Symmetric Stream Cipher using Triple Transposition Key Method and Base64 Algorithm for Security Improvement

Heri Nurdiyanto¹, Robbi Rahim*² and Nur Wulan³

¹Department of Informatics Engineering, STMIK Dharma Wacana, Jl.Kenanga No.03 Mulyojati 16C Metro Barat, Metro Lampung, Indonesia
²School of Computer Engineering and Communication, Universiti Malaysia Perlis, Malaysia
³Department of Informatics Engineering, Sekolah Tinggi Teknik Harapan, Jln. H.M. Joni No. 70 C Medan, Medan, Indonesia

*usurobbi85@zoho.com

Abstract. Symmetric type cryptography algorithm is known many weaknesses in encryption process compared with asymmetric type algorithm, symmetric stream cipher are algorithm that works on XOR process between plaintext and key, to improve the security of symmetric stream cipher algorithm done improvisation by using Triple Transposition Key which developed from Transposition Cipher and also use Base64 algorithm for encryption ending process, and from experiment the ciphertext that produced good enough and very random.

1. Introduction
Data security is very important in maintaining the confidentiality of information, especially containing sensitive information that should only be known by a particular party, especially if the delivery is arranged through the public network, if the data is not secured it will be easy to tap and known contents Information by unauthorized parties[1][2]. One of the ways used to secure data is to use the cryptographic system by encoding the substance of the information (plaintext) into the contents that are not understood through the encryption process and to recover the original information using the decryption process by using the correct key[1].

The symmetric stream cipher method in this research uses triple transposition key and Base64 algorithm to increase the security of symmetric stream cipher algorithm. Improvisation performs on symmetric stream cipher method, in this study received two key types namely the main key and Private Crypto Code. With the result forming three transposition keys containing 463, 251, 181 loop which is interconnected due to feedback output from key processing. With the method of the symmetric stream, cipher will produce output with high speed and have Private Crypto Code.

2. Theory
A. Symmetric Stream Cipher
A stream cipher is a type of symmetry encryption algorithm that is much faster than any block cipher algorithm, and stream ciphers are used for smaller data blocks in bit sizes[3]. A stream cipher produces what is called a key stream. The encryption process is achieved by combining key stream with plaintext
usually with bitwise XOR operations and can be made independent of plaintext and ciphertext which results in what is called a synchronous stream cipher[3][4][5].

B. Triple Transposition Key Algorithm

In transposition cipher, plaintext remains the same, but the order is changed. In other words, this algorithm transfers the character set inside the text. Another name for this method is the permutation[6] because transposed each character in the text is the same as doing the character of the characters, in the results of this study the authors to improvise transposition algorithm that is named Triple Transposition Key specifically used to generate keys[6].

Triple Transposition Key is a technique used to form keys of 463, 251, and 181 which is connected because of the feedback output of key processing, the Triple Transposition Key process that the author uses is as follows:

\[
\text{EncodeByte} = \text{a byte XOR FnULTRA(Feedback)}
\]

\[
\text{Feedback} = \text{EncodeByte}
\]

Keys that have been formed then combined using the following functions:

\[
\text{Output} = (\text{OUT1} + \text{OUT2} + \text{OUT3}) \mod 256
\]

C. Base64 Algorithm

The Base64 transformation is one of the algorithms for encoding and decoding data in ASCII format, which are based on the number 64. The character generated from Base64 consists of A-Z, a-z and 0..9, plus the last two characters of / and + [7].

3. Proposed Method

In this experiment data security is perform using stream cipher algorithm, and the key utilized in the encryption process consists of the primary key and Private Crypto Code and then process with Triple Transposition Key algorithm, and next step is encryption string with Symmetric Stream Cipher, and Base64 algorithm, the use of 2 keys in the encryption process is expected to be able gives better ciphertext results.

The process of completion of cryptography method Symmetric Stream Cipher can be divide into three parts, such as:

a. Key Formation Process
b. Dummy String Formation Process
c. Encryption and Decryption Process.

