Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

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

Rivest Shamir Adleman (RSA) Super Encryption Algorithm with Vigenere Cipher Algorithm Modification for Image Security

Mariza[1*], Achmad Fauzi[2], Yani Maulita[3]

[1, 2, 3]STMIK Kaputama

yulimarisaarora@gmail.com[1], fauzyrivai88@gmail.com[2], yanimaulita26@gmail.com[3]

Abstract

The development of information technology makes information easy to transfer and transmit digital media such as images and videos, but also brings security risks to data that is vulnerable to being infiltrated by irresponsible parties. Digital image security is a major concern in areas such as design security, medical science and architecture. Cryptography is a solution for protecting digital images, where encryption and decryption are the core. This research proposes a double encryption algorithm that combines Rivest Shamir Adleman (RSA) and Vigenere Cipher. The RSA algorithm, as the leading asymmetric cryptography algorithm, is used to secure the Vigenere key, which is then used to encrypt the image. This approach provides multiple layers of security, making attacks on images more difficult. The aim of this research is to analyze and implement the RSA algorithm with the Vigenere Cipher modification in digital image encryption, as well as evaluating the effectiveness, security and efficiency of the algorithm. The results show the success of this approach in protecting digital images from data attacks and hacking, although there is still room for improvement. In addition, this algorithm provides security without sacrificing the quality of the digital image. This research contributes to the development of digital image security methods, which are important in today's information age. By using this dual encryption approach, this research provides a solution that can be applied in a variety of contexts, maintaining the security and integrity of digital data in the face of increasingly complex threats.

Keywords: *Digital Image, Cryptography, RSA, Vigenere Cipher, Information Security*

1. Introduction

In the increasingly developing digital era, data security becomes very important. One of the key aspects of maintaining the confidentiality of information is encryption, which allows messages to be converted into a format that cannot be directly understood by unauthorized parties. In this context, the RSA (Rivest-Shamir-Adleman) algorithm has become one of the most well-known and powerful public key encryption algorithms. RSA is based on the mathematical concept of factorization of large integers. Using a public key and private key pair, this algorithm allows users to encrypt messages using the public key and decrypt them only with the corresponding private key. The main advantage of RSA is the difficulty of solving its private key, which is based on the difficulty of factoring the product of two large prime numbers [1], [2].

However, when we talk about image security, especially in the context of sending images over a network, other aspects also need to be considered. The Vigenere Cipher algorithm offers a more complex way of encryption than the simple Caesar Cipher encryption, because it applies a longer key and shifts the message characters according to the corresponding key characters. By combining the power of the RSA Algorithm with the complexity and key variety of the Vigenere Cipher Algorithm, we can create a strong encryption solution suitable for image security. In this context, a modification of the Vigenere Cipher Algorithm can be applied to strengthen image security, where each pixel in the image is encrypted using a key derived from the RSA key. Thus, although someone may be able to access an encrypted image, without the appropriate key, the contents of the image will remain unintelligible [3], [4].

In this paper, we will explore the concept further, introducing a modification of the Vigenere Cipher Algorithm for image security using keys generated by the RSA Algorithm. We will also analyze the security power of this approach and consider its practical application in the evolving context of digital data security. In doing so, we will offer a comprehensive look at how to utilize two powerful cryptographic algorithms to protect images and other digital data from unauthorized access [5].

2. Theoretical Basis

2.1. Cryptography

The word cryptography comes from Greek, namely "kryptós" which means hidden and "gráphein" which means writing. So cryptography can be interpreted as "hidden writing". According to the RFC (Request for Comments), cryptography is a mathematical science that deals with the transformation of data to hide its meaning, prevent its unauthorized modification, or prevent its unauthorized use. If conversion can be translated, then crypto can also be interpreted as the process of returning encrypted data into an understandable form. In other words, cryptography can be interpreted as a data protection process in a broad sense [6], [7].

