COMBINATION OF CRYPTOGRAPHY AND STEGANOGRAPHY IN IMPROVING TEXT DATA SECURITY USING DES (DATA ENCRYOTION STANDARD) AND LSB (LEAST SIGNIFICANT BIT) METHODS

Fahmi Kurniawan[1]*, Zulham Sitorus[2], Randi Rian Putra [3]
[1-2]Fakultas Sains Dan Teknologi, Sistem Komputer, Universitas Pembangunan Panca Budi
[3]Fakultas Sains Dan Teknologi, Teknologi Informasi, Universitas Pembangunan Panca Budi

Keywords:
Cryptography; Steganography; Text Data Protection; DES; LSB.

***Correspondence Address:**
fahmikurniawan@dosen.pancabudi.ac.id

Abstract: Information security in this globalized era is increasingly becoming a vital requirement in various aspects of life. Information will have a higher value if it concerns aspects of business decisions, security, or public interest. The purpose of this research is to increase the security of text data by using cryptography and Steganography techniques. Cryptography is the science of keeping messages secret by encoding them into a form that is no longer understandable. The DES (Data Encryption Standard) method is a symmetric encryption algorithm, where the message encryption and decryption processes use the same key. Steganography is a technique of disguising messages into a medium without other people realizing that the media has been inserted with a message, because the output of steganography is data that has the same form of perception as the original data when viewed using human senses. The LBS (Least Significant Bit) method is a steganography method that works to insert messages by replacing the lowest bit in a byte of image media as a message carrier. The combination of these two methods is expected to increase the security of text data in the form of messages or information, and make it difficult for irresponsible parties to open the message.

INTRODUCTION

Information security in this globalized era is increasingly becoming a vital requirement in various aspects of life. Information will have a higher value if it concerns aspects of business decisions, security, or public interest (Wijayanti & Romadlon, 2022). Where this information will certainly be in great demand by various parties who also have an interest in it. In everyday life, humans depend a lot on information technology, both from small things to complex problems (Kurniawan & Putra, 2023). Examples of information technology in everyday life are Mobile

Banking, Email, SMS, Chatting and so on. Advances in information technology provide many benefits for human life (Ifan Rizqa & Safitri, 2022).

But the benefits offered by information technology also lead to crimes such as data theft. So that the development of science to secure data is increasingly being improved so that technology users always feel safe (Tjoanda et al., 2024). Various ways are done to maintain data security. One of them is by encoding data into codes that are not understood, so that if tapped it will be difficult to find out the actual information (Suparman & Sewaka, 2022). For many reasons, security and confidentiality are essential in data communication. There are several attempts to deal with the security issues of confidential data sent over the internet, including using cryptography and steganography techniques. Cryptography is the science and art of keeping messages secret by encoding them into a form that is no longer understandable (Buulolo & Sindar, 2020).

Cryptographic techniques can arouse suspicion in third parties who are not entitled to receive information because the message is disguised by changing the original message to make it unreadable(Supiyandi et al., 2020). Furthermore, the third party will have a desire to know the contents of the secret message and try to decipher the actual information (Thahara & Siregar, 2021).

Meanwhile, steganography reduces suspicion because the camouflaged message is hidden in other media. Steganography can disguise a message into a medium without others realizing that the medium has been inserted a message, because the output of steganography is data that has the same form of perception as the original data when viewed using human senses, while changes in cryptography messages can be seen and realized directly by human senses (Yanto et al., 2024). In steganography, secret data is inserted into other data called cover-object and produces a stego-object (steganography result). Commonly used container media in steganography techniques are images, audio, video(Rizal et al., 2021). The stored data can also be in the form of images, audio, video, text. Steganography that is applied is steganography on image documents (images) (Sidiq et al., 2023).

One method that can be used in cryptography and steganography in data security is the DES (Data Encryption Standard) method, which is a symmetric encryption

algorithm, where the message encryption and decryption process uses the same key (Rantelinggi & Saputra, 2020). So even though a cryptographer understands well the algorithm used to encode the message, if he does not know the key used, he will not be able to decrypt the message so that the message is completely safe (Adhar, 2019). The Least Significant Bit (LSB) method is a steganography method that works to insert messages by replacing the lowest bit in a byte of the message carrier media (Minarni & Redha, 2020).

