Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

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

Digital Image Security Analysis using Hill Cipher and AES Algorithm

Dwi Ranti[1]*, Achmad Fauzi[2], Melda Pita Uli Sitompul[3]

[1, 2, 3] *STMIK Kaputama*

dranti743@gmail.com[1], fauzyrivai88@gmail.com[2], meldasitompul19@gmail.com[3]*

Abstract

In today's digital era, digital image exchange has become very common in various industries, but this also increases security risks such as counterfeiting, image manipulation, and information theft. To protect the confidentiality of information in digital images, encryption is a fairly effective method. Hill Cipher, as a classic cryptographic method, offers matrix-based encryption, while *Advanced Encryption Standard* (AES) is known for its high level of security and efficiency. By combining Hill Cipher and AES, encryption systems can leverage the strengths of classic and modern cryptography together, providing an additional layer of protection that strengthens the security of digital data and reduces vulnerabilities that may exist in each method separately. This approach provides a more comprehensive solution for maintaining the confidentiality of digital images in the context of evolving security threats.

Keywords: Encryption, Hill Cipher, *Advanced Encryption Standard* (AES)

1. Introduction

The exchange of information using digital images is becoming increasingly common in the increasingly advanced digital era. However, the increasing use of digital images also increases risks related to the security of the information contained therein, such as data falsification, image manipulation and information theft. Because of this, researchers are increasingly convinced that images or images that are private need to be locked or secured. Hill Cipher is a classic cryptographic method that can be used to encrypt digital images, while Advanced Encryption Standard (AES) is a secure encryption algorithm to protect data. Combining Hill Cipher and AES can provide an additional layer of security to keep digital images confidential.

2. Literature Review

2.1. Cryptography

Cryptography is a science that aims to create secret messages. This process involves converting the original message, called plaintext, into an encrypted message, called ciphertext, through an encryption process. Then, the ciphertext can be returned to plaintext through the decryption process. In cryptography, there are various algorithms that are used as a form of security for information. Cryptographic algorithms can be grouped into two types, namely classical and modern cryptographic algorithms. Classical cryptographic algorithms generally use character mode, while modern cryptographic algorithms use bit mode which is formed from ASCII (American Standard Code for Information Interchange) code [1].

2.2. Image

Image is a picture and similarity of an object or thing. Images as the output of a data recording system can be optical in the form of photos, analog in the form of signals such as images on a television monitor or digital in nature which can be stored directly. Digital images are composed of many pixels. These pixels have values that indicate their level. The data in the form of pixel intensity values is then stored in digital storage media. Images are divided into two, namely analog images and digital images [2].

2.2.1. Analog Image

Analog images are images that are continuous, for example images on television, paintings and natural scenes. Analog images cannot be represented (realized) on a computer so they cannot be processed on a computer directly. Therefore, in order for this image to be processed by a computer, the analog to digital conversion process must be carried out first. Analog images are produced from analog devices, such as analog video cameras and analog photo cameras.

2.2.2. Digital Image

Digital Image are discrete images that can be processed by a computer. This image can be produced using a digital camera or an image that has undergone a digitization process [3].

2.3. Coding

The coding stage is implementing the design results into a form that is understandable and can be understood by a computer. At this stage the results of the design begin to be translated into machine language through a programming language [4].

2.4. Hill Cipher

Hill Cipher is a symmetric cryptographic algorithm that encrypts messages per block of plaintext. Each block has the same size as the key matrix. Before dividing the text into a series of blocks, the plaintext is first converted into number form, with each letter represented as a numeric value, for example A=0, B=1, up to Z=25.

2.4.1. Encryption in Hill Cipher

The steps in Hill Cipher encryption are as follows:

1. Determine the plaintext (original) that will be secured.
2. Determine the key matrix according to the plaintext block.
3. Converting plaintext into numerical form.
4. Divide and arrange the plaintext according to the order of the predetermined key matrix.
5. Carry out the encryption process with the formula:

C = K.P

C = Ciphertext

K = Key

P = Plaintext

So that a new matrix result is obtained from the multiplication. Then, modulate the result of the multiplication matrix by 10/26/256 (numeric count/alphabet count/ASCII value count).

6. After getting the results from modulo, convert them to a new alphabetical form according to the specified numeric form. The ciphertext obtained is the result of the Hill Cipher encryption process.

