LONTAR KOMPUTER VOL. 13, NO. 2 AUGUST 2022 p-ISSN 2088-1541
DOI : 10.24843/LKJITI.2022.v13.i02.p02 e-ISSN 2541-5832
Accredited Sinta 2 by RISTEKDIKTI Decree No. 158/E/KPT/2021

Vigenère Cipher Algorithm Optimization for Digital Image Security using SHA512

Imam Riadi[a1], Abdul Fadlil[b2], Fahmi Auliya Tsani[c3]

[a]Department of Information Systems, Universitas Ahmad Dahlan
Yogyakarta, Indonesia, 55166
[1]imam.riadi@is.uad.ac.id

[b]Department of Electrical Engineering, Universitas Ahmad Dahlan
Yogyakarta, Indonesia, 55166
[2]fadlil@mti.uad.ac.id

[c]Department of Informatics Engineering, Universitas Ahmad Dahlan
Yogyakarta, Indonesia, 55166
[3]fahmi1807048017@webmail.uad.ac.id (Corresponding author)

Abstract

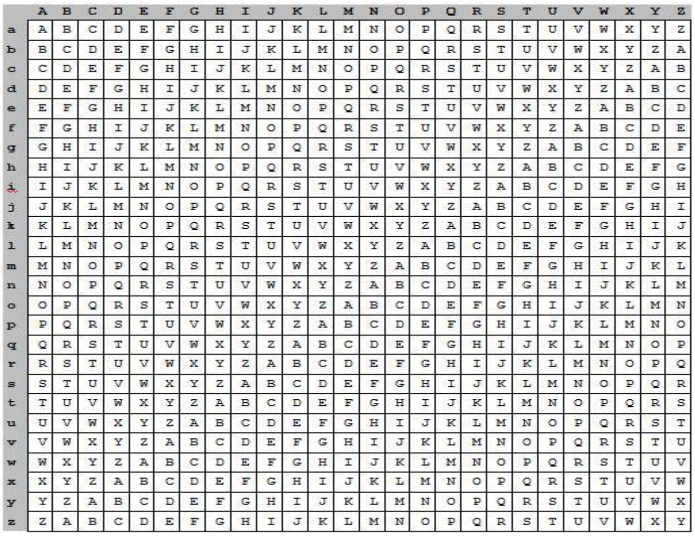
One of the popular cryptographic algorithms is the Vigenère Cipher. This algorithm is included in classical cryptographic algorithms, so its capabilities are limited to text-type data. Through this research, this research try to modify the Vigenère Cipher so that it can be used on digital image media. The improvement is performed using ASCII code as a Vigenère table and the key generated by the SHA512 hash technique with salt. The encryption and decryption process was carried out on ten jpg and ten png files and showed a 100% success rate. Speed and memory consumption tests on the encryption process by comparing it with the AES algorithm show that AES excels in speed with 409,467 Mb/s while Vigenère wins in memory consumption by utilizing only 5,0007 Kb for every Kilobytes of the processed digital image file.

Keywords: *Cryptography, Vigenère Cipher, SHA512, base64 Encoding, AES, Data Security*.

1. Introduction

Human activities lately are related to communication, data, and information [1]. Security is one crucial aspect that information or data should be achieved. The security issue is important because it relates to sensitive data by protecting it from unauthorized access, alteration, or deletion [2]. Data security has several aspects, including authentication, confidentiality/privacy, integrity, and non-repudiation. Some of these points can be solved using cryptographic techniques [3]. Cryptography is a method used to ensure data security through an encryption process so that the data becomes difficult to read or open by someone who is not authorized because they do not have the key to decrypt [4]. In other words, cryptography can change the contents of a data into other random data [5].

Cryptography is broadly classified into two types: classical cryptography and modern cryptography. One of the popular classical cryptography algorithms is the Vigenère Cipher. This algorithm implements a substitution technique, an encoding process, by changing the data contents based on the key used, so it becomes unreadable [6]. The Vigenère Cipher uses Vigenère squares in the encryption and decryption process, thus making this algorithm easily understood and implemented [3]. Figure 1 illustrates a Vigenère square.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
a	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
b	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
c	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
d	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
e	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
f	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
g	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
h	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
i	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
j	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
k	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
l	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
m	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
n	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
o	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
p	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
r	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
s	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
t	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
u	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
v	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
w	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
x	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Figure 1. Vigenère Cipher Tabula Recta Example

Cryptography has not only been used for text-based data but is also applicable to other kinds of data like images, videos, and sounds [7]. A cryptographic algorithm can be categorized as good if it maintains the secret aspect of an encrypted message and cannot be read by someone unauthorized to access the data [8].

Unsafe data or systems will undoubtedly have a harmful impact [8]. One method that can be used to maintain the confidentiality of data is to convert it into meaningless encrypted data. This process can be commonly referred to as cryptography [9]. Cryptography is described as a science and art of securing messages as they travel from a source to a destination. This process consists of three primary functions, among others [10]:

a. Encryption is the process of converting the original message into codes that are difficult or even incomprehensible.
b. Decryption is the reverse encryption process, changing the encrypted message into the original message.
c. Key is a set of parameters used in the encryption and decryption processes.

Cryptography has several purposes on several security aspects as follows [3]:

a. Confidentiality aims to prevent messages from being read by unauthorized parties.
b. Data integrity aims to get guarantee that the message is still original/intact and not manipulated during delivery.
c. Authentication aims to identify the truth of the parties communicating and the message's truth.
d. Non-repudiation aims to avoid denial by the communicating parties.

