Jurnal Nasional Informatika dan Teknologi Jaringan
E-ISSN : 2540-7600
P-ISSN : 2540-7597

HYBRID CHRIPTOGRAPHY STREAM CIPHER AND RSA ALGORITHM WITH DIGITAL SIGNATURE AS A KEY

Grace Lamudur Arta Sihombing
Lecturer of Universitas Sisingamangaraja XII Tapanuli Utara
graceudur@rocketmail.com
Phone. +06285261892932

***Abstract*—Confidentiality of data is very important in communication. Many cyber crimes that exploit security holes for entry and manipulation. To ensure the security and confidentiality of the data, required a certain technique to encrypt data or information called cryptography. It is one of the components that can not be ignored in building security. And this research aimed to analyze the hybrid cryptography with symmetric key by using a stream cipher algorithm and asymmetric key by using RSA (Rivest Shamir Adleman) algorithm. The advantages of hybrid cryptography is the speed in processing data using a symmetric algorithm and easy transfer of key using asymmetric algorithm. This can increase the speed of transaction processing data. Stream Cipher Algorithm using the image digital signature as a keys, that will be secured by the RSA algorithm. So, the key for encryption and decryption are different. Blum Blum Shub methods used to generate keys for the value p, q on the RSA algorithm. It will be very difficult for a cryptanalyst to break the key. Analysis of hybrid cryptography stream cipher and RSA algorithms with digital signatures as a key, indicates that the size of the encrypted file is equal to the size of the plaintext, not to be larger or smaller so that the time required for encryption and decryption process is relatively fast.**

Keywords*— *Hybrid Cryptography, Stream Cipher, RSA, Digital Signature.

I. INTRODUCTION

Using a cryptographic algorithm can often be penetrated by a cryptanalyst easily because they use the same key for decryption and encryption. So with a hybrid cryptography can be utilized as one of the solutions for the security and comfort level transaction information. Hybrid cryptography is often used because it takes advantage of the data processing speed with a symmetric algorithm and easy transfer key using an asymmetric algorithm. This can increase the speed of transaction processing data. Hybrid cryptographic applications that exist today are generally intended for general use, or which is the mainstream computer user.

Cryptographic stream cipher algorithms and the RSA algorithm becomes an issue in research related to information security that will be developed by the researchers. Expected to increase the efficiency of information security and data.

Algorithm stream cipher is a symmetric key algorithm modern encryption method the sender and the recipient have the same key. Cryptographic algorithm operates on plaintext / ciphertext in a single bit in this case a series of bits to encrypt (1 bit each time transformation) or bytes per byte.

Security system relies on generating a stream cipher stream-bit-key (key stream). If the plant issuing flow-bit key wholly-zero then the ciphertext and plaintext in the encryption process becomes meaningless. If the plant issuing flow-bit-key with a repeating pattern of 16 bit encryption algorithm to be the same as with a simple XOR encryption that has a level of security that is nothing. But if the plants emit a stream cipher a truly random (truly random), the same encryption algorithm with a one-time-pad with a perfect level of security. In this case the flow-bit-key as long as the length of plaintext and ciphertext as an unbreakable cipher get. The more random the output produced by the plant-flow-bit key, the more difficult cryptanalyst solves ciphertext [1].

Security RSA algorithm lies in the difficulty of factoring large numbers into prime factors. Factoring conducted to obtain public and private key. During factoring large numbers into prime factors unaccounted algorithm, then during which the security of the RSA algorithm is secure and is recommended to be used in the encoding of the message [2].

RSA can be used for key distribution (including key exchange), as well as for the digital signature. Because it is the first system that can be used for key distribution and digital signatures, RSA public key cryptography into a system that is popular [3].

Each - each of these two algorithms have weaknesses, on symmetric cryptographic algorithms we find weaknesses in the key exchange, due to the use of the same key in the encryption and decryption process. While the weakness in asymmetric cryptography algorithm is the time of encryption that takes quite a long time. Therefore, one of the solutions offered are cryptosystem hybrid (hybrid cryptographic systems) that are substantially the encryption of data is done with symmetric key and decryption key to be

done with an asymmetric key. In addition, the delivery of secure data integrity and verification process is required of the data, so that the author uses digital signatures (Digital Signature) as a key to convert the digital signature into a binary number.

