International Journal of Computer

(IJC)

ISSN 2307-4523 (Print & Online)

http://ijcjournal.org/

Digital Envelope System Based on Optimized NTRU (Number Theory Research Unit) and RC6 Algorithm

Thwe Thwe Ngwe[a*], Su Wai Phyo[b]

[a,b]*Department of Information Technology, Mandalay Technological University*

The Republic of Union of Myanmar

[a]*Email: thwethwengwee@gmail.com*

[b]*Email: suwaiphyo@gmail.com*

Abstract

With the rapid development of technologies, more data are generated and transmitted in the medical, commercial, and military fields, which may include some sensitive information. So, security is essential to transfer the important information securely over the communication channels. To fulfill the information security requirements, many security systems were proposed in many research areas. On the other hand, many cryptographic algorithms are analyzed and optimized to evaluate better performance according to their requirements. This work proposes digital envelope system in order to meet the security requirement such as confidentiality. To create digital envelope system, the original message is encrypted by using Rivest Cipher-6(RC6) with the help of secret key. Then, that secret key is encrypted by using the ONTRU (Optimized Number Theory Research Unit) with the help of Receiver's public key. Moreover, this work also focuses on the optimization of NTRU to obtain better execution time for the digital envelope system. According to the analytical results, it is found that ONTRU is faster than NTRU. The basic idea behind this paper is to provide a good, faster digital envelope system.

Keywords: Digital Envelope System; NTRU Public Key Cryptosystem; MD-5 Hash function; RC6 Symmetric Key Cryptosystem; ONTRU.

1. Introduction

More and more information has been transmitted over the internet. Information security plays a major role for many data communication between people and activities. Cryptography is meaningful mechanism to solve problems of information security.

---

* Corresponding author

With the advent of information technology, ensuring network and data communication security has become a crucial issue. To meet the requirements of network and data communication security, the cryptography science plays a great role. The prime requirements for network and data communication security are privacy, authentication, integrity, maintenance and Non-Repudiation. All these are achieved through cryptographic techniques [1]. The digital envelope consists of an encrypted message using secret-key cryptography and an encrypted key using public key cryptography [2]. Therefore, in this research work, ONTRU asymmetric key algorithm and RC6 symmetric key algorithm is used in this digital envelope. Cryptographic techniques are broadly classified into symmetric key cryptographic techniques (DES, TDES, AES, RC6) and asymmetric key cryptographic techniques (RSA, ECC, HECC, NTRU). In this paper, the implementation details of a digital envelope for network and data communication security using RC6 and ONTRU cryptosystem are presented. By using the ONTRU, this system also fulfils to get faster execution speed than that of original NTRU.

The rest of the paper is organized as follows: section 2 presents related works. Background theories are presented in section 3. The proposed system design is described in section 4 and its implementation details are demonstrated in section 5. Analytical results and conclusion are described in section 6 and 7.

2. Related Work

Many research areas proposed digital envelope systems in order to overcome the security problems. Most of these systems focused on the combination of asymmetric key algorithm and symmetric key algorithm.

In the previous research [3], Mathew and Jacob presented a novel and fast technique for cryptographic applications. It is designed and developed using the symmetric key algorithm “MAJE4” and asymmetric key algorithm “RSA”. They developed a new hybrid system called MARS4 by combining the two encryption methods with an aim to get the advantages of both. Symmetric algorithm MAJE4 is used for encryption and decryption of files because it is much faster and occupies less memory than RSA. The RSA algorithm is used for key exchange and authentication. The performance evaluation of MARS4 is done in comparison with MAJE and RSA. This research showed that MAJE4 is much faster and provides confidentiality as well as message authentication.

Then, Ganesan, R. and Vivekanandan, K. [4] carried out a research to overcome security issues like privacy; authentication etc. digital envelope is a method to minimize these issues. This research developed a digital envelope in java by combining the methodologies of symmetric and asymmetric techniques. The symmetric algorithm used is AES and asymmetric algorithm is HECC (HECC over GF(p) of Genus 2). MD5 hash algorithm is used for the integrity of data. The algorithm is tested and the result illustrates that HECC is the best alternative asymmetric key technique rather than ECC and RSA in the digital envelope.

