Modification of variably modified permutation composition (vmpc) algorithm genetic key algorithm for data security

E S Ompusunggu[1], sawaluddin[2], E B Nababan[3]

[1]Student, Department of Information Technology, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Indonesia
[2]Department of Information Technology, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Indonesia
[3]Department of Information Technology, Faculty of Computer Science and Information Technology, Universitas Katolik Santo Thomas Medan, Indonesia

Email: sastraelvis@gmail.com[1], sawaluddin@usu.ac.id[2], ernastp@yahoo.com[3]

Abstract. Cryptography is the study of mathematical techniques related to information security aspects such as validity, data integrity, and data authentication. In this study, Variably Modified Permutation Composition (VMPC) algorithm is modified by adding complementary methods to the encryption and decryption process. Keys that are generated randomly will be optimized using Genetic Algorithm. The Variably Modified Permutation Composition (VMPC) algorithm is a symmetrical stream cipher algorithm similar to the RC4 cipher designed by Bartosz Zoltak. The Variably Modified Permutation Composition (VMPC) algorithm is an extension of the one-way function VMPC that was developed into a byte-based encryption algorithm. In its use the VMPC is generated by an 8-bit stream of 256 element permutations. The initial state of the permutation is calculated in VMPC Key Scheduling. The Genetic Algorithm produces a unique intermediate key for each algorithm run. The intermediary key is combined with the Ciphertext in the first level which produces the second level Ciphertext, up to the third level Ciphertext. Attackers will not be able to carry out attacks such as brute force, differential attacks or statistical attacks without having knowledge of the key. From the results of the tests carried out, the more characters that are encrypted and decrypted, the longer it will take. On testing the avalanche effect produces an average value of 51.25% for keys without optimization while 53.42% for keys optimized. The effect of key optimization using Genetic Algorithm has increased the value of the avalanche effect and has an effect on the results of changing bits in the ciphertext.

1. INTRODUCTION

Security is a big problem and securing important data is very important, so that the data cannot be tapped or misused for illegal purposes that can harm others. Therefore, many methods are carried out to make data secure from people who are not responsible at the time of data exchange. Data sent is converted into data that cannot be read by the hijacker then the data is changed back in a form that can be read by the recipient. Techniques and knowledge for making data that cannot be read so that only authorized people are able to read data, this is what is called cryptography [1].

The Variably Modified Permutation Composition (VMPC) algorithm is a symmetrical stream cipher algorithm similar to the RC4 cipher designed by Bartosz Zoltak. The Variably

Modified Permutation Composition (VMPC) algorithm is an extension of the one-way function VMPC that was developed into a byte-based encryption algorithm. In its use the VMPC is generated by an 8-bit stream of 256 element permutations. The initial state of the permutation is calculated in VMPC Key Scheduling. The Genetic Algorithm produces a unique intermediate key for each algorithm run. The intermediary key is combined with the Ciphertext in the first level which produces the second level Ciphertext, up to the third level Ciphertext [2]. Attackers will not be able to carry out attacks such as brute force, differential attacks or statistical attacks without having knowledge of the key [2]. Based on the description, the authors are interested in conducting research by modifying the Variably Modified Permutation Composition (VMPC) algorithm by adding the operation of the Complement method to the encryption and decryption process by optimizing the random key using the Genetic Algorithm approach.

2. PROPOSED METHOD

In the process of modifying the Variably Modified Permutation Composition (VMPC) algorithm, the key is generated randomly and then optimizes the key using the Genetic Algorithm which produces a binary key used for the encryption process. The encryption results then undergo a crossover process and mutation process. Furthermore, the results of the process again undergo a process, namely the complement process or the process of adding binary values to produce ciphertext. The decryption process also uses the same key and undergoes the same crossover process and mutation process and experiences the complement process or the process of adding the same binary value to return the plaintext.

