Jurnal Masyarakat Informatika, 16(2), 2025
DOI:10.14710/jmasif.16.2.75091
ISSN: 2777-0648

Performance Analysis Cryptography Using AES-128 and Key Ecnryption Based on MD5

Reza Arista Pratama[1)], Eko Hari Rachmawanto[1)], Candra Irawan[2)] Lalang Erawan[2)] , Deddy Award Widya Laksana[3)] , Rabei Raad Ali[4)]

[1)]*Department of Informatics Engineering, Universitas Dian Nuswantoro, Semarang Indonesia*
[2)]*Department of Informatics Systems, Universitas Dian Nuswantoro, Semarang Indonesia*
[3)]*Department of Visual Communication Design, Universitas Dian Nuswantoro, Semarang Indonesia*
[4)]*Department of Computer Science, Northern Technical University, Mosul, Iraq*
[*] *Corresponding author: eko.hari@dsn.dinus.ac.id*

Abstract

The rampant misuse of data theft has created data security techniques in cryptography. Cryptography has several algorithms that are very strong and difficult to crack, including the AES (Advanced Encryption Standard) algorithm consisting of 128 bits, 192 bits, and 256 bits which have been proven resistant to conventional linear analysis attacks and differential attacks, then there is the MD-5 algorithm (Message-Digest algorithm 5) which is a one-way hash function by changing data with a long size and inserting certain data in it to be recovered. If the two are combined, it is more difficult to crack, therefore to determine its performance, this study conducted a combination experiment of AES-128 with a key encrypted by MD-5 with its avalanche effect test, encryption and decryption execution time, and entropy value of encryption. The types of documents for testing are files with the extensions .docx, .txt, .pptx, .pdf, and .xlsx. After conducting tests on document files obtained from the processing time test, it shows that .txt and .pptx documents dominate with a fast process, while the longest process is obtained by .xlsx and .docx documents for all test files, then the avalanche effect test with an average of 98% and the entropy test is classified as good between values 3 - 7 which are close to value 8. This proves that the combination of the AES - 128 algorithm with the MD-5 key can be used as an alternative in securing documents with stronger security with standard processing time.

Keywords : *Cryptography, AES, MD-5, file, execution time, Avalanche effect, Entropy.*

1 Introduction

Security threats to important information on a system give rise to the desire that information can only be accessed by certain parties. The threats in question are theft or destruction that can endanger or harm a party or group. The existence of these threats has created information security methods using several sciences that function to protect data sent or transmitted via communication networks [1], [2], [3].

One method that can be used to maintain the confidentiality of important information is by using the cryptography method or disguising data. Cryptography is a step to hide logical messages or data from individuals or groups who are not entitled to know [4], [5], [6]. The concept of cryptography techniques is to disguise data before it is transmitted to another network by using mathematical calculations that contain keys to return it to its original data, so that when the data is read by a party who is not entitled to receive it, they will not be able to understand the meaning of the contents of the

Received: 16 Juni 2025 ; Revised: 29 July 2025; Accepted: 31 July 2025; Published: 02 August 2025

data. The main concept of cryptography is divided into two types, namely encryption and decryption [7]. Encryption is the process of disguising important information or data that is changed into something that cannot be recognized or with a new character form with an algorithm. Decryption is the process of changing disguised or encoded information from encryption to the original data or information. There is a term in the data before encryption which is called plaintext then after undergoing the encoding process it becomes ciphertext [8]. Cryptography is divided into two types, namely classical cryptography or security with the process of changing to ciphertext for each character and modern cryptography or the process of security using bit mode [9]. Media that can be encrypted using this cryptographic technique are text, images, document files, audio, and video. In this study, the objects used were files in .docx, .txt, .pptx, .pdf, and .xlsx formats. To maintain the confidentiality of the contents of the document file, an algorithm is needed that is difficult to translate by parties who are not authorized to receive it. Cryptographic algorithms generally [10], [11], [12] have three types of key differences, namely symmetric key cryptography, the opposite is asymmetric key cryptography and hybrid keys.

Cryptography with symmetric keys uses private keys or the same key, methods that use symmetric keys include Advanced Encryption Standard (AES), Blowfish, One Time Pad (OTP), Rivest Cipher 4 (RC4) [13], [14]. Asymmetric keys in cryptography use two types of keys, including private keys and public keys in encryption methods such as the RSA algorithm (Rivest Shamir Adleman), Hill Cipher, Diffie-Hellman. Hybrid key cryptography uses two levels of keys, including a session key which is a symmetric key that is useful for data encryption and a pair between the private key and the public key is used to protect the symmetric key.

The algorithm method chosen in this case is the Advanced Encryption Standard (AES) algorithm for a 128-bit key. The selection of the AES-128 bit algorithm is used to protect the confidentiality of the contents of a file, because the AES-128 bit algorithm has proven its resistance to conventional linear analysis attacks and differential attacks [15]. The process of encryption or decryption of the algorithm is carried out ten rounds or iterations to secure or open it, so it is difficult to crack. In this study, the author also used a combination of the MD-5 algorithm in encoding the AES-128 bit key before encrypting or decrypting, because the AES-128 bit algorithm is a symmetric algorithm with the same private key, which means that if the key has been found, the encryption is easy to attack [16]. The MD-5 algorithm or Message-Digest algorithm 5 is a one-way hash function with a method of changing data with a certain length by inserting data into it so that it is difficult to recover even though it looks at the shape of the hash.

