Journal of Applied Intelligent System (e-ISSN : 2502-9401 | p-ISSN : 2503-0493)
Vol. 8 No. 1, February 2023, pp. 69 – 80
DOI:

# The Implementation of Improved Advanced Encryption Standard and Least Significant Bit for Securing Messages in Images

**Imam Prayogo Pujiono***[1], **Eko Hari Rachmawanto**[2]
*UIN K.H. Abdurrahman Wahid, Pekalongan*[1], Universitas Dian Nuswantoro, Semarang*[2]
*E-mail : imam.prayogopujiono@uingusdur.ac.id*[1], eko.hari@dsn.dinus.ac.id*[2]
*Corresponding author*

**Dicky Anggriawan Nugroho**[3]
*UIN K.H. Abdurrahman Wahid, Pekalongan*
*E-mail : dicky.anggriawannugroho@uingusdur.ac.id*

**Abstract -** Security and confidentiality are essential aspects needed in the process of exchanging messages. Maintaining the message's security and confidentiality can be done by encrypting the message with cryptographic techniques to protect the message. However, the encrypted message (ciphertext) usually raises suspicion from eavesdroppers so that they try to unravel the contents of the message or damage the contents of the message or prevent the message from reaching the intended recipient. To avoid eavesdroppers' suspicion of encrypted messages, after encryption, the ciphertext can then be inserted into an image using steganography techniques so that eavesdroppers do not know whether there is a secret message in an image. In this research, the authors use the Improved AES-128 (Advanced Encryption Standard) Algorithm to encrypt messages and LSB (Least Significant Bit) Algorithm to insert ciphertext into an image. Where the AES Algorithm Improvement is made by adding a sending and receiving applications ID to modify the Key Schedule process, with modifications to the Key Schedule process, messages can only be read on the original recipient's cellphone by entering the correct key. The results of this study show that eavesdroppers do not easily know the existence of the ciphertext, besides that even if the eavesdroppers get the ciphertext and know the encryption key, the message remains unreadable on their cellphone because the ID of the application sending or receiving the message has changed.

**Keywords –** Cryptography, Steganography, Improved AES, Key Schedule, LSB

## 1. INTRODUCTION

The development of various communication media supported by the increasingly widespread internet network has made it easier for people to exchange information [1]. Information security through the internet network needs attention [2]. Poor security can result in information being accessed by unauthorized persons so that the owner of the information can suffer losses [3]. In the process of exchanging information via the internet, two important aspects are needed, namely, aspects of security and confidentiality of information [4][5]. This is because many cyber crimes are developing today with various interruption, wiretapping, modification, and fabrication techniques. Without security guarantees, unauthorized people can easily obtain information sent via the internet [6][7].

Various security techniques have been developed to protect and maintain the confidentiality of messages to avoid eavesdropping, for example, by encrypting messages using cryptographic techniques. One of the strongest cryptographic algorithms is the AES (Advanced Encryption Standard) algorithm. This is proven by the establishment of the AES algorithm by NIST (National Institute of Standards and Technology) since November 2001 as the standard security algorithm in the United States, replacing Data Encryption Standard (DES) Algorithm [8][9][10].

However, even though AES is a strong security algorithm, there are still some weaknesses. For example, if an eavesdropper gets the ciphertext and knows the message key, the meaning of the message can be revealed, besides that, the existence of ciphertext is easily suspected by eavesdroppers as messages containing confidential information. From these suspicions, eavesdroppers usually try to uncover the contents of the message, damage the contents of the message or block the message from reaching the original recipient.

To avoid the suspicion of eavesdroppers, after encryption is carried out, messages can be inserted into an image using steganography techniques so that through unaided eye, eavesdroppers will not know that there is a secret message in an image [11].

One of the steganography techniques is the Least Significant Bit (LSB) algorithm, by hiding the message at the lowest bit location in the digital image. The LSB method converts the message into binary bits and is hidden in a digital image. The implementation of LSB algorithm without being equipped with a security/encryption system can be dismantled easily through the technique of solving frequency analysis by reading the lowest bit [12][13].

Therefore, it is necessary to Improve AES to encrypt messages and LSB to insert messages into an image. These two algorithms will later be implemented in Android-based applications. Improvement of the AES algorithm is made by adding the sending and receiving application ID to the Key Schedule process so that even though it only uses one message security algorithm (without using a public key algorithm), indirectly, the message will be secured with two keys, namely security with a public key originating from the sender and recipient application ID and a private key originating from the message key entered. With these two keys, the meaning of the message can only be read on the original recipient's cellphone by entering the correct key.

