JUFDIKES-Jurnal Ilmu Fisioterapi dan Ilmu Kesehatan Sisthana
Vol.2, No. 1 Januari 2020
e-ISSN: 2828-2469; p-ISSN: 2828-240X, Hal 52-65
DOI: https://doi.org/10.55606/jufdikes.v5i2.380

Addition Of Cryptographic Algorithm For Bitmap Image Security Of Medical Record Information System Electronics (Rme)

Sri Wulandari
Politeknik Indonusa Surakarta

Muhammad Allam Arkani
Politeknik Indonusa Surakarta

Lucky Nurcahya Wibowo
Politeknik Indonusa Surakarta

Karunia Cinta Andini
Politeknik Indonusa Surakarta

Christina Ika Permatasari
Politeknik Indonusa Surakarta

Safitri Ima Yunita
Politeknik Indonusa Surakarta

Alamat: Jl. Palem No. 8 Jati, Cemani, Grogol, Kabupaten Sukoharjo, Jawa Tengah 57552
Author correspondence: sriwulandari@poltekindonusa.ac.id

Abstract *In the daily use of technology, humans cannot be separated from the internet as a need to exchange information. People at this time spend more time in front of computer screens, laptops, even smartphones to find out information, send data to some of their friends. One of the information that is often sought or sent is a file. Apart from that, image files are files that are much searched for and sent, and many also contain important information in them. Image file security is of course very important so that unauthorized parties do not hack or manipulate information from the image. There is a way to secure information so that it does not leak to unauthorized parties, namely by using cryptography and steganography. By combining these two methods, you can maintain the confidentiality and security of a file, especially image files. In this study the algorithm and method used is the AES cryptographic algorithm (Advanced Encryption Standard) 256 and the LSB (Least Significant Bit) steganography method. Data integrity needs to be tested to ensure that the encryption and decryption processes are running properly. Data integrity testing uses the SHA-1 method. Likewise, the image quality after insertion will experience a decrease in quality. To evaluate this, it is necessary to test using the PSNR method. From the results of data integrity testing by comparing the hash value of the decrypted image file with the original image file, there is no difference. So this shows that the encryption and decryption process was successful. While in testing using the PSNR method, the average PSNR value is 44.14086 dB and with an average error value of 2.830403 dB, which means there is a small decrease in quality. From the implementation and testing results, it can be concluded that the AES 256 cryptographic algorithm and the steganography method LSB can be implemented in maintaining the confidentiality and security of secret messages.*

Keywords: *3-5 cryptography, steganography, AES algorithm, LSB method, SHA-1, PSNR.*

Received November 25, 2019; Revised Desember 27, 2019;Accepted Januari 30, 2020

* Sri Wulandari,sriwulandari@poltekindonusa.ac.id

Abstrak. Dalam penggunaan teknologi sehari-hari, manusia tidak lepas dari internet sebagai kebutuhan untuk bertukar informasi. Orang-orang saat ini lebih banyak menghabiskan waktunya di depan layar komputer, laptop, bahkan smartphone untuk mencari tahu informasi, mengirim data ke beberapa temannya. Salah satu informasi yang sering dicari atau dikirim adalah file. Selain itu, file gambar merupakan file yang banyak dicari dan dikirim, dan banyak juga yang berisi informasi penting di dalamnya. Keamanan file citra tentu sangat penting agar pihak yang tidak berkepentingan tidak meretas atau memanipulasi informasi dari citra tersebut. Terdapat cara untuk mengamankan informasi agar tidak bocor ke pihak yang tidak berwenang yaitu dengan menggunakan kriptografi dan steganografi. Dengan menggabungkan kedua cara tersebut, Anda dapat menjaga kerahasiaan dan keamanan suatu file, terutama file gambar. Pada penelitian ini algoritma dan metode yang digunakan adalah algoritma kriptografi AES (Advanced Encryption Standard) 256 dan metode steganografi LSB (Least Significant Bit). Integritas data perlu diuji untuk memastikan bahwa proses enkripsi dan dekripsi berjalan dengan baik. Pengujian integritas data menggunakan metode SHA-1. Begitu juga dengan kualitas gambar setelah disisipkan akan mengalami penurunan kualitas. Untuk mengevaluasi hal tersebut perlu dilakukan pengujian dengan menggunakan metode PSNR. Dari hasil pengujian integritas data dengan membandingkan nilai hash file citra hasil dekripsi dengan file citra asli tidak terdapat perbedaan. Sehingga hal ini menunjukkan bahwa proses enkripsi dan dekripsi berhasil. Sedangkan pada pengujian dengan metode PSNR diperoleh nilai PSNR rata-rata sebesar 44,14086 dB dan dengan nilai error rata-rata sebesar 2,830403 dB yang berarti terjadi penurunan kualitas yang kecil. Dari hasil implementasi dan pengujian dapat disimpulkan bahwa algoritma kriptografi AES 256 dan metode steganografi LSB dapat diimplementasikan dalam menjaga kerahasiaan dan keamanan pesan rahasia.