In this case, several parameters are used to determine that the algorithm used is strong and efficient, namely by determining several types of files of various sizes, encryption and decryption execution times, and avalanche effect values and entropy values. Avalanche effect is the percentage value that indicates changes in plaintext or key that cause the resulting ciphertext, if the value indicates half the number of ciphertext bits (50%) then encryption is difficult to break [17], while entropy is information on the average character content in the codeword [18]. The selection of processing time testing is used to assess how efficient the encryption and decryption time of the combined algorithm is. Avalanche effect and entropy testing are very suitable for testing document file documents because they see the randomness and uncertainty of the results of encrypted document files [19], [20]. Based on the background written, this research conducted a combination analysis of AES - 128 bit with a key

that was previously encrypted using MD-5 and cut by 16 characters. The combination of this algorithm is expected to be one of the alternative choices of algorithms for document file encoding.

2 Method

2.1 Advanced Enryption Standard (AES)

Advanced Enryption Standard (AES) is a block cipher type algorithm that is included in the type of modern cryptography with a security process using bit mode and using symmetric keys [15]. The AES algorithm is the result of a design by Vincent Rijmen and John Daemen (Rijndael) in 2001 which was selected as the winner of a cryptography competition in the United States held by NIST, after several times AES went through a standardization process, then on May 22, 2002 it was officially approved as a standard cryptographic algorithm [13]. The AES algorithm has three types of block sizes, namely 128, 192, and 256 bits. AES performance is classified as very good because there is an encryption method that runs from several networks (Irawan et al. 2020). The differences in each sequence, namely the length of the key and the number of rounds, are illustrated in Table 1.

Table 1 AES (Rijndael) algorithm class

AES Type	Length of Key	Length of Block	Total round
AES-128 bit	4	4	10
AES-192 bit	6	4	12
AES-256 bit	8	4	14

Table 1 above shows the types of AES that are distinguished by their key length, block length, and number of rounds. This study uses the AES - 128 bit type with 10 rounds as in Figure 1. Transformations in AES rounds for encryption include:

1. SubBytes. The function of SubBytes is used to exchange the contents of the bytes with the AES S-Box table.
2. ShiftRows. The function of shifting the block for each row of the state array.
3. MixColumn. The function of multiplying blocks in each existing state array.
4. AddRoundKey. The function of calculating XOR for the state array and round key.

In the AES decryption process, among others:

1. InvShiftRows. The function of shifting bits to the right for each row block.
2. InvSubBytes. Mapping elements for each state using the Inverse S-Box substitution table.
3. InvMixColumn. Function multiplies each state with the matrix in AES.
4. AddRoundKey. Function performs XOR state array with round key.

The AES-128 algorithm uses a key expansion process (key schedule) that forms 10 different keys in each round. ¬ In AES-128 bits, to determine the key expansion, a 16-byte primary key is needed which goes through the RotWord process (one byte left circular shift), Subword (mapping each byte with the AES S-Box table), Rcon (XOR with the Rcon constant matrix) and XOR with the primary key.

Decryption is the process of reversing a cipher transformation, implemented in the opposite direction to the inverse cipher. The transformations used to generate an inverse cipher include InvSubBytes, InvShiftRows, AddRoundKey, and InvMixColumns. The schematic for the decryption calculation process is illustrated in Figure 2.

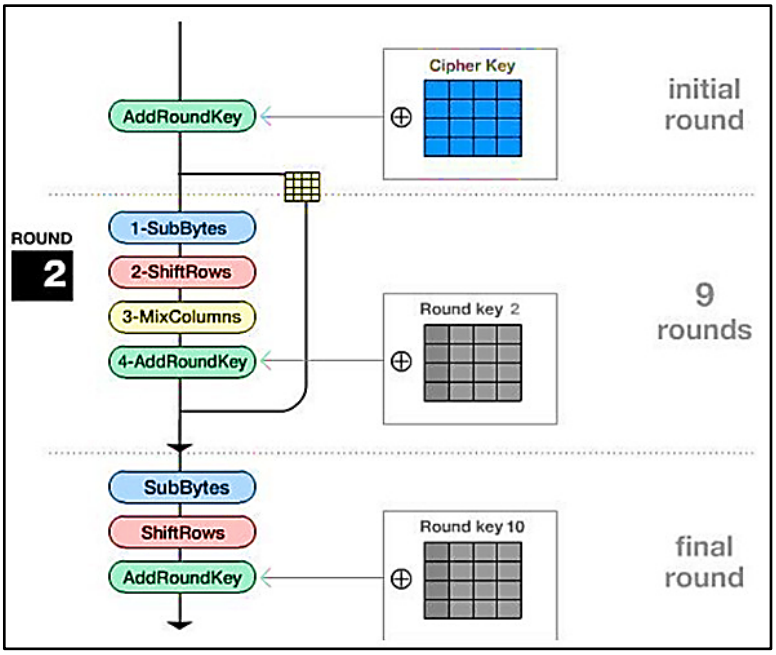

Figure 1 AES Encryption Stages

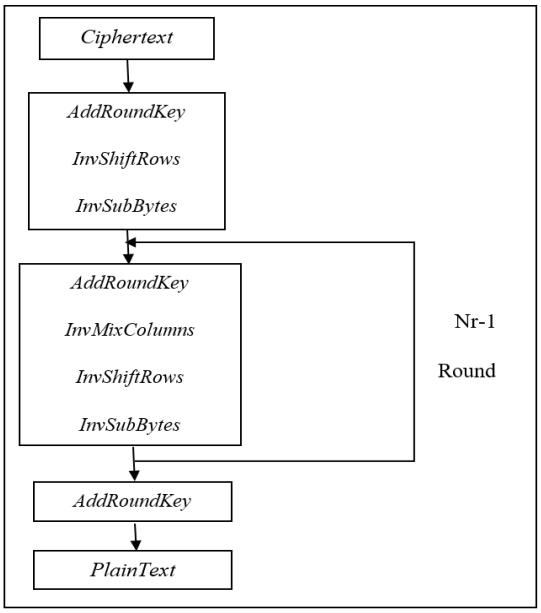

Figure 2 AES decryption

2.2 AES Key Expansion

The AES-128 algorithm uses a key expansion process (key schedule) that generates 10 different keys in each round as in Figure 3. In AES-128, to determine key expansion, a 16-byte primary key is required, which goes through the RotWord process (one-byte left circular shift), Subword (mapping each byte with the AES S-Box table), Rcon (XOR with the Rcon constant matrix), and XOR with the primary key. This process repeats 10 times, as many rounds as there are in AES-128. An illustration of key expansion is shown in Figure 4.

