EMITTER International Journal of Engineering Technology
Vol. 6, No. 2, December 2018
ISSN: 2443-1168

Trusted Data Transmission Using Data Scrambling Security Method with Asymmetric Key Algorithm for Sychronization

Nihayatus Sa'adah, I Gede Puja Astawa, Amang Sudarsono

Electrical Engineering Department
Politeknik Elektronika Negeri Surabaya (PENS)
St. Raya ITS, Keputih, Sukolilo, Surabaya, 60111, Indonesia
Email : nihayatus@pasca.student.pens.ac.id, {puja,amang}@pens.ac.id

Abstract

Security is a major concern of the internet world because the development of the Internet requires the security of data transmission. The security method helps us to store valuable information and send it over an insecure network so that it can not be read by anyone except the intended recipient. Security algorithm uses data randomization method. This method of data information randomization has a low computation time with a large number of bits when compared to other encryption algorithms. In general, the encryption algorithm is used to encrypt data information, but in this research the encryption algorithm is used for synchronization between the sender and the intended recipient. Number of bits on asymmetric key algorithm for synchronization are the 64-bits, 512-bits and 1024-bits. We will prove that security methods can secure data sent with low computational time with large number of bits. In the result will be shown the value of computing time with variable number of bits sent. When data are sent by 50 bytes, encryption time required 2 ms using 1024 bits for synchronization technique asymmetric key algorithm.

Keywords: Data Scrambling, Asymmetric Key, Security, Synchronization.

1. INTRODUCTION

Wireless networks allow users to access wherever the user is located. Development of internet much exploited by hackers to retrieve the information was not on him. Security system is required to give protection for data transmitted in the digital era as now. In wireless communications, we are able to transmit and accept data. Examples of internet applications that require security include online transactions, Internet banking, online store, education, military, etc. The security of transmitted data must comply aspect of authenticity, availability, privacy, and integrity.

Cryptography is one form of transmitted data protection from the sender to the intended recipient [1]. Cryptography consists of the encryption process on the sender and the decryption process at the intended recipient [1]. Encryption is a process of securing a hidden data or data conversion process (plaintext) into a form that is not readable / understandable [2]. Encryption has been used to secure communications in different countries. The original information is referred to as plaintext, and the encrypted form is called ciphertext. The chipertext message contains all the information from the plaintext message, but not in the format readable by human or computer without using the proper mechanism for decryption. While decryption is the reverse of the encryption process is the data conversion process is encrypted (ciphertext) back into the original data (original plaintext) that can be read / understood back. The encryption algorithm performs various substitutions and transformations on the plaintext (the original message before encryption) and converts it into ciphertext (random message after encryption). Many algorithm to encrypt the data being researched and utilized in the security of data transmitted. The plaintext conversion algorithm is grouped into two groups: symmetric key algorithm (also called Symmetric-key encryption) and asymmetric key algorithm (also called Asymmetric-key encryption). In general, the classification of cryptography algorithm can be seen in Figure 1.

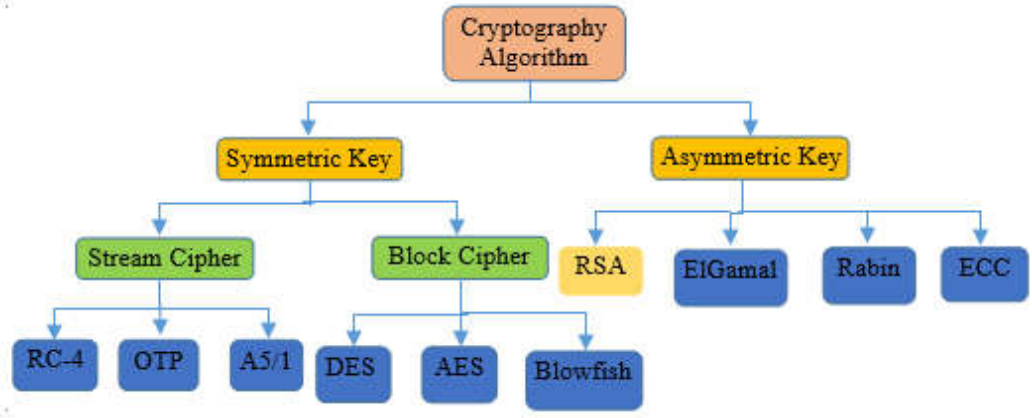

Figure 1. Classification of Cryptography Types [6]

The symmetric algorithm of the encryption and decryption process uses one key or symmetric key, so the sender and receiver must first have the same key that has been agreed to be used so that the sender and receiver can communicate. The plaintext passes through the encryption process and generates the ciphertext, then encoded again using the same key so that it becomes its original form. The symmetric key method is more suitable for use in a symmetric building area because the message delivery does not use message storage, so the security of this symmetric algorithm lies in the security of key delivery and to the length of the key used. The weakness of the algorithm using this symmetric key is the key must be distributed

securely, since it has the same degree of confidentiality as the data sent, and the key can not be revealed at all. Cryptography types using symmetric key are DES, AES, blowfish, etc on block cipher [1] and RC-4, OTP and A5/1 on stream cipher.

In asymmetric algorithm, the use of keys for data encryption process is different with the key for the decryption process so that the encryption method using asymetric key is different when compared with the use of symmetric key method. The basic principle of this algorithm is that every member in the network has 2 keys, namely public key and private key. A public key is a key used to encrypt and a key to decrypt is called a private key. The private key is only owned by the person who does the encryption process only and the public key is known by many people. Asymmetric key algorithms are more commonly referred to as public key, usually these algorithms are used in large and dynamic communications networks. The asymmetric key algorithm is relatively more difficult to solve, because the key used to encrypt it is different from the key to decrypt it. Asymmetric key algorithm is relatively more difficult to solve method, because the key used to encrypt it is different from the key to decrypt it [3]. The disadvantage of this asymmetric key algorithm is slower than the symmetric key algorithm [4], a cryptographic example that uses this public key such as RSA, ElGamal, Rabin and ECC methods.