Kata kunci: kriptografi 3-5, steganografi, algoritma AES, metode LSB, SHA-1, PSNR.

BACKGROUND

Along with the times, nowadays Humans are entering the internet era. Exchange current information takes place not only in everyday life but also in cyberspace. Important information in the form of messages can be sent and accepted via the internet at this time. Now order frequently sent and received via e-mail, social networks such as Facebook, Twitter, and others so many in the form of pictures. Pictures too contains important information in part people to exchange data (Fadlan, Rosmini, and Haryansyah 2021; Pamungkas and Muhammad 2022; Vivi Wahdini, Hartama, and Okta Kirana 2021).

Along with the rapid information needs in humans as is currently the case Of course, security is needed for the message sent or received. such security necessary to prevent wiretapping or image piracy contains important information for its users. Security is necessary to maintain integrity the image to keep it safe.

There are several ways to secure the image message, namely by cryptography and steganography. Cryptography and steganography can implemented in the application for secure images (Cahyo Prabowo and Afrianto 2017).

THEORETICAL STUDY

1. Cryptography

Cryptography comes from two Greek words, viz Crypto which means secret and Grapho which means write. In general, cryptography can be interpreted as the science and art of encoding that aims to maintain the security and confidentiality of data. Cryptography supports the needs of two aspects information security, namely secrecy (protection on confidentiality of information data) and authenticity (protection against counterfeiting and tampering unwanted information). Cryptography is not means only providing information security, but more towards the techniques (Sijabat, Syahputri, and Khairani 2021; Studi et al. 2014)

Cryptographic algorithms can divided into two types, namely symmetrical and asymmetric. Symmetric algorithm (encryption model conventional) is an algorithm that use one key for the encryption process and data description. While the asymmetric algorithm (public key encryption model) using the key which differ in the process of encryption and description message.

2. AES Algorithm

The AES algorithm is a cipher algorithm safe to protect data or information confidential. AES is published by NIST (National Institute of Standards and Technology) at 2001 which was used to replace DES algorithm which is considered ancient and easy to break into. The input and output of the AES algorithm consist of a data sequence of 128 bits. Order data in a group of 128 bits is also known as block of data or plaintext which will later be encrypted into ciphertext. Key length of AES consists of key lengths of 128 bits, 192 bits, and 256 bits. The difference in the length of this key that later affect the number of rounds on the AES algorithm This. The input and output of the AES algorithm consist of data sequence of 128 bits. Order data in one Groups of 128 bits are also known as blocks data or plaintext which will be encrypted later be ciphertext. The key length of AES consists of key lengths of 128 bits, 192 bits, and 256 bits.

The difference in the length of this key that later affect the number of rounds on the AES algorithm This. The number of rounds this algorithm uses There are three kinds as in the table below. Table 1 Number of Rounds on the AES Algorithm (Buulolo and Sindar 2020).

