# AN IMPLEMENTATION OF RABIN CRYPTOGRAPHY AND FIBONACCI CODES ALGORITHM IN IMAGE FILES SECURITY AND COMPRESSION

[1]DIAN RACHMAWATI, [2]AMALIA AMALIA, [3]AKTUALITAS GULO

[1,2,3]Departemen Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi,

Universitas Sumatera Utara, Jl. Universitas No. 9-A, Kampus USU, Medan 20155, Indonesia

E-mail: [1]dian.rachmawati@usu.ac.id

## ABSTRACT

Image files are vulnerable to being fabricated, modified, and duplicated, which violate the copyright. Therefore, it is needed Rabin cryptography method for securing and assuring the confidentiality of the image file. Rabin cryptography algorithm is an asymmetric algorithm. The advantage of this algorithm is a simple encryption process so that it can reduce processing time, even though the used resource is limited. Encryption process in the Rabin algorithm causes enlargement of the data size, moreover, the capacity of the image file that tends to be large, it can cause to the slowing down of the information exchange process. So it is needed Fibonacci codes method to reduce or compress the data size. Fibonacci codes is a compression algorithm that uses an array of the Fibonacci integer to code the bit value from the data or file which compressed. In this research, the writer combines the Rabin cryptography algorithm for securing image files and Fibonacci codes for compressing data. This research will be done in two schemes test. The first scheme which will be done that the image file is encrypted first with Rabin cryptography algorithm, then the result of encryption is compressed with the algorithm of the Fibonacci code. And the second scheme which will be done that image file is compressed first with Fibonacci codes, then the result of the compression is encrypted with the Rabin cryptography algorithm. Image file to be tested with extension are *.JPG, *.PNG, *.BMP. System implementation uses C# programming language. The result of this research shows that the Rabin cryptography method can secure originality and confidentiality of data with the first scheme and the second scheme, whereas the Fibonacci codes method is ineffectively to reduce data size with extension *.JPG and *.PNG in the second scheme. Fibonacci codes method produce RC (Ratio Compression), CR (Compression of Ratio), and SS (Space Saving) with each values are 0.89, 115,52%, -15,53% in average. While in file with *BMP extension Fibonacci codes method can be used effectively to reduce the data size, where this method Rc = 1,4364, Cr = 70,98%, and SS = 27,47 % in average.

**Keywords:** *Fibonacci codes, Rabin, Cryptography, Compression*

## 1. INTRODUCTION

In the current digital era, the exchange of data or information in the form of images is increasingly massive. Image data represents various information or data in the form of image visualization, so that information or data is easier to be understood and remembered by all users of multimedia devices. The more image data is used, the threat of abuse of this data is increasing.[6] For this reason, a data security method or technique is needed, especially image data that is used to prevent acts of abuse by outside parties. This method of securing data is known as cryptography.[5]

Cryptography is the science of encryption techniques in which data is randomized using an encryption key into something that is difficult to read by someone who does not have a decryption key. So, the notion of modern cryptography is a set of techniques that provide information security.[1]

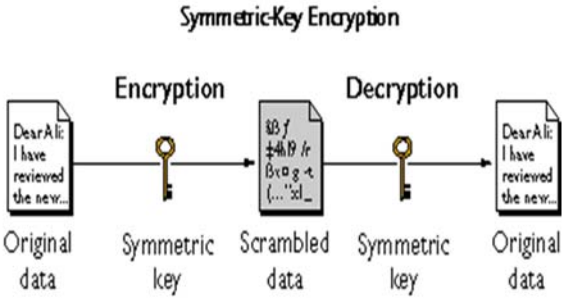
An image is a spatial representation of an actual object in a two-dimensional plane, usually written in a cartesian x-y coordinate, and each coordinate represents one of the smallest signals of an object.
Digital images contain arrays of various pixel values. In pixels digital images, the environment is correlated, and so these pixels contain excessive bits. By using a redundant bit compression algorithm is removed from the picture, so the size of the image size is reduced and compressed images [17].
Data compression is a technique to reduce the size of the data so that storage is much denser and also to reduce the time of data transmission.
The compression algorithm used in this research is Fibonacci codes. The Fibonacci compression method is a type of compression method by forming coding through a universal code system where a positive integer value of data is used to create a binary code-word using the Fibonacci number.
The cryptographic method that will be applied in this study is the Rabin cryptographic system. This cryptographic method is a variant and development of the RSA cryptographic algorithm. The cryptographic system was first published by Michael O. Rabin in 1979. The Rabin cryptographic system is one of the asymmetric cryptographic algorithms that use a key pair, namely the public key and the private key in the encryption and decryption process. The encryption process in this cryptographic method is simpler, so it can save time even if the resources used are limited.

