Analysis A parallel combination between the Number Theorists aRe Us, Riverst Shamir Adleman and Triple Data Encryption Standard methods for measuring the speed of document security

Luken Candawan Hardiaman Hulu, William, David Sipayung, Muhammad Hafis, Christnatalis
Informatics Engineering Study Program, Department of Technology and Computer Science, Universitas Prima Indonesia, Medan, Indonesia
* lukenhulu@gmail.com

Abstract. Computer security is a very important thing in an information system. The strength of the NTRU algorithm is the difficulty of finding a short vector of a lattice (a discrete subgroup of a collection of vectors that includes the entire vector environment) of a random polynomial that has a large degree. The strength of the RSA algorithm lies in the level of difficulty in factoring non-prime numbers into the primary factor. As long as no efficient algorithm has been found to find the prime factors of large integers, the RSA algorithm is highly recommended for message encryption. And finally this is the TRIPLE DES algorithm designed using a 56-bit key, and that size is enough to run a secure encryption technique. This algorithm provides a simple solution that is running the DES Algorithm 3 times for data blocks. The results of our study of measuring speed on encryption and decryption using the same file are superior to the RSA algorithm compared to the NTRU and TRIPLE DES algorithms.

1. Introduction

Computer security is a branch of technology known as information security applied to computers. The goal of computer security is to protect information against information theft. The purpose of implementing computer security is to prevent data loss and prevent intruders (intruders) from performing interruptions (deletion or destruction), interception (access), modification (changing data) and fabrication (adding false data).

NTRU (Number Theorists aRe Us) is one of the public key algorithms that has a higher level of speed. NTRU was first introduced in 1996. This algorithm was created by 3 people namely Jeffrey Hoffstein, Jill Pipher and Joseph Silverman. The strength of this algorithm is where it is difficult to find a short vector of a lattice. [1] [2] [8] [9] [14] [15]

RSA (Riverst Shamir Adleman) is the most frequently used cryptographic algorithm. The RSA algorithm was designed by 3 researchers namely Ron Rivest, Adi Shamir, and Leonard Adleman in 1976. They used public keys and private keys to process encryption and decryption. The advantage of the RSA algorithm is that it is difficult to factor large numbers into prime numbers. [3] [4]

TRIPLE DES (Triple Data Encryption Standard) is a symmetrical algorithm on cryptography that is used to secure data by encoding data. The Triple DES algorithm is an algorithm developed from the DES (Data Encryption Standard) algorithm found in the early 1970s using 56 little keys. The difference between DES and Triple DES lies in the length of the key used. In DES it uses one key that
is 56-bit long, whereas in Triple DES it uses 3 keys up to 168-bits in length (each 56-bit length). This means that the effective key strength for Triple DES is actually 168 bits because each of the three keys containing 8 parity bits is not used during the encryption process.

Narasimham Challa and Jayaram Pradhan's study discussed the comparison between the NTRU and RSA methods for testing speed and efficiency to verify their performance for different text files on variable sizes. [2]

While the research was conducted by Praveen Kumar B and Rajaanadan N.S. using Triple DES Algorithm to analyze the efficiency of the performance of the algorithm. [5]

Based on Urian above in the study we discussed about testing the speed between the 3 NTRU, RSA, and Triple DES methods. The basic idea is to get information on testing 1 document file that you want to identify the NTRU and RSA, RSA and Triple DES methods, Triple DES and NTRU.

2. Methods

The methodology used in this study uses the NTRU, RSA and Triple Des algorithms which focus on the speed of data security. This method is used as a reference in the superiority of speed of security.

2.1. NTRU (Number Theorists aRe Us)

NTRU is based on a certain polynomial ring algebraic structure. Before we continue, we set a number of notations. The following are some domain parameters for NTRU implementation. [3]

\[ n = \text{The dimensions of the polynomial ring used on the NTRU. (this polynomial is n, 1)}. \]
\[ p = \text{Positive integers that determine the ring } \mathbb{Z} / p \mathbb{Z} \text{ where the coefficient of a particular product of the polynomial will decrease during the encryption and decryption process}. \]
\[ q = \text{Positive integers that determine the } \mathbb{Z} / p \mathbb{Z} \text{ ring where the coefficients of a particular product of the polynomial will decrease during the encryption and decryption process, also used in making public keys}. \]
\[ k = \text{Security parameters that control resistance to certain types of attacks, including plaintext awareness}. \]
\[ d_f = \text{The coefficient distribution of the polynomial } f, \text{ below (f is part of the private key)}. \]
\[ d_g = \text{Coefficient distribution of polynomial } g, \text{ below (g used for making public keys)}. \]
\[ d_r = \text{The number 1s and 1s are used in r certain random polynomials, below, in the encryption procedure}. \]