2.2. Discrete Cosine Transform (DCT)

The RSA algorithm was created by three researchers from the Massachusetts Institute of Technology (MIT), namely Ron Rivest, Adi Shamir and Leonard Adleman in 1997. The security of the RSA algorithm lies in the difficulty of factoring large numbers into prime factors. Factorization is carried out to obtain the private key. As long as there is no efficient algorithm for factoring large numbers into prime factors, so long as the security of the RSA algorithm remains guaranteed. The quantities used in the RSA algorithm are [8], [9]:

p and q prime numbers (secret)
n = p . q (not secret)
$\phi(n) = (p - 1)(q - 1)$ (secret)
e (encryption key) (not secret)
d (decryption key) (secret)
m (plaintext) (confidential)
c (ciphertext) (not secret)

The steps for making a key include:

- Choose two prime numbers, for example p and q and $p \neq q$ and randomly and separately for each p and q. Calculate the value of n where the value of n is the product of the numbers p and q.
- Calculate the value nilai $\phi = (p - 1)(q - 1)$.
- Choose an integer between one and ϕ, $(1 < e < \phi)$ which is also a coprime number of ϕ.
- Calculate the value of d until $d\,e = 1 \pmod{\phi}$, and to find d you can use the formula:

$$d = \frac{(1+kN)}{e} \quad (1)$$

The k value is the result of experimental values of 1,2,3, ... so that the resulting d value is a round value.

2.3. Vigenere Cipher Algorithm

The Vigenere Cipher is a classic cryptographic algorithm that was introduced in the 16th century or around 1986. This cryptographic algorithm was published by a French diplomat and cryptographer named Blaise de Vigenère, but actually this algorithm was previously explained in the book La Cifra del Sig [10], [11].
The operation of the vigenere cipher is similar to caesar, namely it encrypts plain message text by shifting the letters of the message to key values in the letter string. Vigenere Cipher is a classic encryption algorithm that uses compound alphabet substitution. Compound alphabet substitution encrypts each letter with a different key, unlike the Caesar cipher which implements single alphabet substitution where all the letters in the message are the same, encrypted with the same key.

The mathematical model of encryption in the Vigenere cipher algorithm is as follows:
Ci = Ek(Mi) = (Mi + Ki) md 26
And the mathematical model for the decryption is:
Mi = Dk(Ci) = (Ci – Ki) mod 26
With C modeling the ciphertext, M modeling the plaintext, and K modeling the key.
However, researchers used a modified Vigenere cipher to complete the research. Where, in the process of modifying the classical vigenere cipher, the modulus value previously used for encryption and decryption of text messages, namely 26 (the number of letters of the alphabet), is changed to the number of intensity values for a pixel in the image, namely 256. Equation (1) is used for the encryption process Ci, while equation (2) is used for the Pi decryption process.
Ci = (Pi + (aZi – 1 + c) mod 256 (1)
Pi = (Pi – (aZi – 1 + c) mod 256 (2)
The digital image encryption process has a range of pixel values between 0 and 255, so in this case, by using a modulus of 256, the pixel value will repeat itself to o if the pixel value has reached 256. The encryption process is carried out on truecolor photo images, grayscale photo images, binary images with jpg, bmp and png storage formats, magnetic resonance images and panchromatic images.
In the process of generating keys in the classic Vigenere cipher, a Linear Congruent Generator is used which generates random numbers with a uniform distribution. The modulus m used to generate the key with the Linear Congruent Generator in this study was also changed to 256 with a multiplier factor a, Zi - 1 being the previous random number and an additional increment c for randomization repetition.
Zi = (aZi – 1 + c) mod m
The Linear Congruent Generator has a full period (m-1) if it meets the following conditions:

a. c is prime relative to m.
b. a – 1 can be divided by all prime factors of m.

c. a – 1 is a multiple of 4 if m is a multiple of 4.
d. m>max (a, c, Z0) e.a>0, c>0.

3. Research Methods

The research methodology is carried out to seek information systematically using scientific methodsand clear sources. In the course of this research is expected to provide useful results for users.
Based on the methodology used in this study, a flow of research work method activities was formed,namely as follows:

1. Preparation
   The initial stage in conducting research begins with compiling the background of the problem, whichis then followed by formulating the problem to be solved and determining the benefits of the research. Once that stage is complete, the author will determine how to encode the image so that it cannot be accessed by third parties.
2. Theory Study
   At this stage, the authors collect various theories as research support. These theories include image security, the discrete cosine transform algorithm, and the use of Visual Basic. Theoretical sourcescome from library books, scientific journals, and various sources on the internet. This theory collection aims to provide a strong and in-depth knowledge base in research development.
3. Theory Collection
   The Library Research stage is an important step in research. This process involves searching for information sources such as books, journals and internet sites that are relevant to the problem you wantto solve. By conducting literature studies, the author can obtain strong references and appropriate methods to support research and enrich the theoretical foundation in this thesis.
4. Design
   At this stage, the author carried out manual calculations using the discrete cosine transform algorithm method. After the calculations are complete, the author designs a system that will be built based on the results of these calculations. This process involves making detailed designs and plans for the implementation of the system to be developed.
5. Testing and Implementation
   a. At this stage, the discrete cosine transform algorithm is implemented into the Microsoft Visual Basic .NET 2010 programming language. Next, testing is carried out to ensure the system functions according to the design and specifications that have been determined previously. This stage is important because it converts the algorithm into executable code and ensures the system is ready foruse by end users.
   b. Do and run the program to see the results of the encrypted image.