2.1. Variably Modified Permutation Composition (VMPC)

Variably Modified Permutation Composition (VMPC) algorithm is a stream cipher similar to the RC4 cipher designed by Bartosz Zoltak. The Variably Modified Permutation Composition (VMPC) algorithm is an extension of the one-way function VMPC that was developed into a byte-based encryption algorithm. In its use the VMPC is generated by an 8-bit stream of 256 element permutations. The initial state of the permutation is calculated in the VMPC Key Scheduling Algorithm. he resulting values must be XOR with the plaintext to get the ciphertext [3]. The VMPC algorithm has two main parts, namely Key Scheduling Algorithm (KSA) and Pseudo-Random Generation Algorithm (PRGA). VMPC Pseudo-Random Generation Algorithm (PRGA) is run after the VMPC Key Scheduling Algorithm (KSA) is executed first.

2.1.1. Key Schedulling Algorithm (KSA)

The VMPC Key Scheduling Algorithm (KSA) algorithm changes keys and additional options for initializing vectors to permutations of 256 elements P and initializes variables s [3]. The scheme of the Key Scheduling Algorithm (KSA) without using additional options for vector initialization is shown as follow:

s = 0
for n from 0 to 255: P[n] = n
for m from 0 to 767: ulangi langkah 4-6:
n = m modulo 256
s = P[(s + P[n] + Key[m modulo keylength]) modulo 256]
Temp = P[n]
P[n] = P[s]
P[s] = Temp

Where

P : 256-byte permutation array

s : 8-bit variable

n, m : 8-bit variable

keylength : password lenght

Key : password storage array in byte

2.1.2. Pseudo-Random Generation Algorithm (PRGA)

VMPC Algorithm Pseudo-Random Generation Algorithm (PRGA) generates 8-bit streams. The scheme from Pseudo-Random Generation Algorithm (PRGA) is shown as follow:

n = 0
Ulangi Langkah 3-6 sepanjang L:
s = P[(s + P[n]) modulo 256]
Keystream= P[(P[P[s]] + 1) modulo 256]
Temp = P[n]
P[n] = P[s]
P[s] = Temp
n = (n + 1) modulo 256

Where :

P : 256-byte Permutation storage table initialized by VMPC KSA

s : 8-bit variable initialized by VMPC KSA

n,m : 8-bit variable

L : The desired plaintext length in bytes

2.2. Genetic Algorithm

Genetic Algorithms are optimization techniques based on natural selection systems. This Genetic Algorithm has three basic operations, namely selection, crossover and mutation [4].

2.2.1. Selection

This technique is used to select individuals or random binary values that will be selected or taken that are used for the mutation process.

2.2.2. Crossover

Crossover technique is a technique to move binary values to form new binary values.

2.2.3. Mutation

This mutation technique is a technique by replacing binary values.

2.2.4. Complementary Method

Complementary method is a mathematical operation on binary used in the process of calculating binary values on a computer that is widely used for the process of cryptographic calculations. The Complementary Method consists of several operations, namely Complement 1 and Complement 2 [5].

2.2.5. Complement 1

Complement 1operation is an operation that subtracts every bit by 1 or changes the binary value 0 to 1 and 1 to 0.

2.2.6. Complemet 2

Complement 2 operation is the operation of adding binary value 1 to the result of complement 1.

2.3. Avalanche Effect

Avalanche Effect is one of the characteristics that becomes a reference to determine whether or not an algorithm on cryptography. A cryptographic algorithm meets the criteria of an avalanche effect if the input bit changes, then the possibility of all bits experiencing change is half. Changes to the input bit will make it difficult for cryptanalysts to solve ciphertexts. Cryptographic algorithms that have a high avalanche effect, the security level of the cryptographic algorithm is good. The higher the value of the avalanche effect, the better the level of security in the algorithm [6]. The calculationstor the value of Avalanche Effect are as follows:

$$avalanche\ effect = \frac{Jumlah\ bit - bit\ yang\ berbeda}{jumlah\ total\ keseluruhan\ bit} \times 100\%$$

3. RESULTS AND ANALYSIS

The system was built using SharpDevelop 4.4. with the programming language is C#. This system is tested with Personal Computer with 1.0 GHz processor specification AMD C-70 APU, 2 GB Memory.