Table 1 Number of Rounds on the AES Algorithm

	Key	Input Block	round
AES-128	128 bits	128 bits	10
AES-192	192 bits	128 bits	12
AES-256	256 bits	128 bits	14

2.2.1 AES Encryption Process

The AES algorithm encryption process consists of 4 bytes transformation type, i.e. SubBytes, ShiftRows, Mixcolumns, and AddRoundKey. At the start of the process encryption, input that has been copied into the state will undergo an AddRoundKey byte transformation. After that, the state will undergo a transformation SubBytes, ShiftRows, MixColumns, and AddRoundKey Nr. This process in the AES algorithm is referred to as round function. The last round is a bit different with previous rounds where in round Finally, the state does not undergo transformation MixColumns (Bhaudhayana and Widiartha 2015).

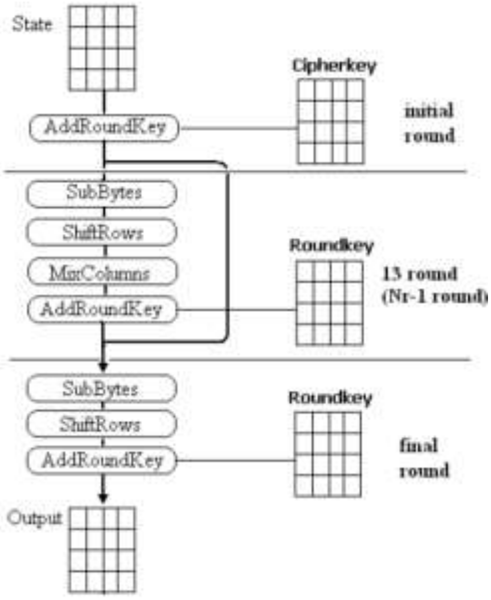


Figure 1 AES Encryption Process

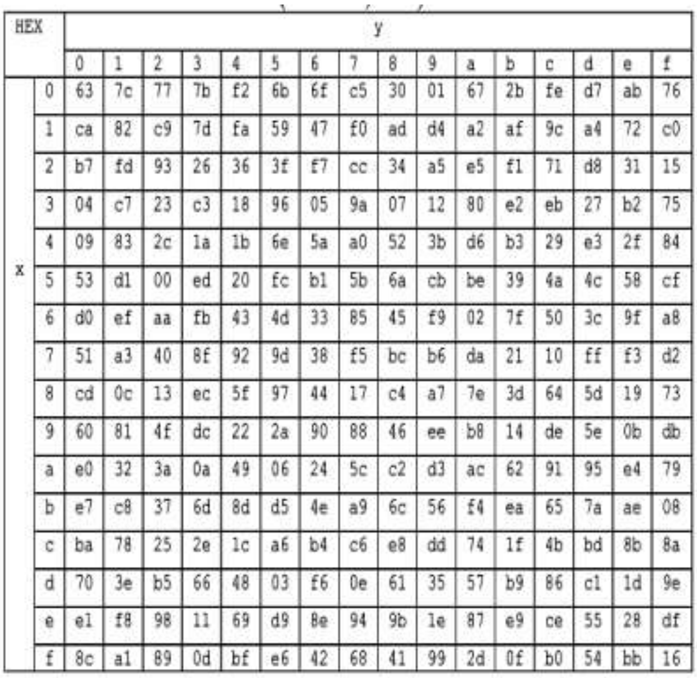
a. AddRoundKey

In the initial round, the transformatio AddRoundKey() is performed on the key main. Meanwhile, in the 10 rounds else, the AddRoundKey process is performed to the round key. Process AddRoundKey is defined as an operation XOR between array state and round key.