i	1	2	3	4	5	6	7	8	9	10
rc_i	01	02	04	08	10	20	40	80	1B	36

Figure 3 RCI value for each round

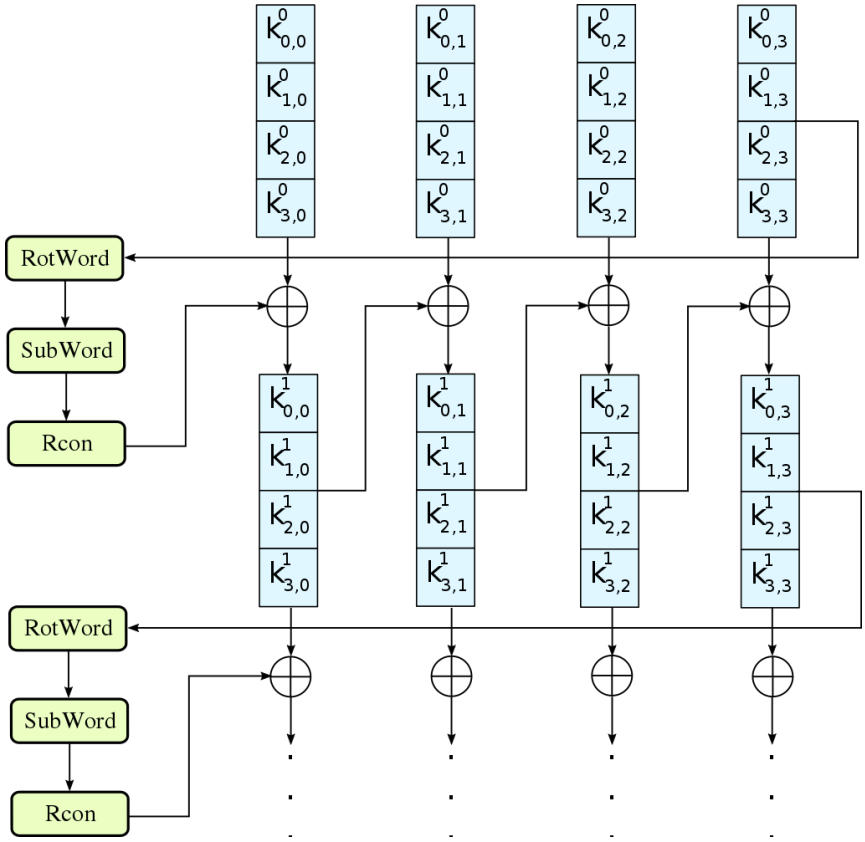

Figure 4 AES Key Expansion

2.3 Message Digest-5 (MD5)

In cryptography, MD5 is one of the cryptographic hash functions with a 128-bit hash value. According to (Yusuf 2020) Message - Digest algorithm 5 is a one-way hash encryption algorithm innovation of MD-4 by adding 1 round or round, namely [1, 3, 10] designed by Ron Rivest with the definition of RFC 1321 which is known to be the most widely used [15], [20]. The algorithm processes a block of 512 bits divided into 16 with each sub-block of 32 bits. The MD-5 output becomes 32 bits as many as 4 pieces with a total of 128 bits or called the hash value [3, 10]. In essence, MD-5 has a message block with a length of 512 bits that goes into 4 rounds and the output is 128 bits from the highest D byte and the lowest A byte. The MD5 work process is as follows :

1. Addition of Padding Bits.
   Addition of padding bits so that the length of the text/message (in bits) corresponds to 448 modulo 512. If the message is 448 bits long, then the message becomes 960 bits if added with 512 bits. The padding bit length is between 1 - 512 bits starting from bit 1 followed by the remaining bits 0.
2. Addition of Message Length Value.

Addition of 64 bits again after going through the process of adding padding bits to the message, if the original length is >264 bits then the next thing that is added is the length value in modulo 264 so that the message is a multiple of 512 bits.

3. Buffer Initialization.
   Message – Digest algorithm 5 requires 4 buffers with a length of 32 bits each so that the total buffer is 128 bits. The MD-5 buffer is expressed in hexadecimal form as follows:
   A = 01234567
   B = ABCDEF
   C = FEDCBA98
   D = 76543210
4. Processing in 512-bit Blocks.
   The division of messages / text into K blocks with a length of 512 bits each (M to MK - 1). Each processing of a 512-bit block that occurs simultaneously with the buffer and becomes its output of 128 bits is called a process for HMD-5.

The 128-bit MD5 hash value is relatively small compared to other, more secure hash algorithms like SHA-256 (256 bits). The compact hash size is advantageous in environments where storage or transmission bandwidth is limited, making it more efficient for data transmission.

2.4 Dataset

Here, we obtained by the author from quantitative data obtained from the author's experiments to solve problems in the study. The data obtained came from the author's personal document files used as objects of study as many as 20 document files consisting of files with the extensions .pdf, .docx, .txt, .pptx, and .xlsx. For detailed data explained in Table 2.