Sinaga et al. combined the Vigenère Cipher algorithm with column transposition to build a strong encryption technique applied to digital image media. This study used a key generated by a random function that contains numbers ranging from 0 to 255 [7]. Gunadhi and Sudrajat, in their research, implemented a modified Vigenère Cipher to secure the patient's medical record data, making the patient's medical record data safer from attacks by cryptanalysts [9]. Mandal and Deepti conducted another study by implementing a multi-level encryption scheme. The method uses a key with the same character length as the plain text to produce the first cipher text. It doesn't stop here; the first cipher text is then encrypted again with the same key as the first cipher text to produce the second cipher text. In conclusion, compared to several other cryptographic algorithms (AES, Blowfish, and RC5), this method has difficult results for cryptanalysts to solve and has lower computational complexity, so it is suitable for lightweight applications and has limited resources [10].

Soofi et al. tried slightly modifying the Vigenère square table by changing the order of each character and adding an "&" character instead of a white space character. This method produces the Vigenère algorithm, which is more robust against attacks by the Kasiski and Friedman methods [11]. Some research on Vigenère Cipher has been conducted by combining it with other approaches, such as the Goldbach Codes compression technique. The merger results produce cipher text that is difficult to predict even using the Kasiski method attack because the resulting set of characters is different from the characters used in plain text [12].

The other combining technique uses encryption, key generation, and steganography. The encryption used is the Vigenère Cipher, modified using a Vigenère square composition according to the arrangement of letters, numbers, and symbols on the keyboard. Meanwhile, the key used is generated through a chaos function. The next process is to compress the encrypted data using Dictionary Based Compression. As the last step, the compressed data is hidden into a digital image using steganography with the Least Significant Bit (LSB) method [13]. Saputra et al. implement the Vigenère Cipher by utilizing a 5 x 5-pixel grayscale image as a key. This grayscale image key is transformed into ASCII characters, resulting in a character arrangement that can be processed into the Vigenère Cipher [14].

Another research that aims to compare the avalanche effect has been carried out by expanding the range of characters that can be accommodated to 128 pieces according to the number of standard ASCII characters and rotating the square matrix. The process implementation produces an avalanche effect value of around 45% to 49% [15]. In their research on the Vigenère Cipher implementation, Fadlil et al. developed a unique technique, merging Artificial Neural Networks (ANN) with Vigenère Ciphers. This study uses ANN as a key generator by entering the parameters of hidden neurons (K), input neurons (N), and weights (L) so that random characters are generated that can be used in the encryption and decryption process. Through this approach, it is claimed to have less possibility of generating the same key even if the same parameter value is entered repeatedly [16].

Hernawandra et al. use digital images in their research to secure data in the form of text by first carrying out the encryption process using Vigenère Cipher and substitution. The encrypted text produced by the encryption procedure is then hidden in digital image media using 4-bit LSB steganography. The output of this research is an application that runs on the Android platform. This research concludes that the built application can secure messages through the 4-bit LSB steganography method combined with substitution encryption and Vigenre Cipher and has an average avalanche effect of 12.77% [17].

There aren't many research projects on implementing Vigenère Cipher on digital image media. One is by substituting the color code for each pixel based on the key entered. As a result, another image with a random color is formed [18]. Another research is to do the encryption process twice using Vigenère Cipher and adopt an expansion key using the RC6 algorithm on text media. This study compares data size before and after encryption (avalanche effect) in several scenarios, such as using the standard Vigenère Cipher, merging with RC6 expansion keys, and others [19]. Riadi conducted similar research by first transforming a digital image with a base64 encoding method with a radix-64 character arrangement used as a Vigenère square. The encryption process was successful and took less than 0.2 seconds, and the decryption took less than 0.19 seconds on ten digital image samples [20].

Digital images are one of the most popular media types used to communicate online and in person [21]. Therefore, this study will use the Vigenère Cipher algorithm to develop digital image security. Previous research has not highlighted the usage of SHA512 as a key generation mechanism in conjunction with salt to prevent key attacks. This research will combine the SHA512 hash method as a key generator with ASCII code as a Tabula Recta. They proved the research results' validity by calculating the Peak Signal-to-Noise Ration (PSNR) between the original and decrypted files. This study will also present the time required for the encryption and decryption processes. Panda research shows that AES exceeds DES, RSA, and Blowfish in terms of encryption and decryption speed [22], so it was chosen as a benchmark cryptographic method in this research.

Furthermore, AES is one of the most extensively utilized modern cryptographic algorithms [23]. The test results will be compared with the AES algorithm regarding processing speed and the amount of resources or memory used. This research is expected to provide insight into the importance of securing data or files, especially digital image data.

2. Research Methods

This research implements the modified Vigenère Cipher algorithm and produces an output in the form of a console-based application. Modifications are made in the form of widening the character range according to the characters used by the ASCII code.

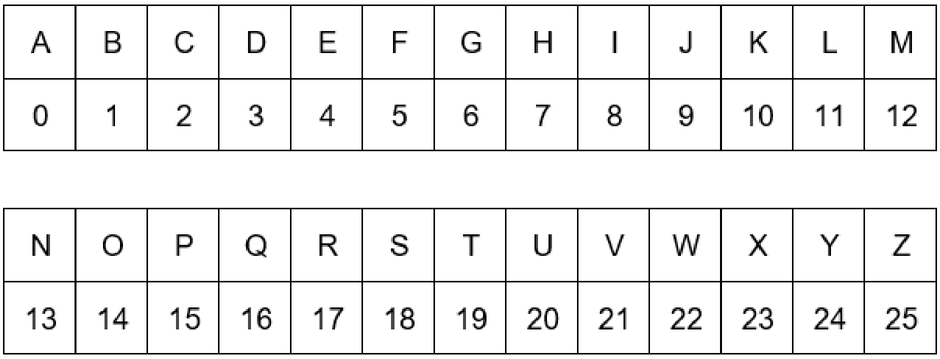
2.1. Vigenère Cipher

The Vigenère Cipher is a further development of the Caesar Cipher and is included in the category of polyalphabetic substitution cipher [24]. Vigenère Cipher can be performed in two ways: manually using a Vigenère square (tabula recta) as shown in Figure 1 or by number substitution (mathematical). Under standard conditions, encryption and decryption using the Vigenère Cipher can be stated as (1), while decryption can be written as (2).

$$C_i = (P_i + K_i) \bmod 26 \quad (1)$$

$$P_i = (C_i - K_i) \bmod 26 \quad (2)$$

Here is an example of using the Vigenère Cipher based on the alphabetical arrangement, as shown in Figure 2.