Creating the NTRU Algorithm Key

To generate the public key pair and the NTRU private key, the following steps are needed.[3]

1. Randomly select 2 small potential polynomial \( f \) and \( g \) which are members of \( R \) (polynomial N-1). The contents of the polynomial \( f \) and \( g \) are kept secret. Polynomial is relatively small against the mod random polynomial \( q \), in a random polynomial, the coefficient will be randomly distributed in mod \( q \), so that in the small polynomial the coefficient will be much smaller than \( q \).

2. Then find the value of \( F_q \) and \( F_p \) using the following two equations:

\[ F_q * f \equiv 1 \pmod{q} \]
\[ F_p * f \equiv 1 \pmod{p} \]

3. Next count

\[ h \equiv p \cdot f_q * g \pmod{q} \]

The public key that will be used for encryption is polynomial \( h \) while the private key for decryption is polynomial \( f \) and \( f_p \).
NTRU Encryption

Encrypt Message M the do the following steps:

1. Change message M to polynomial m (member of polynomial N-1) with the coefficient selected is modulo p which is at intervals \(-q/2 \) to \(q/2\). (m is a small polynomial of mod q).
2. Randomly select a small polynomial r
3. Encrypt using the public key h with the following equation:

\[ e \equiv r^*h + m \pmod{q} \]

polynomial e is an encrypted message.

NTRU Decryption

After the e-encrypted message is received, the decryption can be done with the following steps

1. Calculate the value of a using the private key f as follows:

\[ a \equiv f^*e \pmod{q} \]

where the value of a polynomial is at intervals of \(-q/2 \) to \(q/2\).
2. Using a polynomial, we can get the message m with the following equation

\[ c \equiv f^*_e \cdot a \pmod{p} \]

Polynomial c will be the original message m.

2.2. RSA (Riverst Shamir Adleman)

Security The RSA algorithm uses 2 numbers (e and d) as public keys and private keys. In the RSA algorithm e and n are publicly announced while d is kept confidential. Therefore, the RSA algorithm is useful for short messages. Since the algorithm uses 2 keys for encryption and decryption. The RSA algorithm has the following magnitude:

- p and q are prime numbers (private key)
- \( n = p \times q \) (public key)
- \( \varphi(n) = (p-1)(p-1) \) (private key)
- e = key enkripsi (public key)
- d = key dekripsi (private key)
- m = plaintext (private key)
- c = ciphertext (public key)

To find out how the RSA algorithm works

1. Determine 2 prime numbers, with names p and q
2. Calculate the modulus value (n) to get the value (n)

\[ n = p \times q \]

3. Calculating the total value of n

\[ (n) = (p - 1)x(q - 1) \]

4. Determine the value of e with the terms \( \gcd(e, \varphi(n)) = 1 \)

Where e = prime number, and \( 1 < e < \varphi(n) \)

5. Looking for value the deciphering exponent (d), then

\[ d = \left(1 + \left(k \times \varphi(n)\right)/e\right) \]

6. From the steps outlined above, has been obtained value n, e, dan d so that the key has been formed the public key \((n, e)\), and the private key \((n, d)\).
RSA Encryption
Plaintext \((e) \mod (n) = \text{ascii code}\)

RSA Decryption
\text{ascii code} \((d) \mod (n) = \text{ascii code}\)

2.3. Triple DES (Triple Data Encryption Standard)

Security The Triple DES algorithm runs three times slower than DES, but is much safer if used correctly. 3 DES provides only 112 bits to decrypt something with the same encryption, unless executed in reverse. [5] The way it works as follows:

Run 3 times the DES algorithm, 
If \(K_2 = K_3\), this is DES Decryptions.