## 2. METHODS

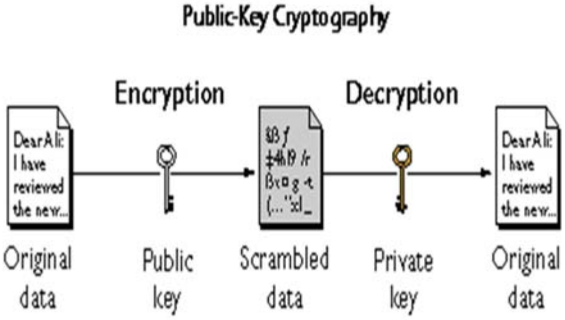
### 2.1 CRYPTOGRAPHY

Cryptography comes from the Greek language "cryptos," which means secret and "graphein" which means writing. So cryptography means secret writing [13]. Cryptography also means the science of using mathematical calculations to convert information to safety and provide immunity against the attackers of that information [15]. Cryptography is crucial because it allows all processes, transactions, and communications to be done electronically. This is very important in communicating personal information that is vulnerable to distortion. It protects your messages or essential information, as well as your personal possessions.
The main objectives of cryptography [7] are:
1. Confidentiality is a service intended to keep messages from being read by unauthorized parties.
2. Integrity is a service that guarantees that messages are still original/intact or have never been manipulated during the transmission.
3. Authentication is a service related to identification, both identifying the truth of the parties communicating and recognizing the fact of the message source [9].
4. Non-repudiation is a service to prevent the communicating entity from denying the information sent [12].
A. Types of Cryptographic Algorithms
In general, there are two types of cryptography based on the key, namely the Symmetric Algorithm and the Asymmetric Algorithm.

1.Symmetric Algorithm
The symmetric algorithm is an algorithm that has the same encryption and decryption key [10]. In a symmetrical cryptographic system, the key for encryption is the same as the decryption key. The method of symmetry cryptography assumes the sender and recipient of the message have shared the same key before exchanging messages. The security of this cryptographic system lies in its key confidentiality.



2. Asymmetric Algorithm
The asymmetric algorithm is an algorithm that has two keys, namely the public key to encrypt while the private key to decrypt it [14]. Another term of this cryptographic scheme is public-key cryptography. In this algorithm, everyone who communicates has a key pair. The sender encrypts the message with the public key, and only the recipient has a private key that can know or open the message's content.



## 2.2 RABIN ALGORITHM

Rabin algorithm is one of the asymmetrical cryptographic systems in which the ability of its capabilities is mathematically evidenced by the rapid method of factoring of numbers to the present unresolved [4]. Rabin algorithm is a variation or development of the RSA cryptographic system. At first, the algorithm is considered to be theoretical only, but in the order of this cryptographic system has a fatal gap, so this system is not good to apply because it is not reliable in the face of a chosen-plaintext attack [11].

A. Key generation
The following is Rabin's algorithm key generation steps [8]:
1. Specify two pieces of prime integer p and q (p ≠ q), where both numbers are congruent with three mod 4, p ≡ q ≡ 3 (mod 4). Both numbers p and q are kept secret and used as private keys
2. Count the number n = pq, where the number n is the public key.

B. Encryption
In the encryption process on the Rabin algorithm, only the public key n is used. The ciphertext is achieved with the equation,
$c = M^2$ mod n, where m as plaintext or message sent, and n is the public key.

C. Decryption
In the decryption process, the Rabin cryptographic system generates 4 number of selections. Among the four numbers, there is one that is the actual result or plaintext value of the previous encryption results [13]. Here are the steps to decrypt the Rabin cryptographic algorithms:
1. Take the private key p and Q, which has been raised and saved.
2. Define the Yp and Yq values that are the GCD (Greatest Common Divisor) of P and Q by using the Extended Euclideaalgorithmhm. Since the GCD of P and Q is 1, it can be written as follows:
Yp * p + Yq * q = 1
3. Calculate the square root value of the ciphertext against the P and Q values with the formula:

$$m_p = c^{(p+1)/4} \; mod \; p$$

and

$$m_q = c^{(q+1)/4} \; mod \; q$$

4.Calculate the value of r, s, t and u by using Chinese Remainder Theorem, with the following equation:

$$r = \left(\left(Y_p * p * m_q\right) + \left(Y_q * q * m_p\right)\right) \; mod \; n$$

$$s = \left(\left(Y_p * p * m_q\right) - \left(Y_q * q * m_p\right)\right) \; mod \; n$$

$$t = \left(-\left(Y_p * p * m_q\right) + \left(Y_q * q * m_p\right)\right) \; mod \; n$$

$$u = \left(-\left(Y_p * p * m_q\right) - \left(Y_q * q * m_p\right)\right) \; mod \; n$$

5. Because Rabin generates four possible outcomes, to determine the correct decryption value, then the first way is to take the smallest value of the four obtained values or by making a bit comparison (after the value is converted to binary), if one of the four values has an identical binary order (if the binary value is divided into two parts, then there is a similarity of value) then it is certain that the value is the result of decryption.

A. Data Compression
In the era of technological development that is developing very rapidly now, and the size of the data used by humans will definitely require a very large place. Today many programmers are trying to reduce data (data compression). The purpose of reducing data is to save storage (memory), so that storage does not exceed capacity, and the process of data transfer will be faster.
Compression is a step or process to change the input data flow to a new data flow that has a smaller data size (compressed data) [2].
The data compression algorithm has two outlines, namely Lossless compression and Lossy compression [3].

1. Lossless compression is a method to do data compression that allows the original data to be reorganized in full, without losing any information. Lossless compression has a lower degree compression, but with data, accuracy maintained between before and after the compression process. Examples of these methods are Elias Delta Code, Elias Gama Code, Levenstein Code, Shanno Fano Coding, Run Length Encoding, and others. The basic concept of Lossless compression can be seen in Figure 1.

