Hybrid cryptosystem implementation using fast data encipherment algorithm (FEAL) and goldwasser-micali algorithm for file security

D Rachmawati*, M A Budiman and W S E Siburian
Departemen Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara, Medan, Indonesia

*Corresponding author’s e-mail: dian.rachmawati@usu.ac.id

Abstract. On the process of exchanging files, security is indispensable to avoid the theft of data. Cryptography is one of the sciences used to secure the data by way of encoding. Fast Data Encipherment Algorithm (FEAL) is a block cipher symmetric cryptographic algorithms. Therefore, the file which wants to protect is encrypted and decrypted using the algorithm FEAL. To optimize the security of the data, session key that is utilized in the algorithm FEAL encoded with the Goldwasser-Micali algorithm, which is an asymmetric cryptographic algorithm and using probabilistic concept. In the encryption process, the key was converted into binary form. The selection of values of x that randomly causes the results of the cipher key is different for each binary value. The concept of symmetry and asymmetry algorithm merger called Hybrid Cryptosystem. The use of the algorithm FEAL and Goldwasser-Micali can restore the message to its original form and the algorithm FEAL time required for encryption and decryption is directly proportional to the length of the message. However, on Goldwasser-Micali algorithm, the length of the message is not directly proportional to the time of encryption and decryption.

1. Introduction
Securing data is important to keep the information of a person or an organization/company is not be used by individuals who are not eligible or misused. Cryptography is the study of the method of delivery of messages in secret [1]. There are two main process in cryptography, the process of transforming file into hidden code that cover the file is called encryption and the process of transforming hidden code into original file is called decryption [2]. The main objectives of cryptography are confidentiality, integrity, authentication, and non-repudiation [3]. FEAL is one example of an algorithm. However, a shortage of algorithm FEAL was in 1988 (a year after the discovery of FEAL), FEAL with version 4 rounds (FEAL-4) already can be attacked with the chosen plaintext attack by den Boer and encryption keys can be found more or less within 4 hours.

To optimize security, then the research will secure encryption key algorithm with a public key algorithm FEAL. Some public key algorithms are already well known general algorithm, such as RSA, ElGamal, and Rabin. However, in the algorithm, called deterministic algorithm, with a public key, the original plaintext message m will always have the same encryption results, namely the secret message (the ciphertext) c. The system has several weaknesses, including the following, namely: it is not safe for all possible message delivery. For example on the RSA algorithm: message in binary 0 and 1 will
have the same ciphertext with itself, so it is easy to detect, and it is easy to recognize a message sent twice. Therefore, this research will use a probabilistic public key encryption. The probabilistic algorithm is an algorithm that adds the random command that has a return value of ' 0 ' or ' 1 ' with each opportunity, i.e., ½. One example of a probabilistic public key algorithm is the Goldwasser-Micali. This algorithm basing on the difficult problems of the quadratic residue modulo a composite number n in the encryption process, i.e., to find a value of q that is equivalent to the square of the value of x modulo a composite number, where q and x is an element of n.

2. Methods

2.1. FEAL Algorithm

FEAL was designed by Akihiro Shimizu and Shoji Miyaguchi, from Nippon Telegraph and Telephone (NTT) of Japan in 1987. FEAL block cipher algorithm is that encrypts data in the 64-bit block and key 64-bit [4].

Algorithm FEAL consists of two processes, namely the encryption keys and block scheduling data. Key generation Process: key 64-bit will be divided into two parts, namely A and B, each 32 bits. On this key generation process will be produced is 16, each 16 bits. Then, calculate the Ki (i = 0 to 15) for r = 1 to 8. Note:

\[ D_r = A_{r-1}; \quad D_0 = 0 \]
\[ A_r = B_{r-1} \]
\[ B_r = f_k (A_{r-1}, B_{r-1} \oplus D_{r-1}) \]

The result of the function is the key to 32 bits, divided into two, namely K2 (r-1) and K2 (r-1) + 1, each 16 bits.