This research aims to increase the security of text data by combining cryptography and Steganography techniques. The combination of these two methods is expected to increase the security of text data in the form of messages or information that uses image media as a message delivery medium and can make it difficult for irresponsible parties to steal data by opening the message (Jum'ah & Sarimuddin, 2024).

RESEARCH METHODS

Making text data security applications in this study uses a research framework that is used as a research method. The research framework can clearly describe the structure of the research plan and help researchers formulate relevant research questions. An inductive case study research framework is different from a deductive case study research framework. The following will describe the research framework carried out in building a text data security system by combining cryptography technology with the DES method and steganography with the LBS method:

1. Literature Study:

   Literature study serves to determine the theory used in this research. The referenced theory is related to cryptography techniques using the DES (Data Encrytion Standard) algorithm and steganography techniques using the LBS (Least Significant Bit) method. Sources are obtained through searching for materials through books, journals and the internet.

2. Analysis:

   Analysis provides an explanation of the flow of problems and solutions in determining the final result of the application of text data security combination of

cryptography and steganography techniques. At this stage of the analysis, we will analyze which technique is used first to provide text data security. This stage is carried out to see the correctness of the design that will be used in the application program then.

3. Design:

   The design is carried out to build a text security data security application combining cryptography and steganography techniques based on the results of the analysis carried out. The design begins with designing the system using UML (Unified Modeling Language) tools and the application of the system design that has been produced is poured into an application program using Microsoft Visual Stuido .Net 2010.

4. Testing:

   Testing is done to determine whether the application is running or not and whether the results are in accordance with the results of the calculation analysis. In testing is done using BlackBox Tasting. Black Box Testing is a test that focuses on the functional specifications of the software, the tester can define a set of input conditions and test the functional specifications of the program.

RESULTS AND DISCUSSION

LSB (Least Significant Bit) method inserts the message into the carrier media at the smallest bits of the content, so the header The use of the LSB method to secure secret messages in digital images is considered insufficient. If the carrier image falls into the hands of someone who should not, then by using the LSB method, that person can know the contents of the secret message stored in the image. To be able to further secure the message embedded in the digital image, DES encryption will be carried out on the secret message before it is embedded in the carrier media. While DES operates using a 64-bit block and a 56-bit key by trying all combinations requiring 256 combinations or about 7x 1017 or 70 million billion combinations. making it easier because it is easy to calculate the bits in the LSB which are small and easy to combine the two.

1. **Key Generation**

Key generation is done by getting input from the user in the form of a 64-bit long key. The key can be obtained from direct input in the form of hex numbers or it can also be from converting characters into hex. There is no standard algorithm for determining this key. Based on the scenario that has been created, the hexa key used is "13 34 57 79 9B BC DF F1". To represent the hexa key into binary, a conversion of the hexa values in the key will be performed. The following is the calculation of the Hexa to Binary Key Conversion Calculation results.

Byte Ke	Hexa	Biner
1	13	0001 0011
2	34	0011 0100
3	57	0101 0111
4	79	0111 1001
5	9B	1001 1011
6	BC	1011 1100
7	DF	1101 1111
8	F1	1111 0001

To perform permutation on a 64-bit hexa key, we will reduce the bits in the key to 56-bit. Where this bit reduction process will place the key bit sequences according to the Permutation Choice One (PC-1) table.

Bit	
Input	Output
1	57
2	49
3	41
4	33
5	25
6	17
7	9
8	1
9	58
10	50
11	42
12	34
13	26
14	18

Bit	
Input	Output
15	10
16	2
17	59
18	51
19	43
20	35
21	27
22	19
23	11
24	3
25	60
26	52
27	44
28	36

Bit	
Input	Output
29	63
30	55
31	47
32	39
33	31
34	23
35	15
36	7
37	62
38	54
39	47
40	38
41	30
42	22

Bit	
Input	Output
43	14
44	6
45	61
46	53
47	45
48	3
49	29
50	21
51	13
52	5
53	28
54	20
55	12
56	4

To calculate 16 subkeys, a Shift Left operation will be performed on C_0 and

D_0. The size of the Shift Left operand is determined based on the Schedule Key table.