A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Figure 2. Index of alphabetical letters

Plaintext : `INFORMATIONSECURITY`
Key : `JOURNALJOURNALJOURN`
Ciphertext : `RBZFEMLCWIEEENDFWKV`

2.2. SHA512

One of the hash functions is SHA (Secure Hash Algorithm). The NSA (National Security Agency) created this algorithm, which was then published by the NIST (National Institute of Standards and Technology) [25]. SHA has now reached the third generation called SHA2, which consists of SHA224, SHA256, SHA384, and SHA512 [26]. SHA has a one-way hash property as a hash function, which implies that it generates a hash result that cannot be decrypted. Another characteristic is that it is very sensitive to changes even though they are minor. Any changes to the input message will give different results [20] [24] [29].

Example
Plaintext : `PLAY WITH CRYPTOGRAPHY`
Hash Value : `bfbf666f835054cb cf77d2e4eb2e0495 0e166791401397c1`
`930cc2a04e9f154b d723f98c0f48eb31 cfc852d043a222dc`
`56cdb964166b0ab6 05e90c97631459c8`

2.3. ASCII Code

ASCII code is a set of codes that bridge the interaction between humans and computers. These codes are 8 bits long, ranging from 00000000 to 11111111. As a result, there are 256-character combinations ranging from 0 to 255 [30]. Text is usually presented by ASCII codes ranging from

p-ISSN 2088-1541
e-ISSN 2541-5832

0 to 127, whereas graphic manipulation is typically represented by sequences ranging from 128 to 255 [31].

2.4. Encryption Process

The process of transforming original data into encrypted data is known as encryption [3]. The encryption process is presented as a flowchart, as shown in Figure 4.

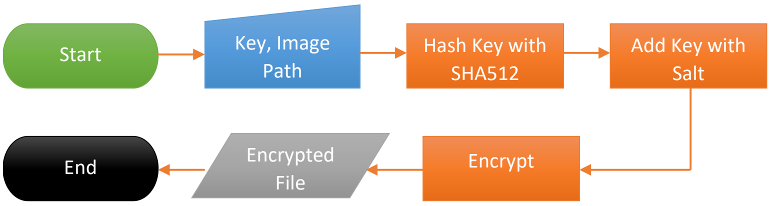


Figure 4. The encryption process

The first step to starting the encryption process is to enter the key and the path to the location of the digital image file. Furthermore, the salt inputted key will be appended both in front of and behind the key, and the key will be converted to a SHA512 hash format. Then the application will perform the encryption process by applying the Vigenère Cipher, which uses a Vigenère table arrangement based on ASCII code and the hashed key. The encryption process is complete until that stage, and the results are issued as a file with the *.vig extension.

2.5. Decryption Process

Decryption is restoring encrypted data into original data [3]. The decryption process is presented in the form of a flowchart, as shown in Figure 5

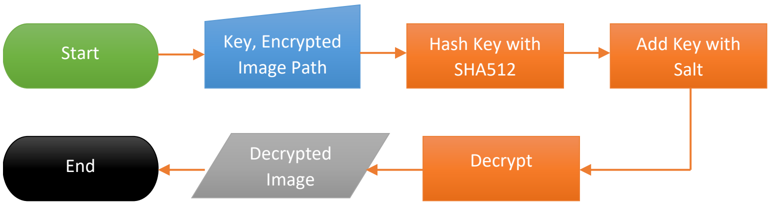


Figure 5. Decryption flowchart

The first step to decrypting is the user entering the encrypted file's key and path. First, the application will add salt both on the front and rear of the key, and the key will be converted to a SHA512 hash format. Then the application will perform the decryption and encryption process, namely applying the Vigenère Cipher using a Vigenère table arrangement based on ASCII code and the hashed key. Next, the application will return the encrypted digital image file into a digital image file according to the original format.

3. Result and Discussion

3.1. Vigenère Cipher Modification

This research uses a modified Vigenère Cipher. The modification was limited to widening the support characters from the original 26 alphabetic to 256 characters in the ASCII table. As a result, the formula for performing the encryption process has altered, as written in (3), whereas the formula for decryption is written in (4).

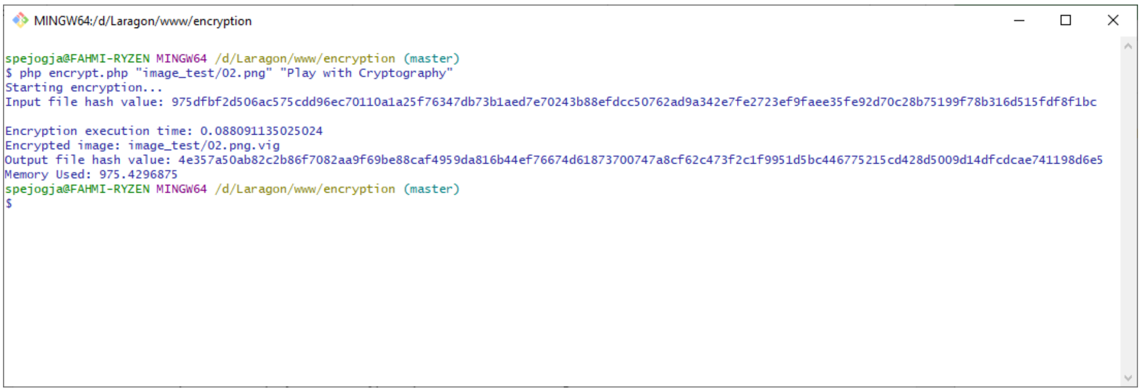
$$C_i = (P_i + K_i) \bmod 256 \quad (3)$$

$$P_i = (C_i - K_i) \bmod 256 \quad (4)$$

The key generation procedure was also changed, with the entered key converted into a 128-character hexadecimal character arrangement using the SHA512 hash method and the insertion of salt (extra characters). Adding salt to the key prevents the attacker from guessing the key [32]. Brute-Force Attack, Rainbow Table Attack, Dictionary Attack, and Online Cracking Attack are examples of key guessing techniques [33]. Although adding salt does not ensure perfect key protection, it can make the computational process for key breaking infeasible [34].