Hybrid cryptographic stream cipher algorithm and uses RSA digital signatures as a key, and Blum Blum Shub generator that serves to lock and to determine the value of p, q on the RSA algorithm. And will provide some of the benefits that can improve information security and efficiency data.

II. MATERIALS AND METHODS

A. Stream Cipher Algortihm

Stream cipher is one kind of modern cryptographic algorithms that encrypt the plaintext into ciphertext bit by bit or byte by byte [4].

Stream cipher or stream cipher is a symmetric key cipher that uses a key stream to then combined with the plaintext to produce a ciphertext. Operations that are commonly used in stream cipher is the XOR operation. Stream cipher algorithm first introduced by Vernam algorithm through which he created, namely Vernam Cipher.

Of the above scheme can be obtained information that the security of the stream cipher key stream is totally dependent on being used. Keystream in the stream cipher key stream generated by the generator. Therefore, the keystream generator is the most important element in stream ciphers.

Generally, there are three cases of keystream generated by a keystream generator, namely:

1. Total: 0
2. Repeats periodically
3. Completely random

If the keystream produced entirely is 0, then the cipher stream becomes useless because there will be no changes, in other words the same plaintext to ciphertext ,

Meanwhile, if the keystream generated by a keystream generator periodically repeated, then the encryption algorithm will do the same with ordinary XOR algorithm that has a low level of security. But if the key stream generator can output keystream are truly random, then the algorithm used has a perfect safety level. Plaintext length equal to the length keystream completely random the so-called unbreakable cipher.

The problem is the key generator may not generate keystream truly random . It's smart to make pseudorandom functions are complex and difficult to predict. The better the pseudorandom functions are used increasingly random keystream generated , the more difficult it is to be solved by the cryptanalyst .

To generate the keystream, keystream generator requires a key U. During evoke keystream, keystream generator will use the U key obtained to randomize the bits or bytes that will be issued for each step so as to produce a keystream in the form of a stream of bits / byte random. U lock is the one that will be owned also by the recipient to generate the same keystream keystream used to scramble data.

In the process of generating the keystream in the keystream generator typically there is an Internal State which, when combined with the U and the function key output will produce bits or bytes that are pseudorandom. This process is repeated each time along the plaintext to be encrypted. Some of the advantages of stream cipher of which is executed by the hardware speed, simple to make, and the ability to encrypt the plaintext length is not known in advance.

Several new stream cipher is devoted to the implementation of the software, for example, is RC4 and SEAL. In addition, the stream cipher is also used in wireless connection because it can encrypt the plaintext with a length that is not yet clear.

$$c_i = (p_i + k_j) \bmod 2$$

which in this case,

p_i: bit plaintext
k_j: bit key
c_i: bit ciphertext

Plaintext is obtained by performing a summation modulo 2 one bit *ciphertext* with the key bit:

$$p_i = (c_i + k_j) \bmod 2$$

Therefore the *stream cipher* is an approximation of the *unbreakable ciphers.*

Modulo 2 summation operation is identical to the operation of the bits with XOR operator, the equation:

encryption:

$$c_i = p_i \oplus k_i$$

decryption:

$$p_i = c_i \oplus k_i$$

In *a stream cipher,* bits have only two values, so that the encryption process causes only two circumstances in bits. Changed or not changed. The two states are determined by the encryption key called *keystream. Keystream* generated from a plant called *keystream generator.*

1) *Types Stream Cipher*

Stream ciphers can be classified into two types based on the internal status which serve as the basis for generating a flow-key [1].

a. Synchronous stream cipher

This is the kind of cipher stream where the flow-independent key from the plaintext and ciphertext, and a stream of key bits be XOR with the plaintext (for encryption) or ciphertext (for decryption). Then, the change in status is not affected by the ciphertext plaintext and ciphertext message. Because combined with XOR operator, then the cipher of this type is also called additive stream ciphers. Mathematically encryption can be expressed as follows:

$$\varphi_{i+1} = f(\varphi_i, U)$$

$$k_i = g(\varphi_i, U)$$

$$c_i = p_i \oplus k_i$$

Both sender and receiver must be synchronized in the sender and receiver of the message because of bits generated key can not be repeated again. Both sender and receiver must have the same key and operates on the same status decryption perfect running order. If synchronization is lost, for example, bits of the ciphertext is lost during transmission, the decryption missing. To get back in sync so special techniques do. For example re-initialization , placing special objects at regular intervals in the ciphertext

b. Attacks on Stream Cipher

- Known-plainttext attack

Attacks that do cryptanalyst who have pieces of the corresponding plaintext and ciphertext , then he can find part - stream corresponding to the key XOR the bits of plaintext and ciphertext.