For example the process of applying the Triple Transposition Key, first determines the primary key and private crypto key as follows:

**Primary Key = IconICT**

**Private Crypto Key = Harapan**

a. **Primary Key = 'IconICT'**

b. **Initialization**  

\[
\text{KEY1} (\text{KEY1} = \text{Primary Key}) = 'IconICT'
\]

\[
\text{KEY1} = (73, 99, 111, 110, 73, 67, 84)
\]

\[
\text{KEY1Len} = 7
\]

c. **Value Initialization K1(),P1,S1**

\[
\text{K1} (0..462)
\]

\[
\text{P1} = 0
\]

\[
\text{S1} = 0
\]

Change K1 (j) with K1 (i) total of 463 times

\[
\text{j} = (\text{j} + \text{K1} (\text{i}) + \text{KEY1} (\text{i} \mod \text{KEY1Len})) \mod 463
\]

\[
\text{Loop of} - 0
\]

\[
\text{j} = (0 + \text{K1} (0) + \text{KEY1} (0 \mod 7)) \mod 463
\]

\[
\text{j} = (0 + 1 + \text{KEY1} (0)) \mod 463
\]
\[ j = (0 + 1 + 73) \mod 463 \]
\[ j = (74) \mod 463 \]
\[ j = 73 \]

Exchange K1 (73) WITH K1 (0) = 0 WITH 73

The above process is done up to 462 times round so as to produce the following value, thus obtained:

\[
K1 = (73, 173, 286, 25, 104, 409, 175, 15, 347, 19, 423, 40, 459, 451, 240, 244, 7, 121, 339, 86, 214, 462, 405, 160, 242, 410, 92, 434, 416, 82, 297, 57, 113, 381, 374, 93, 193, 61, 145, 215, 305, 197, 408, 186, 218, 98, 70, 255, 64, 256, 406, 144, 310, 348, 176, 349, 20, 318, 112, 3, 170, 2, 450, 430, 325, 156, 421, 105, 166, 329, 87, 129, 427, 152, 309, 130, 81, 379, 461, 299, 277, 272, 454, 168, 363, 37, 341, 85, 124, 397, 140, 126, 71, 97, 380, 38, 420, 65, 34, 80, 275, 35, 411, 117, 278, 232, 254, 139, 108, 251, 383, 375, 169, 21, 301, 92, 434, 416, 82, 297, 57, 113, 381, 374, 93, 193, 61, 145, 215, 305, 197, 408, 186, 218, 98, 70, 255, 64, 256, 406, 144, 310, 348, 176, 349, 20, 318, 112, 3, 170, 2, 450, 430, 325, 156, 421, 105, 166, 329, 87, 129, 427, 152, 309, 130, 81, 379, 461, 299, 277, 272, 454, 168, 363, 37, 341, 85, 124, 397, 140, 126, 265, 110, 382, 328, 445, 247, 435, 172, 162, 74, 60, 271, 184, 400, 444, 330, 398, 359, 201, 220, 56, 377, 300, 32, 324, 120, 39, 414, 43, 418, 136, 59, 25, 110, 382, 328, 445, 247, 435, 1
\]

The result of K1 above is the outcome of the process of Triple Transposition Key 462, for keys with 251 and 181 also in the same way to obtain a different result, and each result will combine with function (OUT1 + OUT2 + OUT3) Mod 256 to get a key.

The second process generates the dummy string to add ciphertext security, for the dummy string used is the word HERI, the formulas and processes used for the dummy string are as follows:

a. \( \text{SeedString} = \text{HERI} = (72, 69, 82, 73) \)
Length of SeedString = 4

b. Initialize Random SizeDummy
\[ \text{SizeDummy} = \text{Int}(224*\text{rnd})+32 \]
\[ \text{SizeDummy} = 124 \]

c. Change the value of Variable Size Dummy with \( \text{SizeDummy} = \text{SizeDummy} \text{ Xor Asc(Mid(SeedString, k, 1))} \) as much as Len (seedstring) times
01. \( \text{SizeDummy} = 124 \text{ Xor 72 = 52} \)
02. \( \text{SizeDummy} = 52 \text{ Xor 69 = 113} \)
03. \( \text{SizeDummy} = 113 \text{ Xor 82 = 35} \)
04. \( \text{SizeDummy} = 35 \text{ Xor 73 = 106} \)
SizeDummy = 106