3.2. Implementation

This research produces output from a console-based application built using the PHP programming language version 7.2.19. This application consists of two main files, namely encrypt.php and decrypt.php. While the main core for the encryption-decryption process uses only one file, namely VigenereCipher.php, which is in the source folder. An example of the display in the encryption process and the format of the command that is executed is shown in Figure 6.



```
spejogja@FAHMI-RYZEN MINGW64 /d/Laragon/www/encryption (master)
$ php encrypt.php "image_test/02.png" "Play with Cryptography"
Starting encryption...
Input file hash value: 975dfbf2d506ac575cdd96ec70110a1a25f76347db73b1aed7e70243b88efdcc50762ad9a342e7fe2723ef9faee35fe92d70c28b75199f78b316d515fdf8f1bc

Encryption execution time: 0.088091135025024
Encrypted image: image_test/02.png.vig
Output file hash value: 4e357a50ab82c2b86f7082aa9f69be88caf4959da816b44ef76674d61873700747a8cf62c473f2c1f9951d5bc446775215cd428d5009d14dfcdcae741198d6e5
Memory Used: 975.4296875
spejogja@FAHMI-RYZEN MINGW64 /d/Laragon/www/encryption (master)
$
```

Figure 6. Encryption process

To start this application, two arguments must be filled out, one for each encryption and decryption operation. The first parameter is a string that indicates the file's location to be encrypted or decrypted. The second parameter is the key used. An example of the encryption process using a digital image file in Figure 7 is shown below.

Figure 7. Example of the original file

Encoded File	‰PNG IHDR ——————————ö,—————————— &⍰ € IDATxÚìÝMˆ\÷™?úÚBg&1úêÂþFÐ0„Ñ$‹4h$¢Q-z#'ƒ Ôàð!&)FXöˆx ‹™ŽˆKà®EAW ®ÀP ƒ74W‹ …
Key	sha512(“Th15 Iz Pr3-54lT” + “Play with Cryptography” + “tH1z i5 p0zT-sALt”)
Hashed Key	11bc66a87804925de939d4d2f682cca7db089df65934d223d8b97feec68e734c48e8b931a47f0a4ab6e888e3bed34ab9260e945bbcfd043b0edb40494d064c37

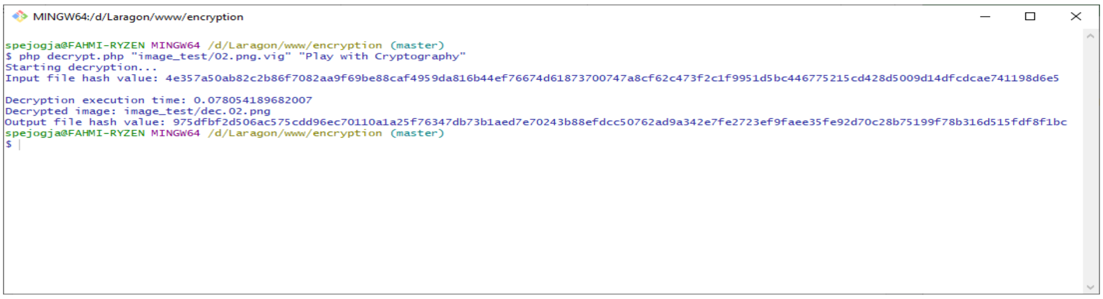
Cipher

```
º�°ªC@{B780A,zy¶e99?d4g(n<82c�d]ób0¸9-ªw‰±

A�º�[Ñ¡3~k€¥�D‹hK.MöPc~2i·…¿kÎFoX————————————————————————————y‡
{V>²ˆZå…8$xƒ_[|ˆ[Á¬?íûñ,ì{JhÞª¥¹?Þô‰?çgHhº¾B

…
```

The result of the encryption process is a file with a .vig extension format, while the result of the decryption process is a file with an extension format according to the initial format of the file. An example of the display in the decryption process is shown in Figure 7.



```
spejogja@FAHMI-RYZEN MINGW64 /d/Laragon/www/encryption (master)
$ php decrypt.php "image_test/02.png.vig" "Play with Cryptography"
Starting decryption...
Input file hash value: 4e357a50ab82c2b86f7082aa9f69be88caf4959da816b44ef76674d61873700747a8cf62c473f2c1f9951d5bc446775215cd428d5009d14dfcdcae741198d6e5

Decryption execution time: 0.078054189682007
Decrypted image: image_test/dec.02.png
Output file hash value: 975dfbf2d506ac575cdd96ec70110a1a25f76347db73b1aed7e70243b88efdcc50762ad9a342e7fe2723ef9faee35fe92d70c28b75199f78b316d515fdf8f1bc
spejogja@FAHMI-RYZEN MINGW64 /d/Laragon/www/encryption (master)
$ 
```

Figure 7. Decryption process

3.3. Testing

The test was carried out using hardware specifications as shown in Table 1 and running on the Windows 10 Pro 64-bit operating system.

