encrypted should lie within the range specified by the modulus (i.e., M lies in the range [0, n-1]), which poses a limitation on the maximum number of characters that can be encrypted at a single time. Our further work will eliminate all the limitations of this algorithm and will implement the RSA digital signature scheme in speech communication systems.

References

[1] W.Diffie and M. Hellman." New Directions in Cryptography". IEEE transactions on Information Theory. IT-22(1978).472-492.

[2] R.L. Rivest, A. Shamir, and L.M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", Communications of the ACM, volume 21, pages 120-126, February1978.

[3] C. Kaufman, R. Perlman, M. Speciner, "Network security," Prentice Hall 1995.

[4] William Stallings, "Cryptography and Network Security Principles and Practices", 4th edition, Pearson Education Inc, 2006.

[5]http://www.akadia.com/services/email_security.html

[6] Ronald L. Rivest, Adi Shamir, Len Adelman, "On Digital Signatures and Public Key Cryptosystems," MIT Laboratory for Computer Science Technical Memorandum 82 (April 1977).

[7] Patrick J. Flinn and James M. Jordan, Using the RSA Algorithm for Encryption and Digital Signatures: Can You Encrypt, Decrypt, Sign and Verify without Infringing the RSA Patent? Alston & Bird LLP, July 9, 1997.

[8] Amogh Mahapatra and Rajballav Dash, "Data Encryption and Decryption by
Using Hill Cipher Technique and Self Repetitive Matrix", A Thesis for the Degree of Bachelor of Technology in Electronics & Instrumentation Engineering, National Institute of Technology, Rourkela, 2007.

[9] S. Gokul, "Multimedia Magic", BPB Publications, B-14, Connaught Place, New Delhi-110001, ISBN 81-7029-972-1.

[10] Dr. Ramesh Chandra Debnath and Md. Farukuzzaman Khan, "Bangla Sentence Recognition Using End-Point Detection", Rajshahi University Studies: Part B, Journal of Science, Vol 32, 2004.

[11] Md. Mijanur Rahman, Fatema Khatun and Dr. Al-Amin Bhuiyan, "Development of Isolated Speech Recognition System for Bangla Words", International Journal of Applied Research on Information Technology and Computing (IJARITAC), Indianjournals.com, Vol-1, No-3, pp 272-278, Sep-Dec 2010.

[12] Behrouz A. Forouzen, "Cryptography and Network Security", Tata McGraw-Hill Publishing Company Ltd, New Delhi, ISBN-13: 987-0-07-066046-5, 2010.

Md. Mijanur Rahman is a faculty member of the Dept. of Computer Science and Engineering, Jatiya Kabi Kazi Nazrul Islam University, Trishal, Mymensingh, Bangladesh. Mr. Rahman obtained his B. Sc. (Hons) degree with first class first and M. Sc degree with first class first in Computer Science and Engineering from Islamic University, Kushtia, Bangladesh. Now he is working as PhD researcher under the supervision of Professor Dr. Al-Amin Bhuiyan in the department of Computer Science and Engineering at Jahangirnagar University, Savar, Dhaka, Bangladesh.

His teaching and research interest lies in the areas such as Database management System, Operating Systems, Web Programming, Compiler Design, Network Security, Artificial Intelligence, Digital Signal Processing, Digital Speech Processing, Pattern Recognition, etc. He has many research articles published in both national and international journals.

Tushar Kanti Saha is working as a lecturer in the Dept. of Computer Science and Engineering at Jatiya Kabi Kazi Nazrul Islam University, Trishal, Mymensingh, Bangladesh. He obtained his B. Sc. (Hons) degree with first class and M. Sc degree with first class in Computer Science and Engineering from Islamic University, Kushtia, Bangladesh. His teaching and research interest lies in the areas such as Internet and Web Programming, Database Programming, Software Engineering, System Analysis and Design, Digital Speech Processing, Pattern Recognition, etc.

Dr. Md. Al-Amin Bhuiyan is a professor of the Dept. of Computer Science and Engineering. Now, he is serving as the chairman of the department. Dr. Bhuiyan obtained his B. Sc. (Hons) degree with first class and M. Sc degree with first class in Applied Physics & Electronics from University of Dhaka, Bangladesh. He completed his PhD study in Information & Communication Engineering from Osaka City University, Japan.

His teaching and research interest lies in the areas such as Image Processing, Computer Vision, Computer Graphics, Pattern Recognition, Soft Computing, Artificial Intelligence, Robotics, etc. He has many articles published in both national and international journals.