2.4.2. Decryption on Hill Cipher

The steps in Hill Cipher encryption are as follows:

1 Changing the ciphertext into numerical form according to what has been determined.

2. Determine the value of the determinant of the key matrix

$$\text{Det (K)} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ((a*d)-(b*c))$$

3. Then determine the inverse value of the key matrix (K-1)

$$K^{-1} = \frac{1}{\det(K)} \bmod 26 \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

4. The result of the inverse value of the key matrix will become a new key in the decryption process.
5. The ciphertext that has been converted into numerical form will be multiplied by the new key in the decryption process. Then modulate it by 26 (the number of letters).
6. The result of the multiplication is numerical. Then it is converted into alphabetical form according to what has been determined.
7. The original plaintext is obtained and the message can be reopened [5].

2.5. Advanced Encryption Standard (AES)

Rijndael is an algorithm designated as AES by NIST in October 2000. Rijndael was discovered by Vincent Rijmen and Joan Daemen who come from Belgium. This algorithm is included in the symmetric cryptography algorithm and block cipher. The AES cipher key is composed of keys with a length of 128 bits, 192 bits, or 256 bits. AES uses an iterative process called a round. This key length variant

will affect the number of rounds applied in the AES algorithm. Below is a table showing the number of rounds (Nr) that need to be implemented for each key length [6].

2.5.1. Encryption on AES

The encryption process in the AES algorithm consists of four types of byte transformation, namely SubBytes, ShiftRows, MixColumns and AddRoundKey. In the initial stage of encryption, the input copied into the state will undergo an AddRoundKey byte transformation. Then, the state will go through a series of transformations, namely SubBytes, ShiftRows, MixColumns and AddRoundKey repeatedly for the number of rounds (Nr). This process in the AES algorithm is known as the round function. The final round is slightly different from the previous round, in that in the last round, the state will not go through the MixColumns transformation.

2.5.1. Decryption on AES

The decryption process in the AES algorithm consists of four types of transformation, namely InvShiftRows which is an inverse process or the opposite of the ShiftRows process, where the byte position changes in the opposite direction. When InvShiftRows is performed, the bits are shifted towards the right as opposed to ShiftRows which are shifted towards the left. InvSubBytes is the opposite stage of SubBytes in the encryption process. InvMixColumns and AddRoundKey.

2.6 . Flowchart

A flowchart is a visual representation that graphically depicts the steps and sequence of procedures of a program or system. By using symbols and arrows, flowcharts clearly visualize how information or processes work [7].

2.7. Unified Modeling Language (UML)

Unified Modeling Language is a series of processes that are generally utilized to detail and simplify object-based systems or software. UML plays an important role in facilitating sustainable application development by providing a clear framework for understanding and planning the structure and interactions between system components. In addition, UML also functions as a means of transferring knowledge about the system or application to be built from one developer to another, ensuring consistency and uniform understanding within the development team.

2.8. Visual Basic 2010. NET

Visual Basic .NET is a programming language developed by Microsoft. This is an evolution of Visual Basic 6.0, which is known for its ease of understanding and reliability in following developments in software technology. One of the differences between Visual Basic .NET and previous versions is the integration of OOP (Object Oriented Programming) capabilities in Visual Basic .NET [8].

3. Analysis and Design

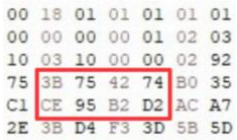
3.1. Research methodology

Security is very necessary in agencies that handle sensitive and confidential data. As in government, to protect state information and sensitive citizen data. This research uses an experimental approach with a focus on testing digital image encryption and decryption methods using a combination of the Hill Cipher and AES algorithms. The experimental design includes selecting digital image samples, encryption and decryption processes.

3.2. Analysis (Problem Topic)

This process will discuss how the combination of the Hill Cipher and AES algorithms works in encrypting and decrypting an image. Sample image file used : (Size: 109 KB). Then, look for the binary value of the image file using HxD software, the results are:

Fig. 1: Image



Hill Cipher Encryption

3B 75 42 74 CE 95 B2 D2

Converted to decimal :

3 11 7 5 4 2 7 4 12 14 9 5 11 2 13 2

Converted to alphabet :

DL HF EC HE MO JF LC NC

Key Hill Cipher (2x2 matrix key) : $\begin{bmatrix}19 & 7\\ 6 & 21\end{bmatrix}$

$\begin{bmatrix}D\\L\end{bmatrix}\rightarrow\begin{bmatrix}19 & 7\\ 6 & 21\end{bmatrix}\bullet\begin{bmatrix}3\\11\end{bmatrix}=\begin{bmatrix}(19x3)+(7x11)\\(6x3)+(21x11)\end{bmatrix}=\begin{bmatrix}134\\249\end{bmatrix}\bmod 26=\begin{bmatrix}4\\15\end{bmatrix}$