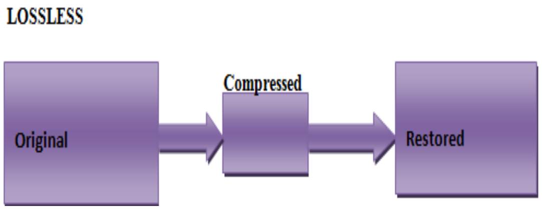


*Figure 1. Lossless compression*

2. Lossy compression is a data method that does not produce original data before compression. Although the difference is quite close, this type of compression is not suitable for data compression such as text data but is often used in audio, file, and image data. This method provides a higher degree of compression. Examples of this method are to Transform Coding, Swallow, and others. The basic concept of loss compression can be seen in Figure 2.

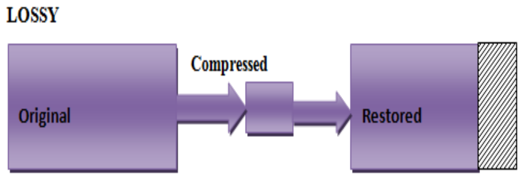


*Figure 2. Lossy Compression*

## 2.3 FIBONACCI CODES ALGORITHM

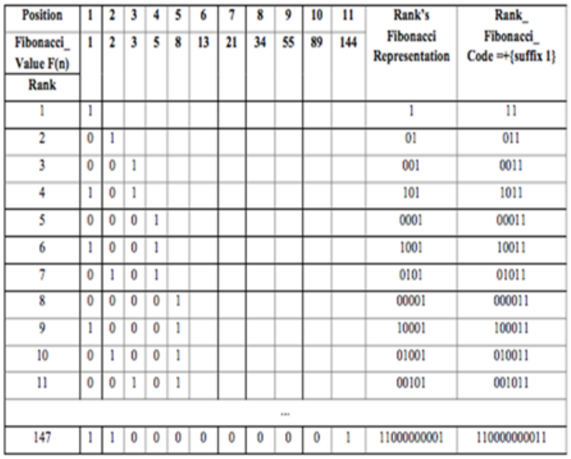
Fibonacci Codes is a universal code that can encode a positive integer into a binary in the form of a code-word. All tokens will end with the number "11" and will not contain the number "11" before the end of the code-word.

To do encoding an integer n with Fibonacci codes can be done with the following stages:

1. Look for a positive integer n more large or equal to 1.
2. Find the largest F Fibonacci number smaller or equal to n, reduce the n value to F, and record the remainder of the N-value reduction with F.
3. If the deductible number is the number contained in the Fibonacci sequence F (i), add the number "1" to I = 2, in the Fibonacci code to be formed.
4. Repeat step 2, change the value of N with the rest of the N value reduction with F until the rest of n value reduction with F is 0.
5. Add the number "1" on the rightmost position of the Fibonacci code to be set.

The integer encoding process in the Fibonacci Codes algorithm can be seen in Table 1 [19].

Table 1. The Fibonacci Codes table



| Position | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | Rank's Fibonacci Representation | Rank_ Fibonacci_ Code =+{suffix 1} |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Fibonacci_ Value F(n) | 1 | 2 | 3 | 5 | 8 | 13 | 21 | 34 | 55 | 89 | 144 | | |
| Rank | | | | | | | | | | | | | |
| 1 | 1 | | | | | | | | | | | 1 | 11 |
| 2 | 0 | 1 | | | | | | | | | | 01 | 011 |
| 3 | 0 | 0 | 1 | | | | | | | | | 001 | 0011 |
| 4 | 1 | 0 | 1 | | | | | | | | | 101 | 1011 |
| 5 | 0 | 0 | 0 | 1 | | | | | | | | 0001 | 00011 |
| 6 | 1 | 0 | 0 | 1 | | | | | | | | 1001 | 10011 |
| 7 | 0 | 1 | 0 | 1 | | | | | | | | 0101 | 01011 |
| 8 | 0 | 0 | 0 | 0 | 1 | | | | | | | 00001 | 000011 |
| 9 | 1 | 0 | 0 | 0 | 1 | | | | | | | 10001 | 100011 |
| 10 | 0 | 1 | 0 | 0 | 1 | | | | | | | 01001 | 010011 |
| 11 | 0 | 0 | 1 | 0 | 1 | | | | | | | 00101 | 001011 |
| ... | | | | | | | | | | | | | |
| 147 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 11000000001 | 110000000011 |

2.4 Digital Image

Digital images are usually referred to as discrete images in which images are generated through the process of digitizing continuous/analog images. The digital image is also a matrix at the row and column index leaving a point on the image, and the matrix element (which is called the image / pixel / image element / pels) determines the color / gray level at that point. Digital image expressed by a matrix set n x m (row = n, column = m) [18].

## 3. IMPLEMENTATION

In this implementation, there are three main menus, consist of Key generator menu to generate the public and private key, the Sender menu for encrypting and compress the image file, and the Recipient menu for decompressing and decrypting data.

The general system architecture is a general description of how the system or software will be built. In this study, the public architecture will be divided into two schemes, namely encryption-compression schemes and compression-encryption schemes, which can be seen in the following figure 3 (a) (b).