4. System Planning

In designing this image application the author uses the DCT algorithm method to solve the problem. This design uses a flowchart to find out how the encryption and decryption process will be designed in a system.

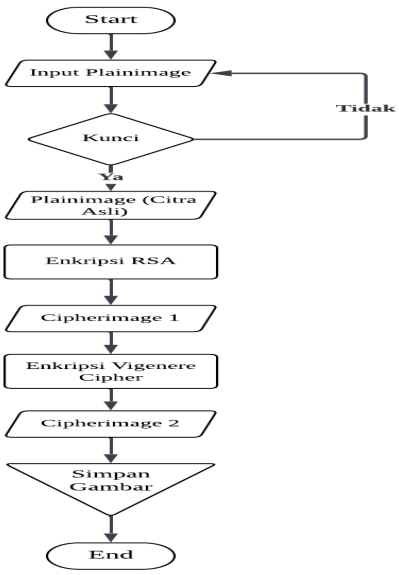

Fig. 1: Encryption Flowchart

In Figure 1 above, the encryption process will carry out several processes, namely:

1. Start.
2. Enter the original image (plain image) and key.

3. If the key does not meet the requirements, then return to image and key input. If you meet the requirements, you can proceed to the next stage.
4. Carry out the RSA encryption process to produce cipher image 1.
5. Then, cipher image 1 will be encrypted again with the vigenere cipher to produce cipher image 2.
6. The image has been successfully encrypted with two RSA algorithms and the Vigenere cipher, then save the image.
7. End

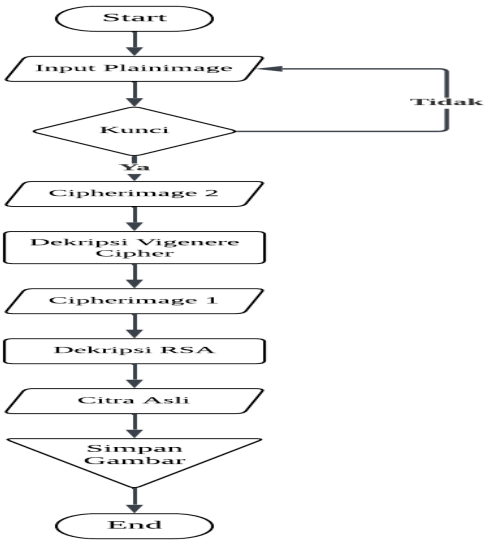

Fig. 2: Flowchart Dekripsi

In Figure 2, several processes will be carried out in the decryption process, namely:

1. Start.
2. Enter the encrypted image (cipher image 2) and key.
3. If the key does not meet the requirements, then return to image and key input. If you meet the requirements, you can proceed to the next stage.
4. Then, the Vigenere cipher decryption process on cipheri mage 2 will produce cipher image 1.
5. Then, cipher image 1 will be decrypted again with RSA to produce plaini mage 1 or the original (original) image.
6. The image has been successfully decrypted using two RSA algorithms and the Vigenere cipher, then save the image.
7. End.

The design is a description of the system and working methods that the author designed, where in designing this image security system the author used the DCT (Discrete Cosine Transform) algorithm to solve the problem. In this research the author used UML, and the explanation can be seen in the nextsub-chapter.

5. Results And Discussion

Image encryption is the process of changing image data into a form that cannot be read or understood directly. The goal is to protect the privacy or confidentiality of image data from unauthorized parties. The following images will be encrypted as samples in this research:

Fig. 3: Images will be encrypted

In Figure 3 is the image (image) that will be encrypted. The image value will be taken from each block (pixel) using the help of additional software, namely Photoshop. The following is the block value taken using Photoshop on a 4x4 image as a sample:

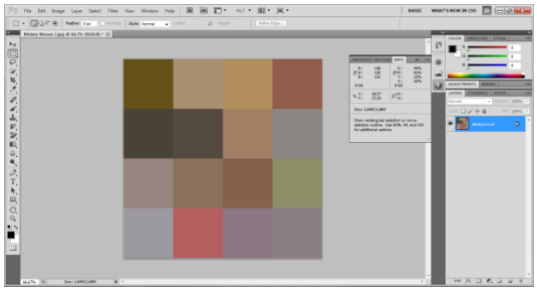

Fig. 4: The RGB values

The image above (Figure 3) shows the RGB values taken for each block which will then be made into a matrix and will form new RGB values to create an encrypted image. The following table below is a table of pixel values that have been taken from the original image:

Table. 1: Pixel Values in Images

Piksel	1			2			3			4		
	R	G	B	R	G	B	R	G	B	R	G	B
1	100	82	25	169	144	113	177	142	91	146	92	75
2	72	66	55	84	74	64	161	127	100	137	134	132
3	153	133	127	139	114	95	133	97	76	142	142	104
4	156	153	160	180	94	96	140	118	131	138	128	132

The table above (Table 1) shows the results of the values for each pixel in the form of RGB values in the digital image.

RSA Key Establishment

The RSA algorithm is a cryptographic algorithm that uses public keys and private keys to encrypt and decrypt data. For the key formation process in this test, the following steps are carried out:

1. Determine 2 prime numbers with the names variable p and variable q.
   For example: p = 101 and q = 151.
   a. Calculating the modulus value (n), where n = p*q, then:
      n = 101 * 151
      n = 15251
   b. Calculating the totient value n, where Φ(n) = (p-1)*(q-1) then:
      Φ(n) = (101-1)*(151-1)
      Φ(n) = (100)*(150)
      Φ(n) = 15000
2. Find the value of e with the condition that gcd (e, Φ(n)) = 1, e = prime number, and 1 <e<Φ(n) in this case, the prime number value of e that will be random is 2.3, 5 and 7.
   The process of finding a suitable e value according to the requirements is as follows: Φ(n) = 15000 (obtained from the previous step).
3. Look for the decipherin_exponent (d) value then:
   d = (1+(k*Φ(n)) : e
   d = (1+(k*15000)) : 7
   The k value is a random value until an integer is generated.
   k = o
   d = (1+(0*15000)) : 7, then d = 0.142 is still a fraction, continue
   k = 1 (1+(1*15000)) : 7, the result obtained is d = 2143, not a fraction, the search results get d = 2143
4. Based on the steps above, the values of n, e and d have been found so that a key pair has been formed.
   Public key pair (n,e) = (15251, 7)
   Secret key pair (n,d) = (15251, 2143).

Image Encryption Using RSA

To carry out the encryption process, the public key that has been formed is used, namely the public key (e,n) = (7.15251), with the formula: C1 = m1e mod n.

• Pixels (1,1)
C1 = 1007 mod 15251 = 6665
C2 = 827 mod 15251 = 10379
C3 = 257 mod 15251 = 4421

• Pixels (1,2)
C1 = 1697 mod 15251 = 12125
C2 = 1447 mod 15251 = 9524
C3 = 1137 mod 15251 = 3169

• Pixels (1,3)
C1 = 1777 mod 15251 = 2245
C2 = 1427 mod 15251 = 3882
C3 = 917 mod 15251 = 11322

• Pixels (1.4)
C1 = 1467 mod 15251 = 7039
C2 = 927 mod 15251 = 1502
C3 = 757 mod 15251 = 14844

• Pixels (2,1)
C1 = 727 mod 15251 = 551
C2 = 667 mod 15251 = 9056

• Pixels (2,2)
C1 = 847 mod 15251 = 12207
C2 = 747 mod 15251 = 10757

C3 = 557 mod 15251 = 11103

• Pixels (2,3)
C1 = 1617 mod 15251 = 5612
C2 = 1277 mod 15251 = 1821
C3 = 1007 mod 15251 = 6665
• Pixels (3,1)
C1 = 1537 mod 15251 = 14322
C2 = 1337 mod 15251 = 5089
C3 = 1277 mod 15251 = 1821

• Pixels (3,3)
C1 = 1337 mod 15251 = 5089
C2 = 977 mod 15251 = 2604
C3 = 767 mod 15251 = 9618

• Pixels (4,1)
C1 = 1567 mod 15251 = 4437
C2 = 1537 mod 15251 = 14322
C3 = 1607 mod 15251 = 10916

• Pixels (4,3)
C1 = 1407 mod 15251 = 12969
C2 = 1187 mod 15251 = 1024
C3 = 1317 mod 15251 = 7521

