Hybrid Cryptosystem Implementation Using IDEA and Knapsack Algorithm for Message Security

Dian Rachmawati¹*, Maya Silvi Lydia¹*, Wynda Arianni Siregar¹*

¹ Departemen Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara, Jl. Universitas No. 9-A, Medan 20155, Indonesia.

*Email: dian.rachmawati@usu.ac.id, maya.silvi@gmail.com, wyndasiregar@gmail.com

Abstract. One way of maintaining the information confidentiality is by using a cryptographic technique such as hybrid cryptosystem. A hybrid cryptosystem is a combination of symmetric and asymmetric algorithms. On symmetric algorithms, the time which it takes to perform encryption and decryption is shorter than an asymmetric algorithm. While the advantages of asymmetric algorithms are only needed to maintain the confidentiality of the private key without the need to frequently change the key as the symmetric algorithm. In this algorithm, there is no need to send the key private key as in the system of symmetry. This research uses the IDEA of symmetry algorithm and algorithm of knapsack asymmetry. The IDEA of the algorithm is to secure a text message, while the knapsack algorithm is to maintain the confidentiality of the message key distribution. Information has acquired in the form of a string. Because the IDEA is a block cipher, then the length of the text characters is out of stock divided by 8. The use of a hybrid cryptosystem with Knapsack algorithm IDEA and managed to keep the integrity of the data, so the data is encoded can return to its original form.

1. Introduction
Cryptography is a technique of securing and ensuring the authenticity of data which is consisting of two processes namely encryption and decryption. Encryption is a process to make a data becomes unreadable by unauthorized parties such as a password or coded language. While decryption is the opposite of encryption, i.e., the process of making the information or data that has been encrypted to be readable by the party entitled.

The use of a symmetric cryptographic algorithm and asymmetric cryptography to secure data, for now, is considered inadequate as cryptanalysis develops. In this research, a cryptographic method used is hybrid cryptosystem. The hybrid cryptosystem is a combination of symmetric algorithms and asymmetric algorithms. Hybrid cryptosystem used in this study is a combination of IDEA and Knapsack algorithms.

The IDEA (International Data Encryption Algorithm) is a symmetric algorithm that operates on an open-width 64-bit message block and a 128-bit key length. While the Knapsack algorithm is an asymmetric cryptographic algorithm that uses a public key to encrypt messages and use private key (private key) to decrypt messages.
2. Method

2.1. Hybrid Cryptosystem

Hybrid Cryptosystem is one of the method in cryptography which works by combining both of symmetric and asymmetric algorithm to secure plaintext [7]. In this hybrid system, encryption/decryption of messages using cryptographic key-symmetry, while the symmetry key is encrypted / decrypted with public-key cryptography. A symmetry key (also called a session key) has generated by either party and encrypted a message with that key. The session key has encrypted with the recipient's public key, and the session key will be sent together with an encrypted message. The recipient initially decrypts the session key with its private key, then decrypts the message with the session key [2].

2.2. International Data Encryption Algorithm (IDEA)

IDEA is one of the most secure block cipher algorithms available to the public [4]. First, The IDEA has discovered by Xuejia Lai and James Massey in 1990. Formerly, it was called PES (Proposed Encryption Standard). The following year, after Biham and Shamir have demonstrated their derivative cryptanalysis, the designers of the algorithm strengthened their ciphers to against the attack and named the new algorithm as the IPES (Improved Proposed Encryption Standard). And in 1992, IPES changed its name to IDEA (International Data Encryption Algorithm) [1].

IDEA is a block cipher, which operates on a 64 bits plaintext block. The key length is 128 bits. The same algorithm has used for encryption and decryption process. IDEA applies the following incompatible algebraic operations:

- XOR or ⊕.
- Added modulo $2^{16}$ or ⊕.
- Multiplication modulo $2^{16} + 1$ or $⨀$ [6].

The IDEA algorithm can be divided into three major parts, namely the key forming algorithm, encryption, and decryption.

1. IDEA Key Formation Process

The IDEA method has an input key 128 bits identical to 32 hexadecimal digits or 16 characters processed to produce 52 subkeys, with the details of each of the six subkeys was to be used in 8 rounds and four subkeys for output transformation.

2. IDEA Encryption Process

The IDEA algorithm encryption process is a 64 bits plaintext and divided it into four sub-blocks with 16 bits long, i.e., X1, X2, X3, X4. These four sub-blocks serve as inputs for the first-phase iteration of the algorithm. In total there are eight iterations. At each iteration, four sub-blocks are XOR-aligned, added, multiplied by the other and with six 16-bit subkeys.