Table 1. Hardware specification for testing

Hardware	Specification
Processor	AMD Ryzen 7 4700U with Radeon Graphics (8 CPUs), ~2.0GHz
Memory/RAM	1228 MB RAM
Harddisk	1 TB
SSD	512 MB

The validity of the research results was tested by calculating the Peak Signal-to-Noise Ratio (PSNR) value between the original and the decrypted file. The test is declared valid if the PSNR value shows an infinite value (zero errors) [35], with the results shown in Table 2.

Table 2. Validity Testing

No	Compared File	PSNR	Compared File	PSNR
1	01.jpg – 01.dec.jpg	Infinite	01.png – 01.dec.png	Infinite
2	02.jpg – 02.dec.jpg	Infinite	02.png – 02.dec.png	Infinite
3	03.jpg – 03.dec.jpg	Infinite	03.png – 03.dec.png	Infinite
4	04.jpg – 04.dec.jpg	Infinite	04.png – 04.dec.png	Infinite
5	05.jpg – 05.dec.jpg	Infinite	05.png – 05.dec.png	Infinite
6	06.jpg – 06.dec.jpg	Infinite	06.png – 06.dec.png	Infinite
7	07.jpg – 07.dec.jpg	Infinite	07.png – 07.dec.png	Infinite
8	08.jpg – 08.dec.jpg	Infinite	08.png – 08.dec.png	Infinite
9	09.jpg – 09.dec.jpg	Infinite	09.png – 09.dec.png	Infinite
10	10.jpg – 10.dec.jpg	Infinite	10.png – 10.dec.png	Infinite

Table 2 shows that all files have an infinite value on PSNR testing, indicating that encrypted files were successfully decrypted into the original file without any changes. Table 3 shows the time needed for the encryption and decryption processes in ten png files.

Table 3. Required Time for Encryption & Decryption Process for png File

No.	File Name	File Size (KB)	Encrypt Time (s)	Decrypt Time (s)

p-ISSN 2088-1541
e-ISSN 2541-5832

1	01.png	66	0,012	0,012
2	02.png	99	0,017	0,018
3	03.png	187	0,034	0,034
4	04.png	212	0,039	0,038
5	05.png	351	0,064	0,065
6	06.png	370	0,067	0,067
7	07.png	393	0,071	0,071
8	08.png	705	0,129	0,127
9	09.png	1.709	0,303	0,304
10	10.png	1.789	0,316	0,316

The chart shown in Figure 9 is based on Table 3 data. The chart shows almost no difference between the encryption and decryption process duration for the png file. The chart also shows that the larger the image size, the greater the time needed for encryption and decryption processes.

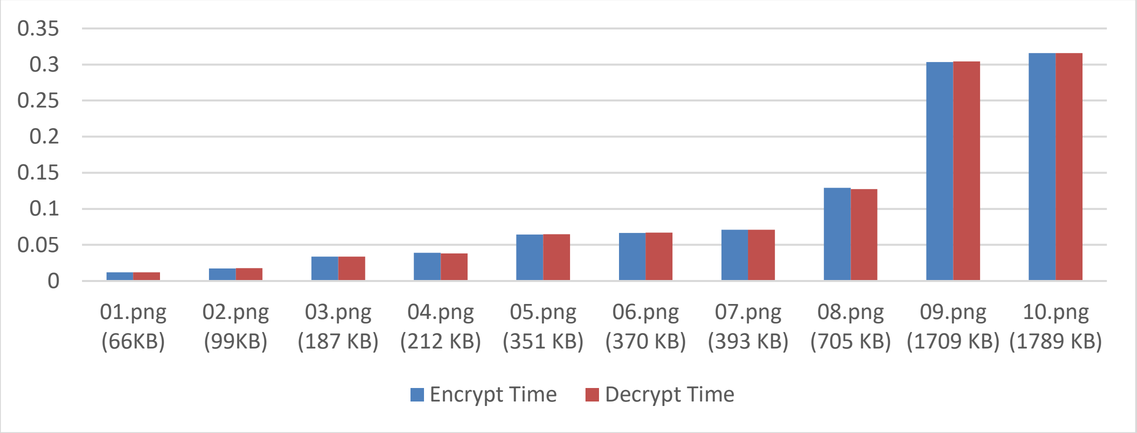


Figure 9. Encryption-Decryption Duration for png Files

Table 4 shows the time needed for the encryption and decryption processes in ten jpg files.

Table 4. Required Time for Encryption & Decryption Process for a jpg file

No.	File Name	File Size (KB)	Encrypt Time (s)	Decrypt Time (s)
1	01.jpg	70	0,013	0,013
2	02.jpg	127	0,023	0,023
3	03.jpg	254	0,045	0,046
4	04.jpg	613	0,113	0,116
5	05.jpg	796	0,145	0,142
6	06.jpg	815	0,150	0,145
7	07.jpg	1.850	0,326	0,327
8	08.jpg	2.475	0,436	0,438
9	09.jpg	5.630	1,015	1,018
10	10.jpg	10.949	2,001	2,005

As in Table 3, Table 4 also shows that the larger the file size, the greater the time required for encryption and decryption processes. The chart is shown in Figure 10. Table 4 and Figure 10 also show no difference between the encryption and decryption process duration for the jpg file.