EhsanMalekian and Ali Zakerolhosseini[5] proposed OTRU, a high speed probabilistic multi-dimensional public key cryptosystem that encrypts eight data vectors in each encryption round. The underlying algebraic structure of the proposed scheme is the power-associative and alternative octonions algebra which can be defined over any Dedekind domain such as convolution polynomial ring.

Their proposed public key cryptosystem relied for its inherent security on the difficulty of the shortest vector problem (SVP) in a non-circular modular lattice. They described the algebraic structure used in their system. By reducing the dimension of the underlying convolution polynomial ring (N) and using parallelism techniques, they could increase the OTRU encryption/decryption speed to a level even higher than NTRU.

According to literature and concepts pointed out from the previous works, this work is intended to provide a powerful digital envelope system by using the new public key algorithm, ONTRU (Optimized Number Theory Research Unit) .

3. Background Theory

The digital envelope consists of a message encrypted using secret-key cryptography and an encrypted secret key. Digital envelopes usually use public-key cryptography to encrypt the secret key. It effectively provides confidentiality (privacy). In proposed scheme, RC6 symmetric key algorithm and ONTRU (Optimized Number Theory Research Unit) are used.

3.1. RC6 block Cipher

RC6 is an improvement to RC5 [6], designed to meet the requirements of increased security and better performance. Like RC5, which was proposed in 1995, RC6 makes use of data dependent rotations. One new feature of RC6 is the use of four working registers instead of two. While RC5 is a fast block cipher, extending it to act on 128-bit blocks using two 64-bit working registers. RC6 is modified its design to use four 32-bit registers rather than two 64-bit registers. This has the advantage that it can be done two rotations per round rather than the one found in a half-round of RC5.

Like RC5, RC6 is a fully parameterized family of encryption algorithms. A version of RC6 is also specified as RC6-w/r/b where the word size is w bits, encryption consists of a number of rounds r, and b denotes the encryption key length in bytes. Since the AES submission is targeted at w = 32 and r = 20, the parameter values specified as RC6-w/r are used as shorthand to refer to such versions. For all variants, RC6-w/r/b operates on four *w*-bit words using the following six basic operations:

a + b: Integer addition modulo 2w

a − b: Integer subtraction modulo 2w

a $\oplus$ b: Bitwise exclusive-OR of w-bit words

a × b: Integer multiplication modulo 2w

a <<< b: Rotate the w-bit word a to the left by the amount given by the least significant lg w bits of b

a >>> b: Rotate the w-bit word a to the right by the amount given by the least significant lg w bits of b (where lg w denotes the base-two logarithm of w)

RC6 exploits data-dependent operations such that 32-bit integer multiplication is efficiently implemented on most processors. Integer multiplication is a very effective diffusion, and is used in RC6 to compute rotation amounts so that these amounts are dependent on all of the bits of another register.

3.1.1. RC6 key schedule

The key expansion algorithm is used to expand the user-supplied key to fill an expanded array S, so S resembles an array of random binary words. The key schedule algorithm for RC6 differs from the RC5 version where more words are derived from the user-supplied key. The user must supply a key of b bytes, where $0 \leq b \leq 255$, and from which (2r+4) words are derived and stored in a round key array S. Zero bytes are appended to give the key length equal to a "non-zero integral number". The key bytes are then loaded in little-endian order into an array L of size c; when b = 0, c = 1 and L[0] = 0. The (2r+4) derived words are stored in array S for later decryption or encryption. Figure1 illustrates the key schedule used in RC6. P_w and Q_w are "magic constants", or word-sized binary constants where Odd(x) is the least odd integer greater than or equal to |x|. The base of natural logarithms and the golden ratio are respectively defined as e = 2.718281828459… and ø = 1.618033988749.