K2 (r-1): K0, K2, K4, K6, K8, K10, K12, K14
K2 (r-1) + 1: K1, K3, K5, K7, K9, K13, K11, K15
R: round to-i, i = 1, 2, 3, ... 8

The input consists of two functions, i.e., α and β, each 32 bits. Inputan α and β is divided into four parts 1-octet, i.e., α0, α1, α2, α3 and; β0, β1, β2, β3 and. Every function will generate a 32 bit key.

\[ f_k 1 = \alpha 1 \oplus \alpha 0 \]
\[ f_k 2 = \alpha 2 \oplus \alpha 3 \]
\[ f_k 1 = S1(f_k 1, f_k 2 \oplus \beta 0, 1) \]
\[ f_k 0 = S0 (\alpha 0, f_k 1 \quad \beta 2, 0) \]
\[ f_k 2 = S0 (f_k 2, f_k 1 \quad \beta 1, 0) \]
\[ f_k 3 = S1 (\alpha 3, f_k 2 \quad \beta 3, 1) \]

Encryption data blocks: step-by-step algorithm FEAL encryption process is as follows:

2.1.1. The encryption process. Begins with the 64-bit data block that is XOR with a 64-bit key, i.e., K8, K9, K10, K11.

2.1.2. Then block data divided into two parts, i.e., the left part (L0) and right (R0) are each 32 bits.

\[ P = (R0, L0) \]

2.1.3. L0 to XOR with the R0 to block the right of new data (R0).

\[ (L0, R0) = (L0, R0) \oplus (L0, R0) \]

2.1.4. Block data L0 and R0 entered n rounds. Block data R0 will be combined with the 16-bit key in function f (things like this also done for each round). The result will be XOR with L0 for form a block of data to the new right, i.e., R = R, while to the left of the data block that is new, is the same with the right data before. Following their definitions:

For r = 1 to 8, calculate

\[ RR = L_{r-1} \oplus f(R_{r-1}, K_{r-1}) \]
LR = R_{r-1}
Then,
\[(R_8, L_8) = (R_8, L_8) \oplus (R_8)\]  

2.1.5. *The second.* Block data are merged forming the block of data with a length of 64-bit again.
\[(R_8, L_8) = (R_8, L_8) \oplus (K_{12}, K_{13}, K_{14}, K_{15})\]
Thus, ciphertext, C, are: \((R_8, L_8)\).

The decryption process is similar to the encryption process (one program can demonstrate two functions, i.e., the encryption and decryption of messages). The key used is also the same. Ciphertext will first on XOR key with \(K_{12}, K_{13}, K_{14}, K_{15}\), will then be divided into two, namely \(C = (R_8, L_8)\).
\[(R_8, L_8) = (R_8, L_8) \oplus (K_12, K_13, K_14, K_15)\]

\[\text{For } i = 8 \text{ to } 1, \text{ count:}\]
\[R_{i-1} = L_i\]
\[L_{i-1} = R_i \oplus f(L_i, K_{i-1})\]

On the final results of the merger of the two block at once:
\[(L_0, R_0) = (R_0, L_0) \oplus (L_0)\]
\[(L_0, R_0) = (R_0, L_0) \oplus (K_8, K_9, K_{10}, K_{11}).\]

2.2. *Goldwasser-Micali Algorithm*

The security on FEAL comes from the key length used, i.e., 64-bit and in the process, the key block used will be variable. But FEAL tends to be easy to attack; this is because the decryption process that occurs in FEAL using the same process with the encryption process. But with the order of use, the subkey on the key schedule is reversed. By using Goldwasser-Micali, there will be message randomization. A binary message will be encrypted bit per bit as follows: "0" encoded by a random x number like \(B(x) = 0\), and "1" encoded with a random x number selected as \(B(x) = 1\). The result, many possible password results for each message. In this research, the FEAL algorithm will be used to secure the message, and Goldwasser-Micali is used to secure the key so that the weakness of the FEAL algorithm will be covered with key randomization.

To encrypt a message \(m\), anyone can easily find out the value of \(E(m)\), but just knowing the trapdoor information that can determine the value of \(m\) from \(E(m)\). Examples of the application of trapdoor function this is the RSA algorithm. For example, in the function of the RSA, the exponent function is used on trap-door against the rank on the public key in the field \(\mathbb{Z}_n\) where \(n = pq\), namely the results of multiplication of two very large primes. Encrypting the original message \(m \in \mathbb{Z}_n^*\) is \(c \equiv me \mod n\). Prime factor of \(n\) is called a trapdoor. However, the downside of being deterministic algorithms are: (1) If the same message posted more than one time, then the attacker will be easy to detect it. (2) Easy to reverse the plain text which has certain forms of RSA algorithm, for example, if the original message in the form of 0 and 1, then the result of encryption will remain the same, namely 0 and 1.

Probabilistic public key encryption has the security parameters, which are: (1) \(K\), key generation: probabilistic algorithms have input \(n\), and outputs a pair \((e, d)\): \(e\) is the public key, and \(d\) is the private key. (2) \(E\), is a function of the input having the encryption public key of \(e\), in the form of the plaintext bit \(b\) \{0,1\}. (3) Decryption function is: has two inputs, i.e., private key \(d\) and the ciphertext \(c\).