Cryptography is used to secure information transmitted from unauthorized recipients by converting them into ciphertext, but in this study cryptography is used as a synchronization between a sender and valid receiver. The data scrambling security method requires synchronization to locate the data scrambling anywhere. The cryptography algorithm uses the asymmetric key, since the public key algorithm does not require a private channel when it is shared and more secure because it uses a public key and a asymmetric key. The RSA algorithm strength lies in the factoring of two large prime numbers RSA public key algorithm is choosen as synchronization between server and client.

The next arrangement of this paper is structured as follows: Section II reviews previous related research on various methods of security to protect transmitted data. Also, it explains the encryption process of random numbers. Section III describes the encryption process using the proposed security method with RSA synchronization. Section IV explains about implementaton security method on server and client. Section V discusses performance analysis and computation time of the proposed algorithm. The conclusions are summarized in Section VI.

2. RELATED WORKS

There are many who examine the security algorithm because of the importance of security in communication systems. There are some studies that were previously associated with security algorithms. The one of studies

on security information is performance analysis of encryption algorithms for security which is investegated by Madhuma Panda [5]. Madhuma has been done evaluation of both cryptography algorithm, AES, DES, Blowfish as symmetric algorithm and RSA as asymmetric algorithm. The result is evaluated by comparising encryption time, decryption time and throughput. Data are taken with difference the types of file, such as binary, text and image data.

In other hand, Bijoy kumar et al have been designed encryption scheme by combining RSA encryption and Diffie-Hellman method to produce higher security level [6]. Diffie-Hellman is used for choosing a pair of RSA's asymmetric key randomly. After this process, data information is encoded using RSA algorithm. The purpose of RSA encryption is intruder can not find the public component of Diffie-Hellman. Combination of Diffie-Hellman and RSA algorithm can increase throughput value and decrease power consumption.

Azzam proposed a new randomized encryption/decryption scheme as security algorithm [7]. The randomized encryption depends on the symmetric key and random number for each encryption process. Subkey for every block of data generate using thhe symmetric key and the random number. Development of the scheme use physical operation concept. Subkey is generated from new angle which is get from turning disk clockwise with assume fixed pointer P.

Souvik S. et al present encoding algorithm by encrypting 8 bit of plaintext binary into 8 bit of ciphertext binary dan decoding algorithm by decrypting 8 bit of ciphertext binary into 8 bit of original plaintext [8]. the use of algorithm can produce public key by swap process. This algorithm is also compatible on 16 and 32 bit binaries.

There was research by Aiswarya et al [9] Binay RSA encryption algorithm (BREA). This encryption method uses a modified RSA algorithm, where the modified RSA algorithm uses 4 prime numbers. The ciphertext is shipped in binary zero and one. Binary ciphertext makes the length of the ciphertext longer. Longer ciphertext size causes overload communication paths. This security scheme is a modification of a security modified RSA algorithm (MREA) scheme. the decimal format (containing 0-9) is the text of the MREA code that is sent, so BREA requires at least 4 binary bits to represent a number.

The Enhanced RSA is proposed by George A. et al [10]. This security method has two value of N, N1 and N2. Four prime numbers is used for encrypting and two large prime numbers for decrypting. Public key pair (e,N1) is used to encrypt data information and asymmetric key pair is (d,N2) which is used for decrpyting output of encryption.

Karthik et al creates an encryption method from a combination of symmetric key and asymmetric key algorithm to increase the level of security in data protection. This method can solve the problem on a symmetric key that sends the security key to use the decryption process and reduces the

computing time on the asymmetric key [11]. The plaintext data is encrypted using the OTP algorithm. The key of OTP is encoded using the RSA asymmetric key algorithm and stores the ciphertext of RSA algorithm in the cloud. The server picks up the passkey in the cloud using a asymmetric key. The key is accessed by receiver, the request message is sent to the client for encrypting plaintext. The receiver gets an encrypted plaintext from the cloud and decrypts it using an OTP key.

In other work on security method, there is a research by Punit Chaudhury et al [12] which is encrypt data with assymetric key using four prime numbers. The security algorithm is called ACAFP. ACAFP can reduce cost computation because it uses four prime numbers with small number. The use of a small prime number will make it easier for intruders to find prime numbers so that intruders can retrieve transmitted information.

Previously, Rizky et al. propose subcarrier randomization for securing data communication [13]. In this research they use scrambling pattern transmission for synchronization to know the scrambled location. It is not secure if they send the scrambling pattern from transmitter to receiver through the channel. We propose method for synchronization between transmitter and receiver to improve the weakness of scrambling pattern transmision method. RSA algorithm is used as synchronization to know the location of randomized data location and restore the data location to its original location.

3. ORIGINALITY

The new security method is proposed on this research. The security method utilizes data scrambling process. This security method requires synchronization between transmitter and receiver. Synchronization between the transmitter and receiver is necessary because both of them must know the location of the scrambled data. There are various synchronization techniques, one of them is scrambling pattern transmission [13]. In this security method using RSA algorithm for synchronization. The RSA algorithm can synchronize without having to send along with the data information.

4. SYSTEM DESIGN

4.1 Asymmetric Key Algorithm

RSA is one of asymmetric key algorithm. Rivest, Adi Shamir and Leonardo Adleman were creator of RSA algorithm [14-16]. RSA is security algorithm using concept of asymmetric key cryptography. The level of RSA security is supported by two large prime number [17]. Asymmetric key algorithm has two key to encrypt transmitted data and decrypt ciphertext in intended recipient. The both of them are totally different. The key for converting transmitted data into ciphertext is public and the key for decrypting ciphertext into preliminary data is asymmetric. RSA is really understandable because this algorithm consist of exponential, eulers's function, fermat

theorem and euclid algorithm. Asymmetric key algorithm has 3 important steps of process, specifically public and asymmetric key generation, ciphertext creation, decrypting ciphertext. Factoring of big number into two large prime numbers is very hard, this is one of RSA excellence. It is also an algorithm which does not require a dedicated channel to transmit the key.