## 2. RESEARCH METHOD

The implementation of the AES algorithm for securing messages and the LSB algorithm for embedding messages in images has been carried out by several researchers before, including:

The first research was conducted by Adit Pabbi et al. in 2021 with the title "Implementation of Least Significant Bit Image Steganography with Advanced Encryption Standard" this research uses the AES algorithm to encrypt messages and the LSB algorithm to insert ciphertext into an image [11]. The second study was conducted by Subhash Panwar et al. in 2020 with the title "Digital Image Steganography Using Modified LSB and AES Cryptography" this study used the AES algorithm for message encryption and modified the LSB algorithm to insert ciphertext into an image. Modifications were made by inverting the LSB of blue pixels with the highest pattern [8]. The third research was conducted by Masumeh Damrudi and Kamal in 2019 with the title "Image Steganography using LSB and encrypted message with AES, RSA, DES, 3DES, and Blowfish" this study used several encryption algorithms separately and used the LSB algorithm to embed ciphertext into an image. The results of this study show that the five encryption algorithms that have been tried are feasible to use and combined with the LSB algorithm [14]. The fourth research was conducted by Fahmi Anwar et al. in 2019 with the

title "StegoCrypt Scheme using LSB-AES Base64" this study used the AES and Base64 algorithms for message encryption and the LSB algorithm for inserting ciphertext into an image [15].

From the research above, the authors try to develop it by implementing Improve AES-128 for message encryption and LSB for inserting ciphertext into images. Improvements are made so only that the rightful recipient can know the message.

### *2.1. AES Algorithm*

The Advance Encryption Standard (AES) was promulgated by the National Institute of Standards and Technology (NIST) in November 2001. NIST announced that AES replaced the older and less secure Data Encryption Standard (DES) encryption algorithm [16]. AES has become a block cipher that can process a block of 128 bits of input in the form of plaintext (original message) at a time. AES also supports 128, 192, and 256-bit key settings [16].

AES has a block length of 128 bits with allowable key lengths of 128, 192, and 256 bits. The encryption process in the AES algorithm consists of 4 types of transformation bytes. Namely SubByte, ShiftRows, Mixcolumns, and AddRoundKey. At the beginning of the encryption process, the input that has been copied into the state will undergo an AddRoundKey transformation. Then the state will undergo SubByte, ShiftRows, Mixcolumns, and AddRoundKey transformations repeatedly as many as Nr [17]. This process in the AES algorithm is called a round function. The last round is slightly different from the previous round, where in this round, the state does not undergo a MixColumns transformation. An illustration of the AES encryption process can be described in Figure 1.

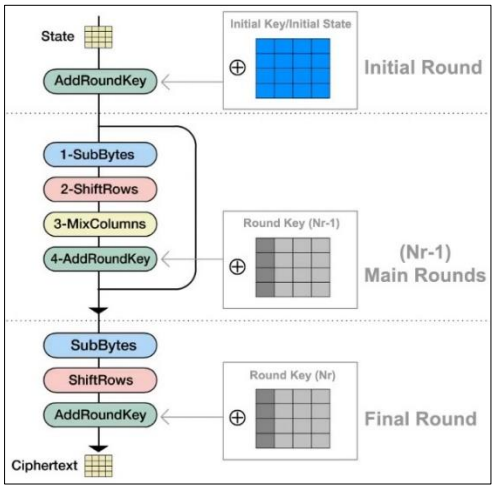



Figure 1. AES encryption process

### *2.2. LSB Algorithm*

The LSB (Least Significant Bit) algorithm is the simplest steganography method, easy to implement, and the most widely used technique in steganography [18]. This method uses a digital image as a cover image. In the arrangement of bits in a byte (1 byte = 8 bits), there are the most significant bits (Most Significant Bits or MSB) and least significant bits (Least Significant Bits or LSB). For example, in byte **1**1010010, bit number **1** (first, underlined) is the MSB bit, and bit number **0** (last, underlined) is the LSB bit. The bit that is suitable to be replaced is the LSB bit because this change only changes the byte value to one higher or one lower than the previous value so that it cannot be distinguished by the human eye.

For example, the image pixel segments before the addition of bits are:

00101011 10100010 11101010 10101011 00101010

10010110 11001101 11101001 10001000 10101011

Secret message (which has been converted to a binary system), for example "1110010111", then each byte of the message replaces the LSB position of the image pixel segments to be as follows (underlined):

00101011 10100011 11101011 10101010 00101010
10010111 11001100 11101001 10001001 10101011

*2.3. Proposed Method*

In this study, the authors propose the Implementation of the Improved AES Algorithm and the LSB Algorithm for Securing Messages in Images. The improved AES algorithm is used to encrypt messages, and then the encryption results (ciphertext) will be inserted into an image using the LSB algorithm. Improvements are made by adding the sending and receiving application IDs to the AES-128 algorithm Key Schedule process so that even though it only uses one message security algorithm and without using a public key algorithm, indirectly the message will be secured with two keys, namely security with a public key originating from the sender and recipient application ID and a private key derived from the message key entered. Meanwhile, the LSB algorithm is used to insert ciphertext into an image so that eavesdroppers are not easily aware of the existence of the ciphertext because it is in an image. The following is the encryption and encoding process that occurs with the proposed method:

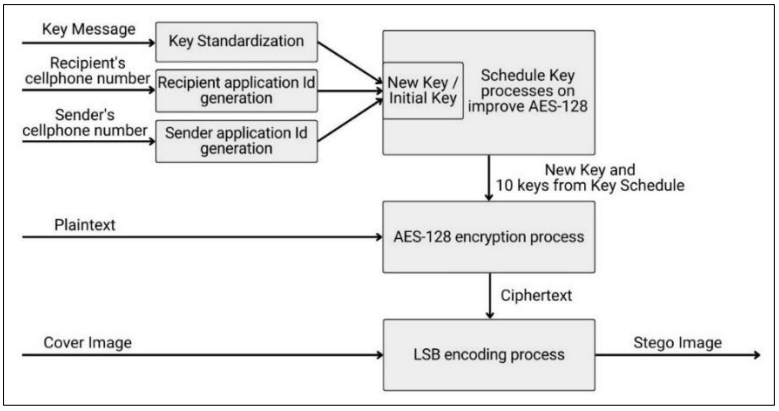



Figure 2. Encryption and encoding process with the proposed method

From Figure 2, it can be seen that the message is not only secured using the message key but also using the sender and recipient application ID for additional security so that even if an eavesdropper gets the ciphertext and knows the message security key, the message remains unreadable on the eavesdropper's cellphone because the sender's application ID or the recipient of the message has changed. Besides that, before the eavesdropper tries to decrypt the encrypted message, the eavesdropper will also have difficulty knowing the existence of the ciphertext because it is in an image. In the picture above, the improvement of the AES algorithm occurs in the Key Schedule process, where the sender and recipient application IDs that are inputted will automatically be combined with the input keys to produce a new key, then the new key will be used as a key in the encryption process with the AES-128 algorithm.

The process of decoding and decryption messages that occur on the recipient's cellphone also has a similar flow as the encryption and encoding process on the sender's cellphone. The recipient only enters the stego image and the correct security key.

## 3. RESULTS AND DISCUSSION

### *3.1. Improved AES Algorithm*

In this study, improvements were made to the Key Schedule process by adding the sending and receiving application IDs (see Figure 2). so that even though it only uses one message security algorithm and without using a public key algorithm, indirectly the message will be secured with two keys, namely security with a public key originating from the sending and receiving application IDs and a private key originating from the message key entered.

#### *3.1.1. The process of generating sender and recipient application IDs.*

In this study, the sender and recipient application IDs are used to modify the Key Schedule process. The application IDs are generated based on the sender and recipient numbers. The following is the process for generating sender and recipient application IDs.
The process of generating the sender application ID:
Input = “085123456789”, Output = “9876543987654398”.

| Sender's Number | 0 | 8 | 5 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sender's Application Id | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 9 | 8 |
| Order of Sender's Application ID | 1st | 2nd | 3rd | 4th | 5th | 6th | 7th | 8th | 9th | 10th | 11th | 12th | 13th | 14th | 15th | 16th |

Figure 3. Sender’s application ID generation process

The process of generating the sender's Application ID is done by taking the seven-digit sender's number starting from the back and then entering it into the sender's Application ID repeatedly until the sender's Application ID is 16 digits long. The seven numbers from the back were chosen because the two front numbers are prone to change. For example, "0" becomes "+62" and the 3rd to 5th numbers are area codes or operator codes which are less unique, besides that seven numbers were also chosen to anticipate short telephone numbers so that no errors occur because only seven numbers are needed, then the application ID is chosen with a length of 16 digits because the state in the Key Schedule has a length of 16 bytes.
Input = “085123456789”, Output = “3987654398765439”.

| Recipient's Number | 0 | 8 | 5 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Recipient's application Id | 3 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 9 |
| Order of Recipient's Application ID | 1st | 2nd | 3rd | 4th | 5th | 6th | 7th | 8th | 9th | 10th | 11th | 12th | 13th | 14th | 15th | 16th |

Figure 4. Recipient’s application ID generation process

The process of making the recipient's application ID is slightly different from generating the sender's Application ID. In making the recipient's Application ID before taking the seven-digit number from the back. First, take the recipient's number in the 7th digit from the back. After that, take the seven-digit recipient's number from the back and enter it into the Recipient’s application ID repeatedly until the recipient's Application ID length becomes 16 digits. The process of generating a slightly different recipient application ID is intended so that the ciphertext can only be described on the original recipient's cellphone and cannot be read if the message is sent back to the sender's cellphone.