p-ISSN 2088-1541
e-ISSN 2541-5832

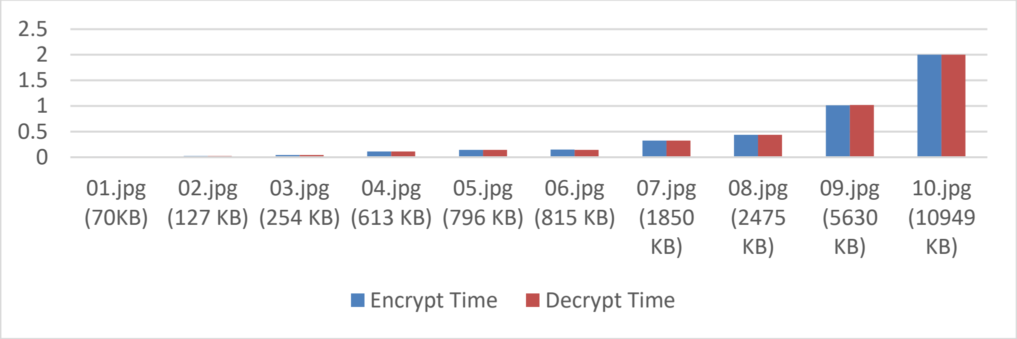


Figure 10. Encryption & Decryption Time

Another test was conducted by comparing the speed of the encryption-decryption process and the memory resources used with one of the modern cryptographic algorithms, the Advance Encryption Standard (AES). The AES algorithm is categorized as symmetric cryptography and uses a block cipher scheme [29]. AES is a new security standard to replace the Data Encryption Standard (DES). This algorithm uses a symmetric key and has been used by the United States government [36]. Table 5 shows the comparison test results for speed for the encryption process between AES and Vigenère Cipher.

Table 5. Speed Comparison Between AES and Vigenère Cipher

No.	File Name	File Size (Kb)	AES Encrypt Time (s)	Vigenère Encrypt Time (s)
1	01.jpg	70	0,00019	0,013
2	02.jpg	127	0,00028	0,023
3	03.jpg	254	0,00056	0,045
4	04.jpg	613	0,00119	0,113
5	05.jpg	796	0,00174	0,145
6	06.jpg	815	0,00176	0,150
7	07.jpg	1.850	0,00484	0,326
8	08.jpg	2.475	0,00620	0,436
9	09.jpg	5.630	0,01603	1,015
10	10.jpg	10.949	0,04331	2,001

Average speed both for the AES and Vigenère Cipher can be calculated using the data from Tabel 5. The AES has an average speed of 409,467 Mb/s, while the Vigenère Cipher has only 5,528 Mb/s. The comparison chart of the encryption process speed test between the AES algorithm and the Vigenère Cipher is shown in Figure 11. Figure 11 shows that The AES algorithm is significantly faster than the Vigenère Cipher algorithm. In fact, a file size of 10,949 KB only takes under one second, much faster than Vigenère Cipher, which takes about two seconds.

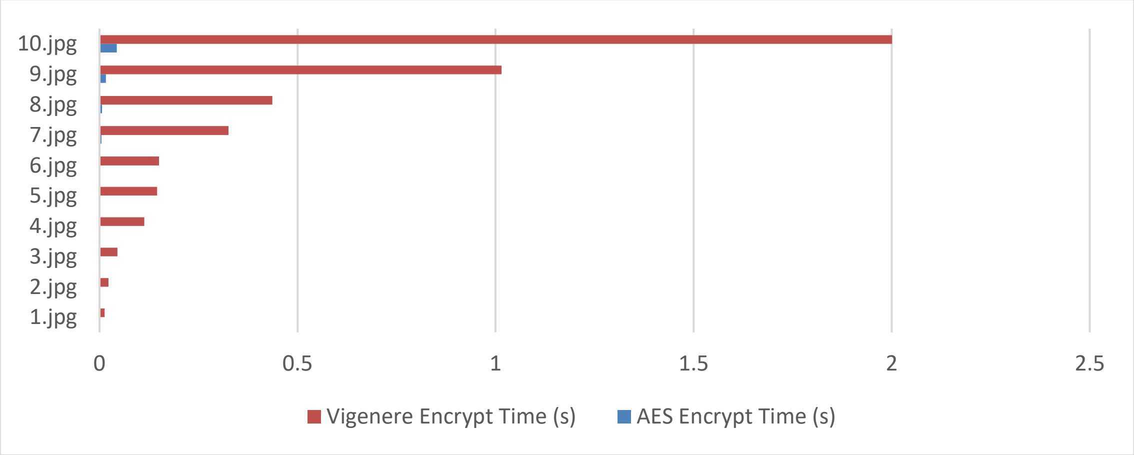


Figure 11. Encryption Speed Comparison Between AES and Vigenère Cipher

Table 6 shows the comparison test results for memory consumption/usage for the encryption

p-ISSN 2088-1541
e-ISSN 2541-5832

process between AES and Vigenère Cipher. Average memory consumption for the AES and Vigenère Cipher was calculated using Table 6 data. The AES has an average memory consumption of about 7,927 Kb for every Kilobytes of the processed file, while the Vigenère Cipher only 5,0007 Kb for every Kilobytes of the processed file.

Table 6. Memory Consumption Comparison Between AES and Vigenère Cipher

No.	File Name	File Size (Kb)	AES Resource Usage (Kb)	Vigenère Resource Usage (Kb)
1	01.jpg	70	837,531	694,633
2	02.jpg	127	1.177,531	862,633
3	03.jpg	254	1.937,531	1.246,633
4	04.jpg	613	4.097,531	2.326,633
5	05.jpg	796	5.189,531	2.866,633
6	06.jpg	815	5.301,531	2.926,633
7	07.jpg	1.850	16.397,602	6.034,633
8	08.jpg	2.475	20.885,648	12.766,703
9	09.jpg	5.630	37.269,648	18.910,703
10	10.jpg	10.949	74.133,648	37.342,703

The comparison diagram of memory consumption for the encryption process between the AES algorithm and Vigenère Cipher can be seen in Figure 12. Figure 12 shows that the Vigenère Cipher outperforms the AES in memory consumption.

