ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

IMPLEMENTATION OF ELGAMAL AND LEAST SIGNIFICANT BIT (LSB) ALGORITHM FOR ENDING AND HIDDEN MESSAGES IN DIGITAL IMAGES

TONNI LIMBONG[1], A M H PARDEDE[2], DESINTA PURBA[3], LAMHOT SITORUS[4]

[1,3,4]Fakultas Ilmu Komputer, Universitas Katolik Santo Thomas Medan, Indonesia
[2]STMIK Kaputama, Jl. Veteran No. 4A-9A, Binjai- Sumatera Utara, Indonesia
E mail: [1]tonni.budidarma@gmail.com, [2]akimmhp@live.com

ABSTRACT

Basically, confidential data needs to be stored or conveyed in a certain way so that it is not known by unauthorized foreign parties. And to overcome this problem, the science of cryptography and steganography was created. Cryptography is the art and science of keeping messages confidential by disguising them in an encoded form that has no meaning, while steganography is the art and science of hiding secret messages inside other messages so that the whereabouts of the secret message cannot be known. Steganography keeps messages secret by hiding messages. The current implementation of steganography uses digital media as a medium for storing or hiding messages, one of which is image media (digital image). The combination of cryptography and steganography can provide better security for secret messages, where secret messages are first encrypted using the ElGamal algorithm, then the ciphertext results from the cryptography are hidden in image media using the Least Significant Bit (LSB) steganography method. The implementation of cryptographic algorithms and steganography methods can further increase the security of secret messages.
Keywords: *Ciphertext, Cryptography, ElGamal, Encryption, Steganography.*

1. INTRODUCTION

Without a guarantee of security in data transmission, of course there will be a risk when sensitive, important and valuable information is accessed by people who are not authorized and responsible, resulting in the data being misused which can be detrimental to the owner of the data. Facing data or information security threats, security techniques are needed, and maintaining the confidentiality of messages using cryptographic and stenographic algorithms.

Basically, confidential data needs to be stored or conveyed in a certain way so that it is not known by unauthorized foreign parties. And to overcome this problem, the science of cryptography and steganography was created. Cryptography is the art and science of keeping messages confidential by disguising them in an encoded form that has no meaning, while steganography is the art and science of hiding secret messages inside other messages so that the whereabouts of the secret message cannot be known. Steganography keeps messages secret by hiding messages. The current implementation of steganography uses digital media as a medium for storing or hiding messages, one of which is image media (digital image).

The ElGamal algorithm is a cryptographic algorithm created by Taher ElGamal in 1984. The ElGamal algorithm is an asymmetric algorithm that has a public key consisting of three pairs of numbers and a secret key consisting of two numbers. For the same plaintext, this algorithm provides a different ciphertext each time the plaintext is encrypted. This is due to the influence of a variable that is randomly determined during the encryption process [1].

One of the digital image steganography methods is the Least Significant Bit (LSB), with the technique of hiding messages at the lowest bit location in a digital image. The message is converted into binary bits and hidden in a digital image using the LSB method [2].

The combination of cryptography and steganography can provide better security for secret messages, where secret messages are encrypted first using the ElGamal algorithm, then the cryptographic ciphertext results are hidden in image media using the Least Significant Bit (LSB) steganography method. The implementation of cryptographic

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

algorithms and steganography methods can further increase the security of secret messages [2].

Cryptography (cryptography) comes from the Greek: "cryptos" means "secret", while "graphein" means "to write" (writing). So, cryptography means "secret writing". There are several definitions of cryptography that have been put forward in various literature. The definition used in old books (before the 1980s) states that cryptography is the science and art of maintaining the secrecy of messages by encoding them into a form that the meaning can no longer be understood. This definition may be appropriate in the past when cryptography was used for the security of important communications such as communications among the military, diplomats, and spies. However, currently, cryptography is more than just privacy, but also for data integrity, authentication, and non-repudiation purposes [3].

In cryptography, there are various terms or terminology. Some important terms to know are [4]:

a. Message, Plaintext, and Ciphertext
   Messages are data or information that can be read and understood. Another name for the message is plaintext or cleartext. Messages can be in the form of data or information sent (via couriers, telecommunications channels, etc.) or stored on recording media (paper, storage, etc.). Stored messages are not only in the form of text, but can also be in the form of images, sound, video, or other binary files. In order for the message to be hidden from other parties, the message is encoded in another form that cannot be understood. The form of the encoded message is called ciphertext (ciphertext) or cryptogram (cryptogram). Ciphertext must be able to be transformed back into the original plaintext so that the received message can be read.
b. Sender and Recipient
   Data communication involves exchanging messages between two entities. The sender (sender) is an entity that sends messages to other entities. The recipient (recipient) is the entity that receives the message. The sender certainly wants the message to be sent safely, but he believes that other parties cannot read the contents of the message he sent. The solution is to encode the message into ciphertext.
c. Encryption and description
   The process of encoding plaintext into ciphertext is called encryption or enciphering (standard name according to ISO 7498-2). Meanwhile, the process of turning ciphertext back into plaintext is called decryption or deciphering (standard name according to ISO 7498-2).
d. Cipher and key
   Cryptographic algorithms are also called ciphers, namely the rules for encrypting and decoding, or the mathematical functions used for encryption and decryption. Some encodings require different algorithms for encoding and decoding.
e. Cryptographic System
   Cryptography forms a system called a cryptographic system. A cryptographic system (cryptosystem) is a collection consisting of cryptographic algorithms, all possible plaintext and ciphertexts, and keys.
f. Tappers
   Eavesdroppers are people who try to catch messages as they are being transmitted. The aim of eavesdroppers is to get as much information as possible about the cryptographic system used to communicate with the intention of breaking the ciphertext.
g. Cryptanalysis and cryptology
   Cryptography developed in such a way that it gave birth to the opposite field, namely cryptanalysis. Cryptanalysis (cryptanalysis) is the science and art of breaking ciphertext into plaintext without knowing the key used. The culprit is called a cryptanalyst. If a cryptographer (cryptographer) transforms plaintext into ciphertext with an algorithm and key, then a cryptographer tries to solve the ciphertext to find plaintext or key. Cryptology (cryptology) is the study of cryptography and cryptanalysis. Both cryptography and cryptanalysis are interrelated. Figure 1 shows the cryptology tree.