Table 2 Dataset for Cryptography Evaluation

No	Dcument name	Size(Kb)	Extentions	Detail
1.	tes_pdf1.pdf	500 Kb	.pdf	Text and images
2.	tes_pdf2.pdf	1000 Kb	.pdf	Text and images
3.	tes_pdf3.pdf	2000 Kb	.pdf	Text and images
4.	tes_pdf4.pdf	3000 Kb	.pdf	Images with large size
5.	tes_doc1.docx	500 Kb	.docx	Text and images
6.	tes_doc2.docx	1000 Kb	.docx	Text and images (Effects)
7.	tes_doc3.docx	2000 Kb	.docx	Text, illustration inserts and large images (Effects)
8.	tes_doc4.docx	3000 Kb	.docx	Text, illustration inserts and large images (Effects)
9.	tes_txt1.txt	500 Kb	.txt	Text
10.	tes_txt2.txt	1000 Kb	.txt	Text
11.	tes_txt3.txt	2000 Kb	.txt	Text
12.	tes_txt4.txt	3000 Kb	.txt	Text
13.	tes_ppt1.pptx	500 Kb	.pptx	Text and images
14.	tes_ppt2.pptx	1000 Kb	.pptx	Text and images
15.	tes_ppt3.pptx	2000 Kb	.pptx	Text, images, inserts of other .ppt documents and illustration inserts
16.	tes_ppt4.pptx	3000 Kb	.pptx	Text, images, inserts of other .ppt documents and illustration inserts
17.	tes_xls1.xlsx	500 Kb	.xlsx	Text (7 sheets)
18.	tes_xls2.xlsx	1000 Kb	.xlsx	Text and formulas (14 sheets)
19.	tes_xls3.xlsx	2000 Kb	.xlsx	Text and formulas (18 sheets)
20.	tes_xls4.xlsx	3000 Kb	.xlsx	Text and formulas (24 sheets)

2.5 Data Analysis

The techniques used to analyze data in the study were avalanche effect value analysis, encryption and decryption time, and entropy value analysis.

1. Avalanche Effect (AE)
   This technique is used to test the quality of the AES-128 bit algorithm encryption results. The process that occurs in this technique is comparing the same encryption file and changing the key by comparing each bit of changing each character to 8 bits. In this process, the encryption characters that are compared are taken 16 characters for each document file and then compared and continued with the next 16 characters. Illustration of the avalanche effect is shown in Figure 5.

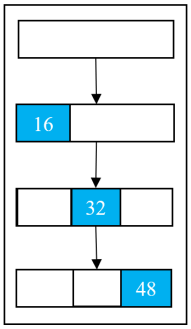

Figure 5 An Illustration of the calculation of the avalanche effect

2. Tame calculation
   This time calculation is used to see the fastest file extension type in encryption and decryption execution. Encryption execution time is obtained from the beginning of the file translation process to binary form and key retrieval in the database to the encryption result in ASCII character form, while decryption time is obtained from the process of translating the encryption file retrieval to binary form and key retrieval in the database until changing back to the original file.
3. Entropy
   Entropy testing is used as a test in determining the best attribute with a measure of uncertainty in a data set, so the higher the entropy result, the better the quality of the attribute uncertainty of a data in cryptographic encryption. The entropy value is obtained by calculating the average value of the probability of the existence of a codeword character.

3 Results

The initial stage carried out in the study was collecting data used in testing the performance of the AES algorithm - 128 bit with the MD-5 key in the form of several document files that have been mentioned previously and preparing the database needed in the encryption and decryption process. Documents were obtained from the author's experiments according to the required size which contained text, images, and diagrams with a size of 500kb - 3000kb as samples as in Figure 6.

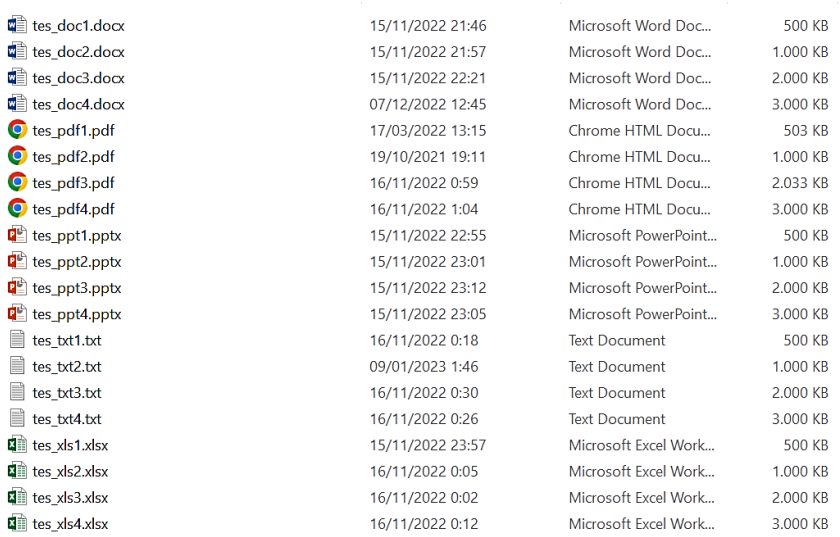

Figure 6 Testing File

The results of this research test include the encryption and decryption process time of the test file, the percentage of the avalanche effect of the encrypted file, and the entropy value of the encrypted file. The test was carried out by calculating the process time when the program was given the command to start until the command was completed. The aim is to see the speed of the algorithm's performance in solving problems. The results of the document file test used in this study can be seen in Table 3.

