HYBRID CRYPTOSYSTEM FOR DATA SECURITY

[1]THWE THWE NGWE, [2]SU WAI PHYO

[1,2]Department of Information Technology, Mandalay Technological University The Republic of the Union of Myanmar
E-mail: [1]thwethwengwee@gmail.com, [2]swp2006@gmail.com

Abstract- With the fast progression of information exchange in electronic way, security is becoming more important in information transmission as well as in storage. To protect the confidential data from unauthorized access, many cryptographic algorithms were used in many research areas. Then, analyzing and optimizing many cryptographic algorithms was also performed to obtain better performance in their systems. Nowadays, hybrid cryptosystems are also developed to fulfill the security requirements. This work also proposes hybrid cryptosystem for data security in order to meet the security requirements: confidentiality, authentication, data integrity and non-repudiation. To create the hybrid cryptosystem, the secret message is encrypted by using Rivest Cipher-6(RC6) and calculated hash value by using Message Digest-5(MD-5). Then, the combination of encrypted message and hash value is encrypted by using optimized NTRU (Number Theory Research Unit) with the help of sender's private key. Moreover, this work also focuses on the optimization of NTRU (Number Theory Research Unit) to obtain better execution speed for the hybrid cryptosystem.

Index Terms- Hybrid cryptographic system, NTRU public key cryptosystem, MD-5 hash function, RC6 symmetric key cryptosystem, optimized NTRU.

I. INTRODUCTION

Today, computer communication is widely used in various fields. Many people send data, information, files, etc, quickly via the communication link. At this point, security plays an important role. Two ways for information security are cryptography and steganography. In cryptography, hybrid cryptosystems are very popular in today's data communication environments. Hybrid cryptography [1] is a technique using multiple ciphers of different types together, each to its best advantage. A hybrid cryptosystem can be constructed using any two separate cryptosystems:

- A key has encapsulation scheme which is a public key or any other type of cryptosystem.
- A data has encapsulation scheme which is a symmetric key cryptosystem.

This system focuses on hybrid cryptographic system by using the combination of symmetric algorithm, public key algorithm and hash function in order to obtain the security requirements: data confidentiality, authentication, data integrity and non-repudiation. In this system, RC6 symmetric algorithm, optimized NTRU public key algorithm and MD5 hash function are effectively organized to meet the security requirements. The rest of the paper is organized as follows: section II presents related works. Background theories are presented in section III. The design of proposed system, implementation and analytical results are described in section IV, V and VI respectively. Finally, conclusion is described in section VII

II. RELATED WORK

Many information security systems have been developed in many application areas such as military, offices and Ecommerce and so on. Ravindra and Parvinder [2] analyzed the principle of the hybrid cryptography based on the combination of symmetric and message digesting. Moreover, the security and performance of their proposed concept would also estimate. In their concept, they designed a new symmetric encryption algorithm and combine with SHA-1 message digesting function to provide hybrid nature. Their proposed system would try to improve existing problem. In their system, symmetric key would use series of logical functions like XOR, Circular Shift (Right, Left), Feastel function and these operations are time efficient operation as compared to mathematical operation for providing confidentiality and authentication. For integrity, their system used SHA-1 algorithm. Due to this reason, they concluded that their system was efficient than existing system [2].

S. Subasree and N. K. Sakthivel [3] presented that the new security protocol has been designed for better security with integrity using a combination of both symmetric and asymmetric cryptographic techniques. To improve the strength of these security algorithms, a new security protocol for online transaction could be designed by using hybrid information security system. This protocol provided three cryptographic primitives such as integrity, confidentiality and authentication. These three primitives can be achieved with the help of Elliptic Curve Cryptography, Dual -RSA algorithm and Message Digest MD5. In their research, they pointed out that Dual RSA is better than RSA algorithm.

Mahmoud Shaar and his fellows [4] also presented an encryption algorithm that could be used for hardware-implemented applications to secure data communications. Their encryption algorithm was based on hiding a number of bits from plain text message into a random vector of bits. The locations of the hidden bits were determined by a key known to the sender and receiver. They called this algorithm

hybrid hiding encryption algorithm (HHEA). This name demonstrated the two basic operations of this algorithm. The name "Hybrid" was used to show that the algorithm had built-in features that were inherited from data hiding techniques or "Steganography." According to the literature and concepts pointed out from the previous works, this work is intended to provide a hybrid information security system by using MD-5 hash function, RC6 and optimized NTRU algorithms.