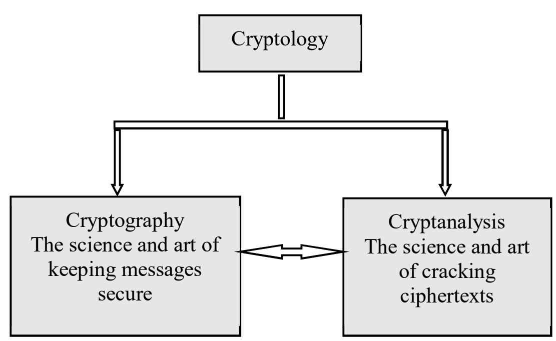

Figure 1: Cryptography and cryptanalysis are branches of cryptology

Currently steganography has been widely implemented in digital media. Digital steganography uses digital media as containers, such as digital images, digital video, or audio. Modified information is also in digital form such as text, images, audio data and video data. Digital steganography can be used in countries where information is strictly censored or in countries where

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

message encryption is prohibited. In such countries confidential information can be hidden using steganography [4].

According to more steganography is done than cryptography [5]. This is because in cryptography the scrambling/encoding of messages will result in the message changing into strange characters, which actually creates hatred for those who read it. However, in steganography, it will not be seen at all that there is a message contained in the image [6].

2. METHODS AND MATERIAL

The analysis of the algorithm used is the analysis of the encryption and decryption process on the ElGamal cryptographic algorithm and the analysis of the process of embedding and extracting messages using the Least Significant Bit (LSB) Steganography algorithm. After that, it will proceed to the system design stage [7].

2.1 How ElGamal Algorithm and Least Significant Bit (LSB) Work

At this stage, an analysis will be carried out on the ElGamal algorithm in carrying out the process of encrypting and decrypting messages, and also an analysis of the Least Significant Bit (LSB) algorithm in carrying out the process of inserting and extracting messages [8].

How the ElGamal Algorithm Works

How the Elgamal Algorithm works is explained starting from the key formation process, the encryption process and also the decryption process.

Key Formation Process, The steps involved in the key formation process are as follows:

1. Choose any prime number p.
2. Choose 2 random numbers g and x provided that $g < p$ and $1 \leq x \leq p - 2$.
3. Calculate y with the formula y = gx mod p.
4. The result of this algorithm is to generate a public key (p, g, y) and a private key: pair (p, x).

The steps in the key formation process in the ElGamal algorithm can be seen in full in Figure 2.

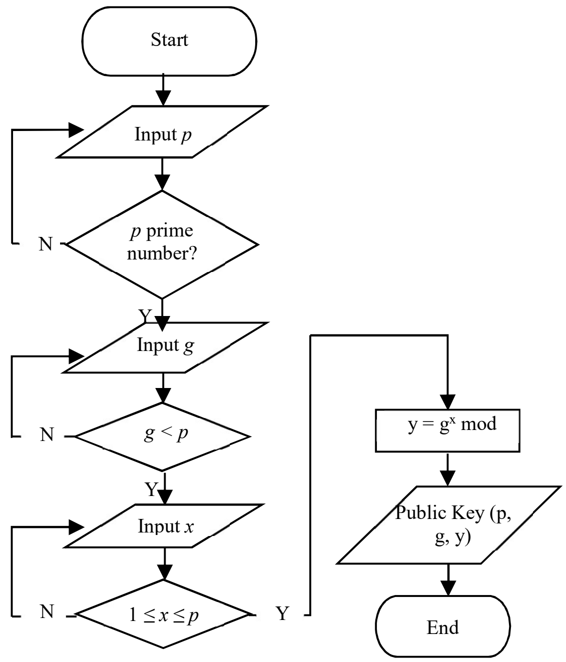

Figure 2: ElGamal key formation process

Encryption Process, The steps to perform the encryption process on the ElGamal algorithm are as follows [9]:

1. Enter the public key (p, g, y) as well as the plaintext to be encrypted.
2. Convert the original message (plaintext) to ASCII.
3. Choose a random number k, which in this case $1 \leq k \leq p - 2$
4. Each plaintext block (m) is encrypted using a public key with the formula: a = gk mod p and b = yk.m mod p
5. Pairs a and b are ciphertext.

The entire encryption process in the ElGamal algorithm can be seen further in Figure 3.

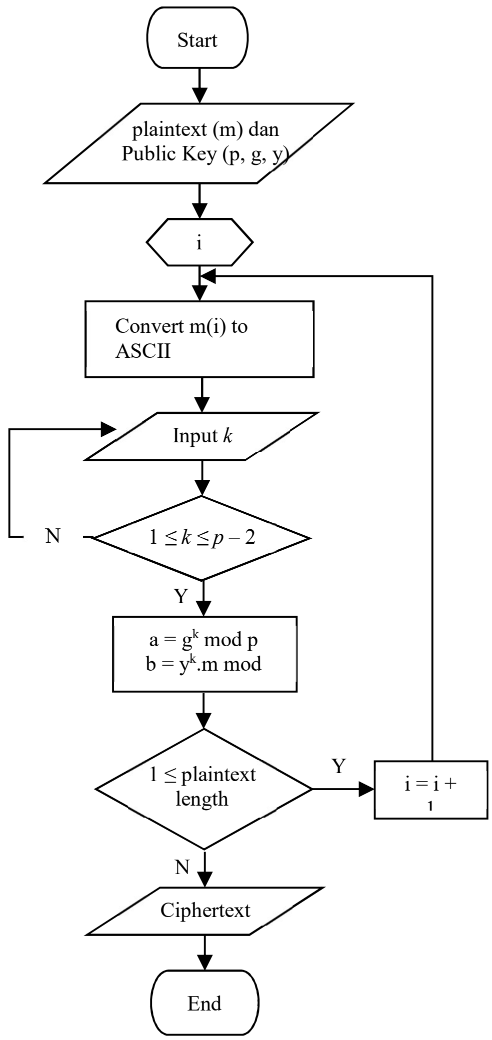

Figure 3: ElGamal algorithm encryption process

Process Description, The steps for decrypting the ElGamal algorithm are as follows [10]:

1. Input password and private key (p, x)
2. Separate the values a and b in the ciphertext, provided that:
   a = Ciphertext of odd order
   b = Even-order ciphertext.
3. Calculate m (original message) using the formula: m = b * a(p-1-x) mod p to generate plaintext.