If SizeDummy > 255 then SizeDummy = SizeDummy - X * 224
If SizeDummy < 32 then SizeDummy = SizeDummy + 224
\[ \text{SizeDummy} = 106 \]

d. Forming Random Dummy with SizeDummy size – 1
\[ \text{Random Dummy} = (220, 143, 128, 6, 28, 205, 12, 204, 197, 35, 93, 30, 61, 169, 59, 115, 180, 37, 103, 87, 19, 43, 46, 178, 47, 137, 61, 77, 67, 225, 183, 146, 157, 152, 133, 147, 151, 239, 237, 79, 201, 112, 103, 48, 125, 140, 172, 215, 99, 130, 15, 28, 134, 130, 16, 81, 127, 19, 47, 199, 233, 178, \]
254, 148, 21, 188, 122, 147, 254, 142, 155, 171, 127, 234, 42, 85, 195, 156, 211, 88, 253, 94, 123, 215, 90, 250, 211, 142, 52, 174, 46, 89, 137, 212, 143, 246, 171, 194, 7, 11, 57, 240, 62, 107, 77)

e. Randomize Dummy with Triple Transposition Key 16 times

1st loop, RndKey Formation Process throughout 16 Bytes

Q = Random 0 s/d 255 = 88
Q = Q Xor SeedString(1) = 88 Xor 72 = 16
Q = Random 0 s/d 255 = 7
Q = Q Xor SeedString(2) = 79 Xor 69 = 10
Q = Random 0 s/d 255 = 149
Q = Q Xor SeedString(3) = 149 Xor 82 = 199
Q = Random 0 s/d 255 = 247
Q = Q Xor SeedString(4) = 247 Xor 73 = 190
Q = Random 0 s/d 255 = 35
Q = Q Xor SeedString(1) = 35 Xor 72 = 107
Q = Random 0 s/d 255 = 217
Q = Q Xor SeedString(2) = 217 Xor 69 = 156
Q = Random 0 s/d 255 = 125
Q = Q Xor SeedString(3) = 125 Xor 82 = 47
Q = Random 0 s/d 255 = 235
Q = Q Xor SeedString(4) = 235 Xor 73 = 162
Q = Random 0 s/d 255 = 129
Q = Q Xor SeedString(1) = 129 Xor 72 = 201
Q = Random 0 s/d 255 = 82
Q = Q Xor SeedString(2) = 82 Xor 69 = 23
Q = Random 0 s/d 255 = 144
Q = Q Xor SeedString(3) = 144 Xor 82 = 194
Q = Random 0 s/d 255 = 188
Q = Q Xor SeedString(4) = 118 Xor 73 = 63
Q = Random 0 s/d 255 = 217
Q = Q Xor SeedString(1) = 217 Xor 72 = 145
Q = Random 0 s/d 255 = 159
Q = Q Xor SeedString(2) = 159 Xor 69 = 218
Q = Random 0 s/d 255 = 242
Q = Q Xor SeedString(3) = 242 Xor 82 = 160
Q = Random 0 s/d 255 = 183
Q = Q Xor SeedString(4) = 183 Xor 73 = 254

rndKey = (16, 10, 199, 190, 107, 156, 47, 162, 201, 23, 194, 63, 145, 218, 160, 254)

The randomization process above is performe as much as 16 times and generate random dummy string as follows:

Random Dummy = (106, 20, 46, 19, 40, 23, 159, 135, 228, 243, 64, 37, 237, 84, 168, 167, 98, 67, 134, 30, 124, 163, 91, 147, 183, 7, 101, 50, 68, 75, 248, 1, 190, 227, 82, 153, 165, 168, 46, 195, 118, 85, 228, 5, 212, 36, 77, 20, 113, 237, 93, 225, 91, 159, 216, 157, 109, 199, 26, 195, 243, 55, 192, 24, 167, 196, 89, 130, 85, 152, 156, 71, 29, 162, 73, 236, 160, 188, 130, 234, 95, 237, 250, 57, 213, 126, 154, 251, 168, 196, 79, 52, 48, 127, 115, 40, 39, 33, 134, 213, 27, 203, 145, 77, 228, 110)