Iterasi	Left Shift	Iterasi	Left Shift
1	1	9	1
2	1	10	2
3	2	11	2
4	2	12	2
5	2	13	2
6	2	14	2
7	2	15	2
8	2	16	1

For the complete left shift calculation results will be presented in the table below.

Iterasi	Left Shift	Ci Di	Key
1	1	C1	11100001100110010101010111111
		D1	10101010110011001111000111110
2	1	C2	11000011001100101010101111111
		D2	01010101100110011110001111101
3	2	C3	00001100110010101010111111111
		D3	01010110011001111000111101010
4	2	C4	00110011001010101011111111100
		D4	01011001100111100011110101010
5	2	C5	11001100101010101111111110000
		D5	01100110011110001111010101010
6	2	C6	00110010101010111111111000011
		D6	10011001111000111101010101010
7	2	C7	11001010101011111111100001100
		D7	01100111100011110101010101100
8	2	C8	00101010101111111100001100110
		D8	10011110001111010101010110010
9	1	C9	01010101011111111000011001100
		D9	00111100011110101010101100110
10	2	C10	01010101111111100001100110010
		D10	11110001111010101010110011000
11	2	C11	01010111111110000110011001010
		D11	11000111101010101011001100110
12	2	C12	01011111111000011001100101010
		D12	00011110101010101100110011110
13	2	C13	01111111100001100110010101010
		D13	01111010101010110011001111000
14	2	C14	11111110000110011001010101010

		D14	11101010101011001100111110001
15	2	C15	11111000011001100101010101111
		D15	10101010101100110011111000111
16	1	C16	11110000110011001010101011111
		D16	01010101011001100111100011111

The next step is to combine the key pairs C_n and D_n and then perform permutations according to the order of the PC-2 table.

Bit	
Input	**Output**
1	14
2	17
3	11
4	24
5	1
6	5
7	2
8	28
9	15
10	6

Bit	
Input	**Output**
11	21
12	10
13	23
14	19
15	12
16	4
17	26
18	8
19	16
20	7

Bit	
Input	**Output**
21	27
22	20
23	13
24	2
25	41
26	52
27	31
28	37
29	47
30	55

Bit	
Input	**Output**
31	30
32	40
33	51
34	45
35	33
36	48
37	44
38	49
39	39
40	56

Bit	
Input	**Output**
41	34
42	53
43	46
44	42
45	50
46	36
47	29
48	32

From the permutation using the PC-2 table, the following results are obtained

K_1 = 000110 110000 001011 101111 111111 000111 000001 110010
K_2 = 011110 011010 111011 011001 110110 111100 100111 100101
K_3 = 010101 011111 110010 001010 010000 101100 111110 011001
K_4 = 011100 101010 110111 010110 110110 110011 010100 011101
K_5 = 011111 001110 110000 000111 111010 110101 001110 101000
K_6 = 011000 111010 010100 111110 010100 000111 101100 101111
K_7 = 111011 001000 010010 110111 111101 100001 100010 111100
K_8 = 111101 111000 101000 111010 110000 010011 101111 111011
K_9 = 111000 001101 101111 101011 111011 011110 011110 000001
K_{10} = 101100 011111 001101 000111 101110 100100 011001 001111
K_{11} = 001000 010101 111111 010011 110111 101101 001110 000110
K_{12} = 011101 010111 000111 110101 100101 000110 011111 101001
K_{13} = 100101 111100 010111 010001 111110 101011 101001 000001

K_{14} = 010111 110100 001110 110111 111100 101110 011100 111010
K_{15} = 101111 111001 000110 001101 001111 010011 111100 001010
K_{16}= 110010 110011 110110 001011 000011 100001 011111 110101

2. **Encryption**

At this stage, it will go through several processes, namely, plaintext representation into binary, permutation based on the Initial Permutation (IP) table and encoding. To present plaintext into binary, ASCII hexa in plaintext will be converted into binary. The following presents the calculation of hexa to binary conversion. The complete calculation results are presented in the table below.

Plaintext	Hexa	Biner
C	43	0100 0011
O	4F	0100 1111
M	4D	0100 1101
P	50	0101 0000
U	55	0101 0101
T	54	0101 0100
E	45	0100 0101
R	52	0101 0010

To get the cihpertext in binary form, the inverse permutation of R_{16}, L_{16}.is performed. The final result is shown in the table below.