All stages of the decryption process in the ElGamal algorithm can be seen in full in Figure 4.

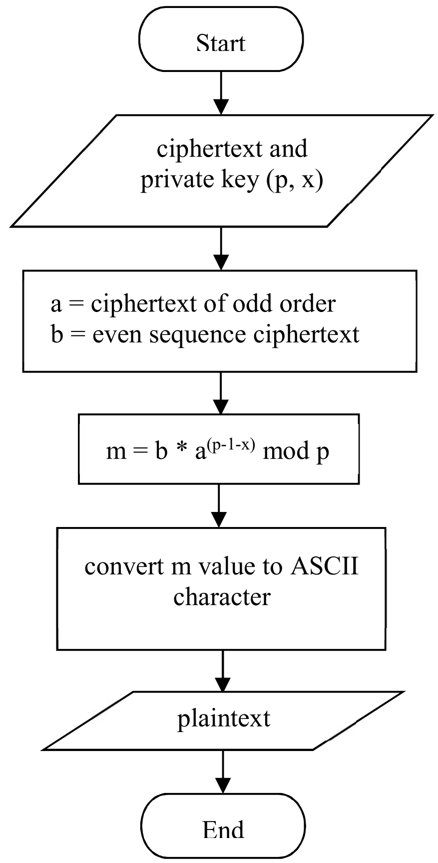

Figure 4: Decryption process in the ElGamal algorithm

2.2 How the Least Significant Bit (LSB) Algorithm Works

How the Least Significant Bit (LSB) Algorithm works is explained starting from the message insertion process and also the message extraction process [11].

1. Message Insertion Process, The steps in carrying out the message insertion process using the LSB algorithm are as follows:
   a. Input text to be inserted into the image.
   b. Select an image file (cover image).
   c. Count the number of pixels of the image file and the length of the text.
   d. Convert each RGB value in each image pixel into 8-bit binary form
   e. Add a marker character (#) at the end of the message to be inserted.
   f. Convert the message to be inserted into 8-bit binary form.

Journal of Theoretical and Applied Information Technology
31st October 2023. Vol.101. No 20

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

g. Replace the last bit of each RGB value in each digital image pixel with the message bit value to be inserted.
h. Convert the digital image binary code that has been inserted into a message into a new RGB image value (stego image).

All stages of message insertion using the Least Significant Bit (LSB) algorithm can be seen in full in Figure 5.

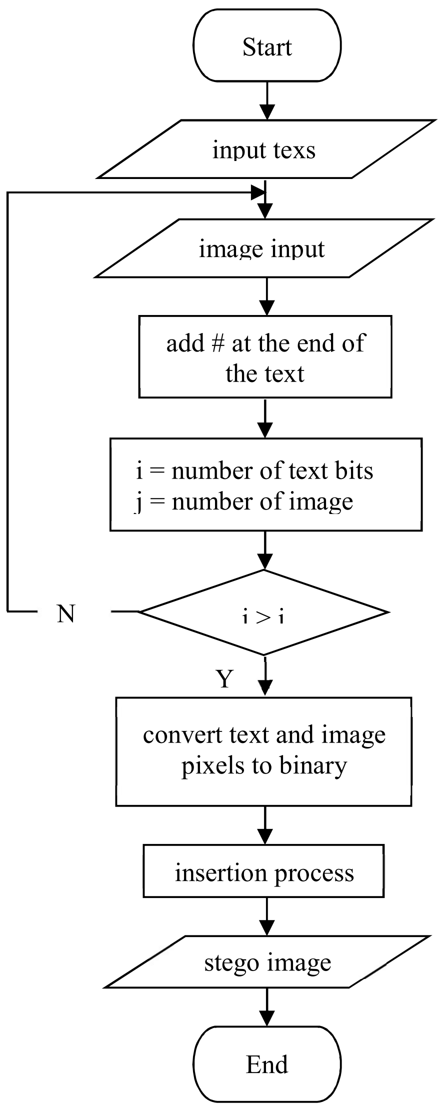

Figure 5 The message insertion process uses the LSB algorithm

2. Message Extraction Process, The steps in carrying out the message extraction process using the LSB algorithm are as follows [12]:
   a. Enter the image file (stego image)
   b. Read each pixel of the image file from start to finish.
   c. Convert each RGB value in each stego image pixel into 8-bit binary form
   d. Take the last bit of each RGB value in each stego image pixel, then divide each into 8 bits, then convert it into a character based on the ASCII table
   e. e. Remove the marker character at the end of the message (#), so you get the original message.

The entire message extraction process using the Least Significant Bit (LSB) algorithm can be seen in full in Figure 6.