There are several parameters which is used on asymmetric key algorithm :

a) x and y prime numbers (asymmetric)
b) z = xy (public)
c) φ (z) = (x-1)(y-1) (asymmetric)
d) p (public key) (public)
e) s (asymmetric key) (asymmetric)
f) m (plaintexts) (asymmetric)
g) c (output of encryption) (public)

The basis of the RSA algorithm is the euler theorem which is shown on (1).

$$a^{\varphi(z)} \equiv 1 \,(mod\ z) \tag{1}$$

Where a is relatively primed with z, and the result of $\varphi(z)$ is

$$\varphi(z) = z\left(1 - {}^{1}/_{y_1}\right)\left(1 - {}^{1}/_{y_2}\right)\ldots\ldots\ldots\left(1 - {}^{1}/_{y_r}\right) \tag{2}$$

$y_1, y_2, \ldots\ldots y_r$ is the prime factor from $\varphi(z)$. $\varphi(z)$ is a function which determine how many of the numbers 1, 2, 3, ..., z are relative primed with z. Based on the characteristic of $a^n \equiv b^n \,(mod\ z)$ where n is integer ≥ 1, equation (1) becomes (3)

$$a^{n\varphi(z)} \equiv 1^n \,(mod\ z) \tag{3}$$

Equation (3) can be expressed with (4) :

$$a^{n\varphi(z)} \equiv 1 \,(mod\ z) \tag{4}$$

If variable a is M, so (2) is changed to (5) :

$$M^{n\varphi(z)} \equiv 1 \,(mod\ z) \tag{5}$$

Based on the characteristic of $ac \equiv bc \,(mod\ z)$, if (5) is multiplied with M and the equation become different in (6) :

$$M^{n\varphi(z)+1} \equiv M \,(mod\ z) \tag{6}$$

The next process is choosing public key P (for encrypting) and asymmetric key S (for decrypting). The values of P and S are selected such as

$$P.S \equiv 1(mod\ \varphi(z)) \quad (7)$$

Or

$$P.S = n\varphi(z) + 1 \quad (8)$$

Substitute (7) into (9) so that the equation becomes:

$$M^{P.S} \equiv M\ (mod\ z) \quad (9)$$

So equation (9) can be rewritten as

$$(M^P)\ ^S \equiv M\ (mod\ z) \quad (10)$$

From the above equation, we get the encryption and decryption equation on RSA algorithm as follows :

$$P_P(M) = C \equiv M^P\ mod\ z \quad (11)$$

$$S_S(C) = M \equiv C^S\ mod\ z \quad (12)$$

Encryption and decryption process using asymmetric key algorithm can be explained by Figure 2.

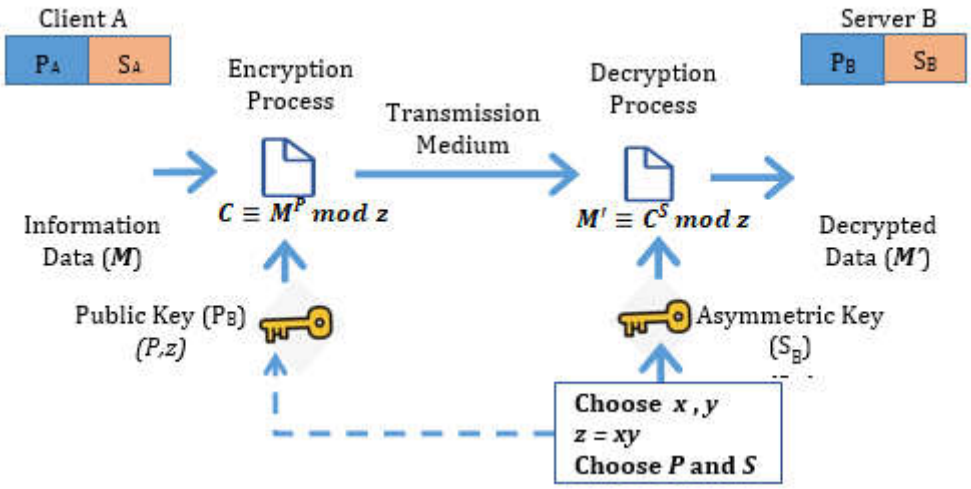

Figure 2. Illustration of Asymmetric Key Cryptosystem

RSA algorithm is not used for encrypting information data, this algorithm will encrypt random number plaintext in proposed method. Asymmetric key algorithm is utilized for synchronization between server and client. Number of *z* bits on asymmetric key algorithm are 64 bits, 512 bits and 1024 bits.

4.2 The Proposed Security Method

The proposed security method is data scrambling using asymmetric key algorithm for synchronization. Data will be scrambled on each sections, where each section consists of *z* bits. For more detail, the pseudocode of proposed security method is shown by Figure 3.

```
%%KEY_GENERATION
x ← Generate large prime number randomly
y ← Generate large prime number randomly
z ← x*y
φ(z) ← (x-1)(y-1)
while TRUE do
    r ← Generate a random number
    P ←r(mod(z-2)) + 2
    if gcd(P, φ(z)) = 1 then
        print P
    end if
end while
PUBLIC_KEY ←(z,P)
(s,k,gd) ← extend_gcd(p, φ(z))
if s < 0 then
    s ← s + z
end if
SECRET_KEY ←(S,z)

%%RSA_ENCRYPT_FOR_SYNCHRONIZATION
M← Generate random plaintext with length z bit
C ← M^p mod z

%%RSA_DECRYPT_FOR_SYNCHRONIZATION
M' ← C^s mod z

%% DATA_SCRAMBLING_SECURITY_METHOD
message ← Generate message in ascii
msgbin ← ascii_to_bin(message)
add_ZP ←sizeof(z) – (sizeof(msgbin) mod size of(z))
FOR n =0 to (sizeof(msgbin) + add_ZP)
    if n >= sizeof(msgbin) then
        tempmsg += '0'
    else
        tempmsg += msgbin(n)
    end if
```

```
30. ptextbin ←hex_to_bin(M)
31. cipherbin ←hex_to_bin(C)
32. indexA ← position of bit one on ptextbin
33. indexKA ←position of bit zero on ptextbin
34. indexB ← position of bit one on cipherbin
35. indexKB ←position of bit zero on cipherbin
36. FOR i =0 to sizeof(msgbin)
37.     result += "0"
38. FOR j =0 to sizeof(indexA)
39.     result [indexB[j]-1]= tempmsg[indexA[j]-1]
40. FOR k =0 to sizeof(indexKA)
41.     result [indexKB[k]-1]= tempmsg[indexKA[k]-1]
42. return result
```