C3 = 647 mod 15251 = 8022

• Pixels (2,4)
C1 = 1377 mod 15251 = 4730
C2 = 1347 mod 15251 = 2015
C3 = 1327 mod 15251 = 92
• Pixels (3,2)
C1 = 1397 mod 15251 = 14384
C2 = 1147 mod 15251 = 146
C3 = 957 mod 15251 = 4177

• Pixels (3,4)
C1 = 1427 mod 15251 = 3882
C2 = 1427 mod 15251 = 3882
C3 = 1047 mod 15251 = 7540

• Pixels (4,2)
C1 = 1807 mod 15251 = 13054
C2 = 947 mod 15251 = 3243
C3 = 967 mod 15251 = 6715

• Pixels (4,4)
C1 = 1387 mod 15251 = 8744
C2 = 1287 mod 15251 = 4999
C3 = 1327 mod 15251 = 92

The encryption value above cannot be directly used as a color index value for encryption. Because the value above has a length of 2 bytes, while the maximum color index value is 1 byte (0-255). Most Significant Byte (MSB) and Least Significant Byte (LSB) will be divided into two blocks, each block has 1 byte, namely 1 byte for MSB and 1 byte for LSB. The explanation can be seen in the following image:

Fig. 5: Representation of the division of encryption values into 8 bits/1 byte

The results of the representation above produce MSB and LSB values, which can be seen in the table below:

Table 2: LSB

Piksel	1			2			3			4		
	R	G	B	R	G	B	R	G	B	R	G	B
1	9	139	69	93	52	97	197	42	58	127	222	252
2	39	96	95	175	5	86	236	29	9	122	223	92
3	242	225	29	48	146	81	225	44	146	42	42	116
4	85	242	164	254	171	59	169	0	97	40	135	92

Table 3: MSB

Piksel	1			2			3			4		
	R	G	B	R	G	B	R	G	B	R	G	B
1	26	40	17	47	37	12	8	15	44	27	5	57
2	2	35	43	47	42	31	21	7	26	18	7	0
3	55	19	7	56	0	16	19	10	37	15	15	29
4	17	55	42	50	12	26	50	4	29	34	19	0

In this process, the pixel values that have been obtained from RSA encryption will continue to encrypt the image using a modified Vigenere cipher algorithm. We can use the encryption formula as follows:
Ci = (Pi + (a.Zi – 1 + c)) mod 256
Information :
Ci = Encrypted pixel value
Pi = The original pixel value of the image, which is usually an integer between 0 and 255 (if the image is an 8-bit grayscale or 24-bit truecolor color scale.
Zi = Pixel index value, ranging from 1 to the total number of pixels in the image.
a and c = Are encryption keys. Both are integers.

Table 4: Pixel (1,1)

Red		Green		Blue	
9	26	139	40	69	17

For Pi = 9
- Zi = 1
- a = 3
- c = 10

Ci = (9 + (3*1–1+10)) mod 256
Ci = (9 + 12) mod 256
Ci = 21 mod 256
Ci = 21

Pi = 26
- Zi = 2
- a = 3
- c = 10

Ci = (26 + (3*2–1+10)) mod 256
Ci = (26 + 15) mod 256
Ci = 41 mod 256
Ci = 41

Pi = 139
- Zi = 3
- a = 3
- c = 10

Ci = (139 + (3*3–1+10)) mod 256
Ci = (139 + 18) mod 256
Ci = 157 mod 256
Ci = 157

Pi = 40
- Zi = 4
- a = 3
- c = 10

Ci = (40 + (3*4–1+10)) mod 256
Ci = (40 + 21) mod 256
Ci = 61 mod 256
Ci = 61

Pi = 69
- Zi = 5
- a = 3
- c = 10

Ci = (69 + (3*5–1+10)) mod 256
Ci = (69 + 24) mod 256
Ci = 93 mod 256
Ci = 93

Pi = 17
- Zi = 6
- a = 3
- c = 10

Ci = (17 + (3*6–1+10)) mod 256
Ci = (17 + 27) mod 256
Ci = 44 mod 256
Ci = 44

The following is done on the next number until it produces a number like the table below:

Image Decryption Using RSA Algorithm

At this stage, the LSB and <SB pixel values will be combined. So the results of the combination are as follows:

Table 5: Piksel (1,1)

Red		Green		Blue	
9	26	139	40	69	17
6665		10379		4421	

Table 6: Piksel (1,2)

Red		Green		Blue	
93	47	52	37	97	12
12125		9524		3169	

Table 7: Piksel (1,3)

Red		Green		Blue	
197	8	42	15	58	44
2245		3882		11322	

Table 8: Piksel (1,4)