Input:
User-supplied b byte key preloaded into the c-word array L[0,…, c - 1]
Number r of rounds
$P_w = Odd((e - 2)2^w)$
$Q_w = Odd((ø - 1)2^w)$

Output:
w-bit round keys S[0,…, 2r + 3]

Procedure:
S[0] = P_w
for i = 1 to (2r + 3) do
S[i] = S[i _ 1] + Q_w
A = B = i = j = 0
v = 3 x max{c, 2r + 4}
for s = 1 to v d
{
A = S[i] = (S[i] + A + B) <<< 3
B = L[j] = (L[j] + A + B) <<< (A + B)
i = (i + 1) mod (2r + 4)
j = (j + 1) mod c
}

Figure 1: Key Schedule for RC6

3.1.2. RC6 encryption and decryption

RC6 encryption works with four w-bit registers A, B, C and D which contain the initial input plaintext [6]. The first byte of plaintext is placed in the least significant byte of A. The last byte of plaintext is placed into the most significant byte of D. The arrangement of (A, B, C, D) = (B, C, D, A) is like that of the paralleled assignment of values (bytes) on the right to the registers on the left. The RC6 encryption algorithm is shown in Figure 2.

RC6 decryption works with four w-bit registers A, B, C, D which contain the initial output ciphertext at the end of encryption. The first byte of ciphertext is placed into the least significant byte of A. The last byte of ciphertext decryption algorithm is illustrated in Figure 3.

```
Input     : Plaintext stored in four w-bit input
registers A, B, C, D
            Number of rounds, r
            w-bit round keys S[0, 1, . . . , 2r + 3]
Output    : Ciphertext stored in A, B, C, D
Procedure: B = B + S[0]
           D = D + S[1]
 for i = 1 to r do
                  {  t = (B × (2B + 1)) <<< 1g w
                     u = (D × (2D + 1)) <<< 1g w
                     A = ((A ⊕ t) <<< u) + S[2i]
                     C = ((C ⊕ u) <<< t) + S[2i + 1]
                     (A, B, C, D) = (B, C, D, A) }
               A = A + S[2r + 2]
               C = C + S[2r + 3]
```

Figure 2: RC6 Encryption Algorithm

```
Input     : Ciphertext stored in four w-bit input
            registers A, B, C, D
            Number of rounds, r
            w-bit round keys S[0, 1, . . . , 2r + 3]
Output    : Plaintext stored in A, B, C, D
Procedure: C = C − S[2r + 3]
           A = A − S[2r + 2]
            for i = r down to 1 do
                 { (A, B, C, D) = (D, A, B, C)
                    u = (D × (2D + 1)) <<< 1g w
                    t = (B × (2B + 1)) <<< 1g w
                    C = ((C − S[2i + 1] >>> t) ⊕ u
                    A = ((A − S[2i]) >>> u) ⊕ t  }
              D = D − S[1]
              B = B − S[0]
```

Figure 3. RC6 Decryption Algorithm

3.2. Number theory research unit(NTRU)

"NTRU" [7] is the acronym for Number Theory Research Unit. NTRU was developed due to the interest in computationally fast and efficient methods to implement public-key cryptography. The encryption process is accomplished through polynomial ring arithmetic modulo two relatively prime values (integer or polynomial). The decryption process reverses the encryption process using probabilistic methods with a chance of failure, although this is minimized or eliminated through selection of the system parameters. The choice of system parameters, as with most cryptosystems, is the main method for determining the projected level of security.

NTRU operations are performed on polynomials of degree N-1 with integer coefficients in a convolution polynomial ring. NTRU is based on polynomial additions and multiplications in the ring of truncated polynomials.

The notation used to represent that operations in the NTRU occur in a convolution polynomial ring is denoted as $Z[X]/(X^N-1)$, where Z[X] represents a polynomial with integer coefficients and the division (X^N-1) represents symbolic reduction of the polynomial [7].

For NTRU cryptosystem, there are some notations. The following are the parameters for an implementation of NTRU.

N - the polynomials in the ring R have degree N-1

q - the large modulus to which each coefficient is reduced

f - a polynomial that is the private key

p- the small modulus to which each coefficient is reduced

g - a polynomial that is used to generate the public key h from f

h - the public key, also a polynomial

r - the random polynomial (secret but discarded after initial use)

3.2.1. NTRU key generation

The key generation scheme is used to generate the private and public key pair. The process begins by choosing two small polynomials f and g, where small is defined as having coefficients much smaller than the large modulo p and modulo q.