The XOR operation is performed on each byte in the array so that generates a new value in the result array with the size of the result array equal to size of initial state array and array key, ie of 4x4. Results for each rows and columns in the result state array obtained from the results of the intermediate XOR operation initial state array with array key for same row and column.

b. SubBytes

The SubBytes() transform maps each bytes from the state array by using S-Box substitution table. Table S-Box can seen in the following.



HEX		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

The way of substitution is as follows: for each byte in the array, say S[r,c] = xy, in which case xy is a digit hexadecimal value of S[r,c], then value substitution, which is expressed by S'[r,c], is an element in the S-Box that is the line x intersects with column y.

c. ShiftRows

ShiftRows() transform performs shift by wrapping (cyclic) at 3 the last row of the state array. Amount the shift depends on the row value (r). Row r = 1 is shifted by 1 byte, row r = 2 shifted by 2 bytes, and row r = 3 shifted as far as 3 bytes. The row r = 0 is not shifted

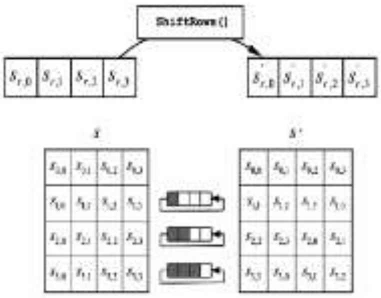


Figure 3 ShiftRows Transformation

d. MixColumns

The MixColumns() transform is performed after the ShiftRows transformation, represents main source of diffusion in the algorithm AES. Diffusion is a principle propagates the effect of one bit of plaintext or key to as many ciphertexts as possible.

The MixColumns() transform multiplies each column of the state array with polynomial a(x) mod (x4 + 1). every column treated as a 4 term polynomial on GF(28). The polynomial a(x) assigned to equation 1 a(x) = {03}x3 + {01} x2 + {01} x + {02} (1)

This transformation is expressed as matrix multiplication as in equation 2

s'(x) = a(x) ⊗ s (x) (2)

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

The result of the matrix multiplication, every bytes in the state array column will be replaced with a new value. Equality math for each byte it is on equation 3

s'0,c = ({02}• s0,c) ⊕ ({03}• s1,c) ⊕ s2,c ⊕ s3,c s'1,c = s0,c ⊕ ({02} • s1,c) ⊕ ({03} • s2,c) ⊕ s3,c s'2,c = s0,c ⊕ s1,c ⊕ ({02}• s2,c)⊕ ({03}• s3,c) s'3,c = ({03} • s0,c) ⊕ s1,c ⊕ s2,c ⊕(*02} • s3,c) (3)

2.2.2 AES Decryption Process

The cipher transformation is reversible and implemented in the opposite direction to produce an easy inverse cipher understood for the AES algorithm. bytes transformation which is used in the inverse cipher is InvShiftRows, InvSubBytes, InvMixColumns, and AddRoundKey. The decryption algorithm can be seen in the following scheme:

a. InvShiftRows

InvShiftRows is a byte transformation opposite of transformation ShiftRows. In the InvShiftRows transformation, shift the bits to the right whereas on ShiftRows it does shift bits to the left. Transformation illustration InvShiftRows is in the following image.

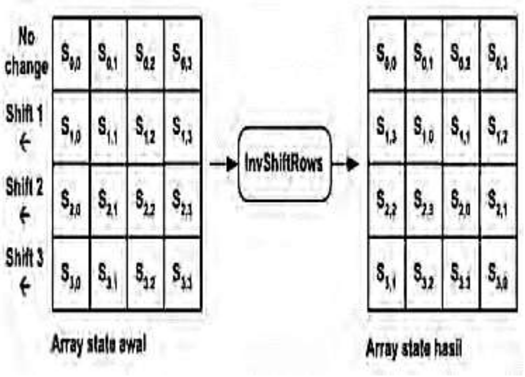
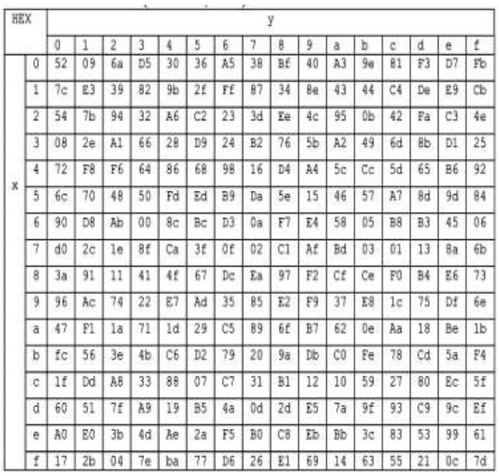


