Vernam Cipher with Complement Method and Optimization Key with Genetic Algorithm

Aulia Rahman Dalimunthe¹, Herman Mawengkang², Saib Suwilo³ and Ahmad Nazam⁴

¹Universitas Sumatera Utara, Medan-Indonesia, ²auliadra@yahoo.com, ³hmawengkang@yahoo.com, ⁴saibwilo@gmail.com, and ⁵ahmadnazam@yahoo.com

Abstract. Integrity is important aspect on the process of exchange data, because the originality of the data sent must be kept confidentiality from the irresponsible attacker. One of the algorithm on cryptography is Vernam Cipher which main focus on XOR operation for encryption and decryption process. The key length and the complexity of algorithm for key determination are very influential to avoid attacks. This aim of study is to modify Vernam Cipher with addition 2’s complement operation on encryption, decryption and key optimizing. Modified Vernam cipher generate random key by genetic approach to improve level of security and complexity of the algorithm.

Keywords: Cryptography, Vernam Cipher, Genetic, XOR

1. Introduction

Data security is very important to be kept its confidentiality. On the process of exchange data, integrity aspect become main focus, therefore the originality of the data sent must be kept maintained from the attacker. One of the algorithm from cryptography is Vernam Cipher which focuses XOR operation on encryption and decryption process.

Vernam Cipher is a symmetric cryptography algorithm, because the encryption and decryption process using the same key. The attacker will get the original message much easier when the key that be using for encryption and decryption are known. The attacker uses brute force method to get the key used.[1]

To avoid attack, the key length and complexity of key algorithm are very influential on the attack.[2] The key determination on Vernam Cipher algorithm using random process to produce key which is likely to be solved by the attacker to get the plaintext. This research, the random generated key will be optimized with using Genetic Algorithm to generate a robust key to make it difficult for attacker to find the key. [3]

2. Theoretical Basis

2.1. Vernam Cipher

Vernam Cipher is a cryptographic algorithm invented by Mayor J. Maugborne and G. Vernam in 1920’s. Vernam Cipher is a symmetric cryptography algorithm, because the encryption and decryption using the same key. This method, operates on using XOR bit plaintext with key. The length of plaintext must be equal with the used key, because on each plaintext character, the process of XOR operation being performed using one key character. Vernam Cipher is also being called One Time Pad algorithm (OTP), because the key is use only once. Therefore, the key distribution must be secured. [4]

Here is the encryption and decryption formula of Vernam Cipher:

\[ C \equiv P \text{ XOR } K \]  \hspace{1cm} (2.1)

\[ P \equiv C \text{ XOR } K \]  \hspace{1cm} (2.2)
Where C is ciphertext, P is Plaintext, K is Key.

2.2. Genetic Algorithm

Genetic Algorithm is optimization techniques by using natural system. On Genetic Algorithm there are 3 basic operation namely selection, crossover and mutation. [2]

2.2.1. Selection

This technique is used for choosing individual or binary value randomly to be selected for the mutation process.

2.2.2. Crossover

This technique is used for shifting binary value thus forming a new binary value. Crossover technique is described in the figure below:

![Figure 2.1 Before Crossover](image1)

![Figure 2.2 Third Locus Separation](image2)

![Figure 2.3 Result After Crossover](image3)

2.2.3. Mutation

Mutation is a technique that replaces the binary value by selecting a randomly binary value. The example of mutation technique are as follows, 010110 binary value mutation on 2 and 5 position generate a new binary value is 00100110.

3. Proposed Method

On this research, the method that we propose consist of 16 randomly characters key generator optimizing by Genetic algorithm. The key will be automatically looping when the plaintext are over 16 characters. Plaintext converted to binary value based on ASCII table then performing encryption and decryption process. Flowchart from the method that we proposing is:

1. Key Generator
   - Output: two different keys, Key1 and Key2.
   - Step 1. Generate 16 key characters randomly (Key1).
   - Step 2. Converting to binary value.
   - Step 3. Forming two parent 64-bit.
   - Step 4. Perform crossover operation randomly between 0-63 bit.
   - Step 5. Perform 10 bit mutation operation randomly.

2. Encryption
   - Output: Ciphertext
   - Step 1. Converting plaintext character to ASCII decimal value.
   - Step 2. Converting to binary value.
   - Step 3. Perform XOR operation on plaintext for Key1.
Step 4. Perform XOR operation result from step 3 to Key2.
Step 5. Perform 2’s complement on result from step 4.
Step 6. Converting to decimal value.
Step 7. Converting to character ASCII.

3. Decryption
Output: Plaintext
Step 1. Converting ciphertext characters to ASCII decimal value.
Step 2. Converting to binary value.
Step 3. Perform 2’s complement.
Step 4. Perform XOR operation result from step 3 to Key2.
Step 5. Perform XOR operation result from step 4 to Key1.
Step 6. Converting to decimal value.
Step 7. Converting to character ASCII.