The user must compute the inverse of f modulo q and the inverse of f modulo p such that $f * f_q = 1 \pmod q$ and $f * f_p = 1 \pmod p$. The inverse of f is calculated both modulo p and modulo q, generating $f_p = f^{-1} \pmod p$ and $f_q = f^{-1} \pmod q$. The values of f and f_p are retained as the private key pair and the public key h is calculated using p, f_q and g [8]. The public key is as follows:

$$h = pf_q * g \pmod q \quad (1)$$

3.2.2. NTRU encryption

The encryption process starts by generating a polynomial message m whose coefficients lie in an interval of length q, which is normally cantered around zero. A small random blinding polynomial, r, is then generated and used to obscure the message [8]. The final encryption uses m, r and the public key h to generate e, the encrypted message that is as follows:

$$e = r * h + m \pmod q \quad (2)$$

3.2.3. NTRU decryption

The decryption process first uses the private key f to calculate:

$$a = f * e \pmod q \tag{3}$$

The coefficients of a must be chosen in the proper interval of length q to ensure the highest probability that the decryption process will be successful. Once the coefficients of a are chosen on the proper interval, a is reduced modulo p and the second private key is used to compute:

$$b = a \pmod p \tag{4}$$

$$c = f_p * b \pmod p \tag{5}$$

If decryption has successfully completed, then the polynomial c will be equal to the original message [8].

3.3. Optimized number theory research unit(ONTRU)

The proposed ONTRU algorithm is based on NTRU. It is an enhancement of NTRU public key cryptosystem based on matrix inverse, additions and multiplications. Unlike NTRU, it uses matrix formulation instead of using polynomial ring. Like NTRU, it has the parameters and three components such as key generation, encryption and decryption. The parameters of ONTRU are as follow:

N - the numbers of entries in the matrices.

q - the large modulus to which each entries in the matrices is reduced.

f - a matrix that is the private key.

p- the small modulus to which each entries in the matrices is reduced.

g - a matrix that is used to generate the public key h from f.

h - a matrix that is the public key.

r - the random matrix.

3.3.1. ONTRU key generation

To generate the private and public key pair, two small matrices f and g, consisting of {-1, 0, 1} must be chosen. To enhance the performance of ONTRU, f_1 is chosen based on f and p. The f_1 is as follows:

$$f_1 = I + pf \tag{6}$$

Where f is a small matrix and I is identity matrix. The user must compute matrix $(f_1)_q$ such that $f_1 *(f_1)_q = I \pmod q$ by using the Gram-Schmidt Process. During the key generation, we threw f_1 away if the inverse didn't exit. The value of matrix f_1 is used as private key. By using matrices p, $(f_1)_q$ and g, the public key H is calculated.

The public key H is as follows:

$$H = p*(f_1)_q * g \pmod q \quad (7)$$

3.3.2. ONTRU encryption

To compute the encrypted message, the user must use a message m matrix and public key H. The user must select a random small matrix, r. The cipher text, E, must be computed as follows:

$$E = r* H + m \pmod q \quad (8)$$

3.3.3. ONTRU decryption

To recover the original message, the decryption procedure requires two steps. These are as follows:

$$a= f_1 * E \pmod q \quad (9)$$

$$b = a \pmod p \quad (10)$$

The value of b is the decrypted ciphertext which should be equal to m. The difference between NTRU and optimized NTRU is illustrated in Table 1.

Table 1: Comparison between NTRU and ONTRU

Components	NTRU(by using truncated polynomial ring)	ONTRU(by using matrix formulation)
Key Generation	1. Private key is (f, f_p). 2. $f_p = f^{-1} \pmod p$ 3. Public key is $h = pf_q * g \pmod q$	1. Private key is f_1 2. $f_1 = 1 + pf$ 3. Public key is $H=p*(f_1)_q*g \pmod q$
Encryption	1. $e = r * h + m \pmod q$	1. $E = r * H + m \pmod q$
Decryption	1. $a = f * e \pmod q$ 2. $b = a \pmod p$ 3. $c = f_p * b \pmod p$	1. $a = f_1 *E \pmod q$ 2. $b = a \pmod p$

4. Comparison Result of NTRU and ONTRU

For performance consideration, running times of NTRU and ONTRU can be analysed by implementing in different processors such as Pentium, core i3 and core i5.