Figure 4 InvShiftRows Transformation

b. InvSubBytes

InvSubBytes is also a transformation bytes as opposed to SubBytes transformation. On InvSubBytes, each element in the state is mapped with using the Inverse S-Box table. Table The inverse S-Box will be shown in the table following.



HEX		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	52	09	6a	D5	30	36	A5	38	Bf	40	A3	9e	81	F3	D7	Fb
	1	7c	E3	39	82	9b	2f	Ff	87	34	8e	43	44	C4	De	E9	Cb
	2	54	7b	94	32	A6	C2	23	3d	Ee	4c	95	0b	42	Fa	C3	4e
	3	08	2e	A1	66	28	D9	24	B2	76	5b	A2	49	6d	8b	D1	25
	4	72	F8	F6	64	86	68	98	16	D4	A4	5c	Cc	5d	65	B6	92
	5	6c	70	48	50	Fd	Ed	B9	Da	5e	15	46	57	A7	8d	9d	84
	6	90	D8	Ab	00	8c	Bc	D3	0a	F7	E4	58	05	B8	B3	45	06
	7	d0	2c	1e	8f	Ca	3f	0f	02	C1	Af	Bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	Dc	Ea	97	F2	Cf	Ce	F0	B4	E6	73
	9	96	Ac	74	22	E7	Ad	35	85	E2	F9	37	E8	1c	75	Df	6e
	a	47	F1	1a	71	1d	29	C5	89	6f	B7	62	0e	Aa	18	Be	1b
	b	fc	56	3e	4b	C6	D2	79	20	9a	Db	C0	Fe	78	Cd	5a	F4
	c	1f	Dd	A8	33	88	07	C7	31	B1	12	10	59	27	80	Ec	5f
	d	60	51	7f	A9	19	B5	4a	0d	2d	E5	7a	9f	93	C9	9c	Ef
	e	A0	E0	3b	4d	Ae	2a	F5	B0	C8	Eb	Bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	D6	26	E1	69	14	63	55	21	0c	7d

Figure 5 Inverse S-Box

2.5 SHA (Secure Hashing Algorythm)-1

SHA-1 is an extension of SHA-0 where SHA-1 fixes the weakness in SHA-0. SHA-1 is the most hash function popular compared to the SHA hash function other. SHA-1 produces 160 bit digest based on the same principle as the algorithm MD4 and MD5 butwith a different design. SHA-1 has a capacity of 264-1 message inputs, with 160 bits hash result and strength evaluation hash 280 Suppose SHA-1 is used to hash a message, M, that has length maximum 264-1 bits. This algorithm uses sequence of 80 times a 32-bit word, using 5 variables that hold 32 bits per variable, and the hash result. (Huda W, 2003)

2.6 PSNR (Peak Signal to Noise Ratio)

The quality of the receiving media after added secret message is not much different with the quality of the previous container media message added. After adding the message secret, the quality of the container image is not far off changed, still looks good. For measuring the quality of the steganographic image is required an objective test. Testing by objective is done by calculating the value PSNR. Peak Signal to Noise Ratio (PSNR) is comparison between the maximum values of the signals which is measured by the magnitude of the noise effect on the signal. PSNR was measured in decibel units. In this study, PSNR used to determine the quality comparison pictures before and after the message is inserted. To determine the PSNR, first must determined MSE (Mean Square Error). MSE basis mathematically can be formulated as follows:

$$MSE = \frac{1}{m\,n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$$

Where :

