A Hybrid Cryptosystem Using Vigenère Cipher and Rabin-\(p\) Algorithm in Securing BMP Files

Mohammad Andri Budiman*, Muhammad Yogi Saputra, Handrizal

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

Abstract. Vigenère cipher is a classical cryptography algorithm and similar to other classical algorithms, it produces smaller but less secure ciphertexts than a public key cryptography algorithm. Meanwhile, Rabin-\(p\) is a public key cryptography algorithm with a stronger encryption than Vigenère cipher. Nevertheless, as a public key algorithm, Rabin-\(p\) is inefficient to encrypt vast amounts of messages such as BMP image files, since the size of the cipherimages will increase manyfold and this would lead to a problem in storing and sending the cipherimages. To overcome these problems, in this study, we combined the Vigenère cipher and the Rabin-\(p\) algorithm in a hybrid cryptosystem scheme. In the experiment, the Vigenère cipher was used to encrypt the BMP files and the Rabin-\(p\) algorithm was used to encrypt the Vigenère keys. The result showed that the size of the cipherimages did not increase and the decryption procedure could recover the original BMP files while maintaining their integrity.

Keyword: Vigenère cipher, Rabin-\(p\), hybrid cryptosystem, BMP file.

Received 3 June 2020 | Revised 26 June 2020 | Accepted 31 July 2020

1 Introduction

A confidential message must be encrypted in such a way so that only intended parties can recover it back and read it. To ensure confidentiality, in the past, people secured data using classical cryptographic algorithms which all belong to the class of symmetric algorithms. Symmetric algorithms are cryptography algorithms that use a single key for both encryption and decryption procedures [1][2]. Their security depends solely on the secrecy of the key; if the key used can be tracked down or guessed methodically by unwanted parties, then the whole message can easily be decrypted by them. This results in messages being vulnerable to theft. Then came the modern time, public key cryptography algorithms are now used everywhere. Public key cryptography algorithm has two distinct keys: private key and public key. The public key is used to encrypt messages into obfuscated form and the private key is used to decrypt the
obfuscated messages back into the original massages [3]. In the public key encryption scheme, both public key and private key are generated by the recipient of the message. The public key should be known to everyone and the private key should be known only to the recipient [4]. The security of the public key encryption algorithm depends on the size of the key: the longer the key, the securer the algorithm. But the size of key also has its consequence: the longer the key, the longer the time to encrypt and decrypt, and the longer the resulting ciphertext. Thus, encrypting vast amount of data with public key encryption scheme is deemed to be inefficient.

Vigenère cipher is a classical polyalphabetic substitution cipher, *i.e.*, a cipher that utilizes multiple substitution characters. Similar to other classical ciphers, Vigenère cipher produces small ciphertexts: the size of its ciphertext is always equal to the size of its plaintext. The drawback of Vigenère cipher is that it is easy nowadays to attack it by some analytical methods, such as the Kasiski method, Friedman test, and the twist algorithm [5].

Rabin algorithm [6] is a public key cryptography algorithm that uses Blum integer $n$ as its modulus. Recall that if $n$ is a Blum integer, then $n = p \times q$, $p, q \in PRIMES$, $p \neq q$, and $p \equiv q \equiv 3 \, (mod \, 4)$. Rabin algorithm has an advantage over the well-used public key algorithm, the RSA, *i.e.* the encryption procedure of the Rabin algorithm is faster than that of RSA algorithm, since Rabin only uses exponent 2 while RSA uses exponent $e$ which is much greater than 2. However, the decryption procedure in Rabin algorithm results in four results, and only one of them is the original message [7]; thus, this may lead to confusion on the side of the recipient.

Rabin-$p$ algorithm is a variant of Rabin algorithm which was designed by M. A. Asbullah and M. R. K. Ariffin [8]. One obvious advantage of this algorithm is that the decryption procedure of the Rabin-$p$ algorithm only results in one result, *i.e.*, the original message; and, therefore, there is no more confusion on the side of the recipient about finding out the original message out of four decryption results as in the Rabin algorithm. The other advantages of the Rabin-$p$ algorithm are that this algorithm does not need to use Jacobi symbol [9] and since it also does not need to use Chinese Remainder Theorem computation, this cryptosystem cannot be compromised by Novak’s attack [8] [10].

Earlier, it has been mentioned that public key cryptography is inefficient to encrypt large messages, such as BMP files, since the size of the cipherimages will increase manyfold and this would lead to a problem in storing and sending the cipherimages. It is also been mentioned that classical cryptography such as Vigenère cipher is prone to various kinds of attacks. In this study, we combined the Rabin-$p$ algorithm and Vigenère cipher in a hybrid scheme in order to solve these problems. In this scheme, the key of the Vigenère cipher is encrypted with Rabin-$p$ algorithm while the BMP file is encrypted with Vigenère cipher.
2 Methods

In this section, we explain the Vigenère cipher, the Rabin-p algorithm, and the proposed hybrid cryptosystem along with the example of encryption and decryption procedures.