Example:
Suppose pieces 01100101 plaintext encrypted with chunks of 00,110,101 flow-bit key.

	P	01100101	character ‘e’)
K		00110101 ⊕	(character ‘5’)
C		01010000	(character ‘P’)

Suppose 01100101 cryptanalyst find pieces of plaintext and corresponding ciphertext, 01010000, cryptanalyst can deduce from two key this information:

C	01010000	(character ‘e’)
P	01100101 ⊕	(character ‘5’)
K	00110101	(character ‘P’)

So, the key is deduced the same as the original encryption key 00110101.

- Ciphertext-only attack

These attacks occur when the same key stream used twice against plaintext berbeda.Serangan pieces of this type is also called keystream reuse attack. For example cryptanalyst has two pieces of different ciphertext (Ci and C2) that is encrypted with the bits of the same key .

$$C_1 \oplus C_2 = (P_1 \oplus K) \oplus (P_2 \oplus K)$$
$$= (P_1 \oplus P_2) \oplus (K \oplus K)$$
$$= (P_1 \oplus P_2) \oplus 0$$
$$= (P_1 \oplus P_2)$$

If P_1 and P_2 are unknown, two XOR plaintext happened with one another can be determined by using a statistical value of the message. For example in the English text, two spaced happened XOR, or a space by the letter 'e' is often Munci, etc. Cryptanalyst smart enough to deduce both the plaintext.

example:

P_1	01100101		(character ‘e’)
K	00110101	⊕	(character ‘5’)
C_1	01010000		(character ‘P’)
P_2	01000010		(character ‘B’)
K	00110101	⊕	(character ‘5’)
C_1	01110111		(character ‘w’)

$P_1 \oplus P_2$ = 01100101 ⊕ 01000010 = 00100111

$C_1 \oplus C_2$ = 01010000 ⊕ 01110111 = 00100111

So, $P_1 \oplus P_2 = C_1 \oplus C_2$

When criptanalys has C1 and C2 , the results are both equal to two pieces of plaintext one another . If P1 and P2 are known, then XOR the plaintext to cipher text corresponds to acquire K. So that the user must have stream cipher key bits that can not be predicted so that you know part of the key bits do not allow the cryptanalyst can deduce the remaining sections.

- Flip-bit attack

These attacks are not aimed at finding the keys or reveal the plaintext and ciphertext, but change certain bits of the ciphertext decryption so that the results changed. Modifier key messages do not need to know, he just needs to know the position of the message that are of interest only. The conversion is done with reverse (flip) certain bits (0 to 1 or 1 to 0).
example:
Plaintext : QT-TRNSFRUS$00,010.00 FRMACCNT
Ciphertext : uthrojLmkyR3j7 U ukdhj38lkkldkytr
Tapper observed that the value of money associated with the character of **U** (in bold). In the course of a message in reverse (flip) a bit 1of character U becomes 0:
00101101
↓ Flip low-bit
00101100
The result is the character of U (00101101) turned into T (00101100)
Ciphertext: uthrojLmkyR3j7 T ukdhj38lkkldkytr #) okRgh
Then the decryption results:
Plaintext: QT-TRNSFR US $ 10,010.00 FRMACCNT
1 bit error in the ciphertext produced only 1 bit error in plaintext encryption results.

B. RSA Algorithm

The RSA algorithm is one of the popular public key algorithms and are still used today . The strength of this algorithm lies in the exponential process and factoring numbers into two primes [5].

This algorithm named after its inventors, Ron Rivest, Adi Shamir and Adleman (Rivest-Shamir-Adleman), published in 1977 at MIT, said the challenges given Diffie Hellman key exchange algorithm.