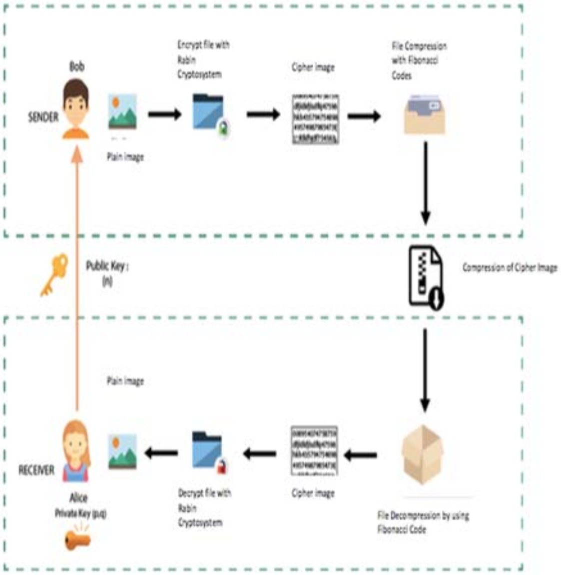

Bob
SENDER
Plain Image
Encrypt file with Rabin Cryptosystem
Cipher image
File Compression with Fibonacci Codes
Public Key : (n)
Compression of Cipher Image
Plain image
RECEIVER
Alice
Private Key (p,q)
Decrypt file with Rabin Cryptosystem
Cipher image
File Decompression by using Fibonacci Code

*Figure 3(A). First Illustrations Of The Application*

Figure 3(a) Describes the general architecture of the system for the first Test, which is the order of the encryption process, compression, decompression, and decryption of the file. In the system architecture visualization above, Bob is represented as the sender/sender, and Alice is represented as a receiver/receiver. The process that occurs in the above scheme is:

1. Bob, as the sender, prepares a message in the form of image/image files to be sent to Alice. The imagery message is then secured or encrypted by Rabin's cryptographic methods, via a previously raised public key. The encrypted file then becomes ciphertext.

2. The ciphertext File is then compressed with Fibonacci Codes method.

3. Once the file is received by Alice as a receiver, the next process is that the receiver decompresses against the ciphertext file that has been

Compressed. The decompression process uses the Fibonacci Codes method.

4. After going through the decompression process, the ciphertext file is then decrypted using the private key previously sent by the sender/Bob, so that the file returns to the original message or the original image (plaintext/plain Image).

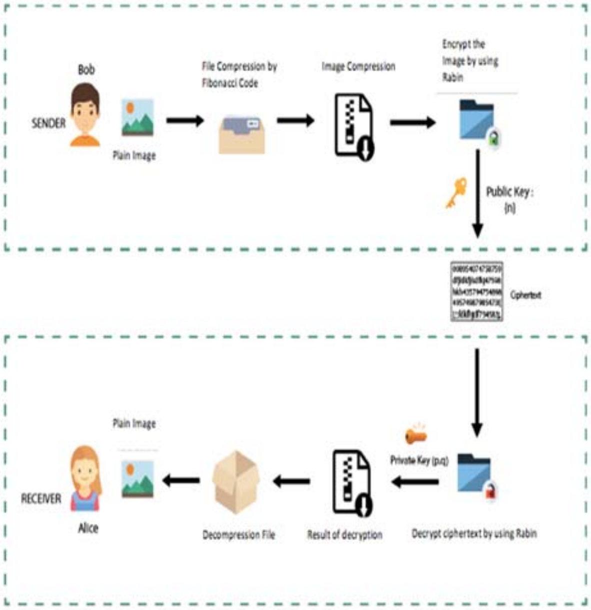


*Figure 3(B). Second Illustrations Of The Application*

Figure 3(b) above describes the general architecture system for the second Test that is the sequence of the compression process, encryption, decryption, and decompression of files. Still the same as on the previous system architecture, the visualization above, shows Bob as the sender/sender and Alice as the receiver/receiver. The process that occurs in the above scheme is:

1. Bob, as the sender, prepares the message of the image file to be sent to Alice. The image message is then compressed with the Fibonacci Codes method.

2. The compressed file is then encrypted by the Rabin cryptographic method, using the previously raised public key, resulting in a ciphertext File.

3. Once the file is received by Alice as a receiver, the ciphertext file is then decrypted using the private key that was given by the previous sender/Bob.

4. Once the file is decrypted, then the next process is to decompress the image file, so that it returns to the original message or the original image.

### 2.1 Steps for generating key:

1. Get two prime number $p$ dan $q$ $(p \neq q)$, which is congruent with 3 mod 4, $p \equiv q \equiv 3 \ (mod\ 4)$ as the private key.
2. Compute $n = pq$ as the public key. The value of $p$ and $q$ is saved from being the private key

### 2.2 Steps for encryption image :

1. Convert the image into the string, get the byte value.
2. Convert byte value into binary and extends with its value.
3. Convert binary into integer and initial it with $m$.
4. Compute $c = m^2 \ mod \ n$, get the ciphertext.

2.3 Steps for compression :

1 Get The byte value of the ciphertext file.
2 Create and save a table that contained the byte value based on frequency by descending order
3 Convert the byte value into binary.
4 Encode the binary value with Fibonacci codes sequences and adds padding bits and flags at the end of code.
5 Convert the binary codes into a hexadecimal number.
6 Calculate Ratio Compression, Compression Of Ratio dan Space-saving

2.4 Steps for decompression :

1. Get the byte value of a compressed file.
2. Convert that value into binary.
3. Check the padding bit and remove it.
4. Check each binary order from left to right, if there is found binary sequences which 11, replace binary code according to Fibonacci Codes that saved in the table.
5. Repeat steps 3, until the last bit.
6. Convert binary values to ciphertext byte values.