Figure 3. Pseudocode of Data Scrambling with RSA Synchronization

The proposed security method is data scrambling using asymmetric key algorithm for synchronization. Data will be scrambled on each sections, where each section consists of *z* bits. For more detail, the proposed security method is shown by Figure 3.

a. Key Generation Process

The key generation process has the complex computation. The purpose of the key generation process is generating two keys of RSA algorithm, these are the public and the asymmetric key. Below is the process to produce a pair of secure RSA key :

1. Generation of Large Prime Number : two large prime numbers *x* and *y* are generated. If the larger prime numbers are used, the system is more considered safe.
2. Modulo : multiply the large prime numbers *x* and *y* will be generated modulo *z.*
3. Euler's totient : the euler's totient of *z*, calculation of $\varphi(z)$.

$$\varphi(z) = (x-1)(y-1) \tag{13}$$

4. Public key : the key *P* is prime number which is calculated between range 3 until $\varphi(z)$. A prime number has a gcd (greatest common divisor) of 1 with $\varphi(z)$. The public key is a pair of the exponent *P* and the modulo *z* (*P,z*)
5. Asymmetric key : Equation (8) is used for calculating the prime number of asymmetric key. Calculation of asymmetric key (*S,z*) can determine by euclidean algorithm.

b. Encrypting and Decrypting Random Number of Plaintext

The length random number plaintext is *z* bits. The value of *z* is 64 bits, 512 bits and 1024 bits. The random number of plaintext is encrypted using asymmetric key algorithm with the formula (11). While the output of encryption is decrypted using formula (12). The result of encryption and decryption will be used for synchronization on server an client.

c. Data Scrambling Method

Data scrambling method is our proposed security method for protecting transmission data. This security method will scramble data each z bits, where the data is split into *n* sections. If the amount of data is not a multiple of z bits, then the data must be added zero padding so that the amount of data modulo z equals 0. The data is randomized by the change of bit position in the output of encryption and output of decryption. The random number plaintext is generated every message is sent, so intruders can not know the position of scrambled data because plaintext and ciphertext are always changing. The output of encryption and decryption must be converted from hexadecimal into biner. On server side, the ciphertext is decoded using asymmetric key algorithm to get the random number plaintext in intended recipient. We use asymmetric key for decrypting the ciphertext, so that this system does not require the private channel.

5. EXPERIMENT AND ANALYSIS

The security algorithm is implemented for communication between server and client. This algorithm utilizing the RSA algorithm as synchronization for scrambling data location, where the number of bits used is a RSA 64 bit, 512 bit and 1024 bit. The experiment of research has two purposes, measure encryption cryptoanalysis process and test the network security of the proposed algorithm. Measurement time on data scrambling with RSA 64 bit, 512 bit and 1024 bit to know the performance of each algorithm. Network security testing is done with 3 scenarios, sniffing scenario, attacker as fake client and attacker as fake server.

5.1 Implementation of System

The testing of security method is done by simulation on linux OS. We use BigDigits library for encrypting random number of plaintext [18]. Synchronization is used for server and client to know the scrambled data's position. Random number of plaintext is encrypted using asymmetric key algorithm. The result of encryption or ciphertext and plaintext represent the change of data's position on client. The result of encryption will be decrypt and produce the random number of plaintext on server. The ciphertext and decrypted data represent the return of data's position on server. We compare the computation time by different number of bits. The number of bits which is used are 64 bits, 512 bits and 1024 bits. We also measure the transmission

time from client to server. The hardware spesification on server and client is shown by Table 1.

Table 1. Specification of H/W on Server and Client

CPU	Intel Core i3-3217U 1.80GHz
O/S	Debian GNU/Linux 7 (wheezy)
RAM	8 GB
Software	gcc-4.7.2, openssl-1.0.1

Communication between client and server uses socket where server and client are connected to LAN network. IP server is 192.168.0.2 and 192.168.0.3 as client's IP. Figure 4 illustrates the communication between client and server.

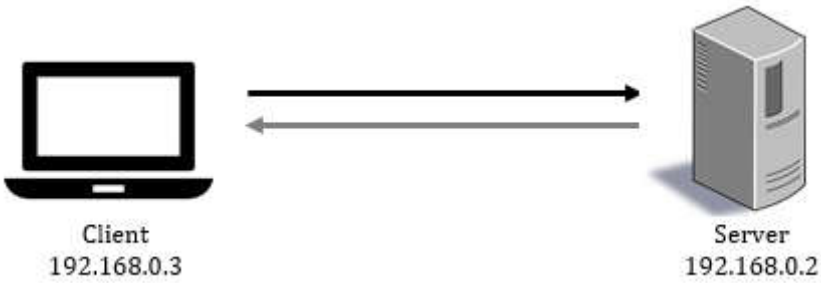

Figure 4. Illustration of Communication Client and Server

5.2 Experimental Result and Analysis

In this section, we analyze the result from the proposed security method. The security method which is proposed is data scrambling. This method requires synchronization between server and client. Many of research use asymmetric key algorithm for protecting data with encryption feature, but in this research we use asymmetric key algorithm for synchronization. we tested by measuring cryptoanalysis time and network security testing with 3 scenarios.

1. Time Measurement

The time is one of the factors in the process of encryption and decryption. With the time, speed of encryption and decryption process can be known. The following will be given table of processing time for encryption and decryption with different file sizes.

Here are the tables which present the relationship between the file size and the time required for the encryption process and the decryption process.

Table 2. The Processing Time Using 64 bits for Synchronization

Size of Information (bytes)	Encryption Time (ms)	Decryption Time (ms)
50	7	10
100	23	30
200	54	70
500	146	165
1000	285	331
2000	515	600

In table 2, there are processing time using 64 bit synchronization. In other words, every 64 bits of data, the bits of information will be randomized according to the bit location changes in the RSA plaintext and ciphertext algorithm. The encryption time consists of random number plaintext encryption time used for synchronization and randomization of bits every 64 bits, while the decrypted time is decrypted ciphertext (from random number plaintext) and returns the original data position. 64 bit are the *z* value of the RSA algorithm. Size of information 50 bytes requires 7 ms encryption time and 10 ms decryption time. On the largest information size at this simulation, 2000 bytes requires 515 ms to encrypt data and 600 ms for the decryption process. The next test is the security method using scrambling data with 512 bit RSA synchronization, the result of encryption and decryption process is shown in table 3.

