File text security using Hybrid Cryptosystem with Playfair Cipher Algorithm and Knapsack Naccache-Stern Algorithm

Amalia¹, M A Budiman² and R Sitepu³
¹,²,³ Department of Computer Science, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Medan, 20155, Indonesia
Email: amalia@usu.ac.id, mandrib@usu.ac.id, ravi@students.usu.ac.id

Abstract. Cryptography is one of the best methods to keep the information safe from security attack by unauthorized people. At present, many studies had been done by previous researchers to generate a more robust cryptographic algorithm to provide high security for data communication. To strengthen data security, one of the methods is hybrid cryptosystem method that combined symmetric and asymmetric algorithm. In this study, we observed a hybrid cryptosystem method contain Modification Playfair Cipher 16x16 algorithm as a symmetric algorithm and Knapsack Naccache-Stern as an asymmetric algorithm. We observe a running time of this hybrid algorithm with some of the various experiments. We tried different amount of characters to be tested which are 10, 100, 1000, 10000 and 100000 characters and we also examined the algorithm with various key’s length which are 10, 20, 30, 40 of key length. The result of our study shows that the processing time for encryption and decryption process each algorithm is linearly proportional, it means the longer messages character then, the more significant times needed to encrypt and decrypt the messages. The encryption running time of Knapsack Naccache-Stern algorithm takes a longer time than its decryption, while the encryption running time of modification Playfair Cipher 16x16 algorithm takes less time than its decryption.

1. Introduction
Nowadays, the growth of data and information exchange on the internet makes us need to increase data protection from compromised security attacks by unauthorized people. One of the solutions is by implementing cryptography algorithm. Cryptography is a study based on mathematical techniques for information security such secrecy, data integrity, and entity authentication [1]. Cryptography can also be used for identification of individuals, authentication and non-repudiation [2].

Cryptography algorithms can be classified into symmetric Key Algorithm and Asymmetric Key Algorithm. The Symmetric Algorithm use a private key [3] or a single key for encryption and decryption process, while the asymmetric algorithm uses public key [3] or use a different key for the encrypt and decrypt process. The advantage of using the symmetric algorithm is the faster time execution. Meanwhile, the most problem with this algorithm determines the efficient method to exchange the key safely. The asymmetric algorithm, based on a mathematic formula, is a robust algorithm because this algorithm is not easy to compromise, but takes longer processing time than the symmetric algorithm[3]. In this study, we combine asymmetric and symmetric algorithm to get advantages of these two various algorithms. We implemented Playfair Cipher algorithm which is a symmetric algorithm to encrypt and decrypt the ciphertext and Knapsack Naccache-Stern algorithm which is an asymmetric algorithm to encrypt and decrypt the key.
Playfair Cipher algorithm developed by Charles Wheatstone in 1854, but popularized by Lord Playfair [4]. This algorithm is substitution digraph which encrypts two letters at once that utilize 5x5 matrix for encryption and decryption process. Playfair Cipher with a 5 x 5 matrix has limitation characters to encrypt and decrypt process, not all ASCII characters are covered in this matrix. So, in this study, we develop Playfair Cipher matrix into a 16 x16 matrix so all of ASCII characters can be used efficiently for encryption and decryption process.

Knapsack Naccache-Stern algorithm is a deterministic public key algorithm based on knapsack problem. This algorithm has an advantage which it can produce a robust encryption result that cannot quickly to destruct. However, this algorithm has a deficiency which is has a long processing time for encryption [5]. Knapsack Naccache-Stern algorithm has three processes, which are key generation process, encryption process, and decryption process. We implemented Agrawal Kayal Saxena (AKS) algorithm, a deterministic algorithm in polynomial time, to generate and examine the prime number [6].

We combined these two cryptography algorithms to cover the weakness of each algorithm. For the experiment, we implement file text .doc, .docx and .thx as plaintext.

2. Method
To encrypt the plaintext with Playfair Cipher algorithm, we arrange the key into 16x16 matrix. The encryption process according to the following rules[7]:

- If the pair of the letters appears on the same row of the table, then the ciphertext of these letters are the letters which are in right position of them.
- If the pair of the letters appears in the same column of the table, then the ciphertext of the letters are the letter to their immediate below respectively.
- If the pair of the letters is not in the same row or column, then the ciphertext of the letters are the letter to their immediate right respectively but at the other pair of corners of the rectangle defined by the original pair. Notice that the order is essential, the first letter of the encrypted pair is the one that lies on the same row as the first letter of the plaintext pair.

The key of Playfair Cipher algorithm that used to encrypt the messages will be encrypted with Knapsack Naccache-Stern algorithm. The asymmetric algorithm has to generate a public and private key first, before encryption and decryption process. To generate the key for Knapsack Naccache-Stern algorithm according to the following rules [5]:

- Pick \( p_i \) the prime number starts from \( i = 0 \) to \( i = 7 \).
- Multiply all the prime numbers formed.
- Pick \( p \), a prime number that greater than the result of the multiplying all of the prime number \( (p_i) \), such that:
  \[
  p > \prod_{i=0}^{n} p_i
  \]
- Pick a secret integer \( s < p-1 \) such that \( \text{GCD} (p-1, s) = 1 \).
- Set \( v_i \) with this rule \( v'_i \equiv p_i \text{mod } p \)

The public key is \{ \( p, n, v_0, \ldots, v_n \) / \} and the private key is \( s \).