2.2.1. *Key generation Algorithm.* First, select two large random primes, \(p\) and \(q\), where \(p \neq q\), then compute \(n = p \times q\). Then, select a number \(y\) that is a non-residue quadratic mod \(n\) \((y\) is non-quadratic residues mod \(p\) and \(y\) is a non-residue quadratic mod \(q\), \(\left(\frac{y}{p}\right) = \left(\frac{y}{q}\right) = -1\)) so that the Jacobi symbol = 1,
\( \begin{pmatrix} y \\ n \end{pmatrix} = 1 \). This number is called the pseudo square. Then set \((n, y)\) as the public key is \((p, q)\) as the private key. Then, send \((n, y)\) to the owner of the message will be encrypted.

2.2.2. Encryption algorithm. Change the message \(m\) into a string of bits, i.e., \(m = m_1 \ m_2 \ m_3 \ldots m_l\). For each \(m_i \in m\), select \(x \in \mathbb{Z}_n^*\). For encryption, we use the public key that is sent \((n, y)\). If \(m_i = 1\), then \(m\) encrypt into a random quadratic residue mod \(n\), i.e., \(c_i = yx^2 \mod n\) otherwise if \(m_i = 0\), then \(m_i\) encrypt into \(c_i = x^2 \mod n\). Then, the message is sent the \(C_m(m) = (c_1 \ c_2 \ c_3 \ldots c_l)\).

2.2.3. Decryption Algorithm. Recipient of the ciphertext (secret message) has factor of \(n\). For each message received, \(c_i \in c\), calculate the \(e_i = Q_n(c_i)\).

- \(Q_n(c_i) = 1\), if the \(c_i\) is a quadratic residue mod \(n\),
- \(Q_n(c_i) = -1\), if the \(c_i\) is a nonquadratic residues mod \(n\).

Because \(n\) is a composite integer numbers \(p\) and \(q\) \((n = p \ * q)\), then \(c_i\) is a quadratic residue mod \(n\), if and only if \(c_i\) is a quadratic residue mod \(p\) and is a quadratic residue mod \(q\). If \(e_i = 1\), then the \(m_i = 0\), and if \(e_i = -1\), then the \(m_i = 1\). So, gained back the original message, i.e. \(m = m_1 \ m_2 \ m_3 \ldots m_l\) [5].

3. Results and Discussion

The relationship between the original text and the length of time the process of with encryption algorithm FEAL: this study tested the encryption process by as much as 5 x with the same key and different size of plaintext. The relationship between duration of encryption and different plaintext length shown in table 1.

| Testing | Size of plaintext(byte) | Encryption time (second) |
|---------|-------------------------|--------------------------|
| I       | 12                      | 0.0089731                |
| II      | 828                     | 0.1620483                |
| III     | 1021                    | 0.2150197                |
| IV      | 2472                    | 0.7545561                |
| V       | 4123                    | 0.6196318                |

Based on Table 1, it can be concluded that the longer the original text, the longer the encryption time. The correlation between the magnitude of the Goldwasser-Micali algorithm's private key value and the cipherkey result shows in Figure 1. The figure shows the relationship between the different Private Key Goldwasser-Micali algorithm values with the size of the cipherkey.
Figure 1. Graph of the relationship between different private key with the size of cipher key

Based on Figure 1, it can be concluded that the greater the private key, the larger the cipher key size. However, although the key and plain key entered are the same, the cipher key result is not necessarily the same. Table 2 shows the different pieces of cipher key results in the two tests.