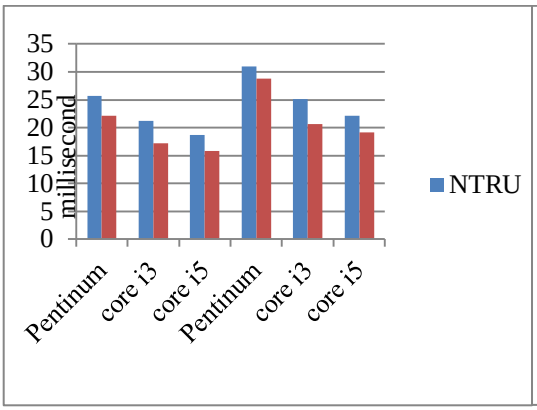


Figure 4: Running times for key generation

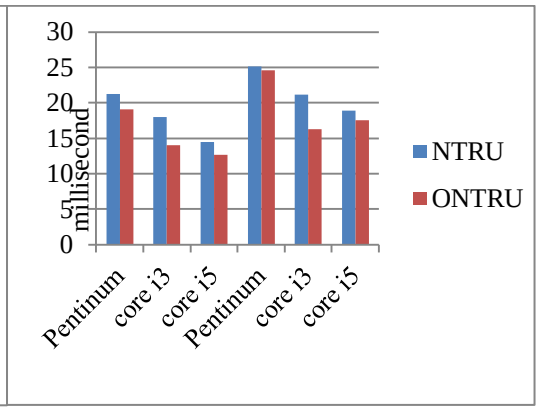


Figure 5: Running times for encryption

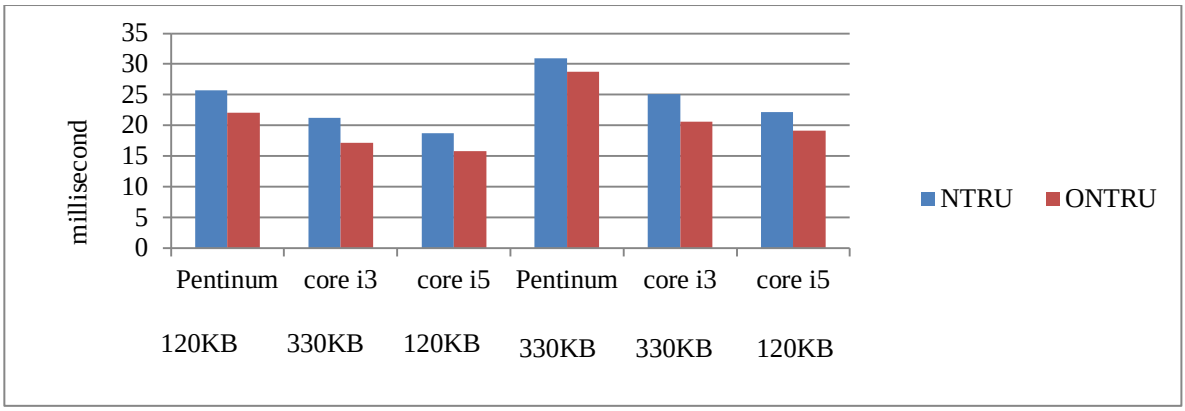


Figure 6: Running times for decryption

Figure 4, 5 and 6 show running times for key generation, encryption and decryption of NTRU and ONTRU that is tested on 120KB and 330KB by implementing in different processors such as Pentium, core i3 and core i5.

5. Proposed System Design

The proposed information security system is based on the digital envelope system that combines the symmetric key algorithm and asymmetric key algorithm in order to provide security requirement such as confidentiality. The proposed system is organized with two portions: proposed system design for sender site and receiver site as illustrated in Figure 7 and Figure 8.