2.1 The Vigenère cipher

The Vigenère cipher was developed by Blaise de Vigenère in 1583. The Vigenère cipher uses a defined square matrix termed as tabula recta (which is also known as Vigenère square or Vigenère table) and a series of keys to encrypt a plaintext. Since during the encryption procedure every character of the plaintext is substituted with different key, the Vigenère cipher belongs to the class of polyalphabetic cipher. By far, the most widely known polyalphabetic substitution cipher is, arguably, the Vigenère cipher [11]. Compared to a monoalphabetic substitution cipher such as Caesar cipher and Affine cipher, the Vigenère cipher is considered securer since it is harder to be compromised by a method called frequency analysis [2]. The formula for encryption and decryption of the Vigenère cipher are as follows.

Encryption:

\[ C_i \equiv (P_i + K_i) \pmod{N} \]

Decryption:

\[ P_i \equiv (C_i - K_i) \pmod{N} \]

Where:

- \( C_i \) = The decimal value of the \( i^{th} \) ciphertext character
- \( P_i \) = The decimal value of the \( i^{th} \) plaintext character
- \( K_i \) = The decimal value of the \( i^{th} \) key character
- \( N \) = The length of the tabula recta

We shall now show below how to encrypt and decrypt one pixel of a BMP file.

A. The Vigenère Cipher Encryption (Example)

An example of using Vigenère cipher to encrypt a pixel of a BMP file is as follows:

1. Suppose that the pixel of the BMP file has these RGB values:

\[ \text{red} = 185; \text{green} = 255; \text{blue} = 125. \]

2. In Vigenère cipher, these values become:

\[ P_1 = 185; P_2 = 255; P_3 = 125 \]

3. We choose some Vigenère keys:
4. Using the encryption formula $C_i \equiv (P_i + K_i) \pmod{N}$, we compute:

\[
C_1 = (P_1 + K_1) \pmod{256} = (185 + 85) \pmod{256} = 270 \pmod{256} = 14
\]
\[
C_2 = (P_2 + K_2) \pmod{256} = (255 + 91) \pmod{256} = 310 \pmod{256} = 90
\]
\[
C_3 = (P_3 + K_3) \pmod{256} = (125 + 155) \pmod{256} = 220 \pmod{256} = 24
\]

The ciphertext $C = (14, 90, 24)$ is the corresponding RGB value in the cipherimage. Note that modulo 256 was used in the computation since we encrypted a pixel of 8-bit BMP where every R, G, and B can have $2^8 = 256$ different values.

B. The Vigenère Cipher Decryption (Example)

The decryption procedure of the cipherimage is as follows:

1. In the cipherimage, we have a pixel with these values:

\[
C_1 = 14; C_2 = 90; C_3 = 24
\]

2. From secured channel, we have the corresponding keys:

\[
K_1 = 85; K_2 = 91; K_3 = 155
\]

3. Using the decryption formula $P_i \equiv (C_i - K_i) \pmod{N}$, we compute:

\[
P_1 = (C_1 - K_1) \pmod{256} = (14 - 85) \pmod{256} = 185
\]
\[
P_2 = (C_2 - K_2) \pmod{256} = (90 - 91) \pmod{256} = 255
\]
\[
P_3 = (C_3 - K_3) \pmod{256} = (24 - 155) \pmod{256} = 125
\]

Thus, $P = (185, 255, 125)$ is the corresponding pixel value of the decrypted image.

2.2 The Rabin-$p$ Algorithm

The Rabin-$p$ algorithm was invented in Malaysia by M. A. Asbullah and M. R. K. Ariffin [8] as a variant of Rabin algorithm. The Rabin algorithm has four decryption results; thus, without additional information, it would be confusing for a recipient to decide which one of these four is the original message. The Rabin-$p$ algorithm solves this problem since its decryption procedure only results in one result, i.e., the original message. The Rabin-$p$ algorithm has other properties: it does not rely on Jacobi symbol [9] and it does not utilize Chinese Remainder Theorem (CRT) computation. Since CRT is avoided, the Rabin-$p$ algorithm cannot be compromised by Novak’s
attack [8] [10]. In the following, we will describe the procedures for key generation, encryption, and decryption of the Rabin-p algorithm.

The Rabin-p key generation procedure is as follows [9]:

1. Choose $k$, the security parameter.
2. Generate two random and different prime numbers, $p$ and $q$, such that $2^k < p, q < 2^{k+1}$, and $p \equiv q \equiv 3 \pmod{4}$.
3. Calculate $N = p^2 q$.
4. Keep the value of $p$ as private key. (Although the authors [9] suggest to only keep the value of $p$ as private key, we recommend to also keep the value of $q$ as secret).
5. Publish $N$ as public key.