2.5 Steps for decryption image:

1. Get a private key (p,q).
2. Get the ciphertext.
3. If Gcd(p,q) = 1 then find yp and yq with Euclidean method.
4. Compute $m_p = c^{(p+1)/4}$ mod p and mq = c(q+1)/4 mod q.
5. Compute v = yp . p . mq and w = $y_q$ . q . $m_p$
6. To Get value of plaintext, find r,s,t,u value by using Chinese Reminder Theorem :

a. P1 = (v + w) mod n
b. P2 = (v - w) mod n
c. P3 = (-v + w) mod n
d. P4 = (-v - w) mod n

Example :

1. Generating key :

a. Get p = 239 dan q = 127
b. Compute n = 239 * 127 = 30353
c. Get public key (n) = (30353) dan private(p,q) = (239,127)

2. Image file :

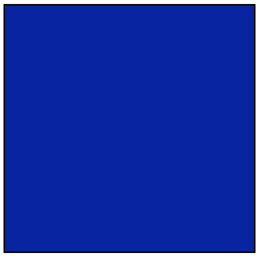
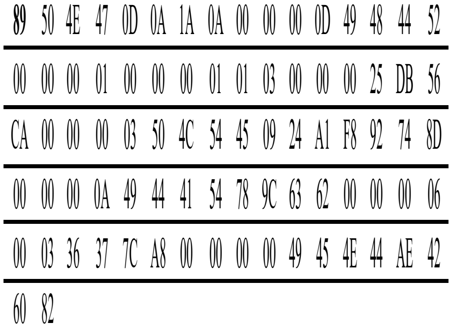


*Figure 4. Sample Of Image File*

Value image file in a byte can be seen in Table 1 as below :

Table 1. Byte Value of Image



| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 89 | 50 | 4E | 47 | 0D | 0A | 1A | 0A | 00 | 00 | 00 | 0D | 49 | 48 | 44 | 52 |
| 00 | 00 | 00 | 01 | 00 | 00 | 00 | 01 | 01 | 03 | 00 | 00 | 00 | 25 | DB | 56 |
| CA | 00 | 00 | 00 | 03 | 50 | 4C | 54 | 45 | 09 | 24 | A1 | F8 | 92 | 74 | 8D |
| 00 | 00 | 00 | 0A | 49 | 44 | 41 | 54 | 78 | 9C | 63 | 62 | 00 | 00 | 00 | 06 |
| 00 | 03 | 36 | 37 | 7C | A8 | 00 | 00 | 00 | 00 | 49 | 45 | 4E | 44 | AE | 42 |
| 60 | 82 | | | | | | | | | | | | | | |

Example: using byte 89

3. Encryption Process :

a. Get public key (n) = (30353)
b. Get byte of 89 and devide into two block $m_1$=8 dan $m_2$ = 9
c. Compute $c = m^2 \ mod \ n$ :

- $m_1$ = 8 :

$m_1 = 8_{16} = 1000_2 = 10001000_2 = 136_{10}$

$c_1 = 136^2 \ mod \ 30353 = 18496_{10} = 0100100001000000_2 = 4840_{16}$

- $m_2$ = 9 :

$m_2 = 9_{16} = 1001_2 = 10011001_2 = 153_2$

$c_2 = 153^2 \ mod \ 30353 = 23409_{10} = 0101101101110001_2 = 5B71_{16}$

d. Cipherimage size 4 times bigger than plainimage :

Table 2. Cipher Image Byte Value

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 48 | 40 | 5B | 71 | 07 | E9 | 00 | 01 | 05 | 10 | 66 | B3 | 05 | 10 | 0F | 81 |
| 00 | 00 | 48 | 38 | 00 | 00 | 70 | E4 | 00 | 09 | 70 | E4 | 00 | 00 | 70 | E4 |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 48 | 38 |
| 05 | 10 | 5B | 71 | 05 | 10 | 48 | 40 | 05 | 10 | 05 | 10 | 07 | E9 | 00 | 64 |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 09 |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 09 |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | E1 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00 | 00 | 00 | 00 | 00 | 64 | 07 | E9 | 48 | 38 | 12 | 8 | 07 | E9 | 0B | 64 |
| 2B | FF | 70 | E4 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00 | 00 | 00 | E1 | 07 | E9 | 00 | 00 | 05 | 10 | 2B | FF | 07 | E9 | 05 | 10 |
| 05 | 10 | 07 | E9 | 00 | 00 | 5B | 71 | 00 | 64 | 05 | 10 | 70 | E4 | 00 | 09 |
| 10 | DF | 48 | 40 | 5B | 71 | 00 | 64 | 0F | 81 | 05 | 10 | 48 | 40 | 48 | 38 |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 70 | E4 |
| 05 | 10 | 5B | 71 | 05 | 10 | 05 | 10 | 05 | 10 | 00 | 09 | 07 | E9 | 05 | 10 |
| 0F | 81 | 48 | 40 | 5B | 71 | 2B | FF | 0B | 64 | 00 | E1 | 0B | 64 | 00 | 64 |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 0B | 64 |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | E1 | 00 | E1 | 0B | 64 | 00 | E1 | 0F | 81 |
| 0F | 81 | 2B | FF | 70 | E4 | 48 | 40 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 05 | 10 | 5B | 71 | 05 | 10 | 07 | E9 |
| 05 | 10 | 66 | B3 | 5 | 10 | 05 | 10 | 70 | E4 | 66 | B3 | 05 | 10 | 00 | 64 |
| 0B | 64 | 00 | 00 | 48 | 40 | 00 | 64 | 02 | | | | | | | |

4. Compression Process :

a. List the byte value by descending.