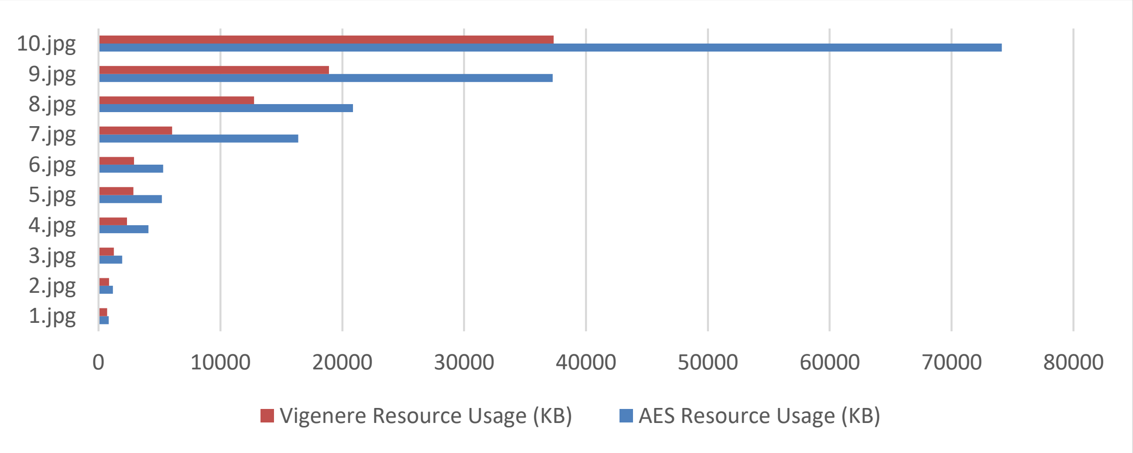


Figure 12. Memory Consumption Comparison Between AES and Vigenère Cipher

4. Conclusion

The Vigenère Cipher algorithm can be used to secure digital images by combining ASCII code and SHA512 as a key generator. Tests conducted on ten png files and ten jpg files showed that the larger the file size, the more time it takes for encryption and decryption.

The comparison test of speed and memory consumed in the encryption process between the AES algorithm and Vigenère Cipher shows that AES is much faster than Vigenère Cipher, even for large image files. However, the Vigenère Cipher managed to win in terms of memory consumption.

References

[1] I. Riadi, R. Umar, and I. M. Nasrulloh, "Experimental Investigation of Frozen Solid State Drive on Digital Evidence with Static Forensic Methods," *Lontar Komputer*, vol. 9, no. 3, pp. 169–181, 2018.

[2] M. Awad, M. Ali, M. Takruri, and S. Ismail, "Security Vulnerabilities Related to Web-based Data," *Telkomnika (Telecommunication, Computing, Electronics and Control)*, vol. 17, no. 2, pp. 852–856, 2019.

[3] R. Munir, *Kriptografi*. Bandung: Informatika, 2006.

[4] Hermansa, R. Umar, and A. Yudhana, “Analisis Sistem Keamanan Teknik Kriptografi dan Steganografi Pada Citra Digital (Bitmap),” in *Seminar Nasional Teknologi Fakultas Teknik Universitas Krisnadwipayana*, 2019, pp. 520–528.

[5] A. Fadlil, I. Riadi, and A. Nugrahantoro, "Data Security for School Service Top-Up Transactions Based on AES Combination Blockchain Technology," *Lontar Komputer*, vol. 11, no. 3, pp. 155–166, 2020.

[6] D. R. I. M. Setiadi, C. Jatmoko, E. H. Rachmawanto, and C. A. Sari, “Kombinasi Cipher Subtitusi (Beaufort Dan Vigenere) pada Citra Digital,” in *Seminar Nasional Multi Disiplin Ilmu*, 2018, pp. 52–57.

[7] D. Sinaga, C. Umam, D. R. I. M. Setiadi, and E. H. Rachmawanto, “Teknik Super Enkripsi Menggunakan Transposisi Kolom Berbasis Vigenere Cipher pada Citra Digital,” *Dinamika Rekayasa*, vol. 14, no. 1, pp. 57–64, 2018.

[8] F. Anwar, E. H. Rachmawanto, C. A. Sari, and D. R. I. M. Setiadi, "StegoCrypt Scheme using LSB-AES Base64," in *International Conference on Information and Communications Technology (ICOIACT)*, 2019, no. July, pp. 85–90.

[9] E. Gunadhi and A. Sudrajat, “Pengamanan Data Rekam Medis Pasien Menggunakan Kriptografi Vigenere Cipher,” *Jurnal Algoritma*, vol. 13, no. 2, pp. 295–301, 2016.

[10] S. K. Mandal and A. R. Deepti, "A Cryptosystem Based On Vigenere Cipher By Using Mulitlevel Encryption Scheme," *International Journal of Computer Science and Information Technologies*, vol. 7, no. 4, pp. 2096–2099, 2016.

[11] A. A. Soofi, I. Riaz, and U. Rasheed, "An Enhanced Vigenere Cipher For Data Security," *International Journal of Scientific & Technology Research*, vol. 5, no. 03, pp. 141–145, 2016.

[12] S. D. Nasution, G. L. Ginting, M. Syahrizal, and R. Rahim, "Data Security Using Vigenere Cipher and Goldbach Codes Algorithm," *International Journal of Engineering Research & Technology*, vol. 6, no. 1, pp. 360–363, 2017.

[13] Rojali, A. G. Salman, and George, "Website-Based PNG Image Steganography Using The Modified Vigenere Cipher, Least Significant Bit, And Dictionary Based Compression Methods," in *International Conference on Mathematics: Pure, Applied and Computation*, 2016.

[14] I. Saputra, N. A. Hasibuan, M. Aan, and R. Rahim, "Vigenere Cipher Algorithm with Grayscale Image Key Generator for Secure Text File," *International Journal of Engineering Research & Technology*, vol. 6, no. 1, pp. 266–269, 2017.