RSA is an asymmetric cryptographic algorithms, where the key used to encrypt different from that used to decrypt. The key used to encrypt the so-called public key, and is used to decrypt the so-called private key. RSA is one of the cryptographic algorithm that uses the concept of public key cryptography. RSA requires three steps in the process, namely

a. key generation
b. encryption
c. Decryption

1) Extended Euclidean Algorithm

On the basis of the extended euclidean previous theorem, developed an algorithm (called the Euclidean algorithm) to find gcd of two integers.

example:

a = 80, b = 12, and is filled condition a ⩾ b

Calculated using the Euclidean algorithm as follows. :

So gcd (80, 12) = gcd (12, 8) = gcd (8,4) = 4

2) Congruence

Definition of Modulo Arithmetic: Given two integers a and m> 0 Operating a mod m gives the remainder when a is divided by m. Number m is called the modulus or modulo, and the results of arithmetic modulo m located in the set {0, 1, ... m-1}

Notation: a mod m = r, such that a = mq + r, with 0 ≤ r <m

example:

23 mod 5 = 3

-41 Mod 9 = 4

Congruent Definition:

Given two integers a and b, and m is the number> 0, then a ≡ b (mod m) if m exhausted divide a - b [6].

3) Primes

Positive integer p (p> 1) is called a prime number if the denominator is only 1 and p. Numbers other than primes called a composite number.

The Fundamental Theorem of Arithmetic:

Every positive integer greater than or equal to 2 can be expressed as the product of one or more primes.

example:

91 = 7 × 13

100 = 2 × 2 × 5 × 5

There are many methods that can be used to test whether a number is prime or not . One of them is the theorem Eratosthenes .

4) RSA Key Generation

The measures used to generate the key pair in the RSA (Kim S., 2013)

1. Choose any two prime numbers p and (Hide p & q).
2. Compute n = p * q, with p ≠ q . n is not a secret.
3. Calculate φ (n) = (p-1) * (q -1)
4. Select the public key e, which is relatively prime to φ (n) or GCD (e, φ (n)) = 1 where e ≠ (p-1) e ≠ (q -1).
5. Generate a private key d by congruence ed Ξ 1 (mod φ (n).

The results of the above algorithm is:

1. The public keys (n, e)
2. The private key (d)

5) Encryption and Decryption Algorithm

The RSA encryption algorithm is as follows:

1. Take the public key of the recipient of the message (n and e).
2. Broke plaintext into blocks m_1 , m_2 , ... , such that each block represents a value in the interval [0, n-1].
3. Each block is encrypted into blocks mi c_i with the formula $c_i = p_i^e \bmod n$

To get back plaintext, ciphertext block ci decrypted into blocks m_i with the formula $p_i = c_i^d \bmod n$.

$C = M^e \bmod n$ (encryption function)

$M = C^d \bmod n$ (decryption function)

Information :

C = ciphertext private key d =

M = Message / plaintext

n = modulo divider

e = key public

6) Examples RSA Encryption and Decryption Algorithm

The RSA algorithm is simulated in a simulation carried messaging between Anie and Bob. Anie allow Bob to send a private message (private message). In the RSA algorithm multiple-key , Alice and Bob will perform the steps on the following: [7].

1. A nie (receiver) and Bob (sender) agreed on two numbers prime as a private key of the message to be delivered. Suppose The key is worth p = 17 and q = 11.
2. Once agreed by the two prime numbers are then used to calculate the value totient with the formula n = p * q, thus obtained value : N = (17) * (11) = 187
3. The next step is to calculate the value of totient formula φ (n) = (p-1) (q-1), so that the obtained values: φ (n) = (17-1) (11-1)) = 160.

   The value of n and the value totient will be used in calculating the value of the encryption key.
4. Of the value totient obtained, then Bob can calculate the value of key encryption e used in

the program on the condition that the value 1 <e < φ (n) and also the value of e relatively prime to φ (n). It can be calculated by computing the GCD (φ (n), e) = 1.