Table 3. Total Bits of Cipher Image

| Byte Value | Freq | Bit | Freq * bit |
|---|---|---|---|
| 00 | 3 | 00000000 | 24 |
| 70 | 2 | 01110000 | 12 |
| E4 | 1 | 11100100 | 8 |
| 09 | 1 | 00001001 | 8 |
| | Total | | 52 |

b. Convert byte value into Fibonacci code

Table 4. Total Bits of Compressed File

| Byte Value | Fibonacci code | Bit | Freq | Freq * bit |
|---|---|---|---|---|
| 00 | 11 | 2 | 3 | 6 |
| 70 | 011 | 3 | 2 | 6 |
| E4 | 0011 | 4 | 1 | 4 |
| 09 | 1011 | 4 | 1 | 4 |
| | | Total | | 20 |

c. List Fibonacci code by the value of byte :
11110110011111011011

d. Add 4 bit to get a 24 bit and add 8 bit as the flag.
11110110011111011011000000000100.

e. Convert compression bit into a byte value

| F6 | 7D | B0 | 04 |
|---|---|---|---|

5. Decompression process:

a. Get the compression byte value

b. Convert byte value into binary

c. Then adjust the sequence of binary value with saved Fibonacci code

Step 1 :

| | | | | | | |
|---|---|---|---|---|---|---|

11110110011111011011000000000100

Step 2 :

| 00 | | | | | | |
|---|---|---|---|---|---|---|

110110011111011011000000000100

Step 3 :

| 00 | 00 | | | | | |
|---|---|---|---|---|---|---|

0110011111011011000000000100

Step 4 :

| 00 | 00 | 70 | | | | |
|---|---|---|---|---|---|---|

00111110110110000000000100

Step 5 :

| 00 | 00 | 70 | E4 | 00 | | |
|---|---|---|---|---|---|---|

1011011000000000100

Step 6 :

| 00 | 00 | 70 | E4 | 00 | 09 | |
|---|---|---|---|---|---|---|

011000000000100

Step 7 :

| 00 | 00 | 70 | E4 | 00 | 09 | 70 |
|---|---|---|---|---|---|---|

000000000100

Step 8 :

| 00 | 00 | 70 | E4 | 00 | 09 | 70 |
|---|---|---|---|---|---|---|

Save byte value, Then remove the padding and flag bit.

6. Decryption Proses :

a. Get byte value of ciphertext, $c_1 = 4840_{16}$ dan $c_2 = 5B71_{16}$

b. Calculate the value of $y_p$ and $y_q$ with Extended Euclidean GCD, and be obtained $y_p = -17$ and $y_q = 32$.

c. Calculate the value of $m_p$ and $m_q$ for $c_1$ and $c_2$

- For $c_1 = 4840_{16} = 18496_{10}$

a. $m_p = c^{(p+1)/4} \bmod p$

$m_p = 18496^{(239+1)/4} \bmod 239 = 136$

b. $m_q = c^{(q+1)/4} \bmod q$

$m_q = 18496^{32} \bmod 127 = 9$

- For $c_2 = 5B71_{16} = 23409_{10}$

a. $m_p = c^{(p+1)/4} \bmod p$

$m_p = 23409^{(239+1)/4} \bmod 239 = 153$

b. $m_q = c^{(q+1)/4} \bmod q$

$m_q = 23409^{(127+1)/4} \bmod 127 = 26$

d. Calculate the value of *v* and *w*, for each $m_p$ and $m_q$ each from $c_1$ and $c_2$.

- For $m_p = 136$ and $m_q = 9$

a. $v = y_p . p . m_q$

$v = -17 . 239 . 9 = -36567$

b. $w = y_q . q . m_p$

$w = 32 . 127 . 136 = 552704$

- For $m_p = 153$ and $m_q = 26$

a. $v = y_p \,.\, p \,.\, m_q$
$v = -17 \,.\, 239 \,.\, 26 = -105638$
b. $w = y_q \,.\, q \,.\, m_p$
$w = 32 \,.\, 127 \,.\, 153 = 621792$

e. Calculate r,s,t,u for each *v* and *w*, then specify plaintext value with comparing what among them from that value have binary sequence which same when both of the sides binary divide by two:

1. Specify $m_1$ with $v = -36567$ and $w = 552704$ :

$r = (v + w) \bmod n = ((-36567) + 552704) \bmod 30353$
$= 136 = 1000|1000$ (SIMILAR)

$s = (v - w) \bmod n = ((-36567) - 552704) \bmod 30353$
$= 17789 = 01000101|01111101_2$(NOT SIMILAR)

$t = (-v + w) \bmod n = (-(-36567) + 552704) \bmod 30353$
$= 12564 = 00110001|00010100_2$(NOT SIMILAR)

$u = (-v - w) \bmod n = (-(-36567) - 552704) \bmod 30353$
$= 3012710 = 01110101|10101111_2$ (NOT SIMILAR)

GET r as m1 $= 136_{10} = 1000|1000_2 = 1000_2 = 8_{16}$

2. Specify $m_2$ with $v = -105638$ and $w = 621792$ :

$r = ((-105638) + 621792) \bmod 30353$
$=153=1001|1001_2$ (SIMILAR)