Table 3. The Processing Time Using 512 bits for Synchronization

Size of Information (bytes)	Encryption Time (ms)	Decryption Time (ms)
50	4	24
100	9	28
200	12	36
500	30	53
1000	61	89
2000	111	147

The security method using 512 bits for synchronization requires 4 ms encryption time for 50 bytes encryption and 24 ms decryption time. While 2000 bytes takes 111 ms for encryption and 147 ms for the decryption process. The encryption time has a value smaller than the decryption time because when the plaintext random number is encrypted using a small number as a public key.

Table 4. The Processing Time Using 1024 bits for Synchronization

Size of Information (bytes)	Encryption Time (ms)	Decryption Time (ms)
50	2	134
100	5	137
200	7	138
500	15	149
1000	34	172
2000	65	205

The largest number of bits which is used for synchronization in this test is 1024 bits. When the security method uses 1024 bits for RSA sync on random location bits the information takes 2 ms encryption time and decryption time is 134 ms. The 2000 bytes encryption process takes 65 ms and the decryption process requires 205 ms. Encryption time is getting shorter at 1024 bits because the public key is small and the time required for randomization is shorter. Among the 64 bits, 512 bits and 1024 bits, 64 bits synchronization has an average value of most small description and 1024 bits synchronization has an average value of most small encryption. 1024 bits has a large decryption value because the private key has a large value. If the number of bits for synchronization is larger, then the computation time of randomization will be shorter because the number of sections less. Table 2, 3 and 4 show the tendency for increasing the time required to perform the encryption and decryption process. If the size of file is larger, so the time required is longer too for performing the process of encryption and decryption process.

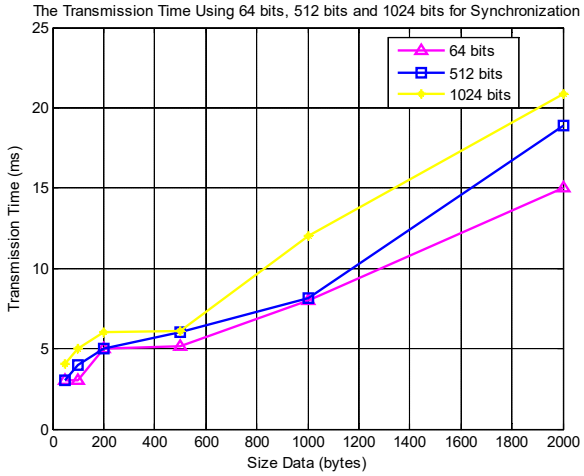

Figure 5. The Transmission Time Using 64 bits, 512 bits and 1024 bits for Synchronization

Figure 5 show the result of transmission time using 64 bits, 512 bits and 1024 bits for synchronization. The transmission time is affected by the number of bits transmitted. Data information 50 bytes with 64 bit synchronization requires 3.003 ms transmission time, with 512 bit synchronization takes longer transmission time 3.025 ms, while 1024 bit transmission time is longer than 512 bit and 64 bit because *z* value is getting larger.

The average length of time required for the decryption process is longer than the length of time the encryption process. This happens because when performing the encryption process using a public key with a small number, unlike the decryption process which uses private key with a large number.

2. Testing of Network Security

Man in The Middle or often called MITM is an attack on the security system. MITM attack is a type of attack that exploits the weakness of Internet Protocol (IP). MITM attack where the attacker is usually the active form makes independent connections with the victims, so it caused a victim who was attacked believe that they are talking directly to each other through personal connections, whereas it is in fact the entire conversation is controlled by the attacker. In the network security testing there are three scenarios of sniffing scenario, attacker acts as a fake client, attacker act as a fake server.

a. Sniffing Scenario

When two host clients and servers communicate, the attacker will perform traffic analysis by running wireshark and analyzing TCP traffic that has been stored. The illustrations of this scenario is illustrated in Figure 6.

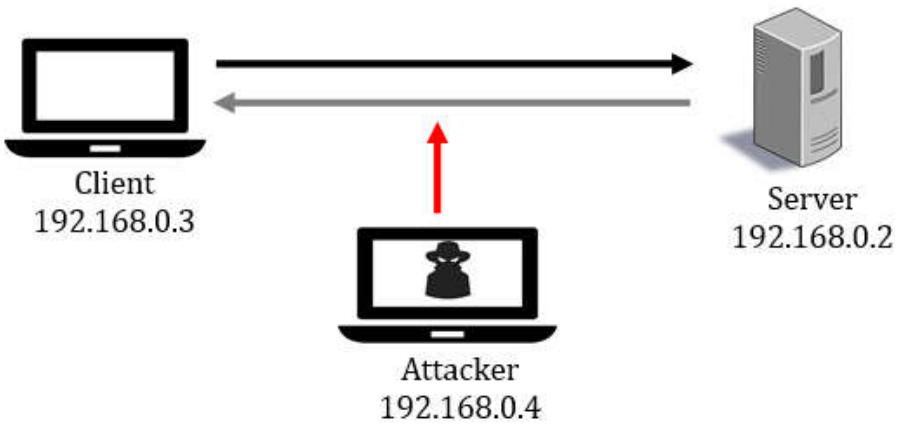

Figure 6. Sniffing Scenario

Client uses IP address 192.168.0.3 and server has IP address 192.168.0.4. Attacker must be in a network in order to see the communication traffic between the client and the server, where the attacker has the IP address

192.168.0.4. Figure 7 is a view of communication traffic between the client and the server at wireshark.

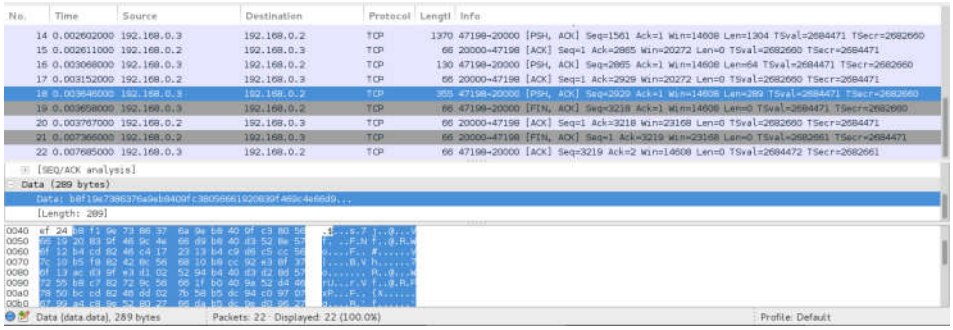

Figure 7. Traffic on Wireshark

The transmitted data on wireshark is ciphertext. Original data is will be encrypted on Figure 8.

```
Write the message: Security is a major concern of the internet world be
cause the development of the Internet requires the security of data tra
nsmission. The security method helps us to store valuable information a
nd send it over an insecure network so that it can not be read by anyon
e except the intended recipient. Security algorithm uses data randomiza
tion method.
```