After determining the key with Triple Transposition Key and dummy string algorithms, next is to encrypt the process for the word "Conference", here is the process:

a. PlainText = 'Conference'
    Primary Key = 'IconICT'
    PCC Key = 'Harapan'
This information is then converted into ASCII form as follows, let call this is PPC Text: (72, 69, 51, 13, 62, 10, 0, 0, 0, 7, 0, 67, 3, 99, 3, 101, 2, 102, 3, 110, 3, 111, 3, 114, 3, 114, 60, 6, 194, 217, 186, 3)

b. Forming DummyString

SeedString = HERI

DummyString in ASCII: (121, 10, 27, 70, 38, 249, 7, 74, 181, 143, 122, 53, 250, 247, 67, 124, 125, 47, 18, 191, 169, 149, 82, 219, 190, 67, 26, 250, 158, 145, 230, 121, 151, 93, 222, 201, 73, 58, 186, 181, 162, 245, 186, 59, 97, 86, 241, 181, 88, 49, 123, 44, 171, 216, 87, 235, 103, 153, 216, 201, 168, 193, 141, 237, 213, 254, 97, 117, 171, 90, 12, 216, 186, 6, 103, 117, 191, 110, 223, 87, 246, 132, 127, 137, 119, 221, 84, 133, 133, 135, 75, 81, 82, 132, 232, 112, 123, 33, 11, 205, 250, 250, 49, 250, 100, 5, 179, 142, 168, 220, 90, 157, 211, 112, 135, 232, 77, 46, 128, 58)

c. Combine DummyString with PPCText

NewPlainText = DummyString & PPCText & (The last 2 bytes of DummyString)

NewPlainText in ASCII: (121, 10, 27, 70, 38, 249, 7, 74, 181, 143, 122, 53, 250, 247, 67, 124, 125, 47, 18, 191, 169, 149, 82, 219, 190, 67, 26, 250, 158, 145, 230, 121, 151, 93, 222, 201, 73, 58, 186, 181, 162, 245, 186, 59, 97, 86, 241, 181, 88, 49, 123, 44, 171, 216, 87, 235, 103, 153, 216, 201, 168, 193, 141, 237, 213, 254, 97, 117, 171, 90, 12, 216, 186, 6, 103, 117, 191, 110, 223, 87, 246, 132, 127, 137, 119, 221, 84, 133, 133, 135, 75, 81, 82, 132, 232, 112, 123, 33, 11, 205, 250, 250, 49, 250, 100, 5, 179, 142, 168, 220, 90, 157, 211, 112, 135, 232, 77, 46, 128, 58)

d. Forming key encryption with Triple Transposition Key

Encryption Process per Byte

Number of Bytes = 154
The 0th iteration
Process FnUltra(FEEDBACK)
Input = Feedback = 0
P1 = (P1+1) Mod 463
P1 = (0+1) Mod 463
P1 = 1
S1 = (S1 + K1(P1) + FB) Mod 463
S1 = (0 + K1(0) + 0) Mod 463
S1 = (0 + 73 + 0) Mod 463
S1 = 73
OUT1 = K1((K1(P1) + K1(S1)) Mod PR1) Mod 256
OUT1 = K1((K1(1) + K1(173)) Mod 463) Mod 256
OUT1 = K1((327 + 173) Mod 463) Mod 256
OUT1 = K1((500) Mod 463) Mod 256
OUT1 = K1(37) Mod 256
OUT1 = 80 Mod 256
OUT1 = 80

P2 = ( P2 + 1 ) Mod 251
P2 = ( 0 + 1 ) Mod 251
P2 = 1
S2 = (S2 + K2(P2) + OUT1) Mod 251
S2 = (0 + K2(0) + 80) Mod 251
S2 = (0 + 152 + 80) Mod 251
S2 = 232
OUT2 = K2((K2(P2) + K2(S2)) Mod PR2) Mod 256
OUT2 = K2((K2(1) + K2(4)) Mod 251) Mod 256
OUT2 = K2((137 + 175) Mod 251) Mod 256
OUT2 = K2(312) Mod 251) Mod 256
OUT2 = K2(61) Mod 256
OUT2 = 231 Mod 256
OUT2 = 231