[15] Rihartanto, R. K. Ningsih, A. F. O. Gaffar, and D. S. B. Utomo, "Implementation of Vigenere Cipher 128 and Square Rotation in Securing Text Messages," *Jurnal Teknologi dan Sistem Komputer*, vol. 8, no. 3, pp. 201–209, 2020.

[16] A. Fadlil, I. Riadi, and A. Nugrahantoro, “Kombinasi Sinkronisasi Jaringan Syaraf Tiruan dan Vigenere Cipher untuk Optimasi Keamanan Informasi,” *Digital Zone: Jurnal Teknologi Informasi dan Komunikasi*, vol. 11, no. 1, pp. 81–95, 2020.

[17] P. Hernawandra, S. Supriyadi, and U. T. Lenggana, “Aplikasi Steganografi Menggunakan LSB 4 Bit Sisipan dengan Kombinasi Algoritme Substitusi dan Vigenere Berbasis Android,” *Jurnal Teknologi dan Sistem Komputer*, vol. 6, no. 2, pp. 44–50, 2018.

[18] Y. A. Gerhana, E. Insanudin, U. Syarifudin, and M. R. Zulmi, "Design of Digital Image Application using Vigenere Cipher Algorithm," in *International Conference on Cyber and IT Service Management*, 2016, pp. 1–5.

[19] A. Subandi, M. S. Lydia, R. W. Sembiring, M. Zarlis, and S. Efendi, "Vigenere Cipher Algorithm Modification by Adopting RC6 Key Expansion and Double Encryption Process," in *2nd Nommensen International Conference on Technology and Engineering*, 2018, pp. 1–6.

[20] I. Riadi, A. Fadlil, and F. A. Tsani, “Pengamanan Citra Digital Berbasis Kriptografi Menggunakan Algoritma Vigenere Cipher,” *JISKA (Jurnal Informatika Sunan Kalijaga)*, vol. 7, no. 1, pp. 33–45, 2022.

[21] T. Zebua and E. Ndruru, “Pengamanan Citra Digital Berdasarkan Modifikasi Algoritma RC4,” *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 4, no. 4, pp. 275–282, 2017.

[22] M. Panda, "Performance Analysis of Encryption Algorithms for Security," in *International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, 2016, pp. 278–284.

[23] Z. El Mrabet, N. Kaabouch, H. El Ghazi, and H. El Ghazi, "Cyber-Security in Smart Grid: Survey and Challenges," *Computers and Electrical Engineering*, vol. 67, pp. 469–482, 2018.

[24] H. E. Prabowo and A. Hangga, “Enkripsi Data Berupa Teks Menggunakan Metode Modifikasi Vigenere Cipher,” in *Seminar Nasional Aplikasi Teknologi Informasi (SNATi)*, 2015, pp. 1–4.

[25] L. G. R. Semesta and S. Amini, “Implementasi One Time Password Dengan Algoritma Secure Hash Algorithm 512 (SHA-512),” *Skanika*, vol. 1, no. 3, pp. 1206–1211, 2018.

[26] M. Sumagita and I. Riadi, "Analysis of Secure Hash Algorithm (SHA) 512 for Encryption Process on Web Based Application," *International Journal of Cyber-Security and Digital Forensics*, vol. 7, no. 4, pp. 373–381, 2018.

[27] R. Fitriyanto, A. Yudhana, and S. Sunardi, “Manajemen jpeg/exif File Fingerprint dengan Algoritma Brute Force String Matching dan Hash Function SHA256,” *Register: Jurnal Ilmiah Teknologi Sistem Informasi*, vol. 5, no. 2, pp. 128–139, 2019.

[28] R. Fitriyanto, A. Yudhana, and S. Sunardi, "Implementation SHA512 Hash Function And Boyer-Moore String Matching Algorithm For Jpeg/exif Message Digest Compilation," *Jurnal Online Informatika*, vol. 4, no. 1, p. 16, 2019.

[29] S. Zhou, P. He, and N. Kasabov, "A Dynamic DNA Color Image Encryption Method Based on SHA-512," *Entropy*, vol. 22, no. 1091, pp. 1–23, 2020.

[30] M. A. Helmiawan, D. I. Juna, and B. Ramdhani, “Pengamanan Sistem dan Data E-Voting Berbasis Network,” *Internal (Information System Journal)*, vol. 1, no. 1, pp. 1–10, 2018.

[31] A. Tantoni and M. T. A. Zaen, “Implementasi Double Caesar Cipher Menggunakan ASCII,” *Jurnal Informatika dan Rekayasa Elektronik*, vol. 1, no. 2, p. 24, 2018.

[32] A. Kushwaha and D. Anil GN, "Securing the Authentication Mechanism for Implementing Secret Password," *International Journal of Scientific Research in Computer Science Applications and Management Studies*, vol. 7, no. 3, pp. 1–4, 2018.

[33] P. J. F. Bemida, A. M. Sison, and R. P. Medina, "Modified SHA-512 Algorithm for Secured Password Hashing," in *Innovations in Power and Advanced Computing Technologies (i-PACT)*, 2021, pp. 1–9.

[34] U. Rathod, M. Sonkar, and B. R. Chandavarkar, "An Experimental Evaluation on the Dependency between One-Way Hash Functions and Salt," in *International Conference on Computing, Communication and Networking Technologies*, 2020.

[35] M. O. Al-Dwairi, A. Y. Hendi, and Z. A. AlQadi, "An Efficient and Highly Secure Technique to Encrypt and Decrypt Color Images," *Engineering, Technology & Applied Science Research*, vol. 9, no. 3, pp. 4165–4168, 2019.

[36] N. Anwar, Munawwar, M. Abduh, and N. B. Santosa, “Komparatif Performance Model Keamanan Menggunakan Metode Algoritma AES 256 bit dan RSA,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 2, no. 3, pp. 783–791, 2018.