Table 3 Results of encryption and decryption time testing

No	File name	Size	Extention	Encryption time	Decription time
1.	tes_pdf1.pdf	500 Kb	*.pdf*	7,9323	8,8285
2.	tes_pdf2.pdf	1000 Kb	*.pdf*	15,038	14,591
3.	tes_pdf3.pdf	2000 Kb	*.pdf*	30,559	29,732
4.	tes_pdf4.pdf	3000 Kb	*.pdf*	45,358	44,179
5.	tes_doc1.docx	500 Kb	*.docx*	7,8081	8,9497
6.	tes_doc2.docx	1000 Kb	*.docx*	15,012	16,386
7.	tes_doc3.docx	2000 Kb	*.docx*	30,488	33,280
8.	tes_doc4.docx	3000 Kb	*.docx*	48,161	45,597
9.	tes_txt1.txt	500 Kb	*.txt*	7,4810	7,2809
10.	tes_txt2.txt	1000 Kb	*.txt*	15,549	14,597
11.	tes_txt3.txt	2000 Kb	*.txt*	29,935	30,576
12.	tes_txt4.txt	3000 Kb	*.txt*	45,150	45,534
13.	tes_ppt1.pptx	500 Kb	*.pptx*	7,5502	8,5678
14.	tes_ppt2.pptx	1000 Kb	*.pptx*	15,000	14,518
15.	tes_ppt3.pptx	2000 Kb	*.pptx*	31,223	29,248
16.	tes_ppt4.pptx	3000 Kb	*.pptx*	45,062	43,623
17.	tes_xls1.xlsx	500 Kb	*.xlsx*	7,5573	7,2992
18.	tes_xls2.xlsx	1000 Kb	*.xlsx*	15,025	14,673
19.	tes_xls3.xlsx	2000 Kb	*.xlsx*	30,232	33,722
20.	tes_xls4.xlsx	3000 Kb	*.xlsx*	45,835	48,908

The fastest time was obtained in the document process with the .txt extension with a size of 500 KB and the longest time was obtained for the document file with the .docx extension with a size of 3000 KB. The .txt file type obtained the fastest time compared to other types because the parsing of the document file was faster than other document files as in Figure 7. Figure 8 shows a graph of the difference in encryption time for each type of extension. The diagram shows an increase in the

difference in encryption time for each size and type of document file except for the .pdf and .pptx extensions with a size difference between 1000 KB and 2000 KB and a size of 2000 KB and 3000 KB which shows a decrease because the parsing of these file types is faster.