Figure 8. Original Data

Figure 9 shows the transmitted data on hexadecimal form from the client to the server. However, the above data is not the original data because before the data transmission process is first encrypted by randomly scaling the location of the data in accordance with the location of the plaintext and ciphertext bits to provide data confidentiality.

```
8C F1 93 D6 87 27 73 95 20 C3 D1 C0 8C 02 6E 91 A1 FB D1 40 8D 3F 66|.....'s. .....n....@.?f
D3 B4 CC 9D 40 DD 1E 23 18 A1 E1 82 70 D5 56 6B 92 B1 F1 D4 40 9F 3F|....@..#....p.Vk....@.?
72 5C B0 40 93 52 8D 07 7B 93 B4 40 D6 60 9C 02 62 99 B8 F1 9E 73 86|r\.@.R..{..@.`..b....s.
37 6A 9E B8 40 9F C3 80 56 66 19 20 83 9F 46 9C 4E 66 D9 B8 40 D3 52|7j..@...Vf. ..F.Nf..@.R
8E 57 6F 12 B4 CD 82 46 C4 17 23 13 B4 C9 D6 C5 CC 56 7C 10 B5 F8 82|.Wo....F..#......V|....
42 8C 56 68 10 B8 CC 92 E3 8F 37 6F 13 AC D3 9F E3 D1 02 52 94 B4 40|B.Vh......7o.......R..@
D3 D2 8D 57 72 55 B8 C7 82 72 9C 56 66 1F B0 40 9A 52 D4 46 78 50 BC|...WrU...r.Vf..@.R.FxP.
CD 82 46 DD 02 7B 58 B5 DC 94 C0 97 07 67 99 A4 C8 9E 52 80 27 66 DA|..F..{X......g....R.'f.
B5 DC 9E D0 96 27 6E DE 20 C1 9F 42 80 4F 6A 9E B0 40 9A C6 80 3F 72|.....'n. ..B.Oj..@...?r
D9 A8 50 92 E3 80 27 67 D3 B4 C9 D6 C5 9C 02 66 D9 B8 ED 9F C5 CD 02|..P...'g.......f.......
7A 5F 20 E4 9A 50 96 02 6F 18 20 C9 92 E3 80 3E 6F D8 20 C8 94 C0 87|z_ ..P..o. ....>o. ....
17 6A 18 20 C8 D8 C0 8C 3E 7E 1F B1 F1 82 52 C6 0F 6B 90 B8 40 D6 60|.j. ....>~....R..k..@.`
9C 02 6E 1E B8 E1 9F 42 9C 16 23 12 B4 C9 9A C4 CC 17 67 D8 31 50 53|..n....B..#.......g.1PS
D2 8D 57 72 55 B8 C7 82 50 D4 1F 6F D2 A5 E4 9A 72 80 57 7A 59 AC 50|..WrU...P..o....r.WzY.P
96 50 96 07 23 12 A4 EA 96 73 DC 27 76 51 B8 C3 9F E3 80 37 6B 98 A1|.P..#....s.'vQ.....7k..
EB 94 63 80 00 00 00 00                                             |..c.....
```

Figure 9. Ciphertext on Hexadecimal and Text Form

b. Attacker Act as Fake Client

This scenario illustrates the attacker act as a fake client, where the attacker pretends to be a client using spoofing methods. IP spoofing is a method of hiding IP addresses by creating IP packets that contain fake IP addresses in an attempt to emulate other connections and hide identities

when you send information. Figure 10 illustrates the attacker scenario as a fake client.