Firstly, the sender has to create digital envelope (message encryption using RC6 and secret key encryption using ONTRU) as shown in Figure 7. The original message (plain text) that is to be transmitted is encrypted using the RC6 algorithm.

The RC6 key which is used to encrypt the plaintext is encrypted with the help of Receiver's public key by using ONTRU algorithm in order to be faster execution time. And then, the encrypted message and encrypted key are sent to the receiver over the network.

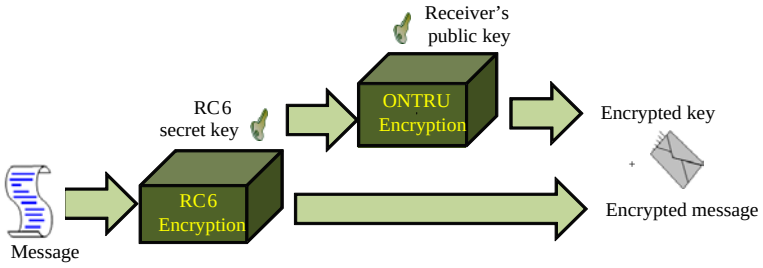


Figure 7: Proposed System Design for Sender Site

In receiver site, the encrypted key must be decrypted with receiver's private key by using ONTRU algorithm in order to obtain the RC6 key. And then, this key is used to decrypt the encrypted message to obtain the plaintext as illustrated in Figure 8.

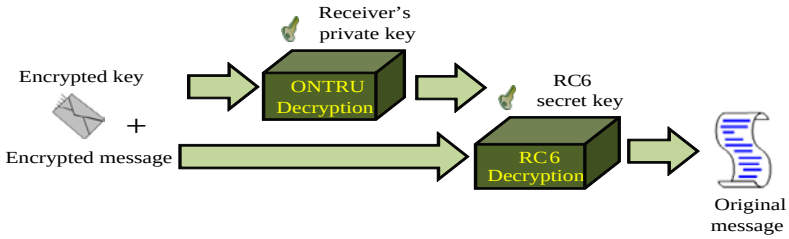
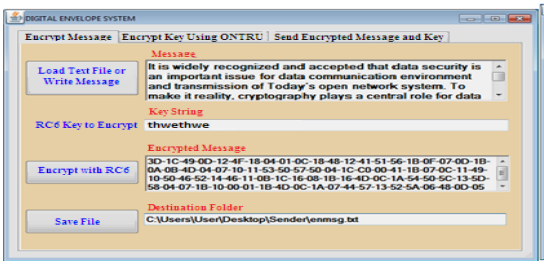


Figure 8: Proposed System Design for Receiver Site

6. System Implementation

The proposed system implementation is represented as a series of interface. It is implemented by using Java programming language. Figure 9 shows that the message files to be encrypted must be selected to create the digital envelope system. Then, the sender has to type any 8 characters in the "RC6 Key" text box. Then, the user has to click "Encrypt with RC6" button to encrypt the message with the RC6 secret key. And then, the user has to click the "Save File" button to save the encrypted message file.



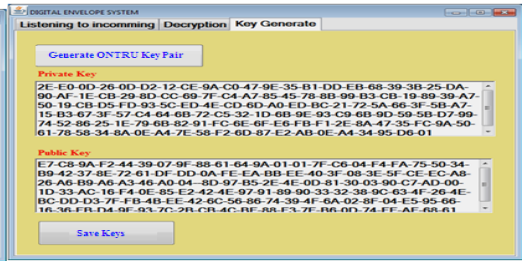


Figure 9: Encryption processes with RC6 **Figure 10**: Key pair of ONTRU

Before the sending and receiving processes of sender and receiver, key generation process is firstly performed to get the key pairs for the sender and receiver. The key generation process is shown in Figure 10.