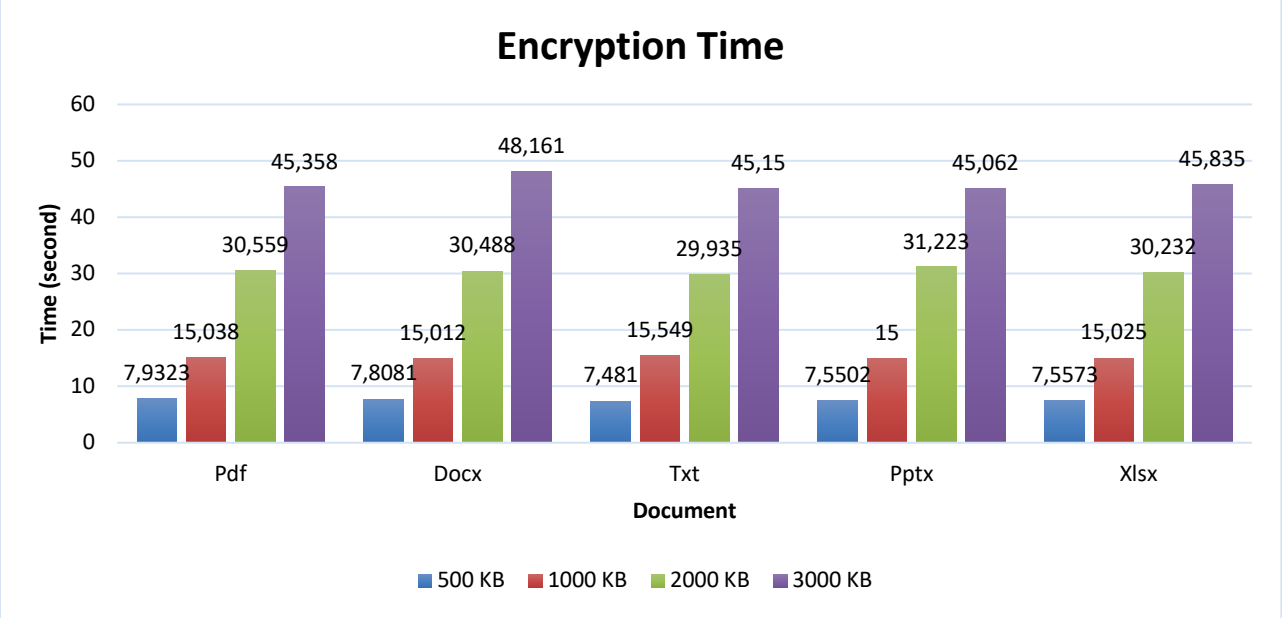

Figure 7 Encryption Time

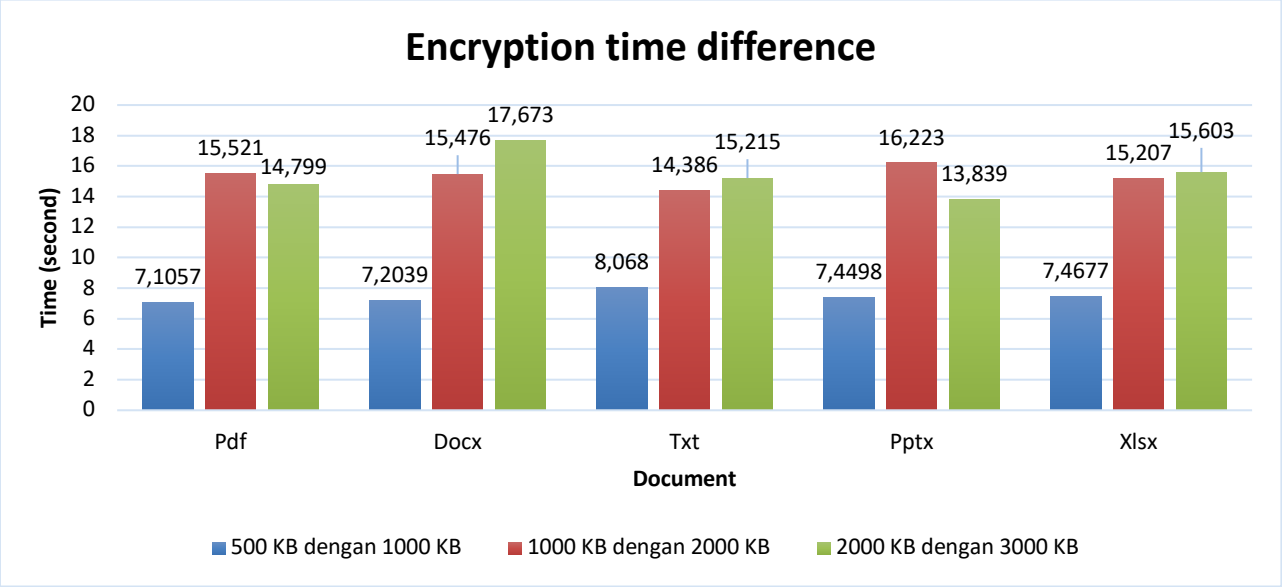

Figure 8 Encryption Time Difference

Figure 9 shown he fastest time was obtained in the document process with the .txt extension with a size of 500 KB and the longest time was obtained for the document file with the .xlsx extension with a size of 3000 KB. The .xlsx file type obtained the longest time compared to other types because the parsing of the document file took longer than other document files. Testing is done by comparing each bit of the encrypted file with the encryption results using a different key, either partially or completely. The goal is to determine the level of binary change. The results of testing the file documents used in this study can be seen in Table 4.