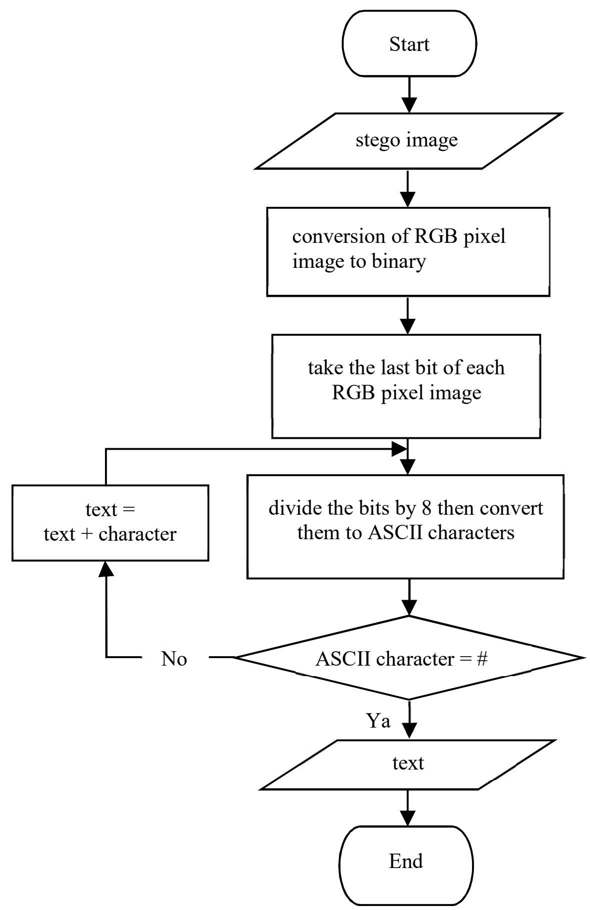

Figure 6 Message extraction process using the LSB algorithm

3. SYSTEM ANALYSIS AND DESIGN

To better understand every process that occurs in an application that is built, in the following the author will provide an example [13].

1. Key formation process
   For example, the value p = 383, g = 148, x = 338 is chosen

Then calculate: $y = g^x \bmod p = 148^{338} \bmod 383 = 295$
Thus, the public key (p, g, y) = (383, 148, 295) and the private key (p, x) = (383, 338)

2. Message encryption process
For example the message to be encrypted is the word "RAPOT", then the encryption process is as follows:
   a. Convert the original message (plaintext) to ASCII, as shown in table 1.

Table 1: Message conversion to ASCII

i	Plainteks	Plainteks m_i	ASCII
1	R	m_1	82
2	A	m_2	65
3	P	m_3	80
4	O	m_4	79
5	T	m_5	84

   b. Choose a random number k, which in this case $1 \le k \le p - 2$
   In this case the k value chosen is $k_1 = 319$, $k_2 = 259$, $k_3 = 353$, $k_4 = 105$, $k_5 = 267$
   c. Each plaintext block (m) is encrypted using a public key with the formula: $a = g^k \bmod p$ dan $b = y^k.m \bmod p$
   $a_1 = 148^{319} \bmod 383 = 197$; $b_1 = 295^{319} * 82 \bmod 383 = 375$
   $a_2 = 148^{259} \bmod 383 = 122$; $b_2 = 295^{259} * 65 \bmod 383 = 43$
   $a_3 = 148^{353} \bmod 383 = 85$; $b_3 = 295^{353} * 80 \bmod 383 = 52$
   $a_4 = 148^{105} \bmod 383 = 379$; $b_4 = 295^{105} * 79 \bmod 383 = 33$
   $a_5 = 148^{267} \bmod 383 = 340$; $b_5 = 295^{267} * 84 \bmod 383 = 272$
   d. Pairs a and b are ciphertext. So that the ciphertext obtained is 197 375 122 43 85 52 379 33 340 272

3. Message insertion process, After the message is successfully encrypted, then the ciphertext will be inserted into a digital image [14].
   a. Suppose an image is 8x12 pixels in size, with RGB values for each pixel in decimal form, as shown in table 2.

Table 2: RGB values in an 8 x 12 image

200, 189, 203	194, 185, 146	192, 170, 87	198, 168, 18	211, 162, 7	200, 189, 203	194, 185, 146	192, 170, 87
198, 168, 18	211, 162, 7	201, 190, 204	199, 190, 151	201, 179, 96	198, 168, 18	209, 160, 5	201, 190, 204
199, 190, 151	201, 179, 96	198, 168, 18	209, 160, 5	193, 189, 203	189, 190, 192	185, 190, 170	196, 170, 111
206, 157, 65	193, 189, 203	189, 190, 192	185, 190, 170	196, 170, 111	206, 157, 65	190, 186, 200	188, 189, 191
191, 196, 176	212, 106, 127	24,162,70	190, 186, 200	188, 189, 191	191, 196, 176	212, 106, 127	24,162,70
196, 190, 190	198, 180, 178	223, 182, 180	232, 160, 148	182, 87,67	196, 190, 190	198, 180, 178	223, 182, 180
232, 160, 148	182, 87,67	200, 189, 203	194, 185, 146	192, 170, 87	198, 168, 18	211, 162, 7	200, 189, 203
194, 185, 146	192, 170, 87	198, 168, 18	211, 162, 7	201, 190, 204	199, 190, 151	201, 179, 96	198, 168, 18
209, 160, 5	201, 190, 204	199, 190, 151	201, 179, 96	198, 168, 18	209, 160, 5	193, 189, 203	189, 190, 192
185, 190, 170	196, 170, 111	206, 157, 65	193, 189, 203	189, 190, 192	185, 190, 170	196, 170, 111	206, 157, 65
190, 186, 200	188, 189, 191	191, 196, 176	212, 106, 127	24,162,70	190, 186, 200	188, 189, 191	191, 196, 176
212, 106, 127	24,162,70	196, 190, 190	198, 180, 178	223, 182, 180	232, 160, 148	182, 87,67	196, 190, 190

   b. Convert each RGB value at each pixel into 8-bit binary form, as shown in table 3.