The scheme of we proposed method, which are described in figure:

![Figure 3.1 Scheme of proposed method](image_url)

4. Implementation
We will do the encryption and decryption of a plaintext with Vernam Cipher with complement an optimization key with Genetic Algorithm that reads “AULIA” uses keys is “abcdefghijklmnopqrstuvwxyz”. Here are the steps to optimized the key using Genetic Algorithm:
1. Generate key randomly = abcdefghijklmnop.
2. Converting to binary value then producing key:
   01100001 01100010 01100011 01100100 01100101 01100110 01100111 01101000 01101001 01101010 01101011 01101100 01101101 01101110 01101111 01110000.
3. Forming two parents 64-bit:
   Parent 1 = 01100001 01100010 01100011 01100100 01100101 01100110 01100111 01101000 01101001 01101010 01101011 01101100 01101101 01101110 01101111 01110000.
   Parent 2 = 01101001 01101010 01101011 01101100 01101101 01101110 01101111 01110000.
4. Performing crossover operation randomly between 0-63 bit, on 42th bit
   Child 1 = 01101001 01101010 01101011 01101100 01100101 01100110 01100111 01101000.
   Child 2 = 01100001 01100010 01100011 01100100 01101101 01101110 01101111 01110000.
   After performing crossover operation, producing:
01101001 01101010 01101011 01101100 01100110 01101000 01100001 01100010 01100011 01100100 01101101 01101110 01101111 01110000

5. Performing 10 bit mutation operation randomly, producing:
01101001 01111010 01101011 11101100 01000101 01101110 01101000 00110011 01100010 01101011 11101000 01101011 01101110 01101111 01110000.

Vernam cipher to be used for generating key randomly for instance “abcdefghijklmnopqrstuvwxyz”, this research reveals that key is optimized by Genetic algorithm which will obtained the new key. There are comparison keys subsequent and preparatory optimized:

Before optimized: 01100001 01100010 01100011 01100100 01100101 01100110 01100111 01101000 01101001 01101100 01101101 01101110 01101111 01110000.

After optimized: 01101001 01111010 01101011 11101100 01000101 01101110 01101000 00110011 01100010 01101011 11101000 01101011 01101110 01101111 01110000.

Obtained that comparison has 16 bit distinction after optimized.

Encryption and Decryption process described in the table below:

### Table 4.1 Encryption process

| ORIGINAL MESSAGE | A | U | L | I | A |
|------------------|---|---|---|---|---|
| ASCII VALUE      | 65| 85| 76| 73| 65|
| BINARY NUMBER    | 01000001| 01010101| 01001011| 01001001| 01000001|
| KEY 1            | 01100001| 01100010| 01100011| 01100100| 01100101|
| RESULT           | 00100000| 00110111| 00101000| 00101101| 00100100|
| KEY 2            | 01101001| 01111010| 01101011| 11101100| 11101100|
| RESULT           | 01001001| 01001101| 01000011| 11000001| 11001000|
| 2'S COMPLEMENT   | 10110111| 10110111| 10111101| 00111111| 00111000|
| DECIMAL VALUE    | 183| 179| 189| 63| 56|
| CIPHERTEXT       | À | Ç | ? | 8 |

### Table 4.2 Decryption Process

| CIPHERTEXT | A | Ç | ? | 8 |
|------------|---|---|---|---|
| ASCII VALUE| 183| 179| 189| 63| 56|
| BINARY NUMBER| 10110111| 10110011| 10111101| 00111111| 00111000|
| 2'S COMPLEMENT| 01001001| 01001101| 01000011| 11000001| 11001000|
| KEY 2       | 01101001| 01111010| 01101011| 11101100| 11101100|
| RESULT      | 00100000| 00110111| 00101000| 00101101| 00100100|
| KEY 1       | 01100001| 01100010| 01100011| 01100100| 01100101|
| RESULT      | 01000000| 01001101| 01000111| 01001001| 01000001|
| DECIMAL VALUE| 65| 85| 76| 73| 65|
| PLAINTEXT   | A | U | L | I | A |
5. Conclusion
This research, new algorithm based on Vernam Cipher is modified with an addition 2’s complement to increase complexity of the algorithm. The used key being optimization using genetic approach to improve the strength and complexity from key determination.

References
[1] Devipriya, M., Sasikala, G. 2015. A New Technique for One Time Pad Security Scheme with Complement Method, International Journal of Advanced Research in Computer Science and Software Engineering, 5(6): 2277-128X.

[2] 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.

[3] Goyat, S., 2012. Cryptography Using Genetic Algorithm (GAs), IOSR Journal of Computer Engineering, 1(5): 2278-0661

[4] Rachmawati, D., M. A. Budiman, and L. Aulya. "Three-pass protocol scheme for bitmap image security by using vernam cipher algorithm." IOP Conference Series: Materials Science and Engineering. Vol. 308. No. 1. IOP Publishing, 2018.

[5] Hatkar, S. S., and Bhagyashri K. Pawar. "Symmetric key algorithm using vernam cipher: VSA." Inventive Computation Technologies (ICICT), International Conference on. Vol. 3. IEEE, 2016.