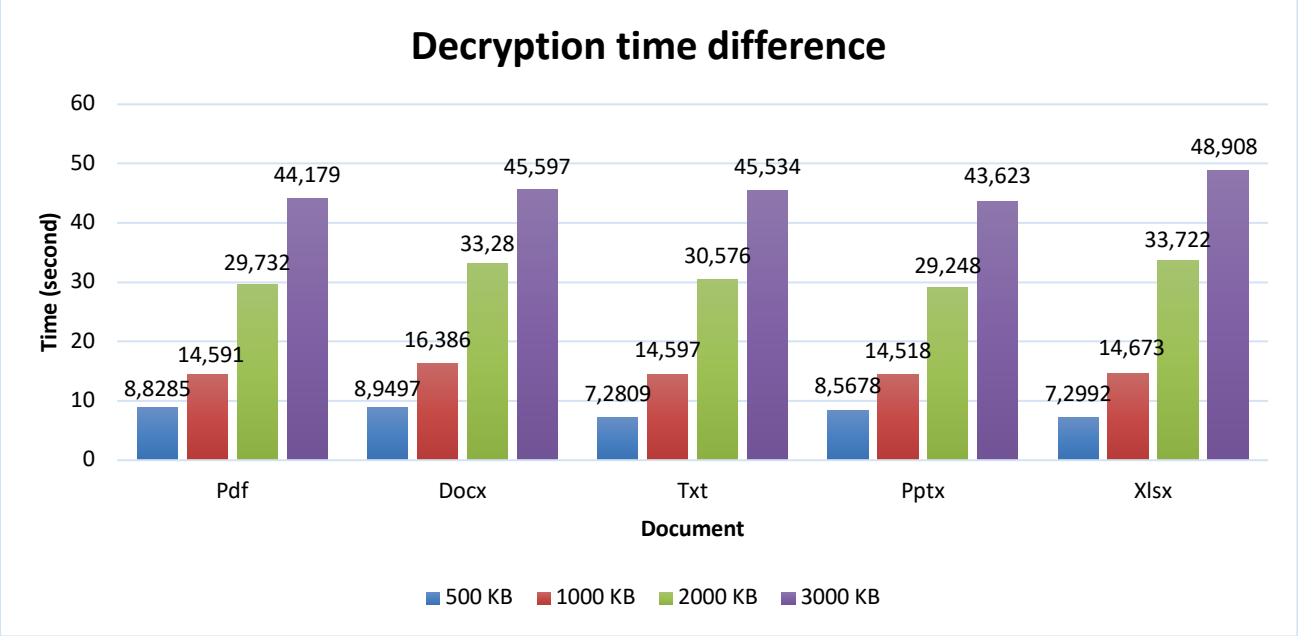

Figure 9 Decryption Time Difference

Table 4 Avalanche effect test results

No	Filename	Size	Extention	Avalanche effect result (%)
1.	tes_pdf1.pdf	500 Kb	*.pdf*	98,826376802586
2.	tes_pdf2.pdf	1000 Kb	*.pdf*	98,866752875359
3.	tes_pdf3.pdf	2000 Kb	*.pdf*	98,843692114952
4.	tes_pdf4.pdf	3000 Kb	*.pdf*	98,87306451949
5.	tes_doc1.docx	500 Kb	*.docx*	98,893021400413
6.	tes_doc2.docx	1000 Kb	*.docx*	99,438619245554
7.	tes_doc3.docx	2000 Kb	*.docx*	98,842261625727
8.	tes_doc4.docx	3000 Kb	*.docx*	98,888807121012
9.	tes_txt1.txt	500 Kb	*.txt*	98,856011074266
10.	tes_txt2.txt	1000 Kb	*.txt*	98,85078125
11.	tes_txt3.txt	2000 Kb	*.txt*	98,846295095289
12.	tes_txt4.txt	3000 Kb	.txt	98,850545898727
13.	tes_ppt1.pptx	500 Kb	.pptx	98,875160161255
14.	tes_ppt2.pptx	1000 Kb	.pptx	98,868013344376
15.	tes_ppt3.pptx	2000 Kb	.pptx	98,86932071313
16.	tes_ppt4.pptx	3000 Kb	.pptx	98,863331527599
17.	tes_xls1.xlsx	500 Kb	.xlsx	99,021917487107
18.	tes_xls2.xlsx	1000 Kb	.xlsx	99,020980796575
19.	tes_xls3.xlsx	2000 Kb	.xlsx	99,004738503367
20.	tes_xls4.xlsx	3000 Kb	.xlsx	99,006831654287

Testing by changing the key that was originally "Cryptography" changed to "Kripto123" was done by changing 5 characters in the key. The highest result was obtained by a file with the extension .docx with a size of 1000 KB with a percentage of 99.44%. The results of the file are due to the parsing of the extension type .docx with a size of 1000 KB has a larger image content so that the bit parsing in the file is more varied. The test is done by looking for the calculation of the average value of the information on the probability of the existence of codeword characters. The results of the document file test used in this study can be seen in Table 4. Using entropy testing, the smallest number

of calculations was obtained with an average of 4.431819981 files with the .pdf extension, meaning that the pdf extension file has a smaller average chance of having the same codeword.

Table 5 Entropy testing

No	Filename	Size	Extention	Entropy
1.	tes_pdf1.pdf	500 Kb	.pdf	4,8073549220576
2.	tes_pdf2.pdf	1000 Kb	.pdf	5,1699250014423
3.	tes_pdf3.pdf	2000 Kb	.pdf	4,00
4.	tes_pdf4.pdf	3000 Kb	.pdf	3,75
5.	tes_doc1.docx	500 Kb	.docx	5,6294072297984
6.	tes_doc2.docx	1000 Kb	.docx	4,744313261737
7.	tes_doc3.docx	2000 Kb	.docx	4,7094909245638
8.	tes_doc4.docx	3000 Kb	.docx	6,3085036466474
9.	tes_txt1.txt	500 Kb	.txt	6,6161243201327
10.	tes_txt2.txt	1000 Kb	.txt	7,6961612041315
11.	tes_txt3.txt	2000 Kb	.txt	7,3067762402127
12.	tes_txt4.txt	3000 Kb	.txt	6,9780389329452
13.	tes_ppt1.pptx	500 Kb	.pptx	6,4591965104472
14.	tes_ppt2.pptx	1000 Kb	.pptx	5,6580157728992
15.	tes_ppt3.pptx	2000 Kb	.pptx	5,2310215847902
16.	tes_ppt4.pptx	3000 Kb	.pptx	5,4034377415252
17.	tes_xls1.xlsx	500 Kb	.xlsx	4,7496867518472
18.	tes_xls2.xlsx	1000 Kb	.xlsx	4,7517270508786
19.	tes_xls3.xlsx	2000 Kb	.xlsx	5,5709595934316
20.	tes_xls4.xlsx	3000 Kb	.xlsx	6,4167162948779

4 Conclusion

The results of the research that has been done in the encryption and decryption of document files using a combination of the AES-128 algorithm with the encrypted key has been done. The results of the encryption time test show that at a size of 500 KB the fastest process is obtained by the .txt extension file followed by .pptx, .xlsx, .docx, then .pdf. For a size of 1000 KB the fastest process is obtained by the .pptx extension file followed by .docx, .xlsx, .pdf, then .txt. Furthermore, the size of 2000 KB the fastest process is obtained by the .txt extension file followed by .xlsx, .docx, .pdf, then .pptx. then the size of 3000 KB the fastest process is obtained by the .pptx extension file followed by .txt, .pdf, .xlsx, then .docx. The results of the decryption time test show that at a size of 500 KB the fastest process is obtained by the .txt extension file followed by .xlsx, .pptx, .pdf, then .docx. For a size of 1000 KB the fastest process is obtained by the .pptx extension file followed by .pdf, .txt, .xlsx, then .docx. Then the size of 2000 KB the fastest process is obtained by the .pptx extension file followed by .pdf, .txt, .docx, then .xlsx. then the size of 3000 KB the fastest process is obtained by the .pptx extension file followed by .pdf, .txt, .docx, then .xlsx. The results of the processing time are not affected by the performance of the device used so that it cannot show a static sequence of document file types. The results of the Avalanche Effect test on all test files show an average of 98.92032616%. The Avalanche Effect test shows the best .xlsx extension file with the highest average value compared to other types, except for the .docx type file size of 1000 KB with large image inserts. The entropy test results show that files with the .txt extension have the highest or best average entropy value compared to other types of test files because the value is close to 8. It can be interpreted that the results of the .txt file encryption are difficult to detect the encryption pattern visually or by calculation. From these results, it shows that the combination of the AES - 128 bit algorithm with the MD-5 encrypted key can

be used as an alternative in securing document files that are quite strong with a relatively fast time, a high avalanche effect value, and an entropy value close to 8 from several test result files.