The Rabin-p encryption procedure is as follows [9]:

1. Obtain the public key, $N$.
2. Take a message $m$ such that $0 < m < 2^{2k-1}$ and $\gcd(m, N) = 1$.
   In the proposed hybrid cryptosystem, $m$ is the key of the Vigenère cipher.
3. Calculate $c \equiv m^2 \pmod{N}$.
4. Send $c$ to the recipient

The Rabin-p decryption procedure is as follows [9]:

1. Obtain the value of $c$.
2. Calculate $w \equiv c \pmod{p}$.
3. Calculate $m_p \equiv w^{-1} \pmod{p}$.
4. Calculate $i = \frac{c - m_p^2}{p}$
5. Calculate $j \equiv \frac{i}{2m_p} \pmod{p}$
6. Calculate $m_1 = m_p + jp$
7. If $m_1 < 2^{2k-1}$ then return $m = m_1$
8. Else, return $m = p^2 - m_1$

A. The Rabin-p Key Generation (Example)
1. The recipient chooses $k = 7$ as the security parameter. (For real-world usage, we recommend choosing much bigger security parameter, e.g., $k > 512$).
2. The recipient generates $p = 131, q = 43$, which satisfy $2^k < p, q < 2^{k+1}$, and $p \equiv q \equiv 3 \pmod{4}$. 
3. The recipient calculates \( N = p^2 q = 131^2 \times 43 = 737923 \).

4. The recipient keeps \( p \) as private key. In addition, the recipient also keeps the value of \( q \) as per recommendation.

5. The recipient publishes the value of \( N \).

B. The Rabin-p Encryption (Example)

1. The sender obtains the value of \( N = 737923 \).

2. The sender chooses a message \( m \). In the proposed hybrid cryptosystem, \( m \) is the key of Vigenère cipher. Let say that \( m = K_1 = 85 \), which satisfies \( 0 < m < 2^{2k-1} \) and \( \text{GCD}(m, N) = 1 \).

3. The sender calculates the cipherkey, \( c = m^2 \mod N = 85^2 \mod 737923 = 7225 \).

4. The sender sends the value of \( c \) to the recipient.

C. The Rabin-p Decryption (Example)

1. The recipient receives the value of \( c = 7225 \).

2. The recipient calculates \( w = c \mod p = 7225 \mod 131 = 20 \).

3. The recipient calculates \( m_p = w^{p+1} \mod p = 20^{131+1} \mod 131 = 46 \).

4. The recipient calculates \( i = \frac{c-m_p^2}{p} \mod \frac{131}{2m_p} = \frac{7225-46^2}{131} = \frac{5109}{131} = 39 \mod 131 = 130 \).

5. The recipient calculates \( j = \frac{39}{92} \mod \frac{131}{131} = 39.92^{-1} \mod 131 = 39.47 \mod 131 = 130 \).

6. The recipient calculates \( m_1 = m_p + jp = 46 + 130.13 = 17076 \).

7. Since \( m_1 > 2^{2k-1} \) then \( m = p^2 - m_1 = 131^2 - 17076 = 85 \).

We have shown that the Rabin-p decryption procedure has recovered back the value of the Vigenère cipher’s key, \( K_1 = 85 \).

3 Result and Discussions

In this section, we will show the result of using the Vigenère cipher and the Rabin-p algorithm in a hybrid scheme to secure some BMP files. The Vigenère cipher’s keys of each BMP file were encrypted and decrypted using the Rabin-p algorithm, while the BMP files were encrypted and decrypted using the Vigenère cipher.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>BMP Encryption Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plainimage</td>
<td>Cipherimage</td>
</tr>
<tr>
<td>image5.bmp</td>
<td>cipherimage5.bmp</td>
</tr>
<tr>
<td>259 x 194 pixel</td>
<td>259 x 194 pixel</td>
</tr>
<tr>
<td>147 KB</td>
<td>147 KB</td>
</tr>
<tr>
<td>Image Name</td>
<td>Cipher Image Name</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>image3.bmp</td>
<td>cipherimage3.bmp</td>
</tr>
<tr>
<td>236 x 236 pixel</td>
<td>236 x 236 pixel</td>
</tr>
<tr>
<td>163 KB</td>
<td>163 KB</td>
</tr>
<tr>
<td>image2.bmp</td>
<td>cipherimage2.bmp</td>
</tr>
<tr>
<td>355 x 245 pixel</td>
<td>355 x 245 pixel</td>
</tr>
<tr>
<td>255 KB</td>
<td>255 KB</td>
</tr>
<tr>
<td>image6.bmp</td>
<td>cipherimage6.bmp</td>
</tr>
<tr>
<td>500 x 300 pixel</td>
<td>500 x 300 pixel</td>
</tr>
<tr>
<td>439 KB</td>
<td>439 KB</td>
</tr>
<tr>
<td>image4.bmp</td>
<td>cipherimage4.bmp</td>
</tr>
<tr>
<td>488 x 488 pixel</td>
<td>488 x 488 pixel</td>
</tr>
<tr>
<td>697 KB</td>
<td>697 KB</td>
</tr>
<tr>
<td>image1.bmp</td>
<td>cipherimage1.bmp</td>
</tr>
<tr>
<td>439 x 600 pixel</td>
<td>439 x 600 pixel</td>
</tr>
<tr>
<td>773 KB</td>
<td>773 KB</td>
</tr>
</tbody>
</table>