5. The decryption key is also directly determined by both parties with the requisite formula $d = e^{-1} \bmod \varphi(n)$. Of the value of e obtained before it can be calculated value d with the following steps:
   $d = e^{-1} \bmod \varphi(n)$ ed = 1 mod 160
   $d = 7^{-1} \bmod 160$ d <160
   d = 23
   Then the public key = (7, 187) and a private key = (23, 187) The decryption key is used to return the value of the ciphertext in the form of plaintext .
6. The encryption process is a process wherein prior messages that are encoded in the form of plaintext into ciphertext . Bob beforehand will make a secret of the message text. In this case the message is to be used Lisda code. From the encryption calculation formula $C = m^e \bmod n$, then the code can be calculated ciphertext of each message.
   Messages will be sent M = GRACE
   GRACE entered into ASCII code into
   G = 71
   R = 82
   A = 65
   C = 67
   E = 69
   Plaintext $P_1 = 71$, then the value of the ciphertext message with the calculation:
   $C = M^e \bmod n$
   $= 71^7 \bmod 187$
   C = 113 converted to ASCII code is q
   Plaintext $P_2 = 82$, then the value of the ciphertext message with the calculation:
   $C = M^e \bmod n$
   $= 82^7 \bmod 187$
   C = 91 conversion to ASCII code is [
   Plaintext $P_3 = 65$, then the value of the cipher-text message with the calculation:
   $C = M^e \bmod n$
   $= 65^7 \bmod 187$
   C = 142 is converted to ASCII code Ä
   Plaintext $P_4 = 67$, then the value of the ciphertext message with the calculation:
   $C = M^e \bmod n$
   $= 67^7 \bmod 187$
   C = 67 conversion to ASCII code is C
   Plaintext $P_5 = 69$, then the value of the ciphertext message with the calculation:
   $C = M^e \bmod n$
   $= 69^7 \bmod 187$
   C = 86 conversion to ASCII code is V
   7) After getting all the ciphertext code can be assembled across the code that generates ciphertext q [Ä CV . This message will be sent to Anie, so that others will not know the true meaning of the message.

   8) Anie can get the real message by performing the decryption process. From decryption calculation formula $P = C^d \bmod n$, then the code can be calculated plaintext of each ciphertext is as follows:
   C = 113 91 142 67 86
   $C_1 = 113$ $C_4 = 67$
   $C_2 = 92$ $C_5 = 86$
   $C_3 = 142$
   The value of each plaintext:
   $P_1 = C^d \bmod n = C_1^d \bmod n$
   $P_1 = 113^{23} \bmod 187$
   $P_1 = 71$ converted to ASCII = G
   $P_2 = 91^{23} \bmod 187$
   $P_2 = 82$ converted to ASCII = R
   $P_3 = 142^{23} \bmod 187$
   $P_3 = 65$ converted to ASCII = A
   $P_4 = 67^{23} \bmod 187$
   $P_4 = 67$ converted to ASCII = C
   $P_5 = 86^{23} \bmod 187$
   $P_5 = 69$ in the conversion to ASCII = E
   After getting all plaintext can be assembled as of the entire code and produce the plaintext is GRACE

   9) From the sample obtained that Anie can reopen the message is encrypted by performing the decryption process.

III. ANALYSIS PROCESS

The algorithm used to enkripi and decryption of messages in this study is a stream cipher algorithm. And hybrid cryptography to increase security message with the RSA algorithm. The process of analysis and problem solving follow these steps:

1. Load plaintext.
2. Plaintext will be modified into in to number binary 1 and 0.
3. Encryption plaintext use algorithm stream cipher.
4. Key stream cipher obtainable with read sign digital signature.
5. Sign digital signatures are used bm is p or jpg format.
6. Bmp or jpg files converted into decimal numbers with add up imagery containing / black (colored Pixel black worth 1 and Pixel color white is 0) then converted into number binary.
7. If long plaintext more big from long key, then digit key will repeated i from early to long plaintext same with long key.
8. And if long plaintext more small from long key, then the digit key will be cut along plaintext.
9. Key stream cipher (image sign digital signature) is encrypted with RSA algorithm.
10. Key Generator : Blum-Blum Shub which will be used to get value p, q on RSA algorithm.

