An implementation of Rabin-p cryptosystem and affine cipher in a hybrid scheme to secure text

M A Budiman, Handrizal, and S Azzahra

Department of Computer Science, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Jl. Universitas No. 9-A, Kampus USU, Medan 20155, Indonesia

mandrib@usu.ac.id, handrizal_tanjung@yahoo.com, and sasmitaazzahra9@gmail.com

Abstract. One way to keep digital information secure is by using cryptographic techniques. With respect to its keys, cryptography generally is divided into two types: symmetric and asymmetric. Symmetric cryptography algorithm uses the same key in both encryption and decryption while asymmetric uses different keys for each of these processes. Asymmetric cryptography algorithm is known to be ineffective to encrypt big messages since it takes more CPU time to do the computation and the resulting ciphertext is usually much bigger than the original message. To eliminate these problems, in this research, we combine Rabin-p cryptosystem, an asymmetric cryptography algorithm, and affine cipher, a symmetric cryptography algorithm, in a hybrid scheme in order to secure messages from unwanted parties. The affine cipher is used for message encryption while Rabin-p is used for affine cipher’s key encryption. The result of this research is a hybrid cryptography scheme that can encrypt and decrypt message in the form of *.pdf files and be able to restore the message without losing integrity.

1. Introduction

In the past to present, a confidential message or information must be kept secret. Only certain individuals or groups may know the contents of the message. Cryptographic helps people to disguise their messages so only intended people can read them. Cryptography has been used since ancient times in Roman times by Julius Caesar by using a device called scytle. Scythe is a tool made of cylindrical papyrus [1]. Cryptography consist of the words “kryptos” and “graphein”. “Kryptos” means “hidden”, while “graphein” means “writing” [2]. The hidden (or “screambled”) message is called ciphertext, while the readable message is called plaintext. The process of scrambling a plaintext into a ciphertext is called encryption while the process of recovering the ciphertext back into the plaintext is called decryption. [3].

Rivest Shamir Adleman (RSA) algorithm was the first asymmetric algorithm to appear in 1978. In 1979 Rabin algorithm was introduced as a variant of RSA which has a faster encryption performance. On the other hand, Rabin algorithm may sometimes result in the failure of the decryption process [6] since we must choose which one is the right plaintext out of four possible decryption results. Rabin-p which is a Rabin-like cryptosystem was recently introduced. In the process of encryption and
decryption Rabin-p uses less storage compared to Rabin-Takagi and HIME(R) which is similar to Rabin's cryptosystem [4].

Caesar cipher is probably one of the oldest algorithms in the history of cryptography that uses substitution technique [5]. But in the Caesar cipher there is a weakness that we can quickly and easily get the original message by using brute force method. Affine cipher is an extension of Caesar cipher which uses transposition technique in addition to substitution [7] [8]. As a result, Affine cipher is considered securer than Caesar cipher.

2. Methods
In this section we explain the Rabin-p algorithm, the affine cipher, and the hybrid scheme for securing *.pdf file. The scheme is shown in Figure 1.

Based on Figure 1, the first step of using hybrid crypotsystem method is to generate Rabin-p public key and the Rabin-p private key by following this step [4].

- Choose two distinct prime numbers, p and q so that
  \[ 2^k < p, q < 2^{k+1} \] and \( p, q \equiv 3 \pmod{4} \). (1)

- Calculate the value of public key N by using this equation:
  \[ N = p^2 q \] (2)

- Save p as private key

After the keys of Rabin-p are generated, th

- e recipient (John) publishes his Rabin-p public key, which is N, so the sender (Rina) can use that public key to encrypt the key of the affine cipher. For encrypting the messages (the *.pdf file), Rina uses affine cipher by following these steps [7]:
  - Choose an integer \( m \) which is co-prime with \( N \) and choose \( b \) as a shifting factor which ranges from 0 until \( N - 1 \). \( N \) is the size of the encoding table. This research uses the 8-bit ASCII (American Standard for Information Interchange) as the encoding table, so \( N = 256 \).
  - Convert every 8-bit of the PDF file into its corresponding ASCII value; let it be \( P \)
  - Calculate the value of \( C \) by using this equation

\[ C = (P + b) \pmod{N} \]
\[ C = mP + b \pmod{N} \]  

After encrypting the whole file, Rina will encrypt the key of the affine cipher by using Rabin-p encryption algorithm by following these steps [4]:

- Let the affine key be a message \( m \) so that \( 0 < m < 2^{2k-1} \) where \( \gcd(m, N) = 1 \).
- Calculate the value of cipherkey \( c \) by using this equation:
  \[ c \equiv m^2 \pmod{N} \]  

After the key has been encrypted, the cipherkey \( c \) and the ciphertext \( C \) are sent to John.