From Table 1, it can be seen that all cipherimages cannot be comprehend by normal eyes and the size of each cipherimage is the same as the size of its corresponding plainimage. Since the size did not increase, there would be an efficiency in storing and sending the cipherimages, and, therefore, one research problem is solved.
From Figure 1, it can be deduced that the correlation between the size of BMP file and encryption time is directly proportional as might be expected.

**Table 2** BMP Decryption Result

<table>
<thead>
<tr>
<th>Cipherimage</th>
<th>Plainimage</th>
<th>Decryption Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>cipherimage5.bmp</td>
<td>plainimage5.bmp</td>
<td>10.3747</td>
</tr>
<tr>
<td>259 x 194 pixel</td>
<td>259 x 194 pixel</td>
<td>147 KB</td>
</tr>
<tr>
<td>cipherimage3.bmp</td>
<td>plainimage3.bmp</td>
<td>10.8662</td>
</tr>
<tr>
<td>236 x 236 pixel</td>
<td>236 x 236 pixel</td>
<td>163 KB</td>
</tr>
<tr>
<td>cipherimage2.bmp</td>
<td>plainimage2.bmp</td>
<td>11.0448</td>
</tr>
<tr>
<td>355 x 245 pixel</td>
<td>355 x 245 pixel</td>
<td>255 KB</td>
</tr>
<tr>
<td>cipherimage6.bmp</td>
<td>plainimage6.bmp</td>
<td>13.0079</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
From Table 2, it can be seen that each cipherimage which cannot be comprehend by normal eyes can be recovered back to its corresponding plainimage. The image integrity has been maintained, and, thus, another research problem is solved. Comparing Table 1 and Table 2, we may deduce that encryption takes less time than decryption.

<table>
<thead>
<tr>
<th>500 x 300 pixel</th>
<th>6.bmp</th>
</tr>
</thead>
<tbody>
<tr>
<td>439 KB</td>
<td>500 x 300 pixel</td>
</tr>
<tr>
<td>439 KB</td>
<td>439 KB</td>
</tr>
<tr>
<td>cipherimage4.bmp</td>
<td>plainimage4.bmp</td>
</tr>
<tr>
<td>488 x 488 pixel</td>
<td>488 x 488 pixel</td>
</tr>
<tr>
<td>697 KB</td>
<td>697 KB</td>
</tr>
<tr>
<td>cipherimage1.bmp</td>
<td>plainimage1.bmp</td>
</tr>
<tr>
<td>439 x 600 pixel</td>
<td>439 x 600 pixel</td>
</tr>
<tr>
<td>773 KB</td>
<td>773 KB</td>
</tr>
</tbody>
</table>

From Table 2, it can be seen that each cipherimage which cannot be comprehend by normal eyes can be recovered back to its corresponding plainimage. The image integrity has been maintained, and, thus, another research problem is solved. Comparing Table 1 and Table 2, we may deduce that encryption takes less time than decryption.

**Figure 2** Decryption time (in milliseconds) of various sizes of cipherimages

From Figure 2, one may deduce that the correlation between the size of cipherimages and decryption time is directly proportional.
4 Conclusion

From the experiment results, we may conclude that:

1. A hybrid cryptosystem using Vigenère cipher and Rabin-p algorithm in securing BMP files has been implemented. In this cryptosystem, the key of the Vigenère cipher is encrypted with Rabin-p algorithm while the BMP file is encrypted with Vigenère cipher.

2. The size of the cipherimage file is the same as the size of its corresponding BMP file. Because there is no increase in the size, there would be an efficiency in storing and transmitting the cipherimages.

3. The correlation between the size of BMP file and its encryption time is directly proportional.

4. The decryption procedure can recover the cipherimage to its corresponding plainimage without loss of integrity.

5. The correlation between the size of cipherimage and its decryption time is directly proportional.

6. The encryption time takes less time than decryption time. This is due to the fact that there are more steps and complexities of computation in the encryption procedure than in the decryption procedure.

REFERENCES


