Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th October 2024. Vol. 4. No. 1; e-ISSN: 2808-4519

Application of Cryptography and Steganography Techniques to Improving the Security of Text Messages with RC4 Algorithm and MSB Method

Ihsan Muchlis[1], Achmad Fauzi[2], Husnul Khair[3]

[1, 2, 3]*Teknik Informatika, STMIK KAPUTAMA*
Jl. Veteran No. 4A-9A, Binjai, Sumatera Utara, Indonesia
ihsanmuchlis27@gmail.com[1], fauzyrivai88@gmail.com[2], Husnul.khair@gmail.com [3]*

Abstract

This study discusses the application of cryptography and steganography techniques to improve the security of text messages using the Rivest Code 4 (RC4) algorithm and the Most Significant Bit (MSB) method. In the ever-growing digital era, data security is a top priority due to the increasing threat of cybercrime that can harm many parties. RC4 is a cryptography algorithm known for its encryption and decryption speed, while the MSB method is an effective steganography technique for hiding information in digital images. This study aims to develop an application that is able to encrypt text messages with the RC4 algorithm and hide them in digital images using the MSB method. With this combination, data is not only encrypted but also hidden, thus providing two layers of security to protect information from unauthorized access. The results of the study show that the combination of RC4 cryptography and MSB steganography techniques successfully improves data security well. The developed application is able to protect sensitive information from the risk of data theft and cyberattacks. In addition, this technique is also easy to implement and can be applied in various sectors, such as banking, health, and business communications, to protect sensitive data from unauthorized access.

Keywords: *Cryptography, Steganography, Most Significant Bit (MSB)*

1. Introduction

Security Digital security is a major issue because digital communication is highly vulnerable to loss of information. The larger the number of users and the more the greater the number of users and the more diverse the types of digital communication services, the greater the chances of the occurrence of cybercrime or what is known as cybercrime and the The culprit is called a “hacker” who is a threat to crime in cyberspace. Generally speaking, a “hacker” is someone who has expertise in using computers and computer networks to explore using computers and computer networks to explore, modify, or hack into a computer system or network in an unauthorized or unauthorized manner. unauthorized or unpermitted.

So that requires a security system that can protect the information that is sent so that it is not easily accessed. protect the information sent so that it is not easily accessed. There are many methods to stay away from these crimes, one of which is with hiding a message in another message by utilizing the Rivest Code 4 (RC4) algorithm combined with Most Significant Steganography. Rivest Code 4 (RC4) algorithm combined with Most Significant Bit (MSB) steganography. (MSB) STEGANOGRAPHY. The RC4 algorithm is known for its high encryption and decryption speed, while stego MSB offers a high speed of encryption and decryption, while MSB stego offers resistance to certain attacks by utilizing digital images. by utilizing digital images. This combination provides fast and secure, making it an efficient and reliable solution for data protection.Combining the cryptographic techniques of the RC4 algorithm and the MSB stego methodis a new approach that has not yet been researched. It makes a unique and innovative contribution unique and innovative contribution to the field of digital security, offering a more comprehensive and secure solution. comprehensive and secure solution.

2. Research Methodology

2.1. cryptography

Cryptography comes from the Greek words crypto and graphia. Crypto means secret and graphia means writing. In general, cryptography can be defined as a science and art that aims to maintain the confidentiality of a message. Cryptography has been known since ancient times, according to the history of cryptography has been used since thousands of years ago, which was introduced by the Egyptians during wartime to send secret messages to a general sent by couriers [1].

2.2. RC4 Algorithm

The Rivest Code 4 (RC4) Cryptographic Algorithm is a symmetric key algorithm algorithm created by RSA Data Security Inc (RSADSI) in the form of a stream chipper [3]. This algorithm was invented in 1978 by Ronald Rivest and became the symbol of RSA security (which stands for the names of its three inventors: Rivest Shamir Adleman). RC4 uses key lengths from 1 to 256 byteswhich is used to initialize a table 256 bytes long. This table table is used for the next generation of pseudo random which uses XOR with plaintext to produce ciphertext. Each element in the table is exchanged at least once.