Red		Green		Blue	
127	27	222	5	252	57
7039		1502		14844	

Table 9: Piksel (2,1)

Red		Green		Blue	
39	2	96	35	95	43
551		9056		11103	

Table 10: Piksel (2,2)

Red		Green		Blue	
175	47	5	42	86	31
12207		10757		8022	

Table 11: Piksel (2,3)

Red		Green		Blue	
236	21	29	7	9	26
5612		1821		6665	

Table 12: Piksel (2,4)

Red		Green		Blue	
122	18	223	7	92	0
4730		2015		92	

Table 13: Piksel (3,1)

Red		Green		Blue	
242	55	225	19	29	7
14322		5089		1821	

Table 14: Piksel (3,2)

Red		Green		Blue	
48	56	146	0	81	16
14384		146		4177	

Table 15: Piksel (3,3)

Red		Green		Blue	
225	19	44	10	146	37
5089		2604		9618	

Table 16: Piksel (3,4)

Red		Green		Blue	
42	15	42	15	116	29
3882		3882		7540	

Table 17: Piksel (4,1)

Red		Green		Blue	
85	17	242	55	164	42
4437		14322		10916	

Table 18: Piksel (4,2)

Red		Green		Blue	
254	50	171	12	59	26
13054		3243		6715	

Table 19: Piksel (4,3)

Red		Green		Blue	
169	50	0	4	97	29
12969		1024		7521	

Table 20: Piksel (4,4)

Red		Green		Blue	
40	34	135	19	92	0
8744		4999		92	

To prove whether the encryption process is correct, the decryption process according to the RSA algorithm must produce the correct value, a secret (private) key is used, namely (n,d) = (15251,2143), with the formula: C1 = m1d mod n.

- Piksel (1,1)
  $C_1 = 6665^{2143} \bmod 15251 = 100$
  $C_2 = 10379^{2143} \bmod 15251 = 82$
  $C_3 = 4421^{2143} \bmod 15251 = 25$

- Piksel (1,2)
  $C_1 = 12125^{2143} \bmod 15251 = 169$
  $C_2 = 9524^{2143} \bmod 15251 = 144$
  $C_3 = 3169^{2143} \bmod 15251 = 113$

- Piksel (1,3)
  $C_1 = 2245^{2143} \bmod 15251 = 177$
  $C_2 = 3882^{2143} \bmod 15251 = 142$
  $C_3 = 11322^{2143} \bmod 15251 = 91$

- Piksel (1,4)
  $C_1 = 7039^{2143} \bmod 15251 = 146$
  $C_2 = 1502^{2143} \bmod 15251 = 92$
  $C_3 = 14844^{2143} \bmod 15251 = 75$

- Piksel (2,1)
  $C_1 = 551^{2143} \bmod 15251 = 72$
  $C_2 = 9056^{2143} \bmod 15251 = 66$
  $C_3 = 11103^{2143} \bmod 15251 = 55$

- Piksel (2,2)
  $C_1 = 12207^{2143} \bmod 15251 = 84$
  $C_2 = 10757^{2143} \bmod 15251 = 74$
  $C_3 = 8022^{2143} \bmod 15251 = 64$

- Piksel (2,3)
  $C_1 = 5612^{2143} \bmod 15251 = 161$
  $C_2 = 1821^{2143} \bmod 15251 = 127$
  $C_3 = 6665^{2143} \bmod 15251 = 100$

- Piksel (2,4)
  $C_1 = 4730^{2143} \bmod 15251 = 137$
  $C_2 = 2015^{2143} \bmod 15251 = 134$
  $C_3 = 92^{2143} \bmod 15251 = 132$

- Piksel (3,1)
  $C_1 = 14322^{2143} \bmod 15251 = 153$
  $C_2 = 5089^{2143} \bmod 15251 = 133$
  $C_3 = 1821^{2143} \bmod 15251 = 127$

- Piksel (3,2)
  $C_1 = 14384^{2143} \bmod 15251 = 139$
  $C_2 = 146^{2143} \bmod 15251 = 114$
  $C_3 = 4177^{2143} \bmod 15251 = 95$

- Piksel (3,3)
  $C_1 = 5089^{2143} \bmod 15251 = 133$
  $C_2 = 2604^{2143} \bmod 15251 = 97$
  $C_3 = 9618^{2143} \bmod 15251 = 76$

- Piksel (3,4)
  $C_1 = 3882^{2143} \bmod 15251 = 142$
  $C_2 = 3882^{2143} \bmod 15251 = 142$
  $C_3 = 7540^{2143} \bmod 15251 = 104$