MSE = Mean Square Error value for steganographic images

m = stego image length (in pixels)

I(i,j) = pixel value of the cover image

n = Stego image width (in pixels)

K(i,j) = pixel value in the stego image

After obtaining the MSE value, the value PSNR can be calculated from the square of the maximum value shared by MSE. Mathematically, the PSNR value formulated as follows:

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right)$$

Where:

MSE = MSE value, MAXi = maximum value of the pixel image used. The lower the MSE Value then the better, and the greater the value PSNR, the better the quality of the steganographic image (Putri and Hidayati 2021).

RESEARCH METHODS

To create an application that can secure image files. Of course there is a need design in the stages of the process. These stages will be represented in flow chart or commonly called flowchart.

The flowchart explains the process in stages encryption and insertion. Likewise for processes extraction and decryption.

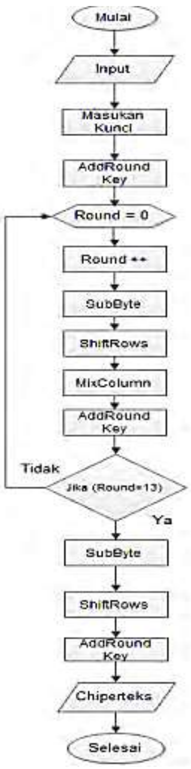


Figure 6 AES Algorithm Encryption Flowchart 256

The flow chart in the image above describes the encryption process using AES cryptographic algorithm. In the first step in the process of encrypting a data must input data to be encrypted. On In this study, the data is in the form of images. Furthermore is to do key input process. Because used in this study are AES 256 algorithm, the key is 256 bits long.

The next step is to carry out the process AddRoundKey. This AddRoundKey process XOR operation is performed between plaintext and key. This encryption process consists of 4 types bytes transformation, i.e. SubBytes, ShiftRows, Mixcolumns, and AddRoundKey. At the start of the process encryption or Round = 0, the input will experience the AddRoundKey byte transformation. After that, states will undergo a SubBytes transformation, ShiftRows, MixColumns, and AddRoundKey repeatedly as many as Round = 13, If Round = 14 then it will do three processes transform SubBytes, ShiftRows, and AddRoundKey and will generate ciphertext. The encryption process is complete.

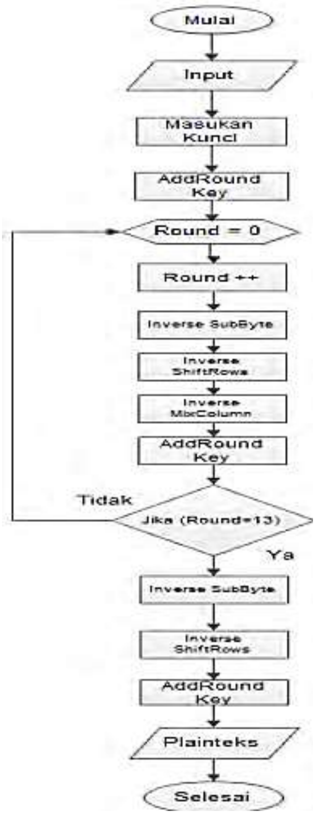


Figure 7 AES Algorithm Description Flowchart 256

The flow chart in the image above describes the decryption process using AES cryptographic algorithm. In the first step in the process of decrypting a ciphertext have to do ciphertext input that will decrypted. From ciphertext results and from the process random key will be combined key in one block that is by doing the process AddRoundKey ie ciphertext added on state with XOR operation with key. Process This decryption consists of 4 types of transformation bytes, i.e. Inverse SubByte, Inverse Shift Rows, Inverse Mix column, and Add Round Key. In the beginning decryption process or Round = 0, input has been state will undergo a byte transformation Add Round Key. After that, the state will experience transform Inverse Sub Bytes, Inverse Shift Rows, Inverse Mix Columns, and Add Round Key respectively repeatedly as many as Round = 13, If Round = 14 then it will carry out three transformation processes Inverse Sub Bytes, Inverse Shift Rows, and Add Round Key and will generate plaintext or original data. The decryption process is complete.