III. BACKGROUND THEORY

In this section, this system explains Message Digest-5 (MD5), Rivest Cipher-6(RC6), NTRU (Number Theory Research Unit) and the optimized NTRU (Optimized Number Theory Research Unit) cryptosystem in detail.

A. MD-5 Algorithm

MD5 is a widely used cryptographic hash function with a 218-bit hash value. A MD5 hash is typically expressed as a 32 digital hexadecimal number. MD5 algorithm takes input message of arbitrary length and generates 128-bit long output hash. The basic operation of MD5 is shown in Fig. 1. MD5 hash algorithm consists of 5 steps that are:

- Append Padding Bits
- Append Length
- Initialize MD Buffer
- Process Message in 16-Word Blocks
- Output

1) Append Padding Bit: The input message is broken up into chunks of 512-bit blocks. The message is "padded" so that its length is congruent to 448, modulo 512. That is, the message is extended so that it is just64 bit shy of being a multiple 512 bits long. Padding is always performed, even if the length of the message is already congruent to 448, modulo 512. Padding is performed as follows: a single "1" bit is appended to the message, and then "0" bits are appended so that the length in bits of the padded message becomes congruent to 448, modulo 512. In all, at least one bit and at most 512 bits are appended [5].

2) Append Length: The remaining bits are filled up with a 64-bit representing the length of the original message or file, in bits. In the unlikely event that the original length is greater than 2 power 64, then only the low-order 64 bits of original length are used. At this point the resulting message has a length that is an exact multiple of 512 bits. This message has a length that is an exact multiple of 15 (32-bit) words [5].

3) Initialize MD Buffer: A four-word buffer (A, B, C and D) is used to compute the message digest. Here each of A, B, C and D is a 32-bit register. These registers are initialized to the following values in hexadecimal, low-order bytes first.

Word A: 01 23 45 67
Word B: 89 AB CD EF
Word C: FE DC BA 98
Word D: 76 54 32 10

4) Process Message in 16-Word Blocks: MD5 uses four auxiliary functions (F, G, H, I) that each take as input three 32-bit words and produce as output one 32-bit word. The four auxiliary functions are as follows:

$$F(X, Y, Z) = (X \wedge Y) \vee (\overline{X} \wedge Z)$$
$$G(X, Y, Z) = (X \wedge Z) \vee (Y \wedge \overline{Z})$$
$$H(X, Y, Z) = X \oplus Y \oplus Z$$
$$I(X, Y, Z) = Y \oplus (X \vee \overline{Z})$$

Where, □, □,- and □represents logical OR, AND, NOT and XOR. The contents of the four buffers (A, B, C and D) are now mixed with words of the input, using the four auxiliary functions (F, G, H and I). There are four rounds, each involves 16 basic operations.

5) Output: After performing all rounds, the buffers A, B, C and D contain the MD5 digest of the original output.

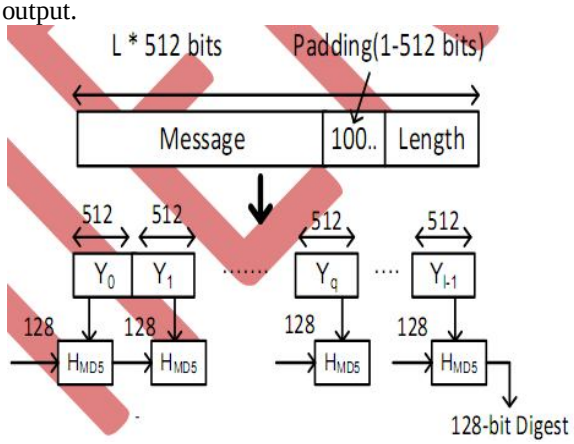


Fig. 1: Basic MD5 operation

B. RC6 Block Cipher

RC6 is a symmetric key block cipher derived from RC5. RC6 proper has a block size of 128 bits and supports key sizes of 128,192 and 256bits. The key size of 128 bits is used in the proposed hybrid cryptosystem for data encryption process.