Table 3: Conversion of RGB values in images into 8-bit binary

R=11001000 G=10111101 B=11001011	R=11000010 G=10111001 B=10010010	R=11000000 G=10101010 B=01010111	R=11000110 G=10101000 B=00010010	R=11010011 G=10100010 B=00000111	R=11001000 G=10111101 B=11001011	R=11000010 G=10111001 B=10010010	R=11000000 G=10101010 B=01010111
R=11000110 G=10101000 B=00010010	R=11010011 G=10100111 B=00000111	R=11001001 G=10111110 B=11001100	R=11000111 G=10111110 B=10010111	R=11001001 G=10110011 B=01100000	R=11000110 G=10101000 B=00010010	R=11010001 G=10100000 B=00000101	R=11001001 G=10111110 B=11001100
R=11000111 G=10111110 B=10010111	R=11001001 G=10110011 B=01100000	R=11000110 G=10101000 B=00010010	R=11010001 G=10100000 B=00000101	R=11000001 G=10111101 B=11001011	R=10111101 G=10111110 B=11000000	R=10111001 G=10111110 B=10101010	R=11000100 G=10101010 B=01101111
R=11001110 G=10011101 B=01000001	R=11000001 G=10111101 B=11001011	R=10111101 G=10111110 B=11000000	R=10111001 G=10111110 B=10101010	R=11000100 G=10101010 B=01101111	R=11001110 G=10011101 B=01000001	R=10111110 G=10111010 B=11001000	R=10111100 G=10111101 B=10111111
R=10111111 G=11000100 B=10110000	R=11010100 G=01101010 B=01111111	R=00011000 G=10100010 B=01000110	R=10111110 G=10111010 B=11001000	R=10111100 G=10111101 B=10111111	R=10111111 G=11000100 B=10110000	R=11010100 G=01101010 B=01111111	R=00011000 G=10100010 B=01000110
R=11000100 G=10111110 B=10111110	R=11000110 G=10110100 B=10110010	R=11011111 G=10110110 B=10110100	R=11101000 G=10100000 B=10010100	R=10111010 G=01010111 B=01000011	R=11000100 G=10111110 B=10111110	R=11000110 G=10110100 B=10110010	R=11011111 G=10110110 B=10110100
R=11101000 G=10100000 B=10010100	R=10111010 G=01010111 B=01000011	R=11001000 G=10111101 B=11001011	R=11000010 G=10111001 B=10010010	R=11000000 G=10101010 B=01010111	R=11000110 G=10101000 B=00010010	R=11010011 G=10100111 B=00000111	R=11001000 G=10111101 B=11001011

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

R=11000010 G=10111001 B=10010010	R=11000000 G=10101010 B=01010111	R=11000110 G=10101000 B=00010010	R=11010011 G=10100111 B=00000111	R=11001001 G=10111110 B=11001100	R=11000111 G=10111110 B=10010111	R=11001001 G=10110011 B=01100000	R=11000110 G=10101000 B=00010010
R=11010001 G=10100000 B=00000101	R=11001001 G=10111110 B=11001100	R=11000111 G=10111110 B=10010111	R=11001001 G=10110011 B=01100000	R=11000110 G=10101000 B=00010010	R=11010001 G=10100000 B=00000101	R=11000001 G=10111101 B=11001011	R=10111101 G=10111110 B=11000000
R=10111001 G=10111110 B=10101010	R=11000100 G=10101010 B=01101111	R=11001110 G=10011101 B=01000001	R=11000001 G=10111101 B=11001011	R=10111101 G=10111110 B=11000000	R=10111001 G=10111110 B=10101010	R=11000100 G=10101010 B=01101111	R=11001110 G=10011101 B=01000001
R=10111110 G=10111010 B=11001000	R=10111100 G=10111101 B=10111111	R=10111111 G=11000100 B=10110000	R=11010100 G=01101010 B=01111111	R=00011000 G=10100010 B=01000110	R=10111110 G=10111010 B=11001000	R=10111100 G=10111101 B=10111111	R=10111111 G=11000100 B=10110000
R=11010100 G=01101010 B=01111111	R=00011000 G=10100010 B=01000110	R=11000100 G=10111110 B=10111110	R=11000110 G=10110100 B=10110010	R=11011111 G=10110110 B=10110100	R=11101000 G=10100000 B=10010100	R=10111010 G=01010111 B=01000011	R=11000100 G=10111110 B=10111110

c. Add a marker character (#) at the end of the message to be inserted. Then the message becomes 197 375 122 43 85 52 379 33 340 272#. Then convert the message to be inserted into 8-bit binary form, as shown in table 4.

Table 4: Convert the message to be inserted into 8 bit binary

No	Character	Ascii	Binari	No	Character	Ascii	Binari	No	Character	Ascii	Binari
1	1	49	00110001	13	4	52	00110100	25	space	32	00100000
2	9	57	00111001	14	3	51	00110011	26	3	51	00110011
3	7	55	00110111	15	space	32	00100000	27	3	51	00110011
4	space	32	00100000	16	8	56	00111000	28	space	32	00100000
5	3	51	00110011	17	5	53	00110101	29	3	51	00110011
6	7	55	00110111	18	space	32	00100000	30	4	52	00110100
7	5	53	00110101	19	5	53	00110101	31	0	48	00110000
8	space	32	00100000	20	2	50	00110010	32	space	32	00100000
9	1	49	00110001	21	space	32	00100000	33	2	50	00110010
10	2	50	00110010	22	3	51	00110011	34	7	55	00110111
11	2	50	00110010	23	7	55	00110111	35	2	50	00110010
12	space	32	00100000	24	9	57	00111001	36	#	35	00100011

d. Replace the last bit of each RGB value in each digital image pixel with the message bit value to be inserted, as shown in table 5.