The test results of the average encryption process are modified Variably Modified Permutation Composition (VMPC) with key optimization using the Genetic algorithm with a number of characters 20, 60 and 100 which are 1.2682 µs, 1.5453 µs and 1.7138 µs. The results of the average test of the encryption process time can be illustrated in a graph of Figure 1:

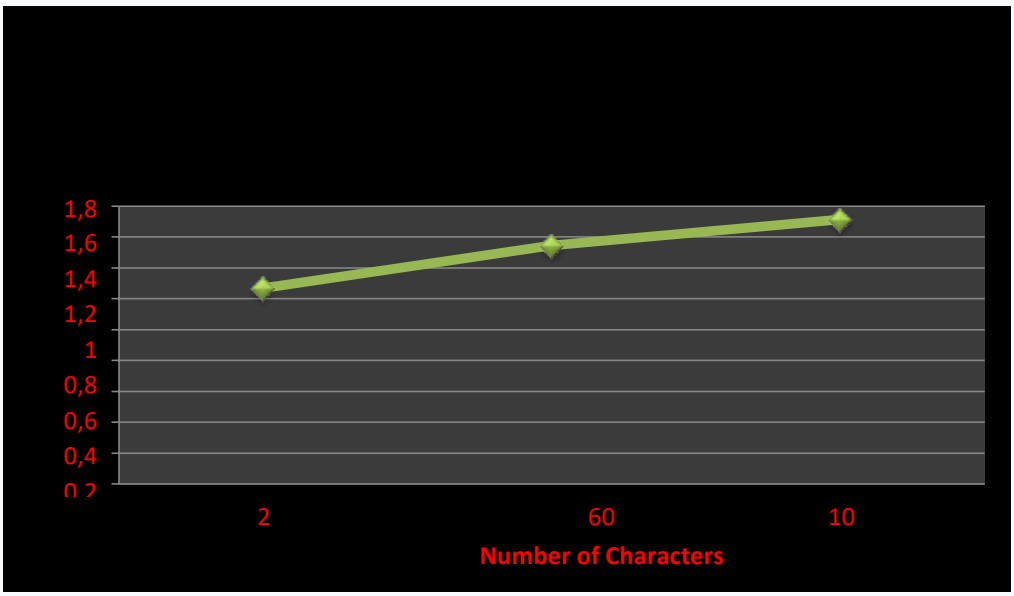

Figure 1. Graph of characters length agains the encryption process time

The test results of the average decryption process are modified Variably Modified Permutation Composition (VMPC) with key optimization using the Genetic algorithm with a

number of characters 20, 60 and 100 which are 0.7751 µs, 1.3302 µs and 1.4532 µs. The results of the average test of the decryption process time can be illustrated in a graph of Figure 2:

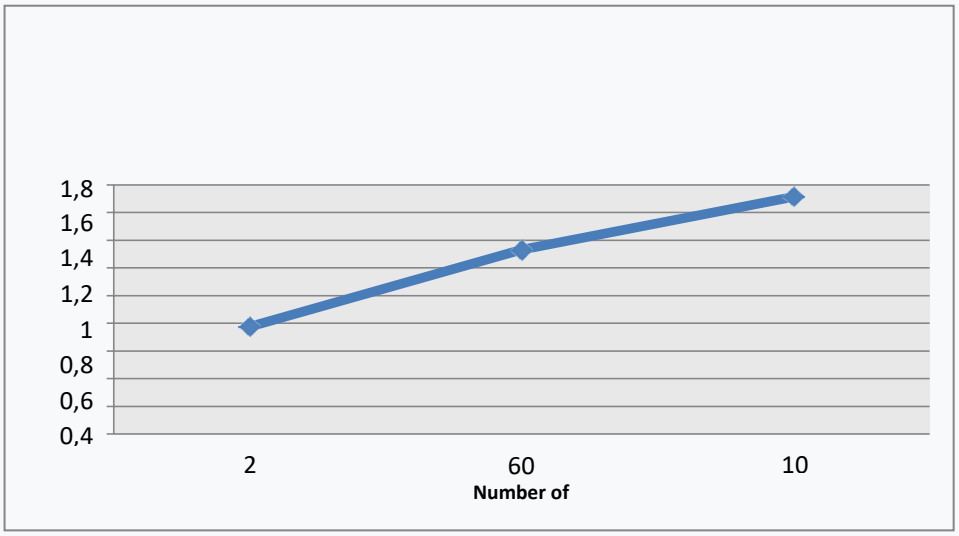

Figure 2. Graph of characters length agains the decryption process time

From the results of 3 tests, the average test results of the avalanche effect were obtained by replacing one character in a randomly generated key of 51.25%. The average avalanche effect results after replacing one character in the key, then the key is optimized using the Genetic algorithm that is 53.42%. The key Avalanche effect comparison before and after optimization can be illustrated in a graph of Figure 3:

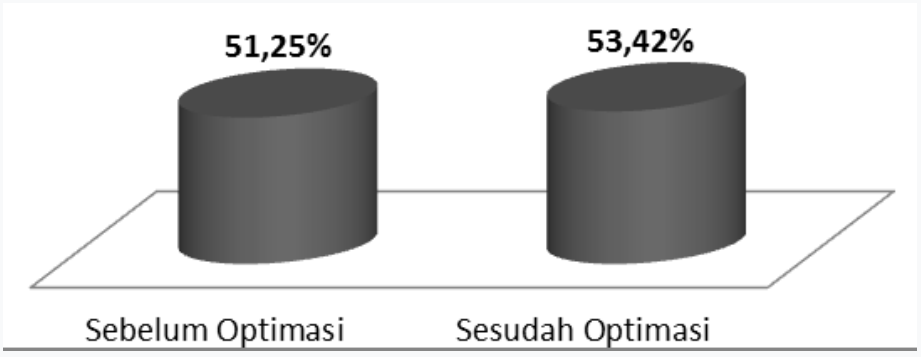

Figure 3. Graph of avarage comparison of the value of avalanche effect

4. CONCLUSION

- The key optimization process uses the Genetic algorithm and adds a complement method to encryption and decryption can be applied to the Variably Modified Permutation Composition (VMPC) algorithm to produce a new variation of the algorithm for modification of the Variably Modified Permutation Composition (VMPC).
- Randomly generated keys are optimized using Genetic algorithms to produce stronger keys.

- From the results of testing the average time of the encryption process with 20, 60 and 100 characters, namely 1.2682 µs, 1.5453 µs and 1.7138 µs. From these results indicate that the length of the character affects the time of the encryption process. The longer the plaintext character is, the longer it takes.
- From the results of testing the average time of the decryption process with 20, 60 and 100 characters, namely 0.7751 µs, 1.3302 µs and 1.4532 µs. From these results indicate that the length of the character affects the decryption process time. The longer the plaintext character is, the longer it takes.
- The Avalanche Effect results carried out on the test obtained an average value of 51.25% for the key without being optimized and 53.42% for the optimized key. From these results it can be concluded that the effect of key optimization using the Genetic algorithm has increased the value of the avalanche effect. So, key optimization using the Genetic algorithm has an effect on the results of changing bits in the ciphertext.

Acknowledgments

The author would like to thank Prof. Dr. Muhammad Zarlis and Dr. Zakarias Situmorang for the guidance given until the research was completed well

REFERENCES

[1] Goyal, Kashish & Kinger, Supriya. 2013. Modified Caesar Cipher for Better Security Enhancement.International Journal of Computer Applications **73**(3): 0975 – 8887.

[2] Sen, A., Ghosh, A., & Nath, A. 2017. Bit Level Symmetric Key Cryptography Using Genetic Algorithm. *International Conference on Communication System and Network Technology (CSNT)* (pp. 193-199). IEEE.Xiong, C., Hua, Z., Lv, Ke. & Li, X. 2016. An Improved K-means text clustering algorithm By Optimizing initial cluster centers. *International Conference on Cloud Computing and Big Data* : 265 - 268.

[3] Zoltak, B. 2004. *VMPC One-Way Function and Stream Cipher*: Delhi, India.

[4] Soni, A., Agrawal, S. 2012. Using Genetic Algorithm for Symmetric Key Generation in Image Encryption, *International Journal of Advanced Research in Computer Engineering & Technology*, **1**(10): 2278-1323.

[5] Mano, M. Morris. *Digital Logic and Computer Design.* Pearson Education India, 2017.

[6] Ramanujam, S. & Karuppiah, M. 2011. Designing an algorithm with high Avalanche Effect. *International Journal of Computer Science and Network Security (IJCSNS).* **11**(1): 106-111.