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. RC6 was submitted to NIST for consideration as the new Advanced Encryption Standard (AES). Since the AES submission is targeted at w = 32 and r = 20, the parameter values of 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, a − b, a ⊕ b, a × b, a <<< b and a >>> b.

RC6 exploits data-dependent operations such that 32-bit integer multiplication is efficiently implemented on most processors. RC6-w/r/b operates on units of four w-bit words using addition (+), subtraction (-),

multiplication (×), exclusive-OR (⊕) and rotations (<<<, >>>) operations. The user supplies a key of b bytes. From this key, 2r + 4 words
(w bits each) are derived and stored in the array S [0, 2r+3]. This array is used in both encryption and decryption [6].

1) RC6 Encryption: RC6 encryption works with four w-bit registers A, B, C, D which contain the initial input plaintext [6]. The RC6 encryption algorithm is shown in Fig. 2.

```
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]
```

Fig. 2: RC6 Encryption Algorithm

2) RC6 Decryption: RC6 decryption works with four w-bit registers A, B, C, D which contain the initial output ciphertext at the end of encryption [6]. RC6 decryption algorithm is illustrated in Fig. 3.

```
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]
```

Fig. 3: RC6 Decryption Algorithm

C. 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.

1) Convolution Polynomial Rings: 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]/ (XN-1), where Z[X] represents a polynomial with integer coefficients and the division (XN-1) represents symbolic reduction of the polynomial [7].

2) 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 inverse of f is calculated both modulo p and modulo q, generating fp * f = 1 (mod p) and fq * f = 1 (mod q). The values of f and fp are retained as the private key pair and the public key h is calculated using p, fq and g [8]. Therefore, the public key is h = pfq * g (mod q).

3) NTRU Encryption: The encryption process starts by generating a polynomial message m whose coefficients lie in an interval of length p, which is normally cantered around zero. A small random blinding polynomial, r, is then generated and used to obscure the message [7]. The final encryption uses m, r and the public key h to generate the encrypted message, e = r * h + m (mod q).

4) NTRU Decryption: The decryption process first uses the private key f to calculate a = f * e (mod q). 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 (mod p) and c = fp * b (mod p). If decryption has successfully completed, then the polynomial c will be equal to m, the original message.

D. Optimized Number Theory Research Unit (OptimizedNTRU)
The proposed optimized number theory research unit (optimized NTRU) algorithm is based on NTRU. The

optimized NTRU is based on polynomial additions and multiplications in the ring of truncated polynomials Z[x]/(xN-1). In the optimized NTRU parameters, N is the polynomials in the ring R have degree N-1. The 'q' is the large modulus to which each coefficient is reduced. The 'p' is the small modulus to which each coefficient is reduced. The values of f and f1 are retained as the private key pair and the public key h is calculated using p, fq and g. The 'g' is a polynomial that is used to generate the public key h from f. The 'h' is the public key, also a polynomial and the 'r' is the random polynomial.

1) Optimized NTRU Key Generation: To generate the private and public key pair, the user must compute the inverse of f modulo q such that f * fq = 1 (modulo q). Then, the values of f and f1 are used as private key pair. By using p, fq and g, the public key h is calculated. To enhance the performance of optimized NTRU, the proposed optimized NTRU calculates the f1 based on polynomial f and p. The f1 is as follows:

f1 = 1 + pf (1)
The public key h is as follows: h=pfq*g (modq) (2)

2) Optimized NTRU Encryption: To compute the encrypted message m, the message must be converted into a polynomial m whose coefficients are chosen (modulo p).The user must select a random small polynomial, r Є R. The cipher text, e, must be computed as follows: e = r * h + m (mod q) (3)

3) Optimized NTRU Decryption: To recover the original message, the decryption procedure requires three steps. These are as follows:
a = f1* e (mod q) (4)
b = a (mod p) (5)
c = (1 + pf) * b (mod p) (6)
The polynomial c will be the original message m.