The above calculation process is done by testing the New Plaintext with the key result of Triple Transposition Key, the above process is continued until all New Plaintext encryption using lock result of Triple Transposition Key, the end result of the process are:

Encrypted String in ASCII: (56, 252, 110, 156, 57, 138, 131, 13, 88, 14, 97, 227, 64, 202, 168, 159, 16, 109, 216, 74, 171, 244, 175, 67, 255, 213, 220, 218, 250, 188, 12, 146, 78, 94, 49, 59, 158, 225, 195, 97, 196, 168, 95, 188, 251, 144, 50, 146, 39, 101, 138, 214, 197, 128, 64, 228, 134, 121, 167, 210, 86, 106, 48, 73, 176, 118, 123, 39, 137, 248, 242, 223, 29, 134, 115, 39, 215, 230, 130, 242, 138, 35, 202, 239, 77, 254, 46, 149, 73, 6, 240, 115, 197, 26, 69, 56, 46, 53, 64, 209, 50, 47, 164, 7, 12, 79, 73, 209, 51, 180, 201, 32, 27, 117, 67, 251, 50, 0, 248, 137, 178, 93, 125, 107, 198, 105, 20, 123, 55, 243, 206, 75, 156, 26, 45, 236, 140, 28, 59, 51, 91, 162, 32, 41, 209, 251, 126, 188, 6, 180, 51, 231, 27, 46, 137

Ciphertext in ASCII is then encoded with Base64 with the following results= OPxunDmKgw1YDmHjQMqonxBt2Eqr9K9D/9Xc2vq8DJOXjE7nuHDYcSoX7zKDKS2W1k1sW AQOSGiaSVMowSB2eyeJ+PfHyZzJ9fmgvKK8rvTF4uIlkG8HF9kU4LjVA0TivpAcMT0nR M7TJjBt1Q/syApJsl19a8ZpFHs3885LnBot7lwOzNbooQp0ft+vAA0M+cbLok

4. Conclusion

Conclusion on the results of this study, the addition of security to Symmetric Stream Cipher with Triple Transposition Key and Base64 produce a ciphertext that is very random and not easily penetrated even with some techniques commonly used by cryptanalyst because the encryption process has some fairly complicated process let alone the key used Raised with the Triple Transposition Key used by the author.

References

[1] R. von Solms and J. van Niekerk, “From information security to cyber security,” Comput. Secur., vol. 38, no. October 2013, pp. 97–102, 2013.
[2] W. Stallings, Network security essentials : applications and standards. 2011.
[3] C. De Canniere, “Trivium: A stream cipher construction inspired by block cipher design principles,” Inf. Secur., vol. 507932, pp. 171–186, 2006.
[4] S. Manna and S. Dutta, “A Stream Cipher based Bit-Level Symmetric Key Cryptographic Technique using Chen Prime Number,” Int. J. Comput. Appl., vol. 107, no. 12, pp. 975–8887, 2014.
[5] M. I. Alam and M. R. Khan, “Performance and Efficiency Analysis of Different Block Cipher Algorithms of Symmetric Key Cryptography,” Int. J. Adv. Res. Comput. Sci. Softw. Eng., vol. 3, no. 10, pp. 2277–128, 2013.
[6] Q.-A. Kester, “A HYBRID CRYPTOSYSTEM BASED ON VIGENÈRE CIPHER AND COLUMNAR TRANSPOSITION CIPHER,” Int. J. Adv. Technol. Eng. Res., vol. 3, no. 1, pp. 141–147, 2013.
[7] A. T. SHOLEH, E. GUNADHI, and A. D. SUPRIATNA, “Mengamankan Skrip Pada Bahasa Pemograman Php Dengan Menggunakan Kriptografi Base64,” J. Algorithm. Sekol. Tinggi Teknol. Garut, vol. 10, no. 1, pp. 1–9, 2013.