The Triple DES algorithm uses three iterations of the DES cipher in general. 168-bit key, which is divided into three 56-bit keys.
- encryption uses the first private key
- decryption using the second private key
- encryption using the third private key

Triple DES Encryption
\[ c = E_3\left(D_2\left(E_1(m)\right)\right) \]

Triple DES Decryption
\[ m = D_1\left(E_2\left(D_3(c)\right)\right) \]

Decryption in the second step during encryption provides backward compatibility with the first and second secret key DES or the second and third secret keys are the same key.

\[ c = E_3\left(D_1\left(E_1(m)\right)\right) = E_3\left(m\right) \]
\[ c = E_3\left(D_3\left(E_1(m)\right)\right) = E_1\left(m\right) \]

the first and third secret keys are the same. This is more powerful than just encrypting DES that is used twice (with two 56-bit keys) because it protects against meet-in-the-middle attacks.

\[ c = E_1\left(D_2\left(E_1(m)\right)\right) \]

Figure 1. How to Work Triple Des The length of the DES triple key contains 168 bits but the security key falls to 112 bits.
3. Testing dan Result

The following is the display of the NTRU program (figure 2,3,4) RSA (figure 5,6,7) and Triple DES (Figure 8,9):

The first step is the process of making keys, after which do the key input (key) press the random button to get the value p, q and n if the value p, q and n has come out then press the validation button. If all values are valid then we enter the value.

Then press the button input polynomial F and press the random key, then press the SET button, then get the value of the polynomial F. Then do the same step again to get the polynomial G value. Then the polynomial value \( F_p, F_q \) and H will be displayed in the column as follows in figure 2.

![Figure 2. Display of the NTRU Key Making Program](image)

The first step in the encryption process, input the file that will be encrypted, then input the key value obtained from the key making in the first image, then press the validation button then press the SET button then press the polynomial input button, then press random and SET again, then press the process button and wait until the time appears from the results of the encryption. Then save the encrypted file as follows in figure 3.

![Figure 3. Display of the NTRU Encryption program](image)
The first step of decryption, input the encrypted file then press the input button, and then enter the values p, q, and n obtained during the key formation process. Then press the polynomial F button, press the random then SET. Do the same steps to get the value of the polynomial $F_p$. Then press the process button to start decryption, after the decryption process is complete, it will take a long time to decrypt, and save the decrypted file as follows in figure 4.

![Figure 4. Display of the NTRU Decryption program](image)

The first step in the key formation process in RSA algorithm. To get p and q values, press the generate button random value then validate whether the value is valid or not. If it is valid then we will get an N value and $T_1(n)$, then to get the value of d press the button generate the random value to get the value then validation then it can also be the value of e as shown below in figure 5.

![Figure 5. Display of the RSA Key Formation program](image)

Then, in the encryption process, the first step is the input file that will be encrypted, then input the same value of e and n during the key formation process, then press the encryption process and wait until the encryption process is complete, then the encryption process will be saved. already encrypted as shown below in figure 6.
The first step is input the file has been encrypt, then input the same value of d and n during the key formation process, after that press the decryption process button and wait until the decryption process is complete, then the decryption process time will be obtained, then save the finished file decrypted as shown below in figure 7.

The triple des algorithm does not require the formation of keywords using only the password specified by the user.

The first step of the triple des encryption process, enter the file that will be encrypted and then input the password in the form of a password determined by the user, after that press the process button to start the encryption process then the encryption process time will be obtained, then save the file that has been encrypted as shown below in figure 8.
The first step in the Triple des decryption process, enter the encrypted file and enter the password in the appropriate password during the encryption process, then press the process button to start the decryption process and then get the decryption process time, then save the decrypted file as shown below in figure 9.

Figure 9. Display of Triple DES decryption program

The following are the test results to compare 3 methods, namely by comparing the NTRU, RSA and TRIPLE DES methods.

a. In the first step, the author first compares the NTRU and RSA methods. From the comparison of the two methods the authors get results, that in terms of speed of data security in the form of JPEG, Movie Clip, Mp4, PDF, Docx and Mp3 files, it turns out that the RSA method is more efficient.