*3.1.2. Key Schedule processes without improvement on AES-128*

The key schedule process begins with the generation of the initial state (initial key) derived from the key input by the user. For example, the user inputs "imamprayogo" as a key, because the contents of each state are 16 bytes (AES-128 has a key length of 16 bytes), so if in the state when forming the key there is an empty space, the empty space can be filled with any character or filled with a word that is input as a key recursively starting from the first letter (In this study, this process is called Key Standardization), more details can be seen in figure 5:

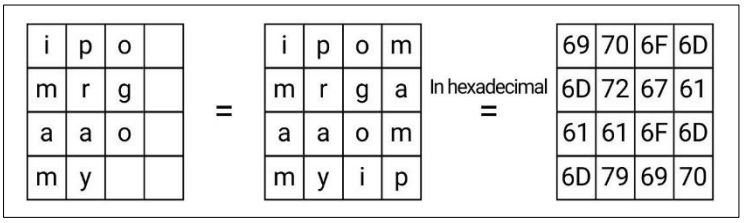



Figure 5. Initial state (initial key) generation on the AES-128 algorithm

So, the contents of $KA_1$ - $KA_{16}$ (Initial State / Initial Key / New Key) are:

| | | | |
|---|---|---|---|
| 69 | 70 | 6F | 6D |
| 6D | 72 | 67 | 61 |
| 61 | 61 | 6F | 6D |
| 6D | 79 | 69 | 70 |

While the process of generating K1 - K10 (First Key to Tenth Key) is as follows:

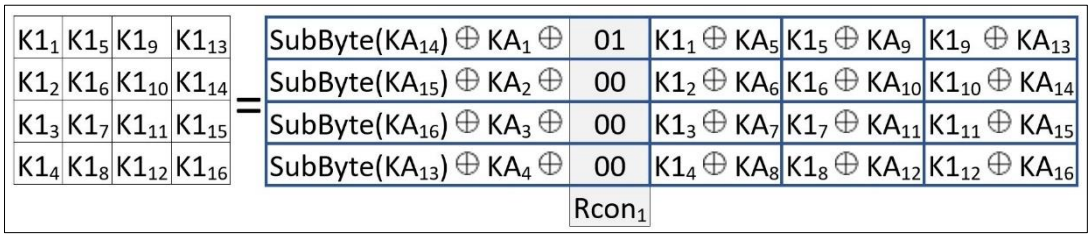



Figure 6. The process of generating K1 (First Key)

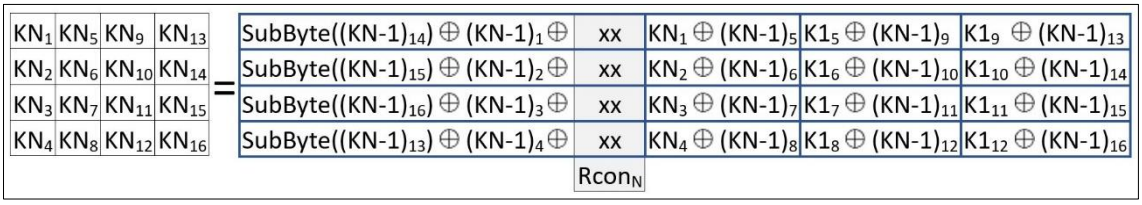



Figure 7. The process of generating K2 - K10

From the key generation process above, ten keys will be generated that will be used in the Addroundkey process from rounds 1-10, while the user input key (KA / Initial Key) is only used in the initial round process, see figure 1.

*3.1.3. Key Schedule processes on improve AES-128.*

The difference between the Key Schedule in the Improve AES algorithm and the Key Schedule in the AES algorithm starts with generating the initial state (initial key). For example, the user inputs "imamprayogo" as a key, because the contents of each state are 16 bytes, so if there is an empty space in the state, the empty space is filled with words which are entered as a key recursively starting from the first letter (In this study, this process is called Key

Standardization). After the state is fully loaded, the contents of each state will be XOR (Exclusive Or) with the sender and recipient application IDs that were previously generated and entered into the state. The following is the process for generating the initial state using the results of key standardization, the sender's application ID, and the receiver's application ID.