RESULTS AND DISCUSSION

To be able to make a security application image file with AES cryptographic algorithm and LSB steganography method, then its implementation made in the form of a program using java language.

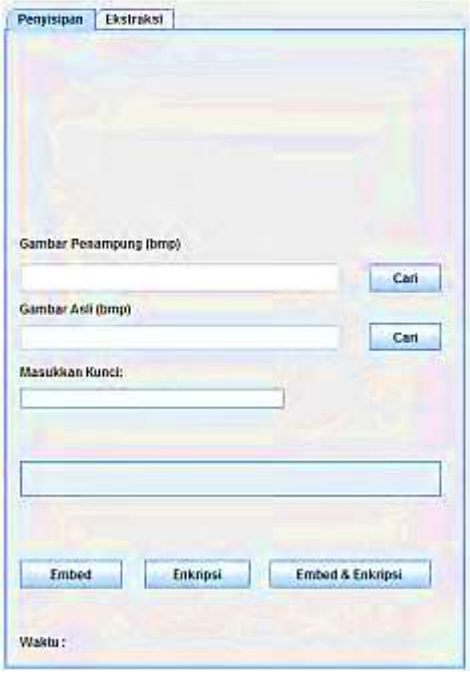


Figure 10 Display of the Encryption Program and Insertion

The image above is a display program for the encryption and insertion process. For perform the first encryption and insertion process take picture. The original image as the message will be encrypted and the container image as media to hold a picture message. Both pictures bmp format. After that enter the key as cipherkey. Then press the embed button and encryption to get program results in the form of image that has been inserted a secret message.

The image above is a display program for the encryption and insertion process. For perform the first encryption and insertion process take picture. The original image as the message will be encrypted and the container image as media to hold a picture message. Both pictures bmp format. After that enter the key as cipherkey. Then press the embed button and encryption to get program results in the form of image that has been inserteda secret message.