Bit	
Input	Output
1	40
2	8
3	48
4	16
5	56
6	24
7	64
8	32
9	39
10	7
11	47
12	15
13	55
14	23
15	63
16	31

Bit	
Input	Output
17	38
18	6
19	46
20	14
21	54
22	22
23	62
24	30
25	37
26	5
27	45
28	13
29	53
30	21
31	61
32	29

Bit	
Input	Output
33	36
34	4
35	44
36	12
37	52
38	20
39	60
40	28
41	35
42	3
43	43
44	11
45	51
46	19
47	59
48	27

Bit	
Input	Output
49	34
50	2
51	42
52	10
53	50
54	18
55	58
56	26
57	33
58	1
59	41
60	9
61	49
62	17
63	57
64	25

$Y = IP^{-1}(R_{16}, L_{16})$

Y = 01010110 11110001 11010101 11001000 01010010 10101111
10000001 00111111

Because the encryption result will be inserted into an image using the LSB method, the ciphertext will still be presented in binary form.

3. **Embedding**

The encryption results that have been obtained in the previous discussion will be inserted into an 8-bit grayscale image with a size of 8×8 pixels. To insert the message bits using the LSB method, each bit of the message will be inserted in the smallest bit of each byte in the image. Thus, 1 byte in the image can only be inserted 1 bit of the message. Based on the results of the previous message encryption, the 64-bit message bits to be inserted are divided into groups of 8-bits in length. The following will present the process of inserting message bits in the image. After getting the embedded bits, the bits will be returned into decimal form. After getting the conversion results in decimal form, it will be represented back into pixel values in the image. The final results can be seen in the table below.

152	159	248	73	194	183	73	214
237	25	69	81	86	212	128	15
239	69	36	3	150	183	180	17
37	181	148	96	103	236	192	234
154	27	230	9	66	142	143	132
189	106	115	24	153	173	85	247
149	188	78	6	156	52	186	71
218	152	93	83	45	17	221	5

4. **Extraction**

From the conversion of pixel values obtained in the previous process, each pixel value becomes one stego bit block with 8-bit length. From each block, the smallest bit will be extracted and then arranged into several blocks with 8-bit length. The results of LSB Extraction from Stego Bit can be seen in the following table.

No	Blok Stego	LSB
1	10011000	0
2	10011111	1
3	11111000	0
4	01001001	1
5	11000010	0
6	10110111	1
7	01001001	1
8	11010110	0
9	11101101	1
10	00011001	1

No	Blok Stego	LSB
33	10011010	0
34	00011011	1
35	11100110	0
36	00001001	1
37	01000010	0
38	10001110	0
39	10001111	1
40	10000100	0
41	10111101	1
42	01101010	0

11	01000101	1	43	01110011	1
12	01010001	1	44	00011000	0
13	01010110	0	45	10011001	1
14	11010100	0	46	10101101	1
15	10000000	0	47	01010101	1
16	00001111	1	48	11110111	1
17	11101111	1	49	10010101	1
18	01000101	1	50	10111100	0
19	00100100	0	51	01001110	0
20	00000011	1	52	00000110	0
21	10010110	0	53	10011100	0
22	10110111	1	54	00110100	0
23	10110100	0	55	10111010	0
24	00010001	1	56	01000111	1
25	00100101	1	57	11011010	0
26	10110101	1	58	10011000	0
27	10010100	0	59	01011101	1
28	01100000	0	60	01010011	1
29	01100111	1	61	00101101	1
30	11101100	0	62	00010001	1
31	11000000	0	63	11011101	1
32	11101010	0	64	00000101	1

The LSB extraction results will be grouped into groups of 8-bit lengths.

$$Cipher\ Bit = 01010110\ 11110001\ 11010101\ 11001000\ 01010010\ 10101111\ 10000001\ 00111111$$

CONCLUSION

The conclusion that can be drawn from the creation of steganography applications using the LSB method for DES encrypted messages in digital images is that the application built is able to insert DES encrypted messages using the LSB method. The number of characters that can be inserted depends on the size of the image dimensions used as a carrier. Where each bit of DES encrypted message will be inserted in each pixel channel.