The RC4 algorithm uses two S-Box arrays with length 256 and containing permutations of numbers from 0 to 255, and a second S-Box, which contains the result of the permutation is a function of the key of varying length. How the RC4 algorithm is the initialization of the first S-Box, S[0], S[1], S[2], ..., S[255], with the numbers 0 to 255. numbers 0 to 255. The first S-Box is filled sequentially with S[0] = 0, S[1] = 1, S[2] = 2, ...,S[255] = 255. Then initialize the second array (S-Box), let's say array K with length 256. Fill array K with the key that repeated until the entire array K[0], K[1], ...,K[255] has been completely filled [10].

Here are the general steps of the RC4 algorithm:
1) Initialize the S-Box (Array S) with the following equation:
for i = 0 to 255
S[i] = i (1)
2) Next the S-Box (Array K) initialization process with the following equation:
Key array with key length “length”
for i = 0 to 255
K[i] = Key [i mod length] (2)
3) Then the S-Box randomization step is formulated with the following equation:
I = 0; j = 0
for i = 0 to 255
j = (j + S[i] + K[i]) mod 256
swap S[i] and S[j] (3)
After that, create a pseudo random byte with the following steps:
i = (i + 1) mod 256
j = (j + S[i]) mod 256
swap S[i] and S[j]
t = (S[i] + S[j]) mod 256
K = S[t]

2.3. Steganography

Steganography is the study of hiding text in another medium that is so that it is intermingled with it. Text, image, audio, and video materials can all be used to hide messages Steganography comes from the Greek word “steganos” which means “secret” and “graphy”. closed/secret and “graphy” which means writing/image. Steganography, thus thus, is a type of writing that is hidden or secret. Steganography works by inserting a message or file into a medium, such as a file, sound, video, or image, so that the presence of the message is not directly visible to the human senses. directly visible to the human senses. Steganography attempts to hide message or information. In steganography, the message is hidden and transformed in such a way that it cannot be seen directly and does not draw attention; for example, if text is buried within an image, the text will not be noticed. immediately visible. The advantage of steganography is that the communication in steganography does not attract the attention of others, in contrast to cryptography which is cannot be hidden and may attract the attention of others. Although cryptographic communication is difficult to decipher, it may draw attention and suspicion. attention and arouse suspicion. Hiding a message or file in another may affect the quality of the material.

2.4. Most Significant Bit (MSB)

The process of inserting a message into a digital image using the Most Significant Bit (MSB) method is the same as the LSB and LSB+1 methods. Most Significant Bit (MSB) method is also the same as the LSB and LSB+1 methods, the difference is in the bit where the message is inserted. is in the bit where the message is inserted. If in the LSB method, the message is inserted at the LSB bit (the 8th bit), in the LSB+1 method, the message is inserted at the 7th bit, while in the MSB method, the message is inserted in the 1st bit.

Table 1: RGB values of 3 pixels in binner form

R	G	B
00110101	11010110	11101010
11110100	00111001	11100001
01110001	10010001	11100001

The message to be inserted is the character “R”, whose binary value is “01010010”. The message will be inserted using the MSB method, so the result image will be generated with the following bit sequence:

Table 2: Three pixel RGB value change in binner form

R	G	B
00110101	**1**1010110	**1**1101010
11110100	**0**0111001	**1**1100001
01110001	**1**0010001	**1**1100001

2.5. Image definition

Images are still images (photos) or moving images such as video recordings, while digital is an image or the results of image processing done digitally using a computer [6]. digitally using a computer [6]. Digital image refers to 2-dimensional image processing using a computer. Digital image is an array that contains real and complex values presented with a certain row of bits. The process of creating a digital image begins with taking pictures with a digital camera or scanner. The image is then converted into a digital format so that it can be processed by a computer. At this stage At this stage, the image is broken down into small pixels that have different color values and light intensities [7].