IV. DESIGN OF THE PROPOSED SYSTEM

In the proposed hybrid cryptosystem, MD5 hash algorithm, RC6 block cipher algorithm and optimized NTRU algorithm are used to secure information during information transmission between sender and receiver. The sender portion of the hybrid cryptosystem is shown in Fig. 4.

At the sender portion, message (M) is firstly encrypted by RC6 encryption algorithm with a secrete key and hashed with MD5 hash function. And then, ciphertext (C1) and hash message H(M) will be got by encryption with RC6 algorithm and by hashing with MD5 hash function respectively. The resulting ciphertext (C1) and hash message H(M) are concatenated. Finally, the concatenated ciphertext is encrypted using optimized NTRU encryption algorithm with the help of sender's private key and then the output (C2) is sent to the intended receiver.

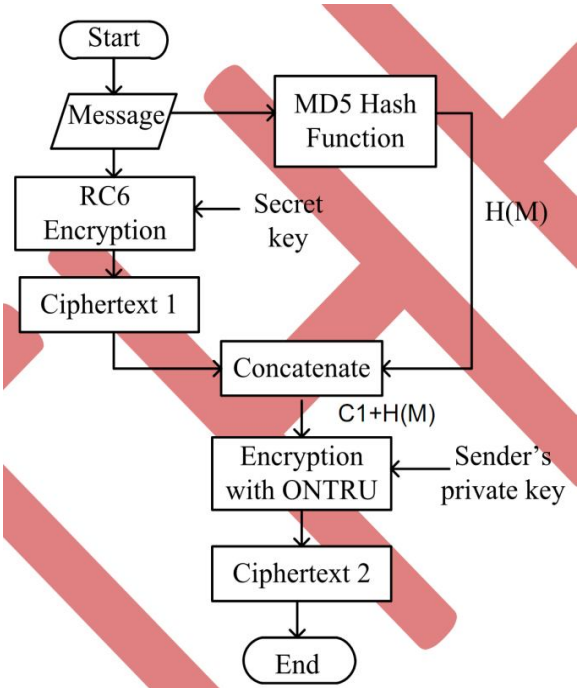


Fig. 4 : Sender Portion of Hybrid Information Security System

At first, the receiver decrypts the ciphertext (C2) using optimized NTRU decryption algorithm with the help of sender's public key and produces the concatenated ciphertext. Secondly, the concatenated cipherext is split into the encrypted message (C1) and hash message H(M). The encrypted message (C1) is decrypted by RC6 decryption algorithm with secret key and the decrypted message is hashed with MD5 hash function to get hash message H(M)'. The resulting two hash message H(M) and H(M)' will be compared to check the data integrity. If it is equal, it is ensured that it meets the data integrity. The receiver portion of the hybrid information security system is also illustrated in Fig. 5.

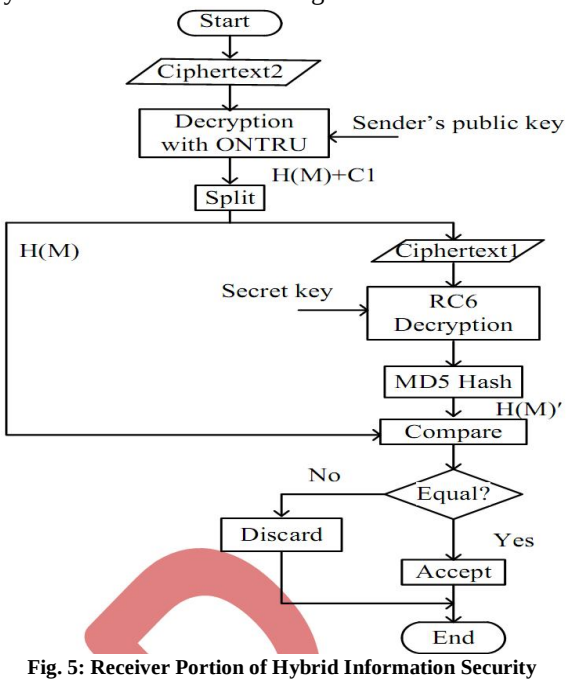


Fig. 5: Receiver Portion of Hybrid Information Security System

V. IMPLEMENTATION OF HYBRID CRYPTOSYSTEM

This system is implemented as a series of interfaces to show the hybrid cryptosystem.The main interface is illustrated in Fig. 6.
At first, the user must load the original message. Before performing any process of the proposed system, key generation process is firstly performed to get the key pairs for the sender and receiver. Key generation process is shown in Fig. 7.