$\begin{bmatrix}H\\F\end{bmatrix}\rightarrow\begin{bmatrix}19 & 7\\ 6 & 21\end{bmatrix}\bullet\begin{bmatrix}7\\5\end{bmatrix}=\begin{bmatrix}(19x7)+(7x5)\\(6x7)+(21x5)\end{bmatrix}=\begin{bmatrix}168\\147\end{bmatrix}\bmod 26=\begin{bmatrix}12\\17\end{bmatrix}$

$\begin{bmatrix}E\\C\end{bmatrix}\rightarrow\begin{bmatrix}19 & 7\\ 6 & 21\end{bmatrix}=\begin{bmatrix}(19x4)+(7x2)\\(6x4)+(21x2)\end{bmatrix}=\begin{bmatrix}90\\66\end{bmatrix}\bmod 26=\begin{bmatrix}12\\14\end{bmatrix}$

$\begin{bmatrix}H\\E\end{bmatrix}\rightarrow\begin{bmatrix}19 & 7\\ 6 & 21\end{bmatrix}\bullet\begin{bmatrix}7\\4\end{bmatrix}=\begin{bmatrix}(19x7)+(7x4)\\(6x7)+(21x4)\end{bmatrix}=\begin{bmatrix}161\\126\end{bmatrix}\bmod 26=\begin{bmatrix}5\\22\end{bmatrix}$

$\begin{bmatrix}M\\O\end{bmatrix}\rightarrow\begin{bmatrix}19 & 7\\ 6 & 21\end{bmatrix}\bullet\begin{bmatrix}12\\14\end{bmatrix}=\begin{bmatrix}(19x12)+(7x14)\\(6x12)+(21x14)\end{bmatrix}=\begin{bmatrix}326\\366\end{bmatrix}\bmod 26=\begin{bmatrix}14\\2\end{bmatrix}$

$\begin{bmatrix}J\\F\end{bmatrix}\rightarrow\begin{bmatrix}19 & 7\\ 6 & 21\end{bmatrix}\bullet\begin{bmatrix}9\\5\end{bmatrix}=\begin{bmatrix}(19x9)+(7x5)\\(6x9)+(21x5)\end{bmatrix}=\begin{bmatrix}206\\159\end{bmatrix}\bmod 26=\begin{bmatrix}24\\3\end{bmatrix}$

$\begin{bmatrix}L\\C\end{bmatrix}\rightarrow\begin{bmatrix}19 & 7\\ 6 & 21\end{bmatrix}\bullet\begin{bmatrix}11\\2\end{bmatrix}=\begin{bmatrix}(19x11)+(7x2)\\(6x11)+(21x2)\end{bmatrix}=\begin{bmatrix}223\\108\end{bmatrix}\bmod 26=\begin{bmatrix}15\\4\end{bmatrix}$

$\begin{bmatrix}N\\C\end{bmatrix}\rightarrow\begin{bmatrix}19 & 7\\ 6 & 21\end{bmatrix}\bullet\begin{bmatrix}13\\2\end{bmatrix}=\begin{bmatrix}(19x13)+(7x2)\\(6x13)+(43x2)\end{bmatrix}=\begin{bmatrix}261\\120\end{bmatrix}\bmod 26=\begin{bmatrix}1\\16\end{bmatrix}$

Ciphertext : 4 15 12 17 12 14 5 22 14 2 24 3 15 4 1 16

AES encryption

Plaintext AES :

4	F	C	11
C	E	5	16
E	2	18	3
F	4	1	10

AES Key :

25	1C	B	C
1D	E	F	10
11	18	13	28
11	15	21	12

Before encrypting, first calculate the key expansion (Key Schedule).

- Round 1

Key				→	Rotation	→	S-Box
25	1C	B	C		10		CA
1D	E	F	10		28		34
11	18	13	28		12		C9
11	15	21	12		C		FE

Key		S-Box		Rcon		Column 1 Results
25 (00100101)		CA (11001010)		01 (00000001)		EE (11101110)
1D (00011101)	Xor	34 (00110100)	Xor	00 (00000000)	=	29 (00101001)
11 (00010001)		C9 (11001001)		00 (00000000)		D8 (11011000)
11 (00010001)		FE (11111110)		00 (00000000)		EF (11101111)

Key		Column 1 Results		Column 2 Results
1C 00011100		EE 11101110		F2 11110010
E 00001110	Xor	29 00101001	=	27 00100111
18 00011000		D8 11011000		C0 11000000
15 00010101		EF 11101111		FA 11111010

Key		Column 2 Results		Column 3 Results
B 00001011		F2 11110010		F9 11111001
F 00001111	Xor	27 00100111	=	28 00101000
13 00010011		C0 11000000		D3 11010011
21 00100001		FA 11111010		DB 11011011