Among the iterations of the second and third sub-blocks are interchangeable. Finally, four sub-blocks have combined with four subkeys in the output transformation. At each stage, the following sequence has done:

1. Multiply X1 with K1 mod $(2^{16} + 1)$.
2. Add X2 with K2 mod $2^{16}$.
3. Add X3 with K3 mod $2^{16}$.
4. Multiply X4 with K4 mod $(2^{16} + 1)$.
5. XOR results from step 1 and 3.
6. XOR results from step 2 and 4.
7. Multiply the result from step 5 with K5 mod $(2^{16} + 1)$.
8. Add the result from step 6 and 7 mod $2^{16}$.
9. Multiply the result from step 8 with K6 mod $(2^{16} + 1)$.
10. Add the result from step 7 and 9 mod $2^{16}$.
11. XOR results from step 1 and 9.
12. XOR results from step 3 and 9.
13. XOR results from step 2 and 10.
14. XOR results from step 4 and 10.
The output of each round is the four sub-blocks generated in steps 11, 12, 13 and 14. The blocks 12 and 13 are swapped (except for the last round) so that the input of the next round is a combination of steps 11, 12, 13, And 14. After eight rounds, the following output transformation will be performed:

1) Multiply X1 with K1 mod subkey \((2^{16} + 1)\).
2) Add X2 with subkey K2 mod \(2^{16}\).
3) Add X3 with K3 mod \(2^{16}\) subkey.
4) Multiply X4 with K4 mod subkey \((2^{16} + 1)\)

3. IDEA Decryption Process
The decryption process is the same as the encryption process. The difference only lies in the rules of the subkey. The order of the subkey was reversed with the encryption process, and its subkey is converted. The subkey on the output transformation step of the encryption process is converted and used as a subkey in round 1 of the decryption process. Subkey of round 8 is converted and used as subkey on round 1 and 2 in the decryption process. Subkeys which has used for encryption are different with subkey which have used for decryption. The first difference is in the order of use of the subkey. The second difference, namely the formation of decryption subkey using multiplexed inverse multiplexing modulo \(2^{16} + 1\) and inverse operations of modulo \(2^{16}\) subkey.

2.3. Knapsack Public Key Algorithm
Many conventional knapsack cryptosystems use particular secret sequences for letting the decryption easy. For example, a super-increasing sequence has used in Merkle-Hellman Knapsack scheme [5]. The superincreasing knapsack algorithm is a weak algorithm since the ciphertext can be decrypted easily into \(O(n)\). The non-super increasing knapsack or Knapsack algorithm is a problematic problem (computational) in knapsack algorithm group because it takes time in an exponential order to solve it. However, superincreasing knapsack can be modified into non-super increasing knapsack by using the public key (for encryption) and private key (for decryption). The public key is a non-super increasing sequence, while the private key remains a superincreasing sequence. This modification has discovered by Ralph Merkle and Martin Hellman [2].

The basic idea behind Merkle-Hellman's encryption scheme is to create a subset of problems that can be solved quickly. This easy-to-solve problem is by hiding the superincreasing sequence by multiplication modulo and permutation [3].

**Algorithm for generating the public key and private key are:**

a. Determine the superincreasing sequence.
b. Multiply each element in the sequence with \(n\) modulo \(m\). The modulus \(m\) should be a number greater than the sum of all elements in the sequence, whereas the multiplier \(n\) should have no union factor with \(m\).
c. Multiplication results will be the public key while the superincreasing sequence became the private key.

**Encryption Process of Knapsack Algorithm**
The encryption process was done with the following stages:

a. Encryption was done in the same way by using previous knapsack algorithm.
b. At first, the plaintext was broken into blocks of bits that are equal in length to the cardinality of the public key row.
c. Multiply each bit in the block with the corresponding element in the public key

**Decryption Process of Knapsack Algorithm**
The process of decryption has done with the following stages: Decryption has done by using the private key. The receiver of the message counts \(n^{-1}\), i.e., inverses \(n\) modulo \(m\), such that \(n \cdot n^{-1} = 1\) (mod \(m\)). Multiply each cryptogram with \(n^{-1}\) mod \(m\), then declare the result as the sum of private key elements to obtain plaintext by using a search algorithm to super increasing Knapsack.
3. Results and Discussions
The illustration of hybrid cryptosystem by using IDEA and Knapsack is shown in figure 1. The recipient shares the Knapsack key to the sender. The sender encrypts the message using IDEA algorithm and encrypts the IDEA key using Knapsack key. After the sending process, the receiver decrypts the IDEA cipher key. The IDEA key is used to open the ciphertext.

![Diagram of hybrid cryptosystem](image)

**Figure 1.** Hybrid cryptosystem IDEA and Knapsack

Data integrity is a parameter used in testing the implementation of hybrid cryptosystem IDEA and knapsack algorithms. There are two tests done. The first is examining the encryption and decryption of messages (files). And the second is IDEA encryption and decryption key testing. Each test has conducted some experiments.