Input:

- Key Message = "imamprayogo"
- Sender Application ID = "9876543987654398"
- Recipient Application ID = "3987654398765439"

First, do the process of standardizing the key length:

| | | | |
|---|---|---|---|
| i | p | o | |
| m | r | g | |
| a | a | o | |
| m | y | | |

*Message Key

=

| | | | |
|---|---|---|---|
| i | p | o | m |
| m | r | g | a |
| a | a | o | m |
| m | y | i | p |

*Key Standardization Results

Then change the standardized key, the sender's application ID, and the receiver's application ID into hexadecimal.

| | | | |
|---|---|---|---|
| i | p | o | m |
| m | r | g | a |
| a | a | o | m |
| m | y | i | p |

*Key Standardization Results

=

| | | | |
|---|---|---|---|
| 69 | 70 | 6F | 6D |
| 6D | 72 | 67 | 61 |
| 61 | 61 | 6F | 6D |
| 6D | 79 | 69 | 70 |

*Key Standardization Results in Hex

| | | | |
|---|---|---|---|
| 9 | 5 | 8 | 4 |
| 8 | 4 | 7 | 3 |
| 7 | 3 | 6 | 9 |
| 6 | 9 | 5 | 8 |

*Sender application ID

=

| | | | |
|---|---|---|---|
| 39 | 35 | 38 | 34 |
| 38 | 34 | 37 | 33 |
| 37 | 33 | 36 | 39 |
| 36 | 39 | 35 | 38 |

*Sender application ID in Hex

| | | | |
|---|---|---|---|
| 3 | 6 | 9 | 5 |
| 9 | 5 | 8 | 4 |
| 8 | 4 | 7 | 3 |
| 7 | 3 | 6 | 9 |

*Recipient application ID

=

| | | | |
|---|---|---|---|
| 33 | 36 | 39 | 35 |
| 39 | 35 | 38 | 34 |
| 38 | 34 | 37 | 33 |
| 37 | 33 | 36 | 39 |

*Recipient application ID in Hex

Next, perform the XOR operation between the standardized key results, the sender's application ID, and the receiver's application ID.

| | | | |
|---|---|---|---|
| 69 | 70 | 6F | 6D |
| 6D | 72 | 67 | 61 |
| 61 | 61 | 6F | 6D |
| 6D | 79 | 69 | 70 |

Standardized Message Key

⊕

| | | | |
|---|---|---|---|
| 39 | 35 | 38 | 34 |
| 38 | 34 | 37 | 33 |
| 37 | 33 | 36 | 39 |
| 36 | 39 | 35 | 38 |

Sender application Id

⊕

| | | | |
|---|---|---|---|
| 33 | 36 | 39 | 35 |
| 39 | 35 | 38 | 34 |
| 38 | 34 | 37 | 33 |
| 37 | 33 | 36 | 39 |

Recipient application Id

=

| | | | |
|---|---|---|---|
| $KA_1$ | $KA_1$ | $KA_1$ | $KA_1$ |
| $KA_1$ | $KA_1$ | $KA_1$ | $KA_1$ |
| $KA_1$ | $KA_1$ | $KA_1$ | $KA_1$ |
| $KA_1$ | $KA_1$ | $KA_1$ | $KA_1$ |

Initial State

So the contents of $KA_1$ - $KA_{16}$ (Initial Key / Initial State / New Key) are:

| | | | |
|---|---|---|---|
| 63 | 73 | 6E | 6C |
| 6C | 73 | 68 | 66 |
| 6E | 66 | 6E | 67 |
| 6C | 73 | 6A | 71 |

While the process for generating K1-K10 (Key One to Key Ten) is as follows:

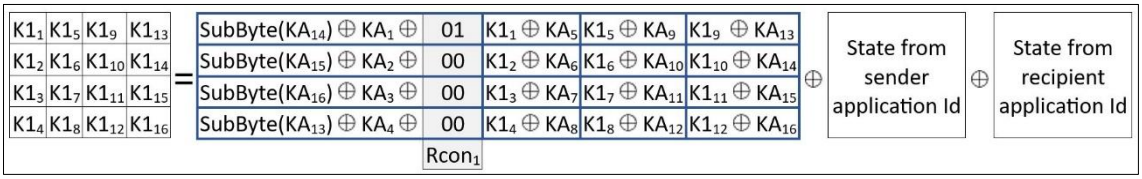



Figure 8. The process of generating K1 (First Key) Improve AES-128

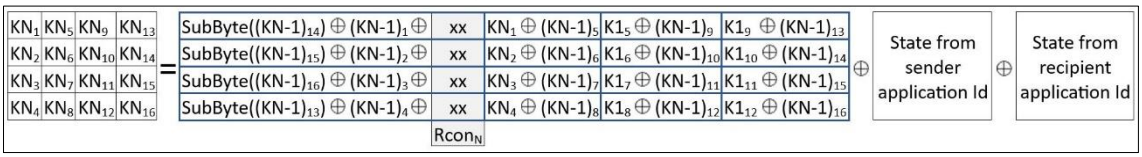



Figure 9. The process of generating K2 - K10 Improve AES-128

From the key generation process above, ten keys will be generated that will be used in the Addroundkey process from rounds 1-10, while the Initial Key (New Key/Initial State) is used in the initial round process. More details can be seen in Figure 1.