3. Results

3.1. RC4 algorithm encryption calculation

Here are the steps of manual calculation for plaintext “Ihsan Muchlis” with the key ‘KEY12345’.
S-Box Array initialization
The following S-Box Array length is 256 bytes:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197,198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255
First iteration (KSA):
 i=0
 j=(0+S[0]+K[0 mod 8])mod 256
 j=(0+0+75)mod 256=75
 Swap S[0] with S[75]
Second iteration:
 i=1
 j=(75+S[1]+K[1mod 8])mod 256
 j=(75+1+69)mod 256=145
 Swap S[1] with S[145]
continue calculation until i = 255……
So that the result of the swap or byte exchange becomes as follows:
The following is the length of the 256 byte S-Box Array after the swap:
26, 145, 236, 32, 139, 185, 70, 86, 134, 165, 87, 68, 96, 19, 120, 118, 163, 249, 100, 243, 33, 51, 128, 226, 183, 36, 75, 38, 154, 234, 89, 102, 192, 11, 191, 81, 79, 197, 76, 119, 17, 88, 219, 122, 64, 135, 212, 164, 4, 131, 104, 239, 40, 82, 155, 251, 116, 45, 143, 91, 0, 153, 245, 186, 232, 189, 65, 18, 238, 140, 112, 12, 138, 106, 194, 101, 6, 109, 133, 105, 127, 188, 227, 94, 21, 206, 27, 130, 49, 207, 201, 93, 173, 43, 205, 152, 46, 39, 107, 246, 83, 172, 117, 253, 95, 141, 115, 160, 2, 69, 240, 210, 25, 166, 20, 199, 208, 90, 123, 3, 28, 203, 55, 250, 136, 53, 156, 162, 204, 221, 217, 44, 150, 98, 108, 196, 151, 167, 220, 223, 15, 200, 157, 247, 121, 158, 61, 228, 244, 31, 254, 24, 97, 187, 7, 237, 190, 34, 195, 41, 125, 180, 175, 60, 241, 211, 179, 42, 67, 78, 132, 229, 84, 56, 231, 5, 222, 29, 13, 30, 124, 218, 224, 213, 216, 214, 182, 255, 85, 178, 198, 54, 50, 174, 23, 126, 62, 230, 181, 59, 169, 209, 71, 22, 52, 103, 113, 66, 1, 48, 248, 177, 147, 47, 142, 176, 37, 148, 99, 92, 225, 149, 57, 14, 233, 16, 63, 202, 110, 72, 193, 74, 235, 161, 77, 129, 10, 137, 171, 242, 8, 159, 170, 35, 144, 252, 73, 114, 58, 9, 111, 146, 215, 184, 80, 168

Pseudo-Random Generation Algorithm (PRGA)
First iteration:
i = 0, j = 0
i = (i + 1) mod 256
i = (0 + 1) mod 256
i = 1 mod 256
i = 1, S[i] = S[1]=145
and
j = (j + S[i]) mod 256j = (0 + S[1]) mod 256
j = (0 + 145) mod 256
j = 145 mod 256
j = 145, S[j] = S[145] = 158
swap S[i] & S[j]
t = (S[158] + S[145]) mod 256
t = (158 + 145) mod 256
t = 303 mod 256
t = 47
K = S[t] = S[47] = 164
So, the first key for encryption is 164

continue until the 13th iteration…..

Table 3: Process XOR Plaintext with Keystream byt

Plainteks	01001001	01101000	01110011	01100001	01101110	00100000
Keystream	10100100	00001011	01000001	10111110	11010001	10110010
P ⊕ K	11101101	01100011	00110010	11011111	10111111	10010010
Cipherteks	ed	63	32	df	bf	92

Plainteks	01001101	01110101	01100011	01101000	01101100	01101001	01110011
Keystream	01000001	01000011	11101011	01010111	00011011	10111110	00100011
P ⊕ K	00001100	00110110	10001000	00111111	01110111	11010111	01010000
Cipherteks	c	36	88	3f	77	d7	50

Ciphertext (Hex) Conclusion: ed6332dfbf920c36883f77d750

3.2. Message Insertion Process into Digital Image

The message in the form of plaintext “Ihsan Muchlis” after being encrypted has been obtained chipertext “ed6332dfbf920c36883f77d750” with binary as in table 3 above. The chipertext will be inserted in the digital image file through the image pixels. image pixels. The following image will be inserted message.