**Table 1.** Encryption and Decryption of the message

| No. | Encryption and Decryption of the message | Encryption | Decryption |
|-----|----------------------------------------|------------|------------|
| 1   | **Input**:                             |            |            |
|     | a. Plain text with 16 characters:      |            |            |
|     | Algoritma IDEA64                       |            |            |
|     | b. DEA Key:                            |            |            |
|     | fasilkom-ti2012C                       |            |            |
|     | **Output**:                            | 8C6070BA2AD9 | 969D404F26D8 |
|     | 3230                                   |            |            |
| 2   | **Input**:                             |            |            |
|     | **Output**:                            | 8C6070BA2AD9CF22 | Algoritma IDEA64 |
|     | 969D404F26D83230                       |            |            |
In Table 1 it can be seen that the result of ciphertext has built in the form of the hexadecimal number. Each time was encrypting the block cipher (requires eight characters each encrypt one block), it will generate 16 digits of the hexadecimal number. In other words, the number of characters from the encryption test results (in hexadecimal digits) is two times the number of input characters. This test successfully returns the ciphertext to plaintext.

| No. | Encryption and Decryption of IDEA Key |
|-----|--------------------------------------|
|     | Encryption                          | Decryption                        |
| 1   | Input:                              | Output:                           |
|     | a.Plain Key:                        | a. Private Key:                   |
|     | fasilkom-ti2012C                    | 2 4 7 15 30 60 120 240            |
|     | b.Public Key:                       | M = 480, N\(^1\) = 343            |
|     | 14 28 49 105 210 420                | b.Cipher key:                     |
|     | 360 240                             | 857 317 782 527 707 887           |
|     | M = 480, N = 7                      | 1307 947 919 602 527 257          |
|     |                                      | 514 154 394 514 628               |
|     | Output (Cipher key):                |                                    |
|     | 857 317 782 527 707 887             | Output (Plain key):               |

Table 2. Encryption and Decryption of IDEA Key
In Table 2 can be seen that the result of the cipher key is a decimal number with the number of cipher key blocks being two times the number of Knapsack public key cards. This test successfully returns the cipher key to plain key (IDEA symmetry key).

From the result of table 1 and table 2, thus, it can be concluded that testing using the hybrid cryptosystem technique of IDEA and Knapsack in this system has met the integrity of the data.

**Table 3.** Encryption and Decryption time

| No. | Length of Message (characters) | Encryption Time (ms) | Decryption Time (ms) |
|-----|--------------------------------|---------------------|---------------------|
| 1   | 8                              | 5                   | 4                   |
| 2   | 16                             | 8                   | 6                   |
| 3   | 24                             | 10                  | 6                   |
| 4   | 32                             | 12                  | 7                   |
| 5   | 40                             | 13                  | 7                   |
| Total| 48                             | 30                  |                     |
| Average | 9.6                         | 6                   |                     |

From table 3, it can be assumed that the longer number of text characters in the message encrypted, the time required to encrypt the message will be longer. Although there is constant time for the greater length of text in the decryption process (e.g., text with characters’ length 16 and 24 have the same decryption time of 6 ms). The system takes less time to decrypt the message than it does the message encryption process.
4. Conclusions
Based on the discussion and the results of the research, the following conclusions have obtained:
1. With the message security mechanism that is hybrid cryptosystem technique, message encryption using IDEA can protect the information contained in the message and encryption key using Knapsack algorithm can secure the distribution of key.
2. IDEA and Knapsack algorithms meet the parameters of data integrity. The result of the message decryption test is ciphertext will be the same as the initial plaintext (original message). And the result of the cipher key decryption test is will be the same as the key that has been used for the message encryption process.
3. From testing five files with text and length of different characters, the average time of encryption is 9.6 ms. While the average decryption time is 6 ms. Therefore, it can be concluded that the decryption time of the message is shorter than the time of message encryption.

Acknowledgments
The authors gratefully acknowledge that the present research has supported by Fund Dissemination IPTEKS Research Results for Lecturers / Researchers Universitas Sumatera Utara.

References
[1] Schneier, B. 1996. Applied Cryptography: Protocols, Algorithms and Source Code in C. Jhon Wiley & Sons, Inc: New York.
[2] Munir, R. 2006. Kriptografi. Informatika Bandung: Bandung.
[3] Agarwal, A. 2011. Encryption messages using the Merkle-Hellman Knapsack cryptosystem. International Journal of Computer Science and Network Security (IJCSNS) Vol. 11, No. 5:12-14.
[4] N. Hamdy, K. Shehata, S. Elagooz and M. Helmy, Design and implementation of fast inverse modulo \(2^{16} +1\) multiplier used in idea algorithm key schedule on FPGA, Electrical, Electronic, and Computer Engineering. 2004. ICEEC’04. 2004 International Conference on, Cairo, Egypt, 2004, pp. 507-511. doi: 10.1109/ICEEC.2004.1374513
[5] S. Hamasho, Y. Murakami and M. Kasahara, A systematic encryption algorithm for knapsack scheme using random sequence, 2012 7th International Conference on Computing and Convergence Technology (ICCCCT), Seoul, 2012, pp. 514-517.
[6] S. L. C. Salomao, V. C. Alves, and E. M. C. Filho, A two-level pipelined implementation of the IDEA cryptographic algorithm, Proceedings. XI Brazilian Symposium on Integrated Circuit Design (Cat. No.98EX216), Rio de Janeiro, 1998, pp. 158-161. doi: 10.1109/SBCCI.1998.715431
[7] Dian Rachmawati, Amer Sharif, Jaysilen, and Mohammad Andri Budiman, Hybrid cryptosystem using tiny encryption algorithm and luc algorithm, IOP Conference Series: Materials Science and Engineering 300 (2018), no. 1, 012042.