11. Key privat obtainable with use RSA algorithm for open key has encrypted.
12. Decryption ciphertext received with algorithm stream cipher so that back to the actual message.
13. arithmetic time process encryption and decryption
14. arithmetic big plaintext already encrypted
15. arithmetic big ciphertext that has been decrypted

A. Analysis Process Digital Signature as key

Steps to change the image of digital signatures becomes Stream Cipher key is**:**

1. Load image mark digital signature.
2. Determine value threshold for the intensity of the image (0 â € "255).
3. Determine number binary from image mentioned with method clustering. If image colored white (Pixel image is empty) then bit = 0 and if the colored image (p ixel image shows) then bit = 1
4. The resulting bit values will be used as key for algorithm stream cipher.

Examples of digital signatures is the key stream cipher:

1. Image sign digital signature

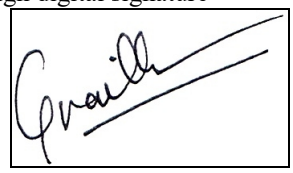

Fig. 1 Digital Signature Image

2. Determine threshold key sign digital signature (0-255) and 120 x 120 pixels.
3. Sign Digital Hands converted to number binary (white image = 0, image black = 1) and determine long key.
4. From the above it is calculated signature key length = 176, with a sequence of binary numbers = 5, 8, 12, 8, 8, 7, 12, 15 ff. This means that there are 5 lines first pixel, pixel that has an image of a black or unbiased.

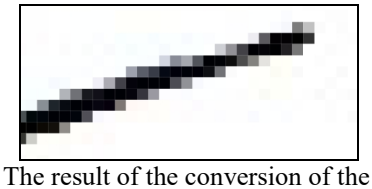

Fig. 2 The result of the conversion of the digital signature image into a binary

B. Analysis Process encryption with Stream Cipher

Steps to change the image of digital signatures becomes Stream Cipher key is:

1. Specify plaintext

Example: plaintext = GRACE

2. plaintext converted to decimal numbers in ASCII code:
   G = 71 , R = 82 , A = 65, C = 67, E = 69
3. The key length stream cipher to be taken along the plaintext

TABLE I
PLAINTEXT AND KEY STREAM CIPHER ALGORITHM

plaintext			Key	
chart	Decimal	binary	Decimal	binary
G	71	01000111	5	00000101
R	82	01010010	8	00001000
A	65	01000001	12	00001100
C	67	01000011	8	00001000
E	69	01000101	8	00001000

5. Plaintext encryption with the following equation:

TABLE II
CIPHERTEXT INTO BINARY STREAM CIPHER ALGORITHM

p_i	k_i	
1000111	00000101	01000010
1010010	00001000	01011010
1000001	00001100	01001101
1000011	00001000	01001011
1000101	00001000	01001101

6. Ciphertext produced, modified to decimal numbers

TABLE III
CIPHERTEXT CONVERTED TO DECIMAL

	Decimal
01000010	66
01011010	90
01001101	77
01001011	75
01001101	77

7. ciphertext form decimal converted to the ASCII table

TABLE IV
CIPHERTEXT CONVERTED TO ASCII TABLE

Decimal	cipher text
66	B
90	Z
77	M
75	K
77	M

C. Analysis Process Generator The value of p and q with Blum Blum shub

The following are the process steps generate value numbers p and q for the RSA:

1. choose prime numbers p and q of number random primed [1..500]
2. Value to p_1 and q_1 in Blum Blum Shub, for **p RSA:**

Values for p_2 and q_2 in the Blum Blum Shub, for **q**

From p_1 q_1 and p_2 q_2, then available Excellent value p and q for RSA

D. Analysis Process Analysis encryption and decryption key with RSA Algoritms

To secure stream cipher key, then the key must be secured with the following steps:

1. p and q
   p = 43 711
   q = 15 467
2. arithmetic value n from formula, n = pxq

   n = 43711x 15467 = 676078037

3. arithmetic m values using theoremeuler with formula, **ϕ(n) =(p–1)(q-1)**

   ϕ(n) = (p–1)(q-1)
   ϕ(n) = (43711 -1) (15467-1)
   ϕ(n) = (43710)(15466)
   ϕ(n) = 676018860