REFERENCE

Adhar, D. (2019). IMPLEMENTASI ALGORITMA DES (DATA ENCRYPTION STANDARD) PADA ENKRIPSI DAN DESKRIPSI SMS BERBASIS ANDROID. *Jurnal Teknik Informatika Kaputama*, *3*(2), 53–60.

Buulolo, N., & Sindar, A. (2020). Analisis dan Perancangan Keamanan Data Teks Menggunakan Algoritma Kriptografi DES (Data Encryption Standard). *Jurnal Teknologi Informasi*, *15*(3), 61–65.

Ifan Rizqa, & Safitri, A. N. (2022). Kriptostegano Menggunakan Data Encryption Standard dan Least Significant Bit dalam Pengamanan Pesan Gambar. *Jurnal Masyarakat Informatika*, *13*(2), 111–120.

Jum'ah, M. N. Al, & Sarimuddin. (2024). Implementasi Steganografi Metode Least Significant Bit (LSB) untuk Menyembunyikan File Pesan dalam Gambar. *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, *6*(1), 102–108.

Kurniawan, F., Putra, R. R., & Wadisma, C. (2023). Village Activity Management Information System with MobileResponsive User Interface Design and Usability Test. *Jurnal Sains, Teknologi Dan Industri*, *20*(2).

Minarni, & Redha, R. (2020). IMPLEMENTASI LEAST SIGNIFICANT BIT (LSB) DAN ALGORITMA VIGENERE CIPHER PADA AUDIO STEGANOGRAFI. *Jurnal Sains Dan Teknologi*, *20*(2).

Rantelinggi, P. H., & Saputra, E. (2020). ALGORITMA KRIPTOGRAFI TRIPLE DES DAN STEGANOGRAFI LSB SEBAGAI METODE GABUNGAN DALAM KEAMANAN DATA. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIIK)*, *7*(4), 661–666.

Rizal, C., Siregar, S. R., Supiyandi, S., Armasari, S., & Karim, A. (2021). Penerapan Metode Weighted Product (WP) Dalam Keputusan Rekomendasi Pemilihan Manager Penjualan. *Building of Informatics, Technology and Science (BITS)*, *3*(3), 312–316. https://doi.org/10.47065/bits.v3i3.1094

Sidiq, R. F., Rahayu, R. E. G., & Supriatna, A. D. (2023). Implementasi Kriptografi Advanced Encryption Standard dan Least Significant Bit untuk Keamanan Pesan Email dalam Gambar. *Jurnal Algoritma*, *20*(2), 305–315.

Suparman, B., & Sewaka. (2022). Aplikasi Pengamanan Data Menggunakan Kriptografi Dengan Metode Wake dan Algoritma Des Bebasis Java Desktop. *OKTAL : Jurnal Ilmu Komputer Dan Science*, *1*(7), 808–817.

Supiyandi, S., Hermansyah, H., & Sembiring, K. A. P. (2020). Implementasi dan Penggunaan Algoritma Base64 dalam Pengamanan File Video. *Jurnal Media Informatika Budidarma*, *4*(2), 340. https://doi.org/10.30865/mib.v4i2.2042

Thahara, A., & Siregar, I. T. (2021). Implementasi Kriptografi untuk Keamanan Data dan Jaringan menggunakan Algoritma DES. *JURTI*, *5*(1), 31–38.

Tjoanda, M., Saputra, R., & Cornelius, A. (2024). METODE ALGORITMA DES UNTUK KEAMANAN DATA. *Journal of Economic, Business and Engineering (JEBE)*, *5*(2), 304–312.

Wijayanti, D. E., & Romadlon, W. (2022). Keamanan Pesan Menggunakan Kriptografi dan Steganografi Least Significant Bit pada File Citra Digital. *EULER : Jurnal Ilmiah Matematika, Sains Dan Teknologi*, *10*(2), 181–192.

Yanto, M. A., Handayani, L., & Pizaini. (2024). Steganografi Gambar Menggunakan Metode Least Significant Bit Pada Citra Dengan Operasi XOR. *Building of Informatics, Technology and Science (BITS)*, *6*(1), 115–124.