$s = ((-105638) - 621792) \bmod 30353$
$= 1042 = 00000100|00010010_2$(NOT SIMILAR)

$t = (-(-105638) + 621792) \bmod 30353$
$= 29311= 01110010|01111111_2$(NOT SIMILAR)

$u = (-(-105638) - 621792) \bmod 30353$
$= 30200 = 01110101|11111000$ (NOT SIMILAR)

Get r as $m_2 = 153_{10} = 1001|1001_2 = 1001_2 = 9_{16}$

## 3. RESULTS AND DISCUSSIONS

The experiments were performed on Windows 8 Notebook with Intel Core i3 processor, 64-bit architecture, and 4096MB RAM. The development environment used for coding C# scripts is SharpDevelop. Experiments are done in two schemes, that encryption-compression scheme, and compression-encryption scheme. To evaluate this scheme to measure the success of cryptography will be checked whether the message encoded can return to the original form while the compression will be done a calculation of the file size before and after the compression, the smaller the file size, then the compromise is more efficient.

a.Encryption-compression scheme
The experiment is done to three file types *.bmp, *.png, *.jpg, and done as many as five times for each file type. The result of the experiment can be seen in the tables below.

| FILE TYPE | IMAGE FILE | FILE SIZE (Kb) | Rc | Cr(%) | SS(%) | Encryption (ms) | Compression (ms) |
|---|---|---|---|---|---|---|---|
| JPG | | 939,507 | 1,415 | 70,653 | 29,346 | 699 | 123 |
| | | 754,66 | 1,417 | 70,528 | 29,471 | 1372 | 272 |
| | | 560,299 | 1,416 | 70,588 | 29,411 | 2191 | 453 |
| | | 345,647 | 1,418 | 70,505 | 29,494 | 3046 | 545 |
| | | 167,667 | 1,404 | 71,209 | 28,79 | 3630 | 702 |
| | | Average | 1,414 | 70,6966 | 29,3024 | 2187,6 | 419 |
| PNG | | 162,894 | 1,42 | 70,42 | 29,579 | 671 | 122 |
| | | 346,977 | 1,412 | 70,343 | 29,656 | 1629 | 245 |
| | | 580,132 | 1,417 | 70,546 | 29,453 | 2444 | 440 |
| | | 784,904 | 1,423 | 70,257 | 29,742 | 3276 | 736 |
| | | 941,327 | 1,425 | 70,166 | 29,833 | 3902 | 744 |
| | | Average | 1,4194 | 70,3464 | 29,6526 | 2384,4 | 457,4 |
| BMP | | 189,066 | 1,717 | 58,238 | 41,761 | 671 | 122 |
| | | 386,706 | 1,722 | 58,053 | 41,496 | 1629 | 245 |
| | | 609,306 | 1,724 | 42,975 | 42,024 | 2444 | 440 |
| | | 780,69 | 1,726 | 57,926 | 42,073 | 3276 | 736 |
| | | 938,918 | 1,726 | 57,907 | 42,092 | 3902 | 744 |
| | | Average | 1,723 | 55,0198 | 41,8892 | 2406,8 | 362,8 |

c. Compression-encryption scheme

The experiment is done as many as five times for each file type. The result of the experiment can be seen from the table below.

| FILE TYPE | IMAGE FILE | FILE SIZE (Kb) | Rc | Cr() | SS(%) | Encryption (ms) | Compression (ms) |
|---|---|---|---|---|---|---|---|
| JPG | | 939,507 | 0,784 | 127,521 | -27,521 | 63 | 87 |
| | | 754,66 | 0,776 | 128,745 | -28,745 | 168 | 1829 |
| | | 560,299 | 0,773 | 129,259 | -29,259 | 190 | 3247 |
| | | 345,647 | 0,772 | 129,507 | -29,507 | 238 | 3654 |
| | | 167,667 | 0,77 | 129,807 | -29,807 | 301 | 4464 |
| | | Average | 0,775 | 128,968 | -28,968 | 192 | 2656,2 |
| PNG | | 162,894 | 0,764 | 130,851 | -30,851 | 671 | 122 |
| | | 346,977 | 0,764 | 130,818 | -30,818 | 1629 | 245 |
| | | 580,132 | 0,766 | 130,359 | -30,539 | 2444 | 440 |
| | | 784,904 | 0,765 | 130,71 | -30,71 | 3276 | 736 |
| | | 941,327 | 0,765 | 130,557 | -30,557 | 3902 | 744 |
| | | Average | 0,7648 | 130,659 | -30,695 | 185,2 | 2858,8 |
| BMP | | 189,066 | 1,152 | 86,777 | 13,222 | 47 | 776 |
| | | 386,706 | 1,151 | 86,872 | 13,127 | 91 | 1430 |
| | | 609,306 | 1,152 | 86,767 | 13,232 | 162 | 1997 |
| | | 780,69 | 1,15 | 86,909 | 13,09 | 173 | 2712 |
| | | 938,918 | 1,144 | 87,406 | 12,593 | 230 | 3041 |
| | | Average | 1,1498 | 86,9462 | 13,0528 | 140,6 | 1991,2 |

## 5. CONCLUSIONS

Built-in applications capable of securing and compresses image files with Rabin and Fibonacci Codes cryptographic algorithms. The process of compression of files performed with algorithms with Fibonacci methods can effectively reduce the file size of the encrypted results. The testing process with the encryption-compression scheme proved to be more effective than the compression-encryption scheme. Because if on the test flow, compression is done first, then it can generate a negative value or not compressed perfectly for some types of image files. Compression test result based on Ratio compression (Rc), Compression ratio (Cr), and SS (Space-saving), shows the following result: For encryption scheme compression on, the total averages for each – each value is, 1.51, 65.35%, and 33.61%.