4. arithmetic key public (e), which are relatively prime to m. e relatively prime to m means factor divider most both is 1, **gcd(e,ϕ(n))=1**gcd(e,ϕ(n)) = 1
   gcd (e, 676018860) = 1
   e =7
   e = 7

5. arithmetic key private-called d way until that **e*d mod ϕ(n)=1**
   e * d mod ϕ(n)=1
   7 * d mod 676018860= 1
   d = 772592983

6. encryption key stream cipher use RSA algorithm, with equation following:
   $C = M^e \bmod n$ (encryption function)
   $M = C^d \bmod n$ (decryption function)

Key stream cipher: 5, 8, 12, 8, 8
$C_1 = 5 \bmod^7 676\,018\,860 = 78\,125$
$C_2 = 8 \bmod^7 676\,018\,860 = 2097152$
$C_3 = 12 \bmod^7 676\,018\,860 = 35{,}831{,}808$
$C_4 = 8 \bmod^7 676\,018\,860 = 2097152$
$C_5 = 8 \bmod^7 676\,018\,860 = 2097152$
Then the stream cipher key that has been encrypted into:
78 125, 2097152, 35831808, 2097152, 2097152
To decrypt the cipher text can be done with the following equation:
$M = C^d \bmod n$ (decryption function)
M_1 78 $125^{772\,592\,983} \bmod = 676018860 = 5 =$ 00000101
$M_2 = 2097152^{772\,592\,983} \bmod 676\,018\,860 = 8 =$ 00001000
$M_3 = 35831808^{772\,592\,983} \bmod 676018860 = 12 =$ 00001100
$M_4 = 2097152^{772\,592\,983} \bmod 676018860 = 8 =$ 00001100
$M_5 = 2097152^{772\,592\,983} \bmod 676\,018\,860 = 8 =$ 00001000

E. Analysis Process decryption with Stream Cipher Algorithm

Stream Cipher key is:

1. ciphertext converted to decimal numbers then continue conversion to number binary

TABLE V
CIPHERTEXT INTO BINARY STREAM CIPHER ALGORITHM

cipher text	Decimal	binary
B	66	01000010
Z	90	01011010
M	77	01001101
K	75	01001011
M	77	01001101

2. encryption plaintext with equation following this:

TABLE VI
PLAINTEXT IN A BINARY NUMBER WITH STREAM CIPHER ALGORITHM

c_i	k_i	$c_i = p_i \oplus k_i$
1000010	101	1000111
1011010	1000	1010010
1001101	1100	1000001
1001011	1000	1000011
1001101	1000	1000101

3. plaintext The resulting, Diko n version to number desimal

TABLE VII
PLAINTEXT CONVERTED INTO A DECIMAL NUMBERS

binary	Decimal
01000111	71
01010010	82
01000001	65
01000011	67
01000101	69

4. plaintext form decimal converted to the ASCII table

TABLE VIII
PLAINTEXT GENERATED WITH STREAM CIPHER ALGORITHM

Decimal	chart
71	G
82	R
65	A
67	C
69	E

IV. RESULT AND DISCUSSION

Hybrid Cryptographic Algorithm Research Stream Cipher and RSA

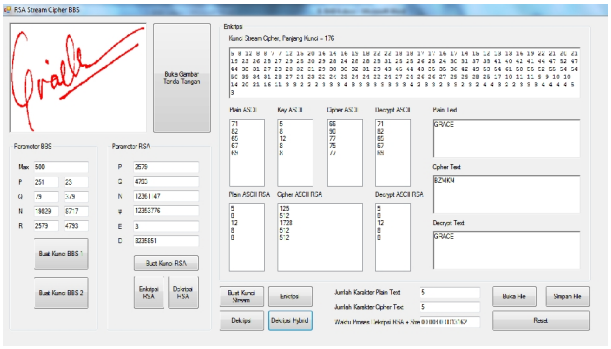

Fig. 3 Hybrid Cryptographic Algorithm Research Stream Cipher and RSA

From some experiments on the encryption process with Hybrid Cryptographic Algorithm Stream Cipher and RSA concluded that:

1. Large file cipher text generated from the encryption process is the same. Not to be smaller or bigger. Because the encryption process in hybrid cryptographic algorithm is a stream cipher and RSA encryption bit by bit, so the length of the plaintext equal to the length of cipher text.
2. The time needed for the encryption process longer than the decryption process for the encryption key generation process with no-Blum Blum Shub to get prime numbers p and q value on the RSA algorithm.
3. Generating key-Blum Blum Shub will generating prime numbers greater values of p and q of random numbers determined by that of the intervals of 1 to 500.
4. The greater the prime numbers generated by the key generating-Blum Blum Shub, the greater the value of p and q for the RSA algorithm.
5. With a stream cipher algorithm hybrid cryptography and RSA can be seen the results of that weakness stream cipher algorithm, namely the possibility of correspondence plaintext and cipher text, can be overcome with a key stream cipher itself is secured with the RSA algorithm. So it will not be easy for the cryptanalyst to find a part-stream of the key.
6. Image digital signature used as a key to generate a long row of binary numbers and it will be difficult to predict.
7. Stream cipher key is secured with RSA Encryption produce a flow-bit key that is longer, very much different from the actual stream cipher key. This certain can enhance the security of the encryption and decryption process when compared to just using the stream cipher algorithm only.

Based on the results of research and discussion can be written the following conclusion:

1. Based on tests performed concluded that time required for process encryption and decryption with hybrid cryptography algorithm stream cipher and RSA is longer if compared with process encryption and decryption that only use algorithm stream cipher. This happen because security key on process hybrid cryptography algorithm stream cipher and RSA requires time for right generation key with Primes more Great.
2. Time processing message proportionate straight with long message and key. The long message and key then time required for process encryption or decryption will longer.
3. The file size on process encryption and decryption hybrid cryptography with algorithm stream cipher is same. Because long key used on process encryption and decryption is same.
4. More and more big primes generated generator key with Blum-Blum Shub, then The resulting value for the value of p and q in RSA algorithm will more big too.
5. The more big the value of p and q in RSA algorithm then time required for produce key too will more and more Great.
6. Time required for process encryption longer if compared with process decryption. Because on process encryption need time for awaken primes and calculate key.
7. Comparison time initialization image sign digital signature with process encryption and decryption is longer. Because calculation imagery black and white on the image.
8. Advantages from hybrid cryptography algorithm stream cipher and RSA is stream cipher key encrypted RSA produce more key long. This will more complicate cryptanalyst for solve real key
9. Security use hybrid cryptography algorithm stream cipher and RSA more good because key used for encryption and decryption is different. In addition to that very difficult guessing key because amount keys sent to

receiver no same with amount character the real key.

Based on the results of research conducted, the researchers gave suggestions as follows:

1. For the development of this research should be continued with security files in a variety of formats.
2. From the results of the evaluation of the implementation of the system is done , it is advisable to conduct further research to obtain a digital signature image size so that the right image specification digital signatures as a key to more accurate .
3. It is recommended to implement the system on a computer with a larger memory and higher processor so that time encryption and decryption process is faster and is able to create an RSA key with a larger number of bits so that the key security better.
4. It is recommended to combine other algorithms to improve the security message with time encryption and decryption faster and smaller file sizes .

REFERENCE

[1] Munir, R. 2006. Kriptografi. Penerbit Informatika : Bandung.

[2] Mahajan Sonam & Sigh Maninder., 2014. Performance Analysis of Efficient RSA.

[3] Kromodimoelyo, S. 2010. Teori dan Aplikasi Kriptografi. *SPK IT Consulting.* ISBN: 78-602-96233-0-7.

[4] Mollin, Richard A. 2003. *RSA and Public-Key Cryptography*. *Discrete Mathematics And Its Applications Series Editor Kenneth H. Rosen*. Chapman & Hall/CRC. A CRC : Press Company Boca Raton London New York Washington, D.C.

[5] Menezes, A. J, Paul C. V. O. & Scott A. V,. 1996. Handbook of Applied Cryptography. *CRC Press.*

[6] Oppliger, R., 2005. *Contemporary Cryptography* (Artech House Computer Security Series). Amazon.

[7] Dooley, J.F. 2013. A Brief History Of Cryptology and Cryptographic Algorithms.