The following is a table of test results from the NTRU and RSA methods

Table 1. Testing Results of the NTRU and RSA Methods

| No | File type | NTRU Encryption Time | NTRU Decryption Time | RSA Encryption Time | RSA Decryption Time |
|----|-----------|----------------------|----------------------|--------------------|--------------------|
| 1. | JPEG      | 00:21:09.577         | 00:09:15.220         | 00:03:27.271       | 00:00:10.581       |
| 2. | JPEG      | 00:00:10.046         | 00:00:03.244         | 00:00:01.408       | 00:00:00.430       |
| 3. | Movie Clip| 00:20:01.763         | 00:05:49.768         | 00:02:03.801       | 00:00:08.252       |
| 4. | Movie Clip| 01:39:41.070         | 00:44:35.020         | 00:13:21.326       | 00:00:37.939       |
| 5. | Movie Clip| 13:10:38.721         | 09:58:27.289         | 02:58:38.980       | 00:07:02.682       |
| 6. | Movie Clip| 06:56:45.112         | 03:22:32.265         | 00:52:10.909       | 00:02:00.276       |
| 7. | PDF       | 01:36:17.856         | 00:30:06.125         | 00:04:13.625       | 00:00:12.795       |
| 8. | PDF       | 00:26:50.270         | 00:12:27.536         | 00:03:27.546       | 00:00:09.798       |
| 9. | Docx      | 00:09:10.790         | 00:03:47.417         | 00:00:26.300       | 00:00:02.350       |
| 10.| Docx      | 00:02:20.852         | 00:02:06.407         | 00:00:15.958       | 00:00:01.622       |
| 11.| Mp3       | 10:15:15.453         | 08:27:34.543         | 01:19:17.131       | 00:03:05.320       |
| 12.| Mp3       | 06:05:41.781         | 04:16:52.171         | 00:18:15.062       | 00:00:36.672       |
b. In the second step the writer compares the NTRU and Triple DES methods. From testing the two methods the authors get the results, that in the speed of securing data in the form of JPEG, Movie Clip, Mp4, PDF, Docx and Mp3 files, the authors conclude that the NTRU method is more efficient than TRIPLE DES.

The following is a table of test results from the NTRU and Triple DES method

| No. | File type | NTRU       | Encryption Time | Decryption Time | Triple DES   | Encryption Time | Decryption Time |
|-----|-----------|------------|----------------|----------------|--------------|----------------|----------------|
| 1.  | JPEG      | 00:21:09.577 | 00:09:15.220   | 01:06:17.271   | 08:39:29.637 | 00:00:13.838   |
| 2.  | JPEG      | 00:00:10.046 | 00:00:03.244   | 00:00:20.842   |              | 00:00:20.842   | 00:00:13.838   |
| 3.  | Movie Clip| 00:20:01.763 | 00:05:49.768   | 00:15:12.798   | 02:19:08.541 | 00:00:13.838   |
| 4.  | Movie Clip| 01:39:41.070 | 00:44:35.020   | 00:36:18.133   | 02:43:49.177 | 00:00:13.838   |
| 5.  | Mp4       | 13:10:38.721 | 09:58:27.289   | 07:46:35.614   | 10:16:40.327 | 00:00:13.838   |
| 6.  | Mp4       | 06:56:45.112 | 03:22:32.265   | 04:51:29.772   | 07:03:38.412 | 00:00:13.838   |
| 7.  | PDF       | 01:36:17.856 | 00:30:06.125   | 00:25:52.825   | 03:04:25.069 | 00:00:13.838   |
| 8.  | PDF       | 00:26:50.270 | 00:12:27.536   | 00:24:55.155   | 02:46:44.146 | 00:00:13.838   |
| 9.  | Docx      | 00:09:10.790 | 00:03:47.417   | 00:03:43.447   | 01:01:46.654 | 00:00:13.838   |
| 10. | Docx      | 00:02:20.852 | 00:02:06.407   | 00:07:11.637   | 00:59:23.476 | 00:00:13.838   |
| 11. | Mp3       | 10:15:15.453 | 08:27:34.543   | 05:33:52.188   | 06:49:20.515 | 00:00:13.838   |
| 12. | Mp3       | 06:05:41.781 | 04:16:52.171   | 03:59:38.689   | 05:05:37.612 | 00:00:13.838   |

c. In the third step the writer compares the RSA and Triple DES methods. From testing the two methods the authors get the results, that in the speed of data security in the form of JPEG, Movie Clip, Mp4, PDF, Docx and Mp3 files, the authors conclude that the RSA method is more efficient than TRIPLE DES.