Key		Column 3 Results		Column 4 Results
C 00001100		F9 11111001		F5 11110101
10 00010000	Xor	28 00101000	=	38 00111000
28 00101000		D3 11010011		FB 11111011
12 00010010		DB 11011011		C9 11001001

Results from Key Schedule Round 1 : EE 29 D8 EF | F2 27 C0 FA | F9 28 D3 DB | F5 38 FB C9
Do it in the same way to produce 10 Key Schedules, namely as follows :
Round 2 : EB 26 05 09 | 19 01 C5 F3 | E0 29 16 28 | 15 11 ED E1
Round 3 : 6D 73 FD 50 | 74 72 38 A3 | 94 5B 2E 8B | 81 4A C3 6A
Round 4 : B3 5D FF 5C | C7 2F C7 FF | 53 74 E9 74 | D2 3E 2A 1E
Round 5 : 11 B8 BD E9 | D6 97 4A 16 | 85 E3 A3 62 | 57 DD 89 7C
Round 6 : F0 1F 9D B2 | 26 88 D7 A4 | A3 6B 74 C6 | F4 B6 FD BA
Round 7 : FE 4B 69 0D | D8 C3 BE A9 | 7B A8 CA 6F | 8F 1E 37 D5
Round 8 : 0C D1 6A 7E | D4 12 D4 D7 | AF BA 1E B8 | 20 A4 29 6D
Round 9 : 5E 74 56 C9 | 8A 66 82 1E | 25 DC 9C A6 | 05 78 B5 CB
Round 10 : D4 A1 49 A2 | 5E C7 CB BC | 7B 1B 57 1A | 7E 63 32 D1

Encryption Stage
AddRoundKey or can also be called Initial Round, that is the plaintext is xored with the key.

Plaintext AES :

4	F	C	11
C	E	5	16
E	2	18	3
F	4	1	10

AES Key :

25	1C	B	C
1D	E	F	10
11	18	13	28
11	15	21	12

4 xor 25 = 00000100 xor 00100101 =00100001, hexa : 21

F xor 1C = 00001111 xor 00011100 = 00010011, hexa : 13

C xor 1D = 00001100 xor 00011101 = 00010001, hexa : 11

E xor E = 00001110 xor 00001110 = 00000000, hexa : 00

E xor 11 = 00001110 xor 00010001 = 00011111, hexa : 1F

2 xor 18 = 00000010 xor 00011000 = 00011010, hexa : 1A

F xor 11 = 00001111 xor 00010001 = 00011110, hexa : 1E

4 xor 15 = 00000100 xor 00010101 = 00010001, hexa : 11

AddRoundKey Result :

21	13	07	1D
11	00	0A	06
1F	1A	0B	2B
1E	11	20	02

1) Round 1

SubBytes

21	13	07	1D
11	00	0A	06
1F	1A	0B	2B
1E	11	20	02

S-Box

FD	7D	C5	A4
82	63	67	6F
C0	A2	2B	F1
72	82	B7	77

ShiftRows

FD	7D	C5	A4		FD	7D	C5	A4
82	63	67	6F	→ 1	63	67	6F	**82**
C0	A2	2B	F1	→2	2B	F1	**C0**	**A2**
72	82	B7	77	→ 3	77	**72**	**82**	**B7**

MixColumns

02	03	01	01
01	02	03	01
01	01	02	03
03	01	01	02

X

FD	7D	C5	A4
63	67	6F	82
2B	F1	C0	A2
77	72	82	B7

Result :

18	D0	62	DB
31	C9	C2	F1
51	75	AC	BB
BA	F5	E4	A2

AddRoundKey (Key Schedule)

MixColumn Results

18	D0	62	DB
31	C9	C2	F1
51	75	AC	BB
BA	F5	E4	A2

Xor

Key Schedule Round 1

EE	F2	F9	F5
29	27	28	38
D8	C0	D3	FB
EF	FA	DB	C9

So you get the results:

F6	22	9B	2E
18	EE	EA	C9
89	B5	7F	40
55	0F	3F	6B

Continue until round 10.