- Piksel (4,1)
  $C_1 = 4437^{2143} \bmod 15251 = 156$
  $C_2 = 14322^{2143} \bmod 15251 = 153$
  $C_3 = 10916^{2143} \bmod 15251 = 160$

- Piksel (4,2)
  $C_1 = 13054^{2143} \bmod 15251 = 180$
  $C_2 = 3243^{2143} \bmod 15251 = 94$
  $C_3 = 6715^{2143} \bmod 15251 = 96$

- Piksel (4,3)
  $C_1 = 12969^{2143} \bmod 15251 = 140$
  $C_2 = 1024^{2143} \bmod 15251 = 118$
  $C_3 = 7521^{2143} \bmod 15251 = 131$

- Piksel (4,4)
  $C_1 = 8744^{2143} \bmod 15251 = 138$
  $C_2 = 4999^{2143} \bmod 15251 = 128$
  $C_3 = 92^{2143} \bmod 15251 = 132$

From the calculations above, the decryption results can be seen in the table below:

Table 21: Final Calculation Results

Piksel	1			2			3			4		
	R	G	B	R	G	B	R	G	B	R	G	B
1	100	82	25	169	144	113	177	142	91	146	92	75
2	72	66	55	84	74	64	161	127	100	137	134	132
3	153	133	127	139	114	95	133	97	76	142	142	104
4	156	153	160	180	94	96	140	118	131	138	128	132

6. Testing

The main form is the form used to call the encryption and decryption forms. The following is a design of the main form that will be built:

1. Encryption Interface Design
   Image Input Form

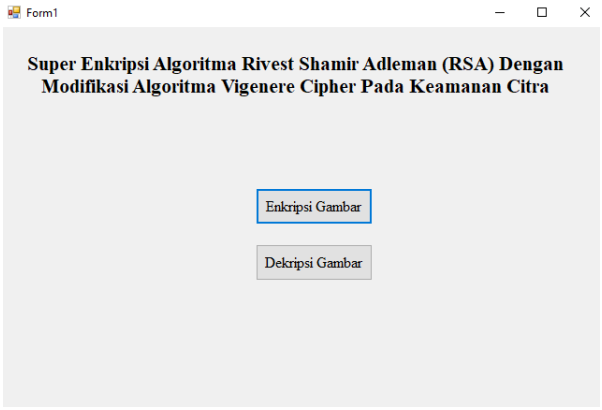

Fig. 6: Home

Page Enkripsi

Fig. 7: Menu Encryption

2. Decryption Interface Design
   Image Input Form

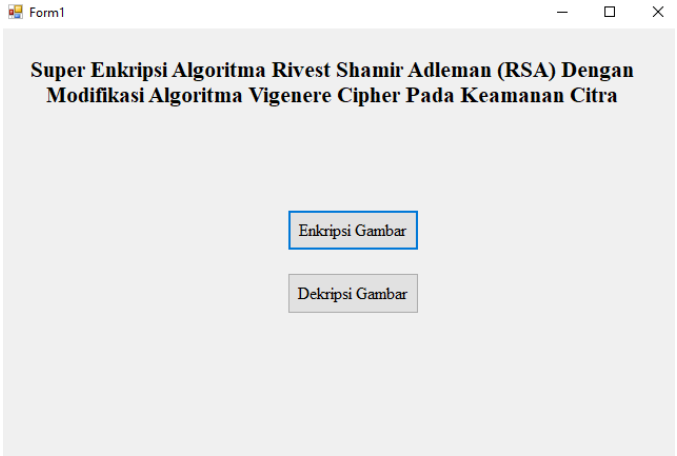

Fig. 7: Input Form

Page Dekripsi

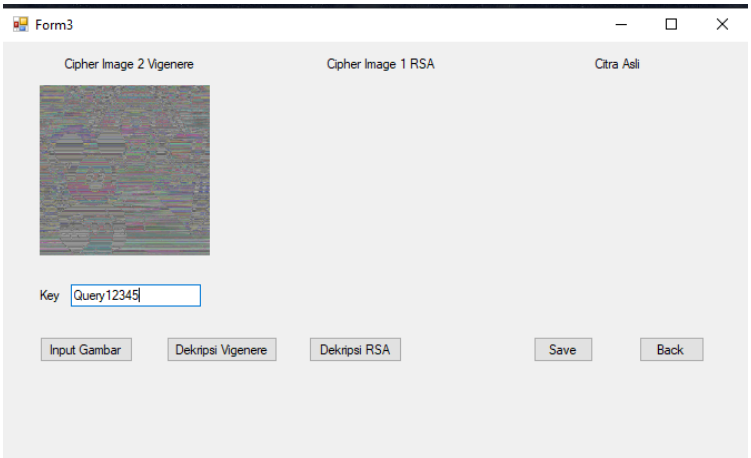

Fig. 7: Dekripsi

7. Conclution

The conclusions of the journal will reflect the results of the analysis and experiments carried out in the research. The following is an example of conclusions that can be drawn from the journal with the title "Rivest Shamir Adleman's Super Encryption Algorithm (RSA) with Modification of the Vigenere Cipher Algorithm for Image Security": In this research, we have succeeded in combining the power of the RSA Algorithm and the flexibility of the Vigenere Cipher Algorithm to improve image security in a digital context. Through a modification of the Vigenere Cipher Algorithm, we apply stronger and more complex encryption to the pixels in the image, using keys generated by the RSA Algorithm. Our experimental results show that this approach effectively increases the security level of images, making them difficult to decrypt without the appropriate key.

We also see that the combination of the RSA Algorithm and the modified Vigenere Cipher provides a fairly high level of security against various attacks, including brute-force attacks and statistical analysis. In addition, this approach can also be applied well in sending images over the network, maintaining the confidentiality and integrity of the image during the transmission process. Nonetheless, there are several areas for improvement and further research. For example, we might want to explore the impact of RSA key size on encryption speed and security, as well as apply compression techniques to reduce the size of the encrypted image. Additionally, it is important to continuously monitor developments in cryptography and computing technology to ensure that proposed solutions remain relevant and secure in the future.

Overall, this research shows that combining the RSA Algorithm with a modified Vigenere Cipher Algorithm can be a powerful and effective solution for improving image security in increasingly complex and risky digital environments. By continuing to develop and refine this approach, we can better meet future data security challenges.

References

[1] Barita, P., Simangunsong, N., & Fitri, K. (2018). *Perancangan Aplikasi Pengamanan Citra Bewarna Dengan Algoritma RSA*. *03*, 99–107.
[2] Dan, E., Citra, D., Marsal, R. U., Arnia, F., & Adriman, R. (2018). *MENGGUNAKAN MODIFIKASI ALGORITMA*. *3*(3), 6–10.
[3] Deskiva, Z. Z., Studi, P., Informatika, T., Digital, C., Password, P., & Aplikasi, P. (2014). *IMPLEMENTASI KRIPTOGRAFI MODERN DENGAN METODE*. 44–49.
[4] https://idmetafora.com/news/read/1894/Mengenal-Apa-itu-Visual-BasicNET-Sejarah-Fitur Kelebihan-dan-Kekurangan.html
[5] https://dasar-pendidikan.blogspot.com/2014/07/pengertian-keistimewaan-dan-sejarah-microsoft-visual-studio-2010.html
[6] Megantara, R. A., Rafrastara, F. A., Studi, P., Informatika, T., Komputer, F. I., Nuswantoro, U. D., Cipher, H., Kolom, T., & Enkripsi, S. (2019). *SUPER ENKRIPSI TEKS KRIPTOGRAFI MENGGUNAKAN ALGORITMA HILL*. 978–979.
[7] Mahesa, K., Sugiantoro, B., & Prayudi, Y. (2019). *Pemanfaatan Metode DNA Kriptografi Dalam Meningkatkan Keamanan Citra Digital*.
[8] Rakhman, A. A., Kurniawan, A. W., Informatika, T., Komputer, F. I., & Nuswantoro, U. D. (2015). *IMPLEMENTASI ALGORITMA KRIPTOGRAFI RIVEST SHAMIR ADLEMAN (RSA) DAN VIGENERE CIPHER PADA GAMBAR BITMAP 8 BIT*. *14*(2), 122–134.
[9] Setyaningsih, E., Iswahyudi, C., Widyastuti, N., & Enkripsi, A. S. (n.d.). *KONSEP SUPER ENKRIPSI UNTUK MENINGKATKAN*.
[10] Syaputra, H., & Herdiyatmoko, H. F. (2012). *Aplikasi enkripsi data pada file teks dengan algoritma rsa (*. *2012*(Semantik), 229–234.
[11] Zainuddin, M. A., Mulyana, D. I., Rivest, R., & Shamir, A. (2016). *PENERAPAN ALGORITMA RSA UNTUK KEAMANAN PESAN INSTAN PADA*. *9*(2), 105–114.