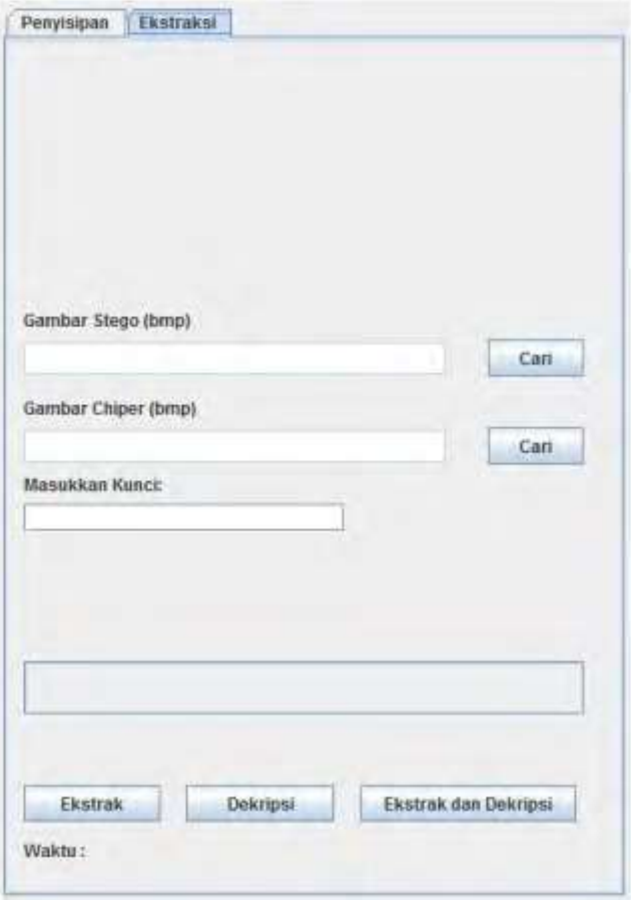


Figure 11 Display of the Decryption Program and Extraction

To determine the integrity of the data on the results decryption of the original file then needs to be done hash value testing using the hashing method SHA-1. This method aims to find out does the decrypted image have a hash value the same as the original picture. If there is Hash value similarity between the decryption and image decryption original image it is certain that the process encryption and decryption have been successful. Here's a table the value of the test results using the SHA-1 method.

CONCLUSSION

Based on the results and discussion that has been obtained, conclusions can be drawn that AES 256 cryptographic algorithm and methods LSB steganography is good to implement in securing the image file confidentiality is strictly guarded. From the results of the process encryption obtained cipher image that can not understandable. With increasing size file the bigger the execution time of the process encryption. As seen in the experimental table which gets the result of its execution time increases as the file size increases picture.

REFERENCE

Bhaudhayana, Gede Wisnu, and I. Made Widiartha. 2015. “Implementasi Algoritma Kriptografi AES 256 Dan Metode Steganografi LSB Pada Gambar Bitmap.” *Jurnal Ilmu Komputer* 8(2):9–25.

Buulolo, Novelius, and Anita Sindar. 2020. “Analisis Dan Perancangan Keamanan Data Teks Menggunakan Algoritma Kriptografi DES (Data Encryption Standard).” *Respati* 15(3):61. doi: 10.35842/jtir.v15i3.373.

Cahyo Prabowo, Egi, and Irawan Afrianto. 2017. “Penerapan Digital Signature Dan Kriptografi PadaOtentikasi Sertifikat Tanah Digital - Teknik Informatika Universitas Komputer Indonesia.” *Jurnal Ilmiah Komputer Dan Informatika (KOMPUTA)* 6(2).

Fadlan, Muhammad, Rosmini Rosmini, and Haryansyah Haryansyah. 2021. “Perpaduan Algoritma Kriptografi Atbash Dan Autokey Cipher Dalam Mengamankan Data.” *Jurnal Media Informatika Budidarma* 5(3):806. doi: 10.30865/mib.v5i3.3019.

Pamungkas, Prima Giri, and Alva Hendi Muhammad. 2022. “Modifikasi Algoritma Kriptografi Caesar Chiper Pada Deretan Simbol Dan Huruf Di Smarphone Dan Laptop.” *Journal of Information Technology* 2(1):1–5. doi: 10.46229/jifotech.v2i1.234.

Putri, Clarissa Amalia, and Meira Hidayati. 2021. “Analisis Kebutuhan Sumber Daya Manusia Petugas Rekam Medis Dengan Menggunakan Metode Analisis Beban Kerja Kesehatan(Abk-Kes).” *Jurnal Manajemen Kesehatan Yayasan RS.Dr. Soetomo* 7(2):257. doi: 10.29241/jmk.v7i2.637.

Sijabat, Lambok Hasudungan, Nenna Irsa Syahputri, and Mufida Khairani. 2021. “Kriptografi Dan Steganografi Penyembunyian Pesan Pada Media Audio Menggunakan Algoritma AES.” *ALGORITMA: Jurnal Ilmu Komputer Dan Informatika* 6341(April):1.

Studi, Program, Teknik Informatika, Fakultas Ilmu, Komputer Universitas, and Dehasen Bengkulu. 2014. “Aplikasi Kriptografi Pesan Menggunakan Algoritma.” 10(2):120–28.

Vivi Wahdini, Sri, Dedy Hartama, and Ika Okta Kirana. 2021. “Pengamanan Data Pelanggan Dan Penjualan Menggunakan Implementasi Algoritma Kriptografi.” *Journal of Informatics Management and Information Technology* 1(3):101–7.