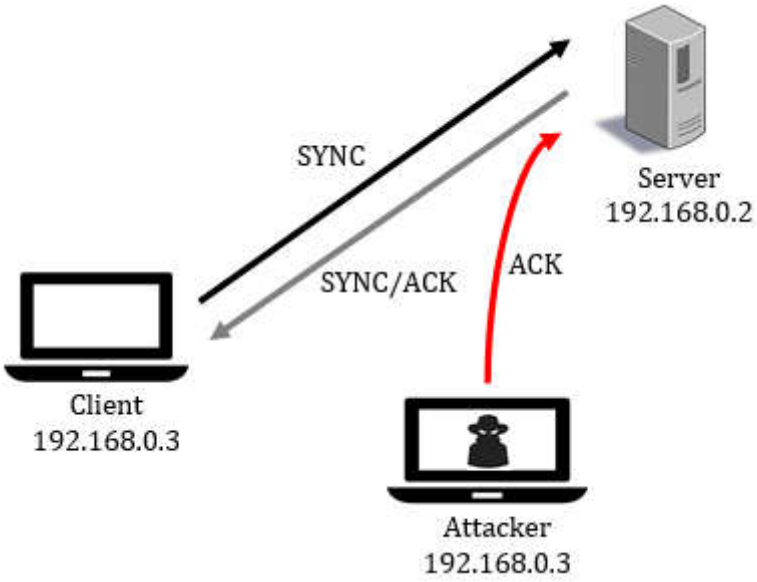

Figure 10. Illustration of Attacker as Fake Client

Assume the fake client knows the encryption algorithm used but the fake client does not know the prime number and the asymmetric key. The algorithm in this scenario uses RSA 64bit for synchronization between transmitter and receiver. Original data encrypted is shown in Figure 8. The prime numbers generated by the fake client are different from the server, so the output of ciphertext decryption in the server does not produce the same information as the information which generated in the fake client. Figure 11 shows the results of decryption on the server.

```
25 B5 75 67 2C A3 4F E2 43 91 44 CB 20 A6 0E 24 77 D1 64 CB 4C AA 4E   %.ug,.O.C.D. ..$w.d.L.N
6D 68 D7 C6 4B 58 0A 8F 50 71 81 25 63 59 C6 8F 61 7D 91 26 CB 5C 8B   mh..KX..Pq.%cY..a}.&.\.
4E DA 68 90 55 6B 08 A6 4F E1 68 94 06 EB 11 A2 0E 30 6D B1 B7 63 C5   N.h.Uk..O.h......0m..c.
87 4E 73 68 B0 D6 23 E0 E2 0E 54 43 11 D6 6F 10 EA 0E 7C 68 B0 55 EB   .Nsh..#...TC..o...|h.U.
40 C7 4F 45 69 C7 14 67 40 22 CF 41 69 D5 06 EF E0 C6 86 C4 7C C5 34   @.OEi..g@".Ai.......|.4
6B 40 C6 06 40 68 F3 14 6B ED A7 4F 45 67 B5 D6 23 B9 82 8C A2 68 94   k@..@h..k..OEg..#....h.
55 EB 68 E2 4E CA 6B A3 15 6B 51 E2 0E 57 68 90 95 63 10 C6 86 C8 69   U.h.N.k..kQ..Wh..c....i
E7 14 6F 18 86 8F 98 7C D7 06 43 38 A3 4F 74 60 D5 97 23 04 82 4E 7D   ..o....|..C8.Ot`..#..N}
7C D7 87 63 34 87 4E 7F 41 91 D6 23 00 CE 4E 73 68 90 94 27 64 AA 4E   |..c4.N.A..#..Nsh..'d.N
F8 64 B0 14 23 A5 A2 4F 6D 69 D5 06 EF B0 A2 0E 3C 69 F3 E6 6F A8 A6   .d..#..Omi......<i..o..
8E 9B 40 93 A5 63 10 87 0F 14 41 D1 14 23 E5 8E 0F 7C 40 D1 06 43 68   ..@..c....A..#...|@..Ch
87 4E 50 40 D1 84 CB 64 AE 0E D7 7D 81 35 63 00 8F CF 60 68 B0 06 EB   .NP@...d...}.5c...`h...
11 86 0E 17 69 B1 F6 6B 50 02 0F 41 69 D5 84 6F 60 82 CF 7C 5C 90 51   ....i..kP..Ai..o`..|\.Q
EB 68 E2 4E CA 6B A3 05 63 50 8E CF 6D 70 97 B5 23 41 E6 4E D8 64 B4   .h.N.k..cP..mp..#A.N.d.
07 63 10 03 4F 41 62 D5 37 6B 95 A2 CE CC 6B B1 D6 23 E5 86 4F 70 73   .c..OAb.7k....k..#..Ops
D1 36 02 81 02 00 00 00                                                .6......
```

Figure 11. The Output of Decryption Process in Server.

c. Attacker Act as Fake Server

The third scenario illustrates the attacker act as a fake server. Attacker pretends to be a server using spoofing method. Figure 12 illustrates the attacker scenario as a fake server.

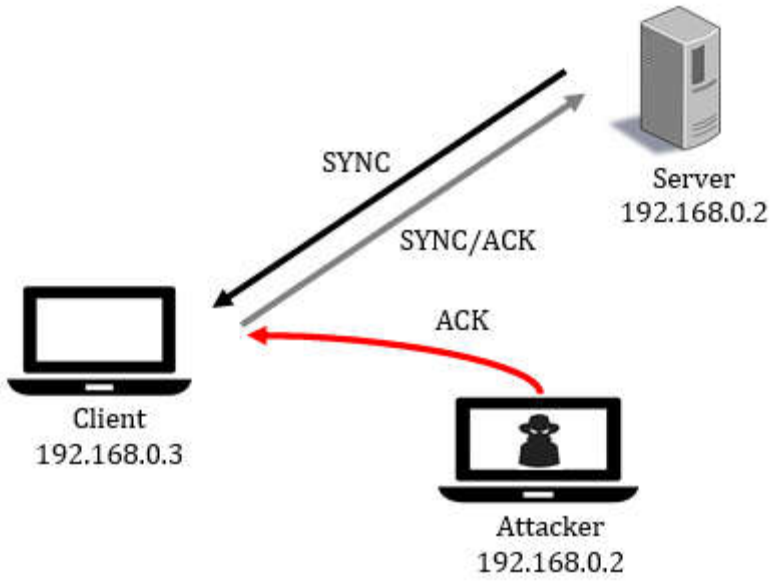

Figure 12. Illustration of Attacker as Fake Server

A legitimated client sends the ciphertext on the fake server because the fake server successfully uses the server IP as a false identity. The fake server knows the encryption algorithm used but the fake server does not know the prime number and the asymmetric key to do the decryption process. The algorithm in this scenario uses RSA 64bit for synchronization between transmitter and receiver. The original data encrypted is shown in Figure 8. The asymmetric key generated by the fake client does not match the prime number on the legitimate client, so the ciphertext decryption received by the fake server does not produce the same information as the information generated on the client. Figure 13 shows the result of decryption on the fake server.

```
86 CF 0A D5 BB 56 53 33 2A 89 08 E5 14 44 50 B5 BA ED 08 65 37 D5 54  .....VS3*....DP....e7.T
B9 A1 AB B8 45 35 E0 74 52 B2 C9 00 17 F1 64 74 3E B3 CD 80 65 37 D7  ....E5.tR.....dt>...e7.
77 50 A1 88 08 55 BD 54 52 3F A1 8A 80 77 14 44 70 71 A7 CD A1 57 9B  wP...U.TR?...w.Dpq...W.
56 55 7C A5 88 A9 D5 49 44 54 D1 2A 09 AA 55 5D C5 70 F1 A5 88 08 75  VU|....IDT.*..U].p....u
DD 56 54 9E A3 AB 12 15 19 70 54 1B A3 AB 83 F5 55 64 16 94 B1 EF 00  .VT......pT.....Ud.....
15 5D 44 14 14 A5 A9 11 D7 3F 56 54 9F AE 8F A9 D7 28 24 73 30 A1 8A  .]D......?VT.....($s0..
08 F5 FD 55 57 11 AF 89 10 17 DD 44 75 D9 A1 88 20 55 D9 65 32 14 A7  ...UW......Du... U.e2..
AB 12 15 3C 65 72 56 B1 AF 90 C5 31 56 70 F3 A0 AB A0 55 8B 55 50 F8  ...<erV....1Vp....U.UP.
B1 AF B0 D5 93 57 71 FC 22 89 A8 55 49 D4 51 7C A1 88 22 D5 0B D5 56  .....Wq."..UI.Q|.."...V
71 A4 8C 01 D7 0B 55 50 BB A3 AB 83 F5 14 45 70 F1 A7 E9 BB D5 34 65  q.....UP......Ep.....4e
53 5D 20 C9 30 55 90 46 70 D6 22 A9 01 D7 0B C5 54 F6 20 A9 80 C5 31  S] .0U.Fp."......T. ...1
56 54 54 20 A9 20 E5 17 C4 57 DD B3 CD 00 15 99 F6 50 36 A5 88 80 77  VTT . ...W.......P6...w
14 44 71 DC A7 C9 A8 55 1D 40 74 1A A3 AB 22 D5 15 75 54 F2 31 84 08  .Dq....U.@t..."..uT.1..
F5 FD 55 57 11 AF 89 10 15 91 F5 74 BE B0 CB 30 57 C9 55 56 55 A4 8E  ..UW.......t...0W.UVU..
80 55 91 52 70 1A A8 EB 81 57 9F 75 72 91 AF 89 A9 D7 0B 54 54 76 BA  .U.Rp....W.ur......TTv.
E9 81 46 08 00 00 00 00                                              ..F.....
```