With the improvement in the Key Schedule process, it can be seen that the message is not only secured using the message key but also matches the sender and recipient's application IDs for additional security so that even if an eavesdropper gets the ciphertext and knows the message key, the message remains unreadable on the eavesdropper's cellphone because the sender application ID or recipient application ID has changed.

### *3.1.4. Encryption-Description Process AES-128*

There are four stages in the message encryption process with the AES Algorithm: Subbyte, Shiftrow, Mixcolumns, and Addroundkey. The encryption process is carried out in each state with a flow as illustrated in Figure 1, while the message description process has a similar flow as the encryption process but in the opposite order. In this study, improvements were not made to the encryption/description process of the AES-128 Algorithm.

## *3.2. LSB Algorithm*

The LSB algorithm is used to embed messages that have been encrypted with the Improved AES-128 Algorithm into an image. Insertion is done by changing the ciphertext in the form of binary bits, then each binary bit of the ciphertext will be inserted at the lowest bit in the image. In this study, improvement was not made to the LSB Algorithm.

## *3.3 System design*

System design is an overview and sketching of several separate elements that are combined into a unified whole. The application that is built generally has two designs, namely the design for generating new keys and the design for encoding-decoding messages.

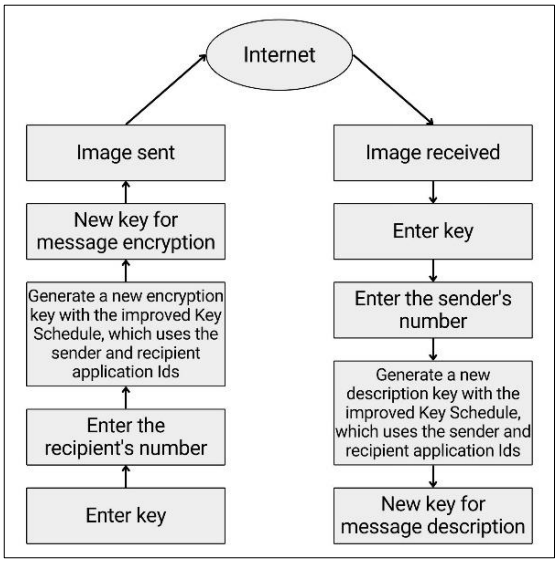



Figure 10. The design of the new key

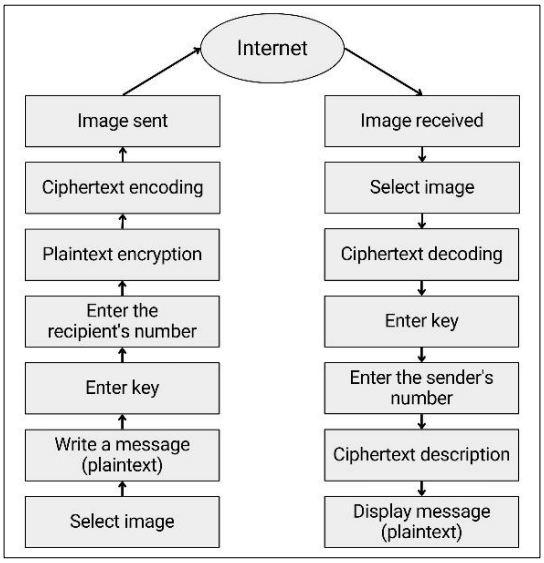



Figure 11. Message encoding-decoding design

Figure 10. is the design of the new key-generating process. When the sender sends an image containing ciphertext (Stego Image), a new key is generated by modifying the AES algorithm's Schedule Key process, the newly generated key will later be used in the message encryption process. Then, after the Stego Image is received on the recipient's cellphone, a new key for the message description is also generated by modifying the Schedule key process as it happens when the sender sends an image.

Figure 11. is a message encoding-decoding design. Initially, the sender chooses the image into which the cyphertext (Cover Image) will be inserted. Then, the sender writes the message, the message key, and the recipient's number. After that, the application will encrypt the message and enter the ciphertext into the Cover Image (Ciphertext encoding). Then, the sender can send an image containing the ciphertext (Stego Image) to the recipient via the internet. After the recipient gets the Stego Image, the recipient can do the ciphertext decoding and enter the correct message key. Finally, the application will perform a ciphertext description and display the original message (plaintext).

### *3.3.1 Application Interface*


Figure 12. Installation interface


Figure 13. Message encoding menu interface


Figure 14. Message decoding menu interface

In Figure 12, the entries in the first row are the entries for the user's cellphone number, then below are the fields for entering the registration code obtained via SMS sent by the application. This registration code will later become the user's application ID.