10) Round 10

SubBytes

FB	A0	9C	1F
D3	AB	E9	79
54	A0	B0	A3
1F	84	4D	35

S-Box

0F	E0	DE	C0
66	62	1E	B6
20	E0	E7	0A
C0	5F	E3	96

ShiftRows

0F	E0	DE	C0
66	62	1E	B6
20	E0	E7	0A
C0	5F	E3	96

→ 1

→ 2

→ 3

0F	E0	DE	C0
62	1E	B6	**66**
E7	0A	**20**	**E0**
96	**C0**	**5F**	**E3**

AddRoundKey (Key Schedule)

ShiftRows results are xorted with Key Schedule Round 10

DB	BE	A5	BE
C3	D9	AD	05
AE	C1	77	02
34	7C	45	32

Ciphertext (Encryption Results) : DB C3 AE 34 | BE D9 C1 7C | A5 AD 77 45 | BE 05 02 32

AES Decryption

Initial Round or often called AddRoundKey, namely Ciphertext xorted with Key Schedule Round 10

DB	BE	A5	BE
C3	D9	AD	05
AE	C1	77	02
34	7C	45	32

Xor

D4	5E	7B	7E
A1	C7	1B	63
49	CB	57	E2
A2	BC	1A	D1

Result :

0F	E0	DE	C0
62	1E	B6	66
E7	0A	20	E0
96	C0	5F	E3

InvShiftRows

0F	E0	DE	C0
62	1E	B6	66
E7	0A	20	E0
96	C0	5F	E3

← 1

← 2

← 3

0F	E0	DE	C0
66	62	1E	B6
20	**E0**	E7	0A
C0	**5F**	**E3**	96

InvSubBytes

0F	E0	DE	C0
66	62	1E	B6
20	E0	E7	0A
C0	5F	E3	96

S-Box

FB	A0	9C	1F
D3	AB	E9	79
54	A0	B0	A3
1F	84	4D	35

1) Round 1

Xor the result of InvSubBytes with Key Schedule 9

A5	2A	B9	1A
A7	CD	35	01
02	22	2C	16
D6	9A	EB	FE

InvMixColumns

0E	0B	0D	09
09	0E	0B	0D
0D	09	0E	0B
0B	0D	09	0E

Xor

A5	2A	B9	1A
A7	CD	35	01
02	22	2C	16
D6	9A	EB	FE

Result :

37	4E	FD	36
15	DE	77	D4
27	F8	FE	C7
D3	37	3F	D6

Continue until round 10.

1) Round 10

Xor the result of InvSubBytes with AES Key

4	F	C	11
C	E	5	16
E	2	18	3
F	4	1	10

The ciphertext returns to the original AES plaintext.

Hill Cipher Decryption

Ciphertext: 4 F C 11 | C E 5 16 | E 2 18 3 | F 4 1 10

Converted to decimal :

4 15 12 17 | 12 14 5 22 | 14 2 24 3 | 15 4 1 16

Key Hill Cipher (2x2 matrix key) : $\begin{bmatrix}19 & 7\\ 6 & 21\end{bmatrix}$

Calculate Determinant Value

Det (k) = det $\begin{bmatrix}a & b\\ c & d\end{bmatrix}$ = $\begin{bmatrix}19 & 7\\ 6 & 21\end{bmatrix}$ = (a*d)-(b*c)

= (19*21)-(7*6)

= 399 – 42

= 357 mod 26

= 19

Compute the inverse of the key matrix

$K^{-1} = \frac{1}{det(k)} \bmod 26 \begin{bmatrix}d & -b\\ -c & a\end{bmatrix}$

$K^{-1} = \frac{1}{19} \bmod 26 \begin{bmatrix}21 & -7\\ -6 & 19\end{bmatrix}$

(19.11 mod 26 = 1)

$K^{-1} = 11 * \begin{bmatrix}21 & -7\\ -6 & 19\end{bmatrix} = \begin{bmatrix}231 & 209\\ 220 & 209\end{bmatrix} \bmod 26 = \begin{bmatrix}23 & 1\\ 12 & 1\end{bmatrix}$

$\begin{bmatrix}23 & 1\\ 12 & 1\end{bmatrix} \bullet \begin{bmatrix}4\\ 15\end{bmatrix} = \begin{bmatrix}(23x4)+(1x15)\\ (12x4)+(1x15)\end{bmatrix} = \begin{bmatrix}107\\ 63\end{bmatrix} \bmod 26 = \begin{bmatrix}3\\ 11\end{bmatrix}$

$\begin{bmatrix}23 & 1\\ 12 & 1\end{bmatrix} \bullet \begin{bmatrix}12\\ 17\end{bmatrix} = \begin{bmatrix}(23x12)+(1x17)\\ (12x12)+(1x17)\end{bmatrix} = \begin{bmatrix}293\\ 161\end{bmatrix} \bmod 26 = \begin{bmatrix}7\\ 5\end{bmatrix}$

$\begin{bmatrix}23 & 1\\ 12 & 1\end{bmatrix} \bullet \begin{bmatrix}12\\ 14\end{bmatrix} = \begin{bmatrix}(23x12)+(1x14)\\ (12x12)+(1x14)\end{bmatrix} = \begin{bmatrix}290\\ 158\end{bmatrix} \bmod 26 = \begin{bmatrix}4\\ 2\end{bmatrix}$