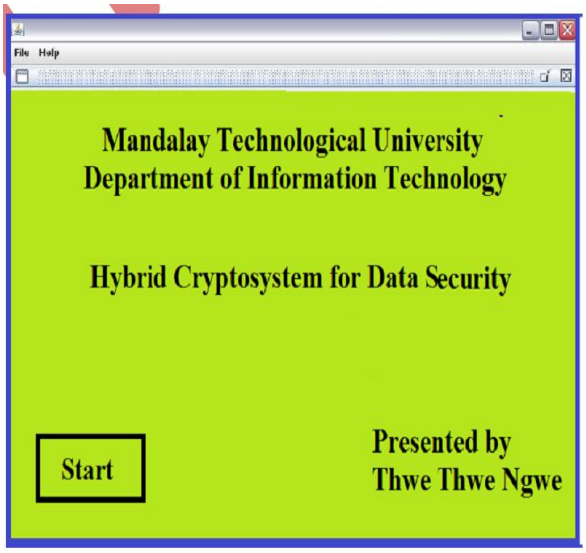


Fig. 6: Main form

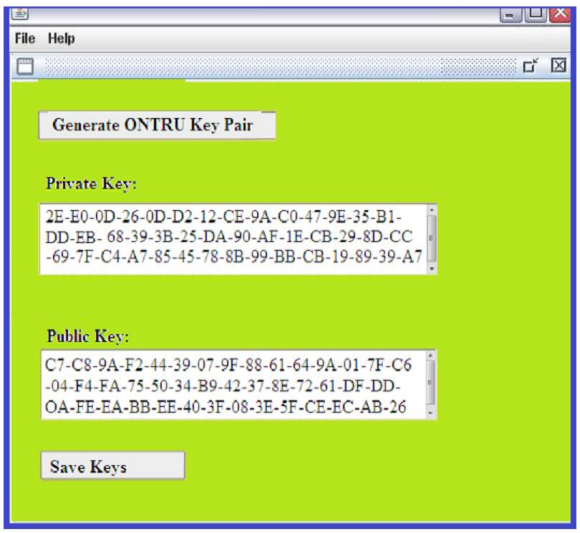


Fig. 7: Key Generation Process

If the user wants to send the message, this message is passed through two encryption processes. The first one is RC6 encryption process with secret key and the next is the encryption process of optimized NTRU with the sender's private key. Before this optimized NTRU encryption process, the encrypted message is concatenated with hash value, which is obtained from the calculation of original message and MD5 hash function. The encryption process is shown in Fig. 8.