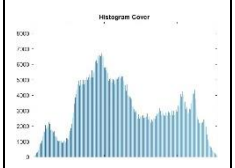
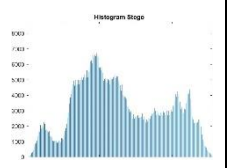
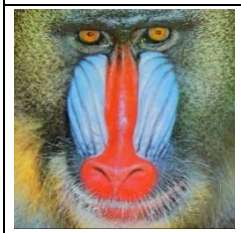
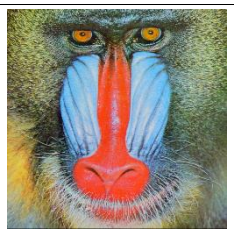
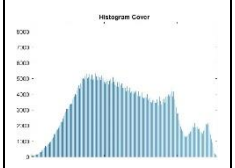
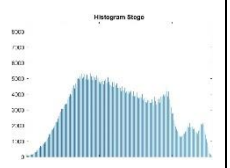
*3.4 Test result*

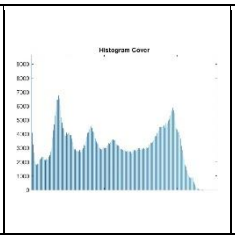
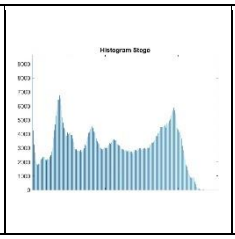
In this test, it is illustrated that Alfa uses a cellphone with the number "085642743334", Beta uses a cellphone with the number "085601144816" and Caca uses a cellphone with the number "089665577638".

Table 1. Black Box Testing

| Test Scenario | Test Case | Ciphertext in Stego Image | Decoding results and descriptions | Information | Conclusion |
|---|---|---|---|---|---|
| Alfa sends the stego image to Beta. | Plaintext (Message): Udinus<br>Sender: Alpha<br>Receiver: Beta<br>Eavesdropper: -<br>Key: Polke | CFA24B4FDE028AA2908B48F0013BDE64 | Udinus | The stego image sent by Alfa to Beta is then decoded and described in Beta's cellphone with the key "Polke" | As Expected. |
| Alfa sends the stego image to Beta. | Plaintext: Udinus<br>Sender: Alfa<br>Receiver: Beta<br>Eavesdropper: -<br>Key: Polke | CFA24B4FDE028AA2908B48F0013BDE64 | ÷¬3úül*í"ÙƒÒo(éË | The stego image sent by Alfa to Beta is then decoded and described in Beta's cellphone with the key "polke" | As Expected. |
| Alfa sends the stego image to Beta, which Caca intercepted. | Plaintext: Udinus<br>Sender: Alfa<br>Receiver: Beta<br>Eavesdropper: Caca<br>Key: Polke | CFA24B4FDE028AA2908B48F0013BDE64 | ¾Y+òoî^ÀĐ»pD$ß | Caca intercepted messages from Alfa to Beta, then decoded and described on Caca's cellphone with the key "Polke" | As Expected. |
| Alpha sends the stego image to Beta, and the stego image is returned to alpha without changes. | Plaintext: Udinus<br>Sender: Alfa<br>Receiver: Beta<br>Eavesdropper: -<br>Key: Polke | CFA24B4FDE028AA2908B48F0013BDE64 | ¿´°b¿&<br>bÆ¶µÑzýÉ | The message sent by Alfa to Beta is returned to Alfa's cellphone and then decoded and described by Alfa with the key "Polke" | As Expected. |

Table 2. Comparison Test of Cover Image and Stego Image

| Cover Image | Stego Image | Cover Image Histograms | Stego Image Histograms | MSE Value | PSNR Value |
|---|---|---|---|---|---|
|  |  |  |  | 0.5079 | 51.0729 dB |
|  |  |  |  | 0.5008 | 51.1342 dB |

| | | | | | |
|---|---|---|---|---|---|
|  |  |  |  | 0.4801 | 51.3174 dB |

## 4. CONCLUSION

Based on the research that has been done, it can be concluded that:

1. The results of the implementation of the improved AES-128 algorithm show that the message can only be read by the rightful recipient even though the eavesdropper gets the ciphertext and knows the message security key because the message is secured with the message key, the sender's application ID and the recipient's application ID.
2. The results of the implementation of the LSB algorithm reveal that the existence of a secret message is difficult to detect with the naked eye because it is in an image, and the resulting stego image meets the imperceptibility requirements (the five senses cannot see the existence of a secret message), as evidenced by the PSNR value analysis which shows the average 51 dB.

In future research, it is hoped that we can use other algorithms which are better or improve the existing algorithms, especially in embedding ciphertext into images.