$\begin{bmatrix}23 & 1\\ 12 & 1\end{bmatrix} \bullet \begin{bmatrix}5\\ 22\end{bmatrix} = \begin{bmatrix}(23x5)+(1x22)\\ (12x5)+(1x22)\end{bmatrix} = \begin{bmatrix}137\\ 82\end{bmatrix} \bmod 26 = \begin{bmatrix}7\\ 4\end{bmatrix}$

$\begin{bmatrix}23 & 1\\ 12 & 1\end{bmatrix} \bullet \begin{bmatrix}14\\ 2\end{bmatrix} = \begin{bmatrix}(23x14)+(1x2)\\ (12x14)+(1x2)\end{bmatrix} = \begin{bmatrix}324\\ 170\end{bmatrix} \bmod 26 = \begin{bmatrix}12\\ 14\end{bmatrix}$

$\begin{bmatrix}23 & 1\\ 12 & 1\end{bmatrix} \bullet \begin{bmatrix}24\\ 3\end{bmatrix} = \begin{bmatrix}(23x24)+(1x3)\\ (12x24)+(1x3)\end{bmatrix} = \begin{bmatrix}555\\ 291\end{bmatrix} \bmod 26 = \begin{bmatrix}9\\ 5\end{bmatrix}$

$\begin{bmatrix}23 & 1\\ 12 & 1\end{bmatrix} \bullet \begin{bmatrix}15\\ 4\end{bmatrix} = \begin{bmatrix}(23x15)+(1x4)\\ (12x15)+(1x4)\end{bmatrix} = \begin{bmatrix}349\\ 184\end{bmatrix} \bmod 26 = \begin{bmatrix}11\\ 2\end{bmatrix}$

$\begin{bmatrix}23 & 1\\ 12 & 1\end{bmatrix} \bullet \begin{bmatrix}1\\ 16\end{bmatrix} = \begin{bmatrix}(23x1)+(1x16)\\ (12x1)+(1x16)\end{bmatrix} = \begin{bmatrix}39\\ 28\end{bmatrix} \bmod 26 = \begin{bmatrix}13\\ 2\end{bmatrix}$

Plaintext (Hill Cipher) : 3 11 7 5 4 2 7 4 12 14 9 5 11 2 13 2
Convert to hexadecimal : 3 B 7 5 4 2 7 4 C E 9 5 B 2 D 2

3.3. Process Design

In designing this image file application, the author used a combination of the Hill Cipher and AES algorithms. Where this design uses a flowchart to find out how the encryption and decryption processes will be designed in the system.