Fig. 1: Digital image to be inserted with text

Table 4: Image Pixels After Message Insertion 5x5 pixels

Pixel dengan nilai desimal	R =198 G =191 B =206	R =97 G =103 B =169	R =28 G =192 B =0	R =251 G =48 B =188	R =255 G =116 B =237
Pixel dengan nilai biner	R =11000110 G =10111111 B =11001110	R =01100001 G =01100111 B =10101001	R =00011100 G =11000000 B =00	R =11111011 G =00110000 B =10111100	R =11111111 G =01110100 B =11101101
Pixel dengan nilai desimal	R =232 G =174 B =240	R =251 G =193 B =131	R =95 G =63 B =191	R =36 G =25 B =229	R =44 G =15 B =200
Pixel dengan nilai biner	R =11101000 G =10101110 B =11110000	R =11111011 G =11000001 B =10000011	R =01011111 G =00111111 B =10111111	R =00100100 G =00011001 B =11100101	R =00101100 G =00001111 B =11001000
Pixel dengan nilai desimal	R =99 G =180 B =2	R =39 G =184 B =48	R =63 G =240 B =217	R =231 G =122 B =243	R =76 G =223 B =220
Pixel dengan nilai biner	R =01100011 G =10110100 B =10	R =00100111 G =10111000 B =00110000	R =00111111 G =11110000 B =11011001	R =11100111 G =01111010 B =11110011	R =01001100 G =11011111 B =11011100
Pixel dengan nilai desimal	R =100 G =117 B =237	R =83 G =100 B =28	R =20 G =113 B =90	R =167 G =58 B =51	R =255 G =179 B =176
Pixel dengan nilai biner	R =01100100 G =01110101 B =11101101	R =01010011 G =01100100 B =00011100	R =00010100 G =01110001 B =01011010	R =10100111 G =00111010 B =00110011	R =11111111 G =10110011 B =10110000
Pixel dengan nilai desimal	R =255 G =180 B =148	R =194 G =118 B =86	R =173 G =77 B =53	R =255 G =202 B =183	R =172 G =84 B =62
Pixel dengan nilai biner	R =11111111 G =10110100 B =10010100	R =11000010 G =01110110 B =01010110	R =10101101 G =01001101 B =00110101	R =11111111 G =11001010 B =10110111	R =10101100 G =01010100 B =00111110

3.3. RC4 Algorithm Message Decryption Process

To decrypt the text that has been encrypted using the RC4 algorithm, we can use the same process as encryption, since the XOR operation used is involutive. used is involutive. Therefore, we only need to XOR between the ciphertext with the same keystream to get back the

original plaintext. Let's perform the decryption process on the ciphertext “ed6332dfbf920c36883f77d750” with the same keystream used during encryption.

Table 4: Process XOR Ciphertext with Keystream byt

Ciphertext	11101101	01100011	00110010	11011111	10111111	10010010
Keystream	10100100	00001011	01000001	10111110	11010001	10110010
C ⊕ K	01001001	01101000	01110011	01100001	01101110	00100000
Desimal	49	68	73	61	6e	20
Konversi ASCII	I	h	s	a	n	(Spasi)

Ciphertext	00001100	00110110	10001000	00111111	01110111	11010111	01010000
Keystream	01000001	01000011	11101011	01010111	00011011	10111110	00100011
C ⊕ K	01001101	01110101	01100011	01101000	01101100	01101001	01110011
Desimal	4d	75	63	68	6c	69	73
Konversi ASCII	M	u	c	h	l	i	s

Plaintext Conclusion: Ihsan Muchlis

4. Discussion

This design will discuss several existing menu displays such as designing the main form, designing the encryption form, and designing the decryption form. design of the decryption form.