Figure 11: Sending encrypted files

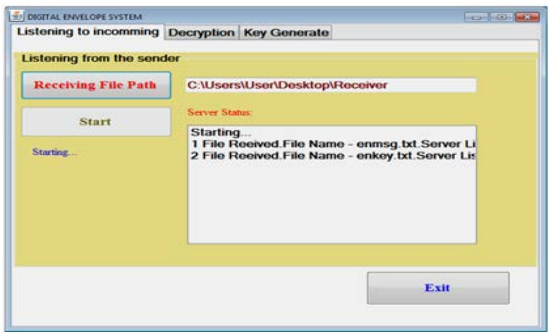


Figure 12: Receiving encrypted files

For the sending process as shown in Figure 11, the encrypted message and encrypted key files are loaded and sent to the receiver using receiver's IP address over the network.

At the receiver side, the receiver has to start server for connection to receive the encrypted files from the sender as shown in Figure 12. For the decryption process, the receiver has to load the encrypted key and receiver's private key file. After clicking on "Decrypt Key" button, the receiver can get the decrypted RC6 key.

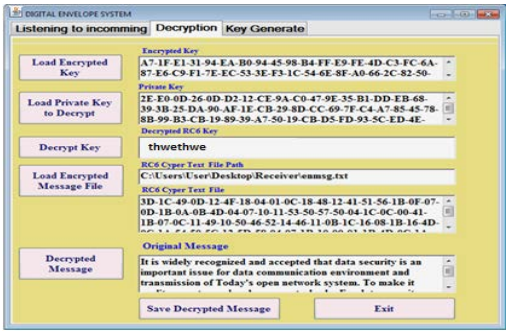


Figure 13: Decryption processes

For the message decryption, the receiver must load the encrypted message. If the user clicks "Decrypted Message" button, the decryption process reads the cipher text from the text box and converts it into plaintext message. Then, the original plaintext message is displayed in the text box as shown in Figure 13.

7. Conclusion

This proposed system focused on the digital envelope system which is related with the concepts of RC6 and optimized NTRU algorithm. The usage of RC6 symmetric algorithm and optimized NTRU public key algorithm gives the security requirement such as confidentiality. By using the optimized NTRU, this system also fulfils to get faster execution speed than that of original NTRU. This fact is intended not to take longer execution time when the larger file sizes are used for encryption/decryption process.

To perform the system operation, the user needs to insert 128-bit key (8 characters) exactly for RC6 symmetric key algorithm. This system only provides data (text) security. As further extension, various files such as image, audio, video files can be added for information security in the hybrid system and new methods can be developed based on the other public key algorithms in order to obtain more efficient system. This digital envelope system can be effectively applied in many information security areas such as Ecommerce, E-mail security system, authentication based security systems and online banking system and so on.

References

[1] Hwang M S, and Liu C Y, "Authenticated encryption schemes: Current status and key issues," International Journal of Network Security, vol. 1, no. 2, pp. 61-73, 2005.

[2] Jeffrey Hoffstein, Jill Pipher, and Joseph H.Silverman. NTRU: A ring-based public key cryptosystem. In *Algorithmic Number Theory*, pages 267- 288 (1998).

[3] Mathew, S Jacob, K.P, "A Novel Fast Hybrid Cryptographic System: MARS4," India Conference, 2006 Annual IEEE, pp.1-5, 15-17 Sept. 2006.

[4] Ganesan, R. and Vivekanandan, K, "A Novel Hybrid Security Model for E-Commerce Channel," Advances in Recent Technologies in Communication and Computing, 2009 ARTCom '09, pp.293-296, 27-28-Oct. 2009 .

[5] E. Malekian and A. Zakerolhosseini, "A non-associative lattice-based public key cryptosystem," Submitted to Security Communication Network, 2010.

[6] Rivest, R.L., Robshaw, M.J.B., Sidney, R., & Yin, Y.L (1998a)."The RC6BlockCipher."URL:ftp://ftp.rsasecurity.com/pub/rsalabs/rc6/rc6v11.pd

[7] J. Hoffstein, J. Pipher and J. H. Silverman, *NTRU- A Ring Based Public Key Cryptosystem*, Lecture Notes in Computer Science, Springer-Verlag, Berlin, pp. 267-288, 1998.

[8] K. Thiagarajan, *NTRU- A Public Key Ring Based Algorithm*, A Thesis for Master of Science, University of Texas, Dallas, 2003.