## *REFERENCES*


[1] Putra, R. A. (2019). Tantangan Media Massa Dalam menghadapi era disrupsi teknologi informasi. JUSIFO (Jurnal Sistem Informasi), 5(1), 1-6.

[2] Fatma, Y., Hafid, A., & Dani, H. O. (2020). Peningkatan Keamanan Pengiriman Pesan Teks: Kombinasi Advanced Encryption Standard (AES) 128 dan Least Significant Bit (LSB). JUSIFO (Jurnal Sistem Informasi), 6(2), 111-120.

[3] Gunawan, C. E., & Fenando, F. (2018). Pengukuran Keamanan Informasi Menggunakan Indeks Keamanan Informasi (KAMI) Studi Kasus di PUSTIPD UIN Raden Fatah Palembang. JUSIFO (Jurnal Sistem Informasi), 4(2), 121-132.

[4] Darmayanti, I., Astrida, D. N., & Ariyus, D. (2019). Penerapan Keamanan Pesan Teks Menggunakan Modifikasi Algoritma Caisar Chiper Kedalam Bentuk Sandi Morse. Jurnal Ilmiah IT CIDA, 4(1).

[5] Alasi, T. S., Wanto, R., & Sitanggang, V. H. (2020). Implementasi Kriptografi Algoritma Idea Pada Keamanan Data Teks Berbasis Android. Jurnal Informasi Komputer Logika, 2(1).

[6] Gustiawan, H., & Rian, H. (2022). Pengamanan Dokumen Digital Perusahaan Menggunakan Metode Least Significant Bit (LSB) Dan Algoritma RC4 Stream Chipher. Jurnal Teknologi Informatika dan Komputer, 8(1), 228-246.

[7] Novianto, D., & Setiawan, Y. (2019). Aplikasi Pengamanan Informasi Menggunakan Metode Least Significant Bit (Lsb) dan Algoritma Kriptografi Advanced Encryption Standard (AES). Jurnal Informatika Global, 9(2).

[8] Panwar, S., Kumar, M., & Sharma, S. (2019, May). Digital image steganography using modified lsb and aes cryptography. In International Conference on Internet of Things and Connected Technologies (pp. 366-375). Springer, Cham.

[9] Irawan, C., & Rachmawanto, E. (2021). Keamanan Data Menggunakan Gabungan Kriptografi AES dan RSA. Proceeding SENDI_U, 567-573.
[10] Fernando, E., Agustin, D., Irsan, M., Murad, D. F., Rohayani, H., & Sujana, D. (2019, September). Performance comparison of symmetries encryption algorithm AES and DES with raspberry pi. In 2019 International Conference on Sustainable Information Engineering and Technology (SIET) (pp. 353-357). IEEE.
[11] Pabbi, A., Malhotra, R., & Manikandan, K. (2021, March). Implementation of Least Significant Bit Image Steganography with Advanced Encryption Standard. In 2021 International Conference on Emerging Smart Computing and Informatics (ESCI) (pp. 363-366). IEEE.
[12] Gustiawan, H., & Rian, H. (2022). Pengamanan Dokumen Digital Perusahaan Menggunakan Metode Least Significant Bit (LSB) Dan Algoritma RC4 Stream Chipher. Jurnal Teknologi Informatika dan Komputer, 8(1), 228-246.
[13] Wiranata, A. D., & Aldisa, R. T. (2021). Aplikasi Steganografi Menggunakan Least Significant Bit (LSB) dengan Enkripsi Caesar Chipper dan Rivest Code 4 (RC4) Menggunakan Bahasa Pemrograman JAVA. Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi), 5(3), 277-281.
[14] Damrudi, M., & Aval, K. J. (2019). Image Steganography using LSB and Encrypted Message with AES, RSA, DES, 3DES and Blowfish. International Journal of Engineering and Advanced Technology (IJEAT), 8(6S3).
[15] Anwar, F., Rachmawanto, E. H., & Sari, C. A. (2019, July). Stegocrypt scheme using LSB-AES base64. In 2019 International conference on information and communications technology (ICOIACT) (pp. 85-90). IEEE.
[16] Yusfrizal, Y., Meizar, A., Kurniawan, H., & Agustin, F. (2018, August). Key management using combination of Diffie–Hellman key exchange with AES encryption. In 2018 6th International Conference on Cyber and IT Service Management (CITSM) (pp. 1-6). IEEE.
[17] Al-Hilali, A. A., Jumma, L. F., & Amory, I. A. (2019). High-Quality Image Security Implementation Using 128-Bit Based on Advanced Encryption Standard algorithm. Journal of Southwest Jiaotong University, 54(6).
[18] Hamdy, A. (2021). Image Processing and AES for Secure Communications. (Bachelor Thesis, The German University in Cairo).