As for the compression scheme – encryption, the total average obtained is 0.89, 115.52%,-15.53%. From the average of the above compression results, it is proven that testing with the compression-encryption method using the Fibonacci codes algorithm is not effective to compress image files in particular PNG and JPG files. Wherefrom the results, the files are thus experiencing magnification, which causes the value of space-saving (SS) to be negative. In BMP type files, if viewed specifically, the compression process for these types of files proved to be effective, either with an encryption-compressive scheme or vice versa. Total average compression result on this file is Rc = 1.4364, Cr = 70.98% and SS = 27.47%. Based on the test result graph of processing time either with the encryption scheme – compression or compression – encryption applied to the file type of imagery (*. BMP,*. png, *. jpg) indicates that the size of the file is directly proportional to the processing time. So the larger the file size, the more processing time it takes. From the test result chart for the encryption scheme compression – encryption, the file size of the encryption or compression process results directly proportional to the size of the original file or the tested image file. The encryption result file generates the largest size, and the compressed file generates the smallest size.

## REFERENCES

[1] Diffie, Whitfield, and Martin Hellman. “New directions in cryptography.” *IEEE transactions on Information Theory* 22.6 (1976): 644-654.

[2] Rivest, Ronald L., Adi Shamir, and Leonard Adleman. “A method for obtaining digital signatures and public-key cryptosystems.” *Communications of the ACM* 21.2 (1978): 120-126.

[3] Pomerance, Carl. “The quadratic sieve factoring algorithm.” *Workshop on the Theory and Application of Cryptographic Techniques*. Springer, Berlin, Heidelberg (1984).

[4] Boneh, Dan, and Ramarathnam Venkatesan. “Breaking RSA may not be equivalent to factoring.” *Advances in Cryptology—EUROCRYPT'98* (1998): 59-71.

[5] Katz, Jonathan, A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone. *Handbook of applied cryptography*. CRC press. (1996).

[6] Yang, Li, Ling-An Wu, and Songhao Liu. "Quantum three-pass cryptography protocol." *Quantum Optics in Computing and Communications*. Vol. 4917. International Society for Optics and Photonics (2002).

[7] Smart, Nigel P. “Cryptography made simple”. Springer International Publishing, (2016).

[8] Rachmawati, Dian, and Mohammad Andri Budiman. “An implementation of the H-Rabin algorithm in the Shamir three-pass protocol.” *2nd International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT)* IEEE (2017): 28-33.

[9] Schneier, Bruce. *Applied cryptography: protocols, algorithms, and source code in C*. Indianapolis, IN. Wiley (2015).

[10] Carlsen, Ulf. "Cryptographic protocol flaws: know your enemy." *Computer Security Foundations Workshop VII*. Proceedings. IEEE (1994).

[11] Ambika, R., S. Ramachandran, and K. R. Kashwan. "Securing Distributed FPGA System using Commutative RSA Core." *Global Journal of Research In Engineering* (2013).

[12] Hsu, Jen-Chieh, Raylin Tso, Yu-Chi Chen, and Mu-En Wu. "Oblivious Transfer Protocols Based on Commutative Encryption." *International Conference on New Technologies, Mobility and Security (NTMS), 2018 9th IFIP*. IEEE (2018).

[13] Budiman, Mohammad Andri, Dian Rachmawati, and M. R. Parlindungan. “An implementation of super-encryption using RC4A and MDTM cipher algorithms for securing PDF Files on android.” *Journal of Physics: Conference Series*. Vol. 978. No. 1. IOP Publishing (2018).

[14] Rachmawati, Dian, Mohammad Andri Budiman, and Indra Aulia. “Super-Encryption Implementation Using Monoalphabetic Algorithm and XOR Algorithm for Data Security.” *Journal of Physics: Conference Series*. Vol. 979. No. 1. IOP Publishing (2018).

[15] Budiman, Mohammad Andri, Amalia, and N. I. Chayanie. “An Implementation of RC4+ Algorithm and Zig-zag Algorithm in a Super Encryption Scheme for Text Security.” *Journal of Physics: Conference Series*. Vol. 978. No. 1. IOP Publishing (2018).

[16] Agrawal, Manindra, Neeraj Kayal, and Nitin Saxena. “PRIMES is in P.” *Annals of mathematics* (2004): 781-793.

[17] Sonal Chawla et al, International Journal of Computer Science and Mobile Computing, Vol.3 Issue.8, August- 2014, pg. 291-296

[18] Gonzalez R, Woods R (2007) Digital image processing, 3rd edn. Prentice-Hall, Cambridge

[19] Saptarshi Bhattacharyya. "Complexity Analysis of a Lossless Data Compression Algorithm using Fibonacci Sequence". International Journal of Information Technology (IJIT) V3(3): Page(77-82) May - Jun 2017. ISSN: 2454-5414 www.ijitjournal.org.Published by Eighth Sense Research Group.