Table 5: Replacing the last bit in each image pixel with message bits

R=11001000 G=10111100 B=11001011	R=11000011 G=10111000 B=10010010	R=11000000 G=10101011 B=01010110	R=11000110 G=10101001 B=00010011	R=11010011 G=10100010 B=00000110	R=11001001 G=10111100 B=11001010	R=11000011 G=10111001 B=10010010	R=11000001 G=10101011 B=01010111
R=11000110 G=10101000 B=00010011	R=11010010 G=10100110 B=00000110	R=11001000 G=10111110 B=11001100	R=11000110 G=10111111 B=10010111	R=11001000 G=10110010 B=01100001	R=11000111 G=10101000 B=00010010	R=11010001 G=10100001 B=00000100	R=11001001 G=10111111 B=11001101
R=11000110 G=10111110 B=10010111	R=11001001 G=10110010 B=01100001	R=11000110 G=10101001 B=00010010	R=11010000 G=10100001 B=00000100	R=11000000 G=10111100 B=11001010	R=10111100 G=10111110 B=11000000	R=10111001 G=10111111 B=10101010	R=11000100 G=10101010 B=01101011
R=11001110 G=10011100 B=01000001	R=11000001 G=10111100 B=11001010	R=10111101 G=10111110 B=11000000	R=10111000 G=10111111 B=10101011	R=11000100 G=10101010 B=01101011	R=11001110 G=10011100 B=01000000	R=10111111 G=10111010 B=11001000	R=10111100 G=10111110 B=10111110
R=10111110 G=11000100 B=10110001	R=11010101 G=01101010 B=01111111	R=00011000 G=10100010 B=01000110	R=10111110 G=10111010 B=11001001	R=10111100 G=10111100 B=10111111	R=10111111 G=11000100 B=10110000	R=11010101 G=01101010 B=01111110	R=00011000 G=10100010 B=01000110
R=11000100 G=10111110 B=10111111	R=11000111 G=10110101 B=10110010	R=11011110 G=10110110 B=10110100	R=11101000 G=10100001 B=10010101	R=10111010 G=01010111 B=01000010	R=11000101 G=10111110 B=10111110	R=11000111 G=10110100 B=10110010	R=11011110 G=10110110 B=10110100
R=11101000 G=10100000 B=10010101	R=10111011 G=01010110 B=01000001	R=11001000 G=10111101 B=11001010	R=11000010 G=10111001 B=10010011	R=11000000 G=10101010 B=01010111	R=11000110 G=10101000 B=00010010	R=11010011 G=10100110 B=00000110	R=11001000 G=10111110 B=11001010
R=11000010 G=10111000 B=10010011	R=11000001 G=10101010 B=01010110	R=11000111 G=10101001 B=00010010	R=11010010 G=10100111 B=00000111	R=11001000 G=10111111 B=11001101	R=11000111 G=10111110 B=10010110	R=11001001 G=10110011 B=01100001	R=11000110 G=10101000 B=00010011
R=11010000 G=10100000 B=00000101	R=11001000 G=10111110 B=11001100	R=11000110 G=10111110 B=10010110	R=11001000 G=10110011 B=01100001	R=11000110 G=10101000 B=00010011	R=11010001 G=10100000 B=00000100	R=11000001 G=10111101 B=11001010	R=10111100 G=10111111 B=11000001

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

R=11010100 G=01101010 B=01111111	R=10111110 G=10111010 B=11001001	R=10111000 G=10111110 B=10101011
R=00011001 G=10100010 B=01000111	R=10111101 G=10111100 B=10111110	R=11000100 G=10101010 B=01101110
R=11000101 G=10111111 B=10111110	R=10111110 G=11000100 B=10110000	R=11001110 G=10011100 B=01000000
R=11000110 G=10110101 B=10110011	R=11010100 G=01101011 B=01111110	R=11000000 G=10111101 B=11001011
R=11011110 G=10110110 B=10110101	R=00011000 G=10100010 B=01000110	R=10111100 G=10111110 B=11000001
R=11101000 G=10100000 B=10010100	R=10111110 G=10111010 B=11001000	R=10111001 G=10111110 B=10101010
R=10111011 G=01010110 B=01000010	R=10111101 G=10111101 B=10111110	R=11000101 G=10101011 B=01101110
R=11000100 G=10111111 B=10111111	R=10111110 G=11000101 B=10110000	R=11001111 G=10011100 B=01000000

e. Finally, convert the digital image binary code that has been inserted into a message into a new RGB image value (stego image), as shown in table 6.

Table 6: Conversion of the inserted binary image into the new RGB image value

200, 190, 203	195, 184, 146	192, 171, 86	198, 169, 19	211, 162, 6	201, 188, 202	195, 185, 146	193, 171, 87
198, 168, 19	210, 162, 6	200, 190, 204	198, 191, 151	200, 178, 97	199, 168, 18	209, 161, 4	201, 191, 205
198, 190, 151	201, 178, 97	198, 169, 18	208, 161, 4	192, 188, 202	188, 190, 192	185, 191, 170	196, 170, 111
206, 156, 65	192, 189, 203	188, 190, 193	185, 190, 170	197, 170, 110	206, 156, 64	191, 186, 200	188, 188, 190
190, 196, 177	213, 106, 127	24,162,70	190, 187, 201	188, 188, 191	191, 196, 176	213, 106, 126	24,162,70
196, 190, 191	199, 181, 178	222, 182, 180	232, 161, 149	182, 87,66	197, 190, 190	199, 180, 178	222, 182, 180
232, 160,149	183, 86,67	200, 189,202	194, 185,147	192, 170,87	198, 168,18	211, 162,6	200, 188,202
194, 184,147	193, 170,86	199, 169,18	210, 162,7	200, 191,205	199, 190,150	201, 179,97	198, 168,19
208, 160,5	200, 190,204	198, 190,150	200, 179,97	198, 168,19	209, 160,4	193, 189,202	188, 191,193
184, 190,171	196, 170,110	206, 156,64	192, 189,203	188, 190,193	185, 190,170	197, 171,110	207, 156,64
190, 186,201	189, 188,190	190, 196,176	212, 107,126	24,162,70	190, 186,200	189, 189,190	190, 197,176
212, 106,127	25,162,71	197, 191,190	198, 181,179	222, 182,181	232, 160,148	183, 86,66	196, 191,191

4. Message extraction process, After the message insertion process is successfully carried out, the next step is to carry out the message extraction process.
   a. Convert each RGB value in each stego image pixel into 8-bit binary form, as shown in table 7.