The following is a table for testing the RSA and Triple DES methods

| No. | File type | RSA       | Encryption Time | Decryption Time | Triple DES | Encryption Time | Decryption Time |
|-----|-----------|-----------|----------------|----------------|------------|----------------|----------------|
| 1.  | JPEG      | 00:03:27.271 | 00:00:10.581   | 01:06:17.271   | 08:39:29.637 | 00:00:13.838   |
| 2.  | JPEG      | 00:00:01.408 | 00:00:00.430   | 00:00:20.842   | 00:00:13.838 | 00:00:13.838   |
| 3.  | Movie Clip| 00:02:03.801 | 00:00:08.252   | 00:15:12.798   | 02:19:08.541 | 00:00:13.838   |
| 4.  | Movie Clip| 00:13:21.326 | 00:00:37.939   | 00:36:18.133   | 02:43:49.177 | 00:00:13.838   |
| 5.  | Mp4       | 02:58:38.980 | 00:07:02.682   | 07:46:35.614   | 10:16:40.327 | 00:00:13.838   |
| 6.  | Mp4       | 00:52:10.909 | 00:02:00.276   | 04:51:29.772   | 07:03:38.412 | 00:00:13.838   |
| 7.  | PDF       | 00:04:13.625 | 00:00:12.795   | 00:25:52.825   | 03:04:25.069 | 00:00:13.838   |
| 8.  | PDF       | 00:03:27.546 | 00:00:09.798   | 00:24:55.155   | 02:46:44.146 | 00:00:13.838   |
| 9.  | Docx      | 00:00:26.300 | 00:00:02.350   | 00:03:43.447   | 01:01:46.654 | 00:00:13.838   |
| 10. | Docx      | 00:00:15.958 | 00:00:01.622   | 00:02:11.637   | 00:59:23.476 | 00:00:13.838   |
| 11. | Mp3       | 01:19:17.131 | 00:03:05.320   | 05:33:52.188   | 06:49:20.515 | 00:00:13.838   |
| 12. | Mp3       | 00:18:15.062 | 00:00:36.672   | 03:59:38.689   | 05:05:37.612 | 00:00:13.838   |
d. In the next step the writer compares the three methods simultaneously, namely NTRU, RSA and Triple DES. From the testing of the three methods the authors get results, that in the speed of data security in the form of JPEG, Movie Clip, Mp4, PDF, Docx and Mp3 files, the authors conclude that the RSA method is more efficient than NTRU and TRIPLE DES.

The following is a table compared to the NTRU, RSA and TRIPLE DES methods.

**Table 4. Results of the speed comparison between the NTRU, RSA and TRIPLE DES methods**

| No | File Type | NTRU Encryption Time | NTRU Decryption Time | RSA Encryption Time | RSA Decryption Time | Triple DES Encryption Time | Triple DES Decryption Time |
|----|-----------|----------------------|----------------------|--------------------|--------------------|---------------------------|---------------------------|
| 1  | JPEG      | 00:21:09.577         | 00:09:15.220         | 00:03:27.271       | 00:00:10.581       | 01:06:17.271              | 08:39:29.637              |
| 2  | JPEG      | 00:00:10.046         | 00:00:03.244         | 00:00:01.408       | 00:00:00.430       | 00:00:20.842              | 00:00:13.838              |
| 3  | Movie Clip | 00:20:01.763         | 00:05:49.768         | 00:02:03.801       | 00:00:08.252       | 00:15:12.798              | 02:19:08.541              |
| 4  | Movie Clip | 01:39:41.070         | 00:44:35.020         | 00:13:21.326       | 00:00:37.939       | 00:25:52.285              | 02:43:49.177              |
| 5  | MP4       | 13:10:38.721         | 09:58:27.289         | 02:58:38.980       | 00:07:02.682       | 07:46:35.614              | 10:16:40.327              |
| 6  | MP4       | 06:56:45.112         | 03:22:32.265         | 00:52:10.909       | 00:02:00.276       | 04:51:29.772              | 07:03:38.412              |
| 7  | PDF       | 01:36:17.856         | 00:30:06.125         | 00:04:13.625       | 00:00:12.795       | 00:25:52.285              | 03:04:25.069              |
| 8  | PDF       | 00:26:50.270         | 00:12:27.536         | 00:03:27.546       | 00:00:09.798       | 00:24:55.155              | 02:46:44.146              |
| 9  | Docx      | 00:09:10.790         | 00:03:47.417         | 00:00:26.300       | 00:00:02.350       | 00:03:43.447              | 01:01:46.654              |
| 10 | Docx      | 00:20:28.852         | 00:02:06.407         | 00:00:15.958       | 00:00:01.622       | 00:02:11.637              | 00:59:23.476              |
| 11 | MP3       | 10:15:15.453         | 08:27:34.543         | 01:19:17.131       | 00:03:05.320       | 05:33:52.188              | 06:49:20.515              |
| 12 | MP3       | 06:05:41.781         | 04:16:52.171         | 00:18:15.062       | 00:00:36.672       | 03:59:38.689              | 05:05:37.612              |