Before decrypting the file, John must decrypt the affine key by using Rabin-p decryption algorithm by following these steps [4]:

- Calculate the value of \( w \) by using this equation:
  \[ w \equiv c \pmod{p} \]  

- Calculate the value of \( m_p \) by using this equation:
  \[ m_p \equiv w^{p+1} \pmod{p} \]  

- Calculate the value of \( i \) by using this equation:
  \[ i = \frac{c - m_p^2}{p} \]  

- Calculate the value of \( j \) by using this equation:
  \[ j \equiv \frac{i}{2m_p} \pmod{p} \]  

- Calculate the value of \( m_1 \) by using this equation:
  \[ m_1 = m_p + jp \]  

- If \( m_1 < 2^{2k-1} \) then \( m = m_1 \), else \( m = p^2 - m_1 \).

After the key has been decrypted, John can do the decryption of the *.pdf file by following these steps [7]:

- Calculate \( m^{-1} \pmod{N} \)
- Calculate the value of \( P \) by using this equation:
  \[ P = m^{-1} (C - b) \pmod{N} \]  

The value of each \( P \) is the ASCII value of each corresponding byte of the original message.

### 3. Results and Discussions

These experiments were conducted on the Android Studio 3.1.2 and the version of android is 7.1.2 (Xiaomi Redmi 5) CPU octa-core MAX 1.80GHz and RAM 2 GB. The result of the affine encryption is presented in Table 1 as follows:

| Sample Text in the PDF files | Cipher text |
|------------------------------|-------------|
| Science                      | Ñï_b=6[L    |
| Computer Science             | Ñï_b=6[LÔ!q/xq| |
| Computer Science             | Ñï_b=6[LÔ!q/xq|Ô_'_|
| USU                          | Ñï_b=6[LÔ!q/xq|Ô_'_|

The result of Rabin-p encryption is presented in Table 2 as follows:
Table 2. The result of encryption of affine key with the Rabin-\(p\) algorithm

| Sample Affine Key | Cipherkey |
|-------------------|-----------|
| 9                 | 3249      |
| 83                | 3136 2601 |
| 117               | 2401 2401 3025 |

Table 2 shows that the cipherkey (the encrypted affine key) is larger than the affine key. It makes the encryption time longer. The running time for encryption and decryption of affine cipher based on the size of the *.pdf files is presented in Figure 2 as follows:

Figure 2. Running time of affine cipher in the encryption and decryption of *.pdf files

In Figure 2, the size of documents are 182 kb (195 words), 298 kb (1525 words), 378 kb (24303 words), and 443 kb (34467 words). The result running time encryption and decryption of affine cipher is the bigger the size, the longer time it takes. The encryption and the decryption time is almost the same. The running time of encryption and decryption of Rabin-\(p\) is presented in Figure 3:

Figure 3. Running time of Rabin-\(p\)

In Figure 3, the key lengths of affine cipher are 2, 3, and 4 digits.
Figure 4 (a) Before encryption (*.pdf) (b) After encryption (saved as *.txt) (c) After decryption (*.pdf)
Figure 4 shows the content of a PDF before encryption, after encryption, and after decryption. The detailed are the encrypted content of the *.pdf file is saved as *.txt, and the decrypted *.txt file is restored to the original *.pdf file while maintaining its integrity.

4. Conclusions
In this research, we have developed a hybrid scheme where affine cipher is used to encrypt and decrypt the *.pdf file, while Rabin-p algorithm is used to encrypt and decrypt the key of affine cipher. The result both of encryption and decryption is that the longer the message length, the longer the time it takes for the encryption and decryption process in affine cipher. The *.pdf file is restored without loss of integrity.

References

[1] Ariyus D 2008 Pengantar Ilmu Kriptografi. (Yogyakarta:Andi Offset)
[2] Schneier B 1996 Applied Cryptography 2nd Ed John Wiley & Sons, Inc
[3] Mollin R A 2007 An Introduction to Cryptography 2nd Ed Chapman & Hall/CRC: Boca Raton,Florida
[4] Asbullah M A, Arifin M R K and Zahari M 2016 Analysis on The Rabin-p Cryptosystem AIP Conference Proceedings 1787
[5] Mohammed B B 2013 Automatic Key Generation of Caesar Cipher Int. J. of Eng. Trends and Technology (IJETT) 6 6
[6] Asbullah M A and Ariffin M R K 2014 Rabin-p Cryptosystem: Practical and Efficient Method for Rabin based Encryption Scheme 1-13
[7] Tudunkaya S M 2015 Design of the Affine Ciphers of Rhotrices IEEE Transaction on Information Theory 8 79-82
[8] Rachmawati D and Budiman M A 2017 New approach toward data hiding by using affine cipher and least significant bit algorithm 2017 4th Int. Conf. on Comp. Applications and Information Processing Technology (CAIPT)