Table 7: Conversion of stego RGB image values into 8-bit binary

R=11010100 G=01101010 B=01111111	R=10111110 G=10111010 B=11001001	R=10111000 G=10111110 B=10101011	R=11010000 G=10100000 B=00000101	R=11000010 G=10111000 B=10010011	R=11101000 G=10100000 B=10010101	R=11000100 G=10111110 B=10111111	R=10111110 G=11000100 B=10110001	R=11001110 G=10011100 B=01000001	R=11000110 G=10111110 B=10010111	R=11000110 G=10101000 B=00010011	R=11001000 G=10111100 B=11001011
R=00011001 G=10100010 B=01000111	R=10111101 G=10111100 B=10111110	R=11000100 G=10101010 B=01101110	R=11001000 G=10111110 B=11001100	R=11000001 G=10101010 B=01010110	R=10111011 G=01010110 B=01000011	R=11000111 G=10110101 B=10110010	R=11010101 G=01101010 B=01111111	R=11000001 G=10111100 B=11001010	R=11001001 G=10110010 B=01100001	R=11010010 G=10100110 B=00000110	R=11000011 G=10111000 B=10010010
R=11000101 G=10111111 B=10111110	R=10111110 G=11000100 B=10110000	R=11001110 G=10011100 B=01000000	R=11000110 G=10111110 B=10010110	R=11000111 G=10101001 B=00010010	R=11001000 G=10111101 B=11001010	R=11011110 G=10110110 B=10110100	R=00011000 G=10100010 B=01000110	R=10111101 G=10111110 B=11000000	R=11000110 G=10101001 B=00010010	R=11001000 G=10111110 B=11001100	R=11000000 G=10101011 B=01010110
R=11000110 G=10110101 B=10110011	R=11010100 G=01101011 B=01111110	R=11000000 G=10111101 B=11001011	R=11001000 G=10110011 B=01100001	R=11010010 G=10100111 B=00000111	R=11000010 G=10111001 B=10010011	R=11101000 G=10100001 B=10010101	R=10111110 G=10111011 B=11001001	R=10111000 G=10111111 B=10101011	R=11010000 G=10100001 B=00000100	R=11000110 G=10111111 B=10010111	R=11000110 G=10101001 B=00010011
R=11011110 G=10110110 B=10110101	R=00011000 G=10100010 B=01000110	R=10111100 G=10111110 B=11000001	R=11000110 G=10101000 B=00010011	R=11001000 G=10111111 B=11001101	R=11000000 G=10101010 B=01010111	R=10111010 G=01010111 B=01000010	R=10111100 G=10111100 B=10111111	R=11000100 G=10101010 B=01101111	R=11000000 G=10111100 B=11001010	R=11001000 G=10110010 B=01100001	R=11010011 G=10100010 B=00000110
R=11101000 G=10100000 B=10010100	R=10111110 G=10111010 B=11001000	R=10111001 G=10111110 B=10101010	R=11010001 G=10100000 B=00000100	R=11000111 G=10111110 B=10010110	R=11000110 G=10101000 B=00010010	R=11000101 G=10111110 B=10111110	R=10111111 G=11000100 B=10110000	R=11001110 G=10011100 B=01000000	R=10111100 G=10111110 B=11000000	R=11000111 G=10101000 B=00010010	R=11001001 G=10111100 B=11001010
R=10111011 G=01010110 B=01000010	R=10111101 G=10111101 B=10111110	R=11000101 G=10101011 B=01101110	R=11000001 G=10111101 B=11001010	R=11001001 G=10110011 B=01100001	R=11010011 G=10100110 B=00000110	R=11000111 G=10110100 B=10110010	R=11010101 G=01101010 B=01111110	R=10111111 G=10111010 B=11001000	R=10111001 G=10111111 B=10101010	R=11010001 G=10100001 B=00000100	R=11000011 G=10111001 B=10010010
R=11000100 G=10111111 B=10111111	R=10111110 G=11000101 B=10110000	R=11001111 G=10011100 B=01000000	R=10111100 G=10111111 B=11000001	R=11000110 G=10101000 B=00010011	R=11001000 G=10111100 B=11001010	R=11011110 G=10110110 B=10110100	R=00011000 G=10100010 B=01000110	R=10111100 G=10111100 B=10111110	R=11000100 G=10101010 B=01101111	R=11001001 G=10111111 B=11001101	R=11000001 G=10101011 B=01010111

b. Take the last bit of each RGB value in each pixel of the stego image, then divide each

Journal of Theoretical and Applied Information Technology
31st October 2023. Vol.101. No 20

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

into 8 bits, then convert them into characters based on the ASCII table, as shown in table 8.

Table 8: Convert the last bit of stego image to ASCII character

No	Binari	Ascii	Character	No	Binari	Ascii	Character	No	Binari	Ascii	Character
1	00110 001	49	1	13	00110 100	52	4	25	00100 000	32	space
2	00111 001	57	9	14	00110 011	51	3	26	00110 011	51	3
3	00110 111	55	7	15	00100 000	32	space	27	00110 011	51	3
4	00100 000	32	space	16	00111 000	56	8	28	00100 000	32	space
5	00110 011	51	3	17	00110 101	53	5	29	00110 011	51	3
6	00110 111	55	7	18	00100 000	32	space	30	00110 100	52	4
7	00110 101	53	5	19	00110 101	53	5	31	00110 000	48	0
8	00100 000	32	space	20	00110 010	50	2	32	00100 000	32	space
9	00110 001	49	1	21	00100 000	32	space	33	00110 010	50	2
10	00110 010	50	2	22	00110 011	51	3	34	00110 111	55	7
11	00110 010	50	2	23	00110 111	55	7	35	00110 010	50	2
12	00100 000	32	space	24	00111 001	57	9	36	00100 011	35	#

c. Remove the marking character (#) at the end of the message, so that the message (ciphertext) is obtained, namely 197 375 122 43 85 52 379 33 340 272

5. Message decryption process, After the message extraction process has been successfully carried out, the next step is to perform the message decryption process using the private key (383, 338) as follows:
   a. Separate the values a and b in the ciphertext, provided that:
      a = Ciphertext of odd order
      b = Even-order ciphertext
      So obtained:
      $a_1 = 197$, $a_2 = 122$, $a_3 = 85$, $a_4 = 379$, $a_5 = 340$
      $b_1 = 375$, $b_2 = 43$, $b_3 = 52$, $b_4 = 33$, $b_5 = 272$
   b. Calculate m (original message) using the formula: m = b * a(p-1-x) mod p, then convert it into ASCII characters to produce plaintext, as shown in table 9.