| Testing I | Testing II |
|-----------|------------|
| 19790     | 89737      |
| 43579     | 88702      |
| 43955     | 5219       |
| 97305     | 91405      |
| 30401     | 34602      |
| 76136     | 96975      |
| 29001     | 78426      |
| 35552     | 93205      |
| 80302     | 76924      |
| 85934     | 22363      |
| 8094      | 53612      |
| 20230     | 32636      |
| 79244     | 64483      |
| 88369     | 74655      |
| 89539     |            |
| 66327     | 56543      |
| 30667     | 65129      |
| 80966     | 92146      |
| 19139     | 1383       |
| 39832     | 80283      |
| 17150     | 54121      |
| 21511     | 74999      |
| 66327     | 84987      |
| 30099     | 39192      |
| 24284     | 85980      |
| 16901     | 18455      |
| 45655     | 77181      |
| 79720     | 53026      |
| 5665      | 1560       |
| 80966     | 3609       |
| 30967     | 33593      |
| 28037     | 41683      |
| 53495     | 43934      |
| 89737     | 69737      |
| 88702     | 94043      |
| 5219      | 91028      |
| 91405     | 13051      |
| 34602     | 91405      |
| 96975     | 34602      |
| 78426     | 96975      |
| 93205     | 76924      |
| 22363     | 53612      |
| 32636     | 64483      |
| 64483     | 74655      |
| 39192     | 85980      |
| 85980     | 18455      |
| 11162     | 77181      |
| 85775     | 53026      |
| 1560      | 3609       |
| 33593     | 41683      |
| 43934     | 69737      |
| 94043     | 91028      |
| 13051     | 91405      |
| 91405     | 34602      |
| 76924     | 53612      |
| 22363     | 64483      |
| 74655     | 39192      |
| 85980     | 18455      |
| 77181     | 53026      |
| 1560      | 3609       |
| 41683     | 43934      |
| 69737     | 94043      |
| 94043     | 91028      |
| 13051     | 91405      |
| 91405     | 34602      |
| 76924     | 53612      |
| 22363     | 64483      |
| 74655     | 39192      |
| 85980     | 18455      |
| 77181     | 53026      |
| 1560      | 3609       |
| 41683     | 43934      |
| 69737     | 94043      |
| 94043     | 91028      |
| 13051     | 91405      |
| 91405     | 34602      |
| 76924     | 53612      |
| 22363     | 64483      |
| 74655     | 39192      |
| 85980     | 18455      |
| 77181     | 53026      |
| 1560      | 3609       |
| 41683     | 43934      |
| 69737     | 94043      |
| 94043     | 91028      |
| 13051     | 91405      |
| 91405     | 34602      |
| 76924     | 53612      |
| 22363     | 64483      |
| 74655     | 39192      |
| 85980     | 18455      |
| 77181     | 53026      |
| 1560      | 3609       |
| 41683     | 43934      |
| 69737     | 94043      |
| 94043     | 91028      |
| 13051     | 91405      |
| 91405     | 34602      |
| 76924     | 53612      |
| 22363     | 64483      |
| 74655     | 39192      |
| 85980     | 18455      |
| 77181     | 53026      |
| 1560      | 3609       |
| 41683     | 43934      |
| 69737     | 94043      |
| 94043     | 91028      |
| 13051     | 91405      |
| 91405     | 34602      |
| 76924     | 53612      |
| 22363     | 64483      |
| 74655     | 39192      |
| 85980     | 18455      |
| 77181     | 53026      |
| 1560      | 3609       |
| 41683     | 43934      |
| 69737     | 94043      |
| 94043     | 91028      |
| 13051     | 91405      |
| 91405     | 34602      |
| 76924     | 53612      |
| 22363     | 64483      |
| 74655     | 39192      |
| 85980     | 18455      |
| 77181     | 53026      |
| 1560      | 3609       |
| 41683     | 43934      |
| 69737     | 94043      |
| 94043     | 91028      |
| 13051     | 91405      |
Based on table 3, it can be concluded that the duration of the encryption and decryption process is independent of the value of the private key.

4. Conclusion

Implementation of hybrid cryptography, i.e., FEAL and Goldwasser-Micali algorithm in text security successfully done, which can encrypt the message and session key and original message can be recovered (decryption) appropriately.

The longer the message will be encrypted with the FEAL algorithm, the longer it will take to process the encryption.

The greater the private key value of the Goldwasser-Micali algorithm, the larger the resulting cipher key size.

The resulting cipher key will be different for every encryption process with the Goldwasser-Micali algorithm, even using the same private key and the same message (session key). This is due to the random selection of $x$ by the system for every resulted binary conversion from the plain key.

Key randomization performed by the Goldwasser-Micali algorithm improves the security of Feal algorithm. The Feal has the weakness, i.e., the operations involved in the encryption and decryption processes are the same. Only the order of its operations is reversed. Thus it is easy to hack.

5. References

[1] Mollin Richard A 2007 *An Introduction to Cryptography 2nd Ed* (Francis: Taylor and Francis Group Boca Raton)

[2] Smart N 1999 *Cryptography: An Introduction, 3rd Ed* (Bristol City)

[3] Garg 2014 Comparison of Asymmetric Algorithms in Cryptography *International Journal of Computer Science and Mobile Computing* 3 4 1190-1196

[4] Miyaguchi S and Shimizu A 1987 *Fast Data Encipherment Algorithm FEAL* http://link.springer.com

[5] Goldwasser S and Micali S 1984 Probabilistic Encryption *Journal of Computer and System Sciences* 28 270-299

Acknowledgments

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