Figure 12. The Output of Decryption Process in Fake Server.

6. CONCLUSION

The security method in this research uses the process of randomizing the data location. We propose a new synchronization technique using a asymmetric key algorithm. The synchronization technique utilizes the plaintext random number and the ciphertext of its. The location change of the plaintext and ciphertext bits represents the location of the scrambled data. From the results obtained can be concluded that the number of bits used for synchronization is greater, then the encryption time is shorter because the number of sections less. This is also due to small public key numbers. The encryption time require 2 ms using 1024 bits for synchronization technique asymmetric key algorithm when data 50 bytes are transmitted. Decryption time contrary to encryption time, the bit used for synchronization is bigger then the decryption time is also bigger. The decryption time gets larger as the number of bits gets larger because the asymmetric key number is very large for decrypting ciphertext. The advantage of this method is the plaintext generated every message sent so that the ciphertext is always different, resulting in the result of randomization of data positions always changing. In the future work, this security method added digital signature after the data scrambling process to make the system more secure.

Acknowledgements

This research was partially supported by 2018 Program of Ristek-Dikti Indonesian Government Grant.

REFERENCES

[1] W. Stallings, **Crypto. and Network Security: Principles and Practice, 5th edition**, Prentice Hall, 2010.

[2] B. Aiden A., and M. A. Forcinito. ***Cryptography, information theory, and error-correction: a handbook for the 21st century***. Vol. 68. John Wiley & Sons, 2011.

[3] R. L. Rivest, A. Shamir, and L. Adleman**. "A method for obtaining digital signatures and public-key cryptosystems."** *Communications of the ACM* 21.2 (1978): 120-126.

[4] S. Gurpreet. **"A study of encryption algorithms (RSA, DES, 3DES and AES) for information security."** *International Journal of Computer Applications* 67.19 (2013).

[5] P. Madhumita. **"Performance analysis of encryption algorithms for security."** *Signal Processing, Communication, Power and Embedded System (SCOPES), 2016 International Conference on*. IEEE, 2016.

[6] M. B. Kumar, D. Bhattacharyya, and S. Kumar Bandyopadhyay. **"Designing and performance analysis of a proposed symmetric cryptography algorithm."** *Communication Systems and Network Technologies (CSNT), 2013 International Conference on*. IEEE, 2013.

[7] A. Jamal Abelfatah Morad. **"A Randomized Encryption Scheme."** *Computational Science and Computational Intelligence (CSCI), 2015 International Conference on*. IEEE, 2015.W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

[8] S. Souvik, and M. Sen. **"Encoding algorithm using bit level encryption and decryption technique."** *Computer, Electrical & Communication Engineering (ICCECE), 2016 International Conference on*. IEEE, 2016.

[9] Aiswarya, P. M., et al. **"Binary RSA encryption algorithm."** *Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2016 International Conference on*. IEEE, 2016.

[10] A. I. George, and H. M. Leena. **"Enhanced RSA Algorithm with Varying Key Sizes for Data Security in Cloud."** *Computing and Communication Technologies (WCCCT), 2017 World Congress on*. IEEE, 2017.

[11] Karthik et al. **"Hybrid Cryptographic Technique Using OTP:RSA."** *International Conference On Intelligent Techniques In Control, Optimization And Signal Processing 2017 International Conference on*. IEEE, 2017.

[12] C. Punit, et al. **"ACAFP: Asymmetric key based cryptographic algorithm using four prime numbers to secure message communication. A review on RSA algorithm."** *Industrial Automation and Electromechanical Engineering Conference (IEMECON), 2017 8th Annual*. IEEE, 2017.

[13] R.P. Hudhajanto, I.G.P. Astawa, and A. Sudarsono. **"Covert Communication in MIMO-OFDM System Using Pseudo Random Location of Fake Subcarriers."** *EMITTER International Journal of Engineering Technology* 4.1 (2016): 150-163.

[14] C. Aumüller, P. Bier, W. Fischer, et al **"Fault attacks on RSA with CRT: Concrete results and practical countermeasures."** *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, Berlin, Heidelberg, 2002.

[15] A. Berzati, C. Canovas-Dumas., L. Goubin, **"A survey of differential fault analysis against classical RSA implementations."** *Fault Analysis in Cryptography*. Springer, Berlin, Heidelberg, 2012. 111-124.

[16] R. Bhaskar, G. Hegde, and P. R. Vaya. **"An efficient hardware model for RSA Encryption system using Vedic mathematics."** *Procedia Engineering* 30 (2012): 124-128.

[17] J. H. Hong, and C. W. Wu. **"RSA public key crypto-processor core design and hierarchical system test using IEEE 1149 family."** *National Tsing-Hua University, Taiwan, Doctoral dissertation* (2000).

[18] https://www.di-mgt.com.au/bigdigits.html.