In this research, for the key generation steps, we tried some of the prime numbers randomly. We implemented AKS to examine the validity of this prime number. AKS algorithm has the following rules [6]:

- Let \( p \), a number that will be checked a prime or not.
- Pick a positive integer \( z \) where \( 2 \leq z < p-2 \) and \( \text{GCD} (p,z) = 1 \)
then $p$ is a prime number if and only if:

$$(1 + z)^p \equiv 1 + z^p \pmod{p}$$

The next step is a process to encrypt the Playfair Cipher Algorithm’s key with a preformed public key of Knapsack Naccache-Stern. This public key is generated according to the following formula [5]:

$$c = \prod_{i=0}^{n} v_i^{m_i} \mod p$$

where:
- $c$ = ciphertext
- $m_i$ = the $i$th bit of the message $m$
- $n, p, v_i$ = public key

The decryption process of Knapsack Naccache-Stern algorithm is used to get the Playfair Cipher algorithm’s key back. So the process is continued by using the preformed public key according to the following rules [5]:

$$m = \sum_{i=0}^{n} \frac{2^i}{p_i - 1} \times (\gcd(p_i, c^s \mod p) - 1)$$

where:
- $m$ = message
- $c$ = ciphertext
- $s$ = private key

A decrypted process can be done to get back the key of the Playfair Cipher algorithm. The rules for decryption process of Playfair Cipher algorithm has the same rules as its encryption process. We observed running time for all processes in each algorithm with various characters length and various length of the private key. We also noted size of ciphertext that is produced by these process. The aim of this process, we can see the relation between the length of plaintext with running time for encryption process of Playfair Cipher Algorithm.

3. Results and Discussion

3.1 Encryption Process for Playfair Cipher Algorithm

To measure the running time for Playfair Cipher Algorithm encryption process, we do encryption experiments with various length of plaintext. In this study, we tried 10, 100, 1000, 100000 characters with the length of the symmetric key that are 10, 20, 30 and 40 characters. The result of this process can be described in Figure 1.
According to Figure 1, we can see the encryption processing time increase linearly with the increasing length of plaintext. Besides that, encryption running time process also affected by the length of the symmetric key. The longer symmetric keys used, then the longer times needed to encrypt.

### 3.2 Decryption Process for Playfair Cipher Algorithm

For decryption process, we do the same experiment like encryption process. We tried various length of ciphertext with various length of the symmetric key like a. We decided 10, 100, 1000, 10000 and 100000 characters of ciphertext and 10, 20, 30, and 40 characters of the symmetric key. The results are shown in Figure 2.
Based on figure 2, we can see the result of decryption processing time has the same pattern like encryption processing time which is decryption processing time increase linearly with the increasing length of ciphertext and decryption running time process affected by the length of the symmetric key.

3.3 Encryption Process of Knapsack Naccache-Stern Algorithm
In this study, we also calculated the running time for encryption process of Knapsack Naccache-Stern algorithm. We tried for some various length of the symmetric key which is 10, 20, 30, and 40 characters. The results are shown in the figure 3.

![Encryption Process](image.png)

**Figure 3.** The Relation Between The Length of Plainkey with Running Time for Encryption Process of Knapsack Naccache-Stern Algorithm

Based on figure 3 the encryption processing time of plainkey using Knapsack Naccache-Stern algorithm is affected by its length. The longer characters of the plainkey, then the longer periods needed to encrypt.

3.4 Decryption Process for Knapsack Naccache-Stern Algorithm
The running time for decryption process of Knapsack Naccache-Stern calculated with the cipher key that already formed on the previous stage. The results are shown on the figure 4.
3.5 Size of Ciphertext

In this work, we also observe the relation between the size of plaintext with the size of ciphertext. We tested some of the various sizes of plaintext that is encrypted with 10 characters length of the symmetric key. The results are shown in figure 5.

Based on figure 5, we can see encrypted process produced smaller length size of the file. It means the size of ciphertext is smaller than the size of the plaintext.
4. Conclusion
Based on this research of cryptography using Playfair Cipher algorithm and Knapsack Naccache-Stern algorithm, it can be concluded:

- Implementation of file text security using hybrid cryptosystem with Playfair Cipher algorithm and Knapsack Naccache-Stern algorithm can be a choice to secure a plaintext.
- Modification of Playfair Cipher from 5x5 matrix into 16x16 matrix will increase the quality of cryptography because more characters of key combinations can be formed than using Playfair Cipher 5x5.
- The running time for encryption and decryption process each algorithm is linearly proportional, it means the longer messages character then, the more significant times needed to encrypt and decrypt the messages.
- The encryption running time of Knapsack Naccache-Stern algorithm takes a longer time than its decryption, while the encryption running time of modification Playfair Cipher 16x16 algorithm takes less time than its decryption.
- The size of ciphertext is smaller than the size of its plaintext.

References
[1] W Mao and Wenbo Modern cryptography : theory and practice. Prentice Hall PTR, 2004.
[2] S Subasree, N K Sakhivel, and T Nadu 2010 Design of a New Security Protocol Using Hybrid Cryptography Algorithms Comput. vol. 2 pp. 95–103.
[3] E Fujisaki and T Okamoto 2013 Secure Integration of Asymmetric and Symmetric Encryption Schemes J. Cryptol 26(1) pp. 80–101.
[4] Alam AA, Khalid BS, Salam CM 2013 A Modified Version of Playfair Cipher Using 7 x 4 Matrix Int. J. Comput. Theory Eng. 5(4) pp. 626–628.
[5] D Naccache and J Stern 1997 A New Public-Key Cryptosystem Springer, Berlin, Heidelberg pp. 27–36.
[6] Agrawal M, Kayal N, Saxena N 2002 PRIMES is in P Dep. Comput. Sci. Eng. Indian Inst. Technol. Kanpur Kanpur-208016.