4. Conclusions
From the results of testing on the NTRU, RSA and Triple Des methods, we can conclude that using the RSA method is more efficient than NTRU and TRIPLE DES, fast and slow encryption and decryption processes that depend on files and computer specifications that will help facilitate the encryption process and time descriptions.

References
[1] Hoffstein J., Lieman D., Pipher J., Silverman J. “NTRU: A Public Key Cryptosystem”, NTRU Cryptosystems, Inc. (www.ntru.com).
[2] Hoffstein J., Lieman D., Pipher J., Silverman J. “NSS: An NTRU Lattice-Based Signature Scheme”, NTRU Cryptosystems, Inc., 5 Burlington Woods, Burlington, MA 01803USA (www.ntru.com).
[3] Challa N and Pradhan J., “Performance Analysis of Public key Cryptographic Systems RSA and NTRU”, IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.8, August 2007.
[4] A. Huffman “A method for the construction of minimum redundancy codes Proc. IRE, vol. 40, pp. 1098–1101, Sept. 1952.
[5] Kumar P B and Rajaanadan N.S. “Data Encryption and Decryption Using By Triple DES Performance Efficiency Analysis of Crypto system”, International Journal of Engineering Research and Development, vol. 4, Issue 3, March 2016.
[6] Zalekian A, Esmaeildoust M, and Kaabi A. “Efficient Implementation of NTRU Cryptography using Residue Number System”, International Journal of Computer Applications (0975 – 8887), Volume 124 – No.7, August 2015.
[7] Diffie W., Hellman M. “New directions in cryptography” IEEE Information theory, June 23-25, 1975 and IEEE International Symposium on Information theory, Sweden, June21-24, 1976.

[8] Hoffstein J., Silverman J. “Optimizations for NTRU” Proceedings of conference on Public key Cryptography and Computational number theory, Warsaw, De Gruyter ,2000 (Sep 11-15), 77-88.

[9] O’Rourke C M. “Efficient NTRU Implementation” A thesis For Master of Science at Worcester Polytechnic Institute, Apr 2002.

[10] J. Hoffstein., J. Pipher., J. H. Silverman. “NTRU: A new high speed public key cryptography system in Algorithmic number theory” (ANTS III), Portland , OR, June 1998, Lecture Notes in Computer Science 1423 (J. P. Buhler, ed.) Springer-Verlag , Berlin 1998, 267-288.

[11] Thiagarajan K “NTRU – A Public key ring based algorithm” A thesis for Master of Science, University of Texas, Dallas, 2003.

[12] C. Narasimham., Pradhan J. “Performance analysis: RSA and Truncated polynomials”

[13] R.L.Rivest, A.Shamir, L.Adleman “A method for obtaining digital signatures and Public-Key Cryptosystems”, Communications of the ACM 21 (1978),120-126.

[14] Hoffstein J., Pipher J., Silverman J. “NTRU-A Ring based public key cryptosystem” Lecture notes in Computer Science, Springer-Verlag, Berlin 1433(1998),267-288.

[15] Kumar, R. G. V. S., Jumar, N. K., Sekhar, C. P., Numma, B. V. V. S., Kumar, V. B., "Modified Mutual Authentication and Key Agreement Protocol Based on NTRU Cryptography for Wireless Communications", International Journal of Computer Science and Network (IJCSN),Volume 1, Issue 4, August, 2012.