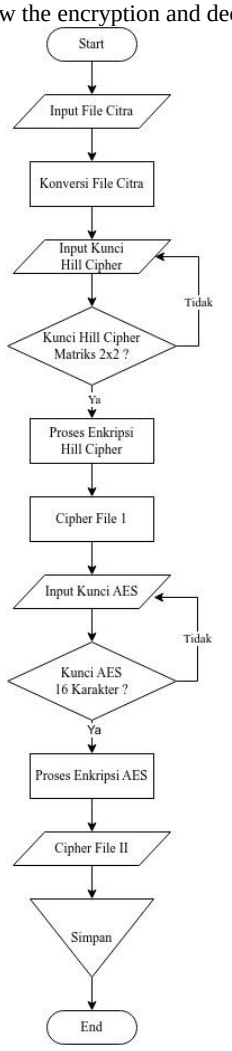


Fig.2: Encryption Flowchart

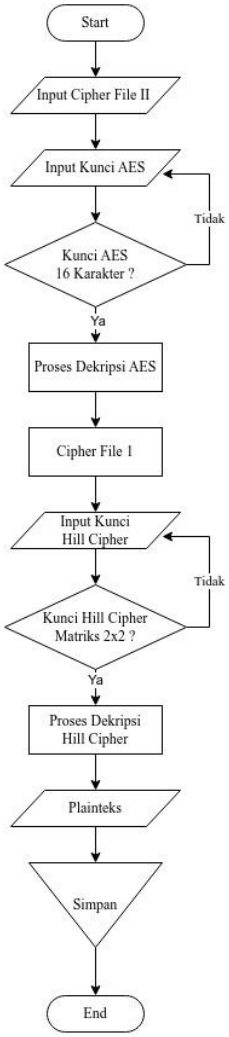


Fig. 3: Decryption Flowchart

3.4. System Modeling

Use Case Diagram

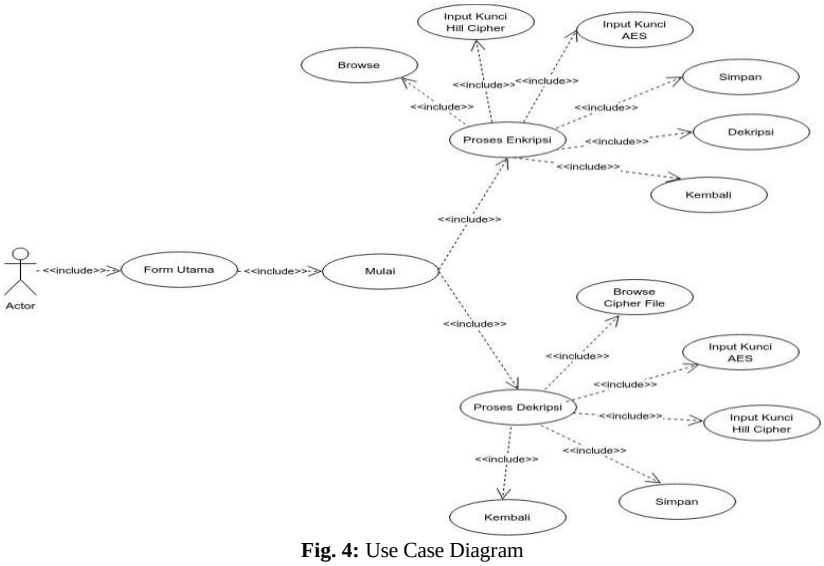


Fig. 4: Use Case Diagram

3.5. Interface Design

Main Form

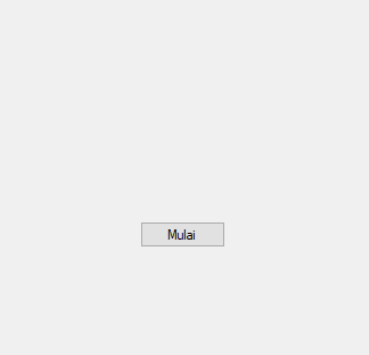


Fig. 5: Main Form

Encryption Process Form

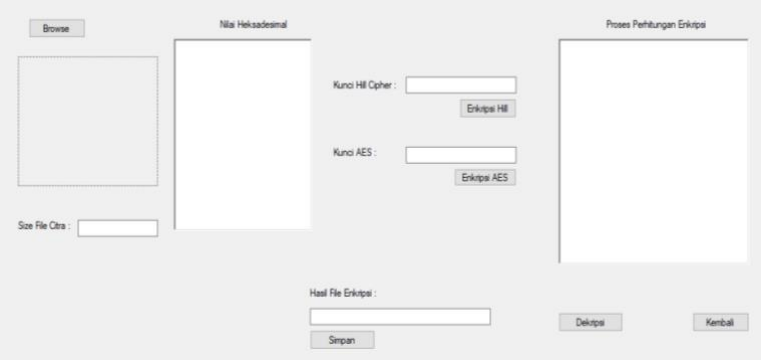


Fig. 6: Encryption Process Form

Decryption Process Form

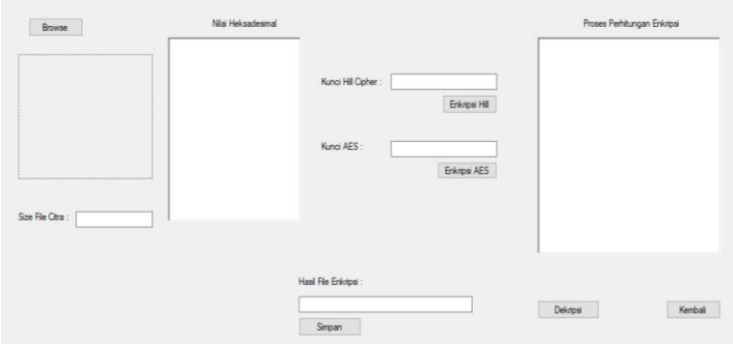


Fig. 7: Decryption Process Form

4. Implementation and Discussion

4.1. Main Form

This Main Form is the main interface form that provides access to various features and functions of the System. The following is the main form of the system.