4.1. Encryption and Decryption form

Here is the design of the main form:

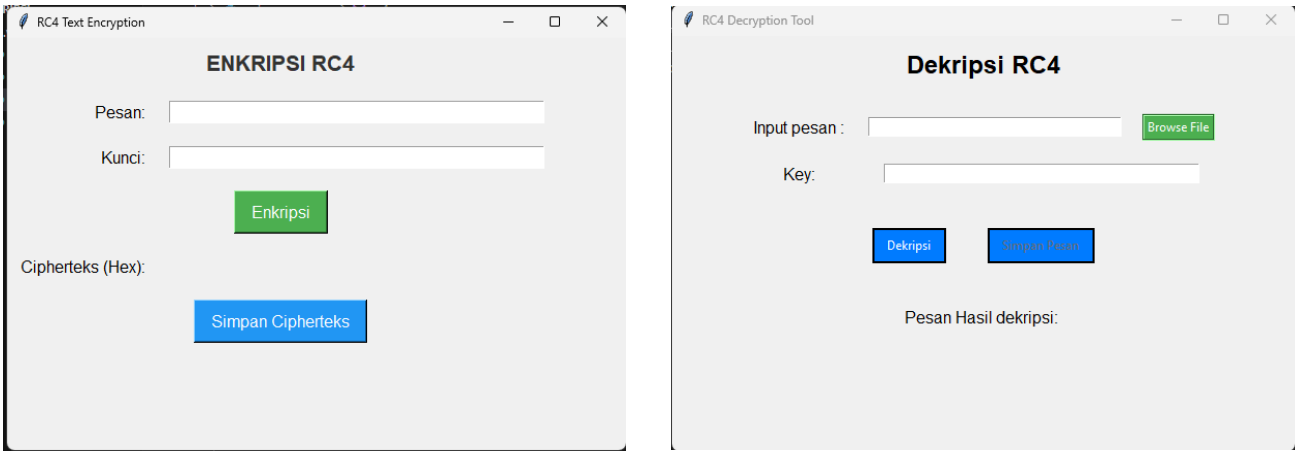

Fig. 2: Encryption and Decryption form

4.2. Encoding and Decoding form

Here is the design of the main form:

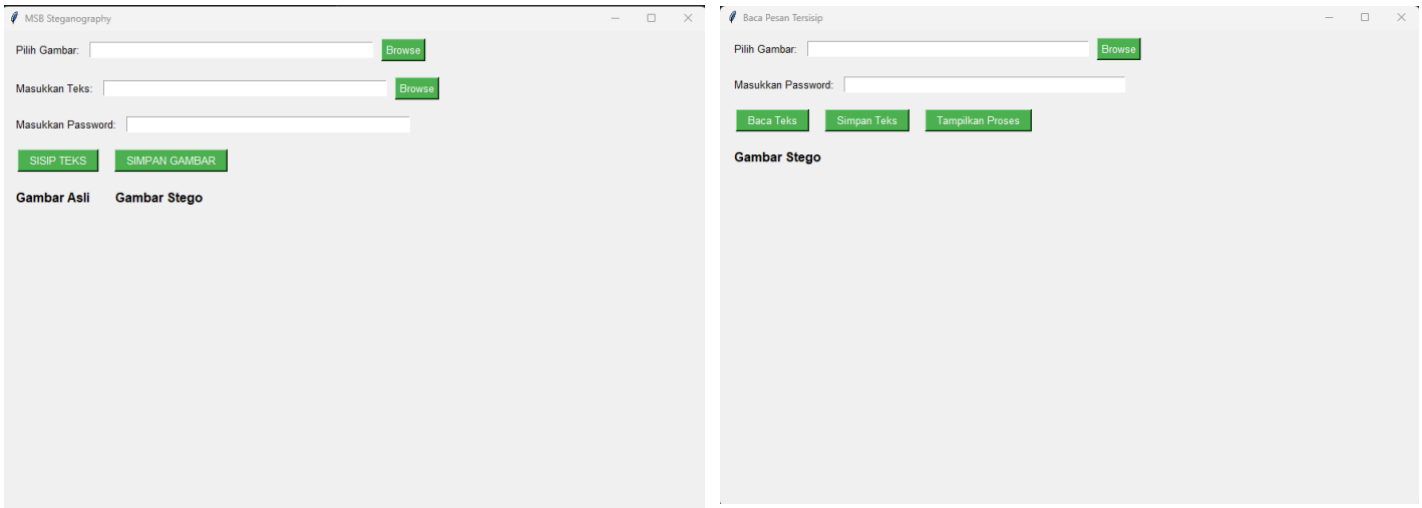

Fig. 3: Encoding and Decoding form

5. Conclusion

Based on the results of research and system testing, the authors draw conclusions which include:

1. From the test results of the encryption system using the RC4 (Rivest Code 4) algorithm, the running application is successful in implementing the RC4 (Rivest Code 4) algorithm in encrypting a text message. 2.
2. The application designed to implement text message insertion with the Most Significant Bit (MSB) method has been tested and succeeded in applying the Most Significant Bit (MSB) method in inserting text messages in digital images.
3. In testing encryption and message insertion, it has been carried out and succeeded in designing an application for hiding messages in digital images using the MSB method steganography technique as a message insertion and the RC4 algorithm as a message locker.

Referensi

[1] Jatmoko, C., Handoko, L. B., Sari, C. A., Ignatius, D. R., & Setiadi, M. (2018). U JI P ERFORMA P ENYISIPAN P ESAN DENGAN M ETODE LSB DAN MSB. 14(1), 47–56.

[2] Purba, B., Apustriani Gulo, F., Indah Utami, N., & Annisa Sihotang, Y. (2020). Pengamanan File Teks Menggunakan Algoritma RC4. Seminar Nasional Teknologi Komputer & Sains (SAINTEKS).

[3] Romadhan, D., & Ferdiansyah, F. (2022). Implementasi Keamanan Database Menggunakan Kriptografi Rc4 Pada Sistem Milik Pt. Torop Sumber Makmur. Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI) Jakarta-Indonesia, September, 368–376.

[4] Basim, Z., & Painem. (2020). Implementasi Kriptografi Algoritma Rc4 Dan 3Des Dan Steganografi Dengan Algoritma Eof Untuk Keamanan Data Berbasis Desktop Pada Smk As-Su'Udiyyah. Skanika, 3(4), 54–60.

[5] Eko Setiawan, A., & Pasaribu, A. (2020). Penerapan Steganografi Pada Citra Digital Menggunakan Metode Least Significant Bit (LSB) Kombinasi RC4 Berbasis Mobile Android. Aisyah Journal Of Informatics and Electrical Engineering (A.J.I.E.E), 2(1). https://doi.org/10.30604/jti.v2i1.27

[6] Hidayat, M., Tahir, M., Sukriyadi, A., Sulton, A., A, C. A. S., & F, S. A. (2023). Penerapan Kriptografi Caesar Chiper Dalam Pengamanan Data. Jurnal Ilmiah Multidisiplin, 2(03), 35–41. https://doi.org/10.56127/jukim.v2i03.619

[7] Irdayani. (2019). Keamanan Citra Menggunakan Algoritma Route Cipher. Majalah Ilmiah INTI, 14(1), 82–85.

[8] Jatmoko, C., Handoko, L. B., Sari, C. A., Ignatius, D. R., & Setiadi, M. (2018). U JI P ERFORMA P ENYISIPAN P ESAN DENGAN M ETODE LSB DAN MSB. 14(1), 47–56.