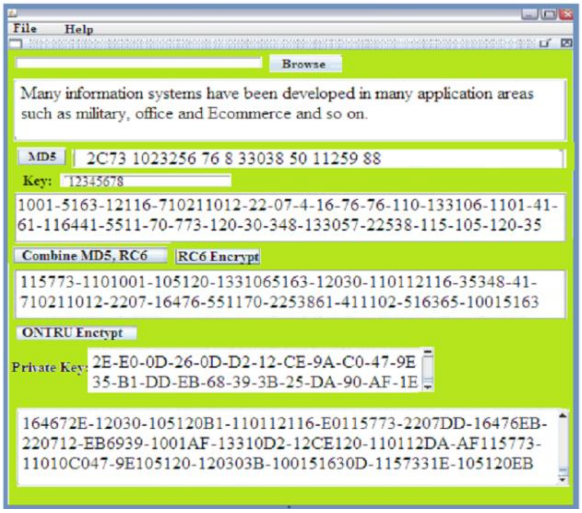


Fig. 8 : Encryption Processes

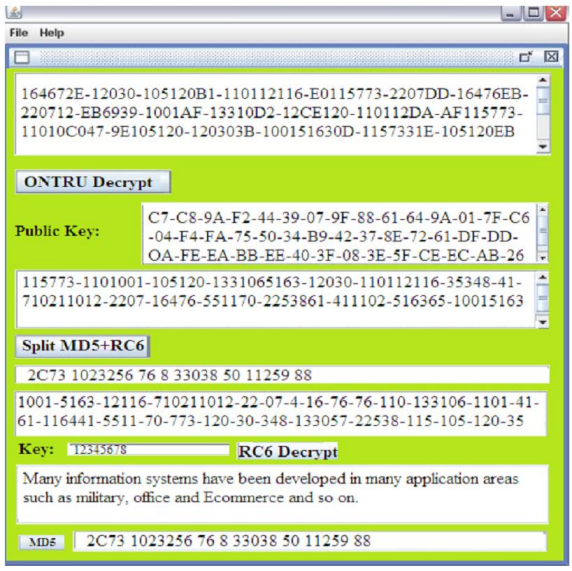


Fig. 9 : Decryption Processes

At the receiver site, optimized NTRU decryption with the sender's public key and RC6 decryption with the secret key must be performed. The decryption processes are shown in Fig. 9. After finishing the decryption processes, it must be compared that the decrypted hash value H (M) and the calculated hash value H (M)′, which is obtained from the decrypted message and MD5 hash function in order to check data integrity. If it is correct, it is ensured the data integrity.

VI. ANALYTICAL RESULTS

The analytical results of the original NTRU and optimized NTRU in terms of execution time are shown in Table I. According to the analytical results, the performance of optimized NTRU is always faster than the performance of NTRU.
These running time results are obtained by testing on different text file sizes. So, the optimized NTRU is used to encrypt and decrypt the combination of hash value and encrypted message in the proposed system.

TABLE I : RUNNING TIME RESULTS OF ENCRYPTION AND DECRYPTION PROCESS

Test ID	Text Size (KB)	Encryption (seconds)		Decryption (seconds)	
		NTRU	Optimized NTRU	NTRU	Optimized NTRU
1	80	0.045	0.036	0.070	0.041
2	112	0.072	0.067	0.087	0.058
3	130	0.097	0.085	0.104	0.089
4	170	0.112	0.107	0.138	0.106
5	220	0.149	0.137	0.165	0.127

CONCLUSION

This proposed system focused on the hybrid cryptosystem which is related with the concepts of RC6, MD5 hash function and optimized NTRU algorithm. The usage of RC6 symmetric algorithm and optimized NTRU public key algorithm gives the security requirements: confidentiality, authentication and non-repudiation. Moreover, MD5 hash function also provides to check the data integrity. 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. This hybrid cryptosystem 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] Lili Yu ; Zhijuan Wang ; Weifeng Wang "The Application of Hybrid Encryption Algorithm in Software Security "Computational Intelligence and Communication Networks (CICN), 2012 Fourth International Conference on Page(s): 762 - 765 Print ISBN: 978-1-4673-2981-1.

[2] Ravindra, K. G. and Parvinder, S.: A New Way to Design and Implementation of Hybrid Crypto System for Security of the Information in Public Network, International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com ISO 9001: 2008 Certified Journal, Volume 3, Issue 8, August(2013).

[3] S. Subasree and N. K. Sakthivel, "Design of a New Security Protocol using Hybrid Cryptography Algorithms", IJRRAS, vol. 2, February, 2010, pp. 95-103.

[4] Mahmoud Shaar and Magdy Saeb "A Hybrid Hiding Encryption Algorithm (HHEA) For Data Communication Security" Cairo University, Faculty of Science Mathematics Dept., Computer Science Division, Cairo, Egypt and Arab Academy for Science, Technology & Maritime Transport School of Engineering, Computer Department.

[5] R. Rivest, The MD5 Message-Digest Algorithm, MIT Laboratory for Computer Science and RSA Data Security, Inc., April, 1992.

[6] Man Young Rhee "Internet Security Cryptographic Principles, Algorithms and Protocols" School of Electrical and Computer Engineering Seoul National University, Republic of Korea, 2003.

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

[8] K. Wilhelm, Aspects of Hardware Methodologies for the NTRU Public-Key Cryptosystem, Computer Engineering Department, RIT-Main Campus, 2008. Available: http://scholarworks.rit.edu/theses/3214.

★★★