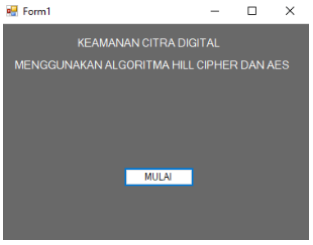


Fig. 8: Main Form Display

4.2. Encryption Process Form

Fig. 9: View After Encryption

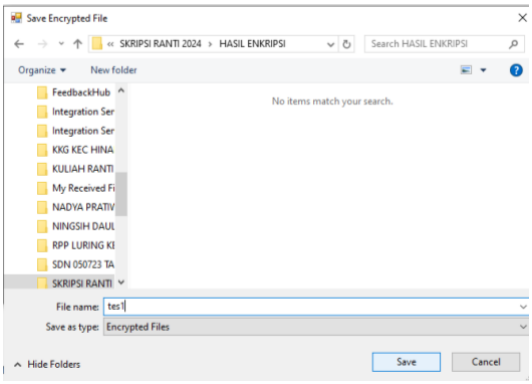


Fig. 10: Display Save Encryption Results

4.3. Decryption Process Form

Fig. 11: Decryption Results Display

In the Decryption form, when you want to restore the initial image file but the Hill Cipher decryption results do not return to the Initial Hexadecimal Value which is the plaintext of the image file. Because the results are not returned, the original image also does not appear.

5. Conclusion

In this research, the Hill Cipher and AES algorithms are applied to encrypt digital images by converting the image into hexadecimal form which is used as plaintext. However, the research results show that the combination of the two algorithms cannot run optimally when applied to image files. Problems arise mainly at the decryption stage. After the AES decryption process, the Hill Cipher decryption results cannot be returned to the correct hexadecimal value format to reconstruct the original image. As a result, the original image cannot be displayed again correctly. This error occurs due to differences in the nature of the Hill Cipher algorithm which operates on decimal values with a certain modulus and the AES algorithm which works on blocks of data in binary form, causing incompatibilities when processing image hexadecimal values. This conclusion is proof that there needs to be a more suitable approach or method if you want to combine the two algorithms for digital image security.

References

[1] S. P. Ananda and S. Lukman, "Analisa Metode Kriptografi Modern Advance Encryption Standard (AES) 128 Bit dalam Mengenkripsi dan Mendekripsi File Dokumen Digital," *J. Ilm. Komputasi*, vol. 21, no. 3, pp. 333–344, 2022, doi: 10.32409/jikstik.21.3.2973.

[2] A. S. Irtawaty and R. Jayanti, "Implementasi Pengolahan Citra Pada Analisis Ciri Bakteri Yogurt," *JST (Jurnal Sains Ter.*, vol. 2, no. 2, pp. 83–87, 2016, doi: 10.32487/jst.v2i2.179.

[3] R. Perangin-angin and E. J. Gunawati Harianja, "Comparison Detection Edge Lines Algoritma Canny dan Sobel," *J. TIMES (Techonology Informatics Comput. Syst.*, vol. 8, no. 2, pp. 35–42, 2019.

[4] D. Syahrul Suci Romadhon1, "Vol . 3 No . 1 Februari 2019 ISSN : 2597-3673 (Online) ISSN : 2579-5201 (Printed) ISSN : 2597-3673 (Online) ISSN : 2579-5201 (Printed)," *Peranc. WEBSITE Sist. Inf. SIMPAN PINJAM MENGGUNAKAN Framew. CODEIGINTER PADA Kop. BUMI ISSN 2579-5201 Peranc. Sejah. JAKARTA Syahrul*, vol. 3, no. 1, pp. 21–28, 2019.

[5] Y. W. Hasibuan, R. B. Veronica, J. Matematika, U. N. Semarang, K. S. Gunungpati, and I. Artikel, "How to Cite," vol. 11, no. 1, pp. 54–68, 2022.

[6] B. E. Widodo and A. S. Purnomo, "Implementasi Advanced Encryption Standard Pada Enkripsi Dan Dekripsi Dokumen Rahasia Ditintelkam Polda Diy," *J. Tek. Inform.*, vol. 1, no. 2, pp. 69–77, 2020, doi: 10.20884/1.jutif.2020.1.2.21.

[7] A. Eka Putri, A. Kartikadewi, and L. A. Abdul Rosyid, "Implementasi Kriptografi dengan Algoritma Advanced Encryption Standard (AES) 128 Bit dan Steganografi menggunakan Metode End of File (EOF) Berbasis Java Desktop pada Dinas Pendidikan Kabupaten Tangerang," *Appl. Inf. Syst. Manag.*, vol. 3, no. 2, pp. 69–78, 2021, doi: 10.15408/aism.v3i2.14722.

[8] N. T. Sidik and K. Kusmadi, "Rancang Bangun Sistem Peringatan Dini Banjir Dengan Menggunakan Arduino Uno Dan Monitoring Level Ketinggian Air Pada Pc Dengan Aplikasi Visual Basic," *Infotronik J. Teknol. Inf. dan Elektron.*, vol. 5, no. 1, pp. 17–23, 2020, doi: 10.32897/infotronik.2020.5.1.3.