Table 9: Convert m value to ASCII character

i	Plainteks m_i	$m = b * a^{(p-1-x)}$ mod p	Character
1	m_1	$375 * 197^{(383-1-338)}$ mod 383 = 82	R
2	m_2	$43 * 122^{(383-1-338)}$ mod 383 = 65	A
3	m_3	$52 * 85^{(383-1-338)}$ mod 383 = 80	P
4	m_4	$33 * 379^{(383-1-338)}$ mod 383 = 79	O
5	m_5	$272 * 340^{(383-1-338)}$ mod 383 = 84	T

Based on the decryption process above, the initial plaintext is obtained, namely the word "RAPOT".

From the results of calculations and complete steps which have been described in detail, it can be concluded that the implementation has been successful in returning the text inserted in the image.

After the analysis and design stages of the system have been completed, the next stage is system implementation. This system was built using the Visual Basic.NET programming language, with Microsoft Visual Studio 2010 software. This system consists of 6 (six) forms, including the intro form, main form, key generation form, encryption and insertion form, extraction form and description, form about me and form about the application.

4. CONCLUSIONS

Based on the discussion and results of the research, the following conclusions are obtained:

1. The built system can perform the process of encrypting text files, inserting, extracting and decrypting text files again so that they return to their original form.
2. The size of the text file inserted into the image must be smaller than the size of the image (cover image).
3. The image file size after insertion is larger than the original image size.
4. The existence of a secret message embedded in an image is difficult for the sense of sight to see because visually the two images look the same.
5. The initial message will be overwritten if another message is inserted.
6. For further research, it is important to discuss maintaining the size of the inserted image file the same as the previous file size.
7. Application performance needs to be improved so that it can receive different message inserts.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFRENCES:

[1] A. Widarma, "Kombinasi Algoritma Aes, Rc4 Dan Elgamal Dalam Skema Hybrid Untuk Keamanan Data," *Cessjournal Comput. Eng. Syst. Sains*, Vol. 1, No. 1, 2016.

[2] Handrizal, F. Nurahmadi, And S. D. Siregar, "Hybrid Cryptosystem Using Elgamal Algorithm And Beaufort Cipher Algorithm For Data Security," *J. Theor. Appl. Inf. Technol.*, Vol. 100, No. 7, 2022.

[3] K. Vanitha, K. Anitha, Z. Rahaman, And M. Musthafa, "Analysis_Of_Cryptographic_Techniques_In Network Security," *J. Appl. Sci. Comput.*, Vol. 5, No. 8, 2018.

[4] R. Munir, "Algoritma Enkripsi Citra Digital Dengan Kombinasi Dua Chaos," *Chaos*, Vol. 2012, No. Snati, 2012.

[5] Krisnawati, "Metode Least Significant Bit (Lsb) Dan End Of File (Eof)," *Seminar*, Vol. 2008, No. Semnasif, 2008.

[6] A. A. Alarood, A. A. Manaf, M. J. Alhaddad, And M. S. Atoum, "Hiding A Message In Mp3 Using Lsb With 1, 2, 3 And 4 Bits," *Int. J. Comput. Networks Commun.*, Vol. 8, No. 3, 2016, Doi: 10.5121/Ijcnc.2016.8305.

[7] J. R. Jayapandiyan, C. Kavitha, And K. Sakthivel, "Enhanced Least Significant Bit Replacement Algorithm In Spatial Domain Of Steganography Using Character Sequence Optimization," *Ieee Access*, Vol. 8, 2020, Doi: 10.1109/Access.2020.3009234.

[8] F. Deeba, S. Kun, F. A. Dharejo, And H. Memon, "Digital Image Watermarking Based On Ann And Least Significant Bit," *Inf. Secur. J.*, Vol. 29, No. 1, 2020, Doi: 10.1080/19393555.2020.1717684.

[9] K. K. Jabbar, M. B. Tuieb, And S. A. Thajeel, "Digital Watermarking By Utilizing The Properties Of Self-Organization Map Based On Least Significant Bit And Most Significant Bit," *Int. J. Electr. Comput. Eng.*, Vol. 12, No. 6, 2022, Doi: 10.11591/Ijece.V12i6.Pp6545-6558.

[10] S. A. Nie, G. Sulong, R. Ali, And A. Abel, "The Use Of Least Significant Bit (Lsb) And Knight Tour Algorithm For Image Steganography Of Cover Image," *Int. J. Electr. Comput. Eng.*, Vol. 9, No. 6, 2019, Doi: 10.11591/Ijece.V9i6.Pp5218-5226.

[11] E. H. J. Halboos And A. M. Albakry, "Hiding Text Using The Least Significant Bit Technique To Improve Cover Image In The Steganography System," *Bull. Electr. Eng. Informatics*, Vol. 11, No. 6, 2022, Doi: 10.11591/Eei.V11i6.4337.

[12] S. K. Salim, M. M. Msallam, And H. I. Olewi, "Hide Text In An Image Using Blowfish Algorithm And Development Of Least Significant Bit Technique," *Indones. J. Electr. Eng. Comput. Sci.*, Vol. 29, No. 1, 2023, Doi: 10.11591/Ijeecs.V29.I1.Pp339-347.

[13] Z. Bin Faheem *Et Al.*, "Image Watermarking Using Least Significant Bit And Canny Edge Detection," *Sensors*, Vol. 23, No. 3, 2023, Doi: 10.3390/S23031210.

[14] H. H. Liu, P. C. Su, And M. H. Hsu, "An Improved Steganography Method Based On Least-Significant-Bit Substitution And Pixel-Value Differencing," *Ksii Trans. Internet Inf. Syst.*, Vol. 14, No. 11, 2020, Doi: 10.3837/Tiis.2020.11.016.