For future research, the author has suggestions so that the research written in the next study is better and more relevant to increase the use of document files in testing that are more varied in size, file type, and content. MD5 rentan terhadap serangan tabrakan, maksudnya dimana dua input berbeda bisa menghasilkan nilai hash yang sama. Untuk mengoptimalkan hal tersebut, peneliti selanjutnya dapat menggunakan SHA-256, Bcrypt, atau Argon2. In another hand, improve the device used because in the process the condition of the device is very influential. Reserachers can evaluate the same research results using the TeslaCrypt tool.

Bibliography

[1] Pujitha Manepalli Dharani, "Security Enhancement Using Caesar Cipher," *International Journal of Research Publication and Reviews*, vol. 3, no. 11, pp. 9–16, 2022, [Online]. Available: www.ijrpr.com

[2] R. Verma, A. Kumari, A. Anand, and V. S. S. Yadavalli, "Revisiting Shift Cipher Technique for Amplified Data Security," *Journal of Computational and Cognitive Engineering*, 2022, doi: 10.47852/bonviewJCCE2202261.

[3] I. Stepheng, C. A. Sari, E. H. Rachmawanto, and F. O. Isinkaye, "A Combination of Vigenere Cipher and Advanced Encryption Standard for Image Security," *Advance Sustainable Science Engineering and Technology*, vol. 5, no. 3, p. 0230305, Oct. 2023, doi: 10.26877/asset.v5i3.17150.

[4] S. M. Suhael, Z. A. Ahmed, and A. J. Hussain, "Proposed Hybrid Cryptosystems Based on Modifications of Playfair Cipher and RSA Cryptosystem," *Baghdad Science Journal*, May 2023, doi: 10.21123/bsj.2023.8361.

[5] S. M. Suhael, Z. A. Ahmed, and A. J. Hussain, "Proposed Hybrid Cryptosystems Based on Modifications of Playfair Cipher and RSA Cryptosystem," *Baghdad Science Journal*, May 2023, doi: 10.21123/bsj.2023.8361.

[6] O. F. AbdelWahab, A. I. Hussein, H. F. A. Hamed, H. M. Kelash, and A. A. M. Khalaf, "Efficient Combination of RSA Cryptography, Lossy, and Lossless Compression Steganography Techniques to Hide Data," *Procedia Comput Sci*, vol. 182, pp. 5–12, 2021, doi: 10.1016/j.procs.2021.02.002.

[7] F. Mushtaq Sher Ali and F. Hassan Sarhan, "Enhancing Security of Vigenere Cipher by Stream Cipher," *Int J Comput Appl*, vol. 100, no. 1, pp. 975–8887, 2014.

[8] M. Uthman Azlan and M. Shamian Zainal, "An IoT Based Home Security System With ESP32, Video Monitoring, and Blynk Integration," vol. 5, no. 1, pp. 238–244, 2024, doi: 10.30880/peat.2024.05.01.024.

[9] A. Susanto, D. R. I. Moses Setiadi, E. H. Rachmawanto, C. A. Sari, R. R. Ali, and I. U. Wahyu Mulyono, "Dual Security Method for Digital Image using HBV Encryption and Least Significant Bit Steganography," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, May 2019. doi: 10.1088/1742-6596/1201/1/012024.

[10] B. Deepa, V. Maheswari, and V. Balaji, "An Efficient Cryptosystem Using Playfair Cipher Together with Graph Labeling Techniques," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Jul. 2021. doi: 10.1088/1742-6596/1964/2/022016.

[11] K. Prasad and H. Mahato, "Cryptography using generalized Fibonacci matrices with Affine-Hill cipher," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 25, no. 8, pp. 2341–2352, Nov. 2022, doi: 10.1080/09720529.2020.1838744.

[12] K. Prasad and H. Mahato, "Cryptography using generalized Fibonacci matrices with Affine-Hill cipher," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 25, no. 8, pp. 2341–2352, 2022, doi: 10.1080/09720529.2020.1838744.

[13] C. A. Sari, G. Ardiansyah, D. R. I. Moses Setiadi, and E. H. Rachmawanto, "An improved security and message capacity using AES and Huffman coding on image steganography," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 17, no. 5, pp. 2400–2409, 2019, doi: 10.12928/TELKOMNIKA.v17i5.9570.

[14] E. Hari Rachmawanto *et al.*, "Testing Data Security Using a Vigenere Cipher Based on the QR Code," *Computer Network, Computing, Electronics, and Control Journal*, vol. 8, no. 4, pp. 701–708, 2023, doi: 10.22219/kinetik.v8i4.1734.

[15] L. Khakim, M. Mukhlisin, and A. Suharjono, "Analysis of password after encryption by using the combination of AES256 and MD5 algorithm methods," in *AIP Conference Proceedings*, American Institute of Physics, Apr. 2024, p. 030023. doi: 10.1063/5.0198840.

[16] E. H. Rachmawanto, K. Prasetyo, C. A. Sari, I. M. S. De Rosal, and N. Rijati, "Secured PVD Video Steganography Method based on AES and Linear Congruential Generator," in *2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, IEEE, Nov. 2018, pp. 163–167. doi: 10.1109/ISRITI.2018.8864466.

[17] K. Mohamed, M. N. M. Pauzi, F. H. H. M. Ali, and S. Ariffin, "Analyse On Avalanche Effect In Cryptography Algorithm," in *Proceedings of the International Conference on Sustainable Practices, Development and Urbanisation (IConsPADU 2021), 16 November 2021, Universiti Selangor (UNISEL), Malaysia*, European Publisher, Oct. 2022, pp. 610–618. doi: 10.15405/epms.2022.10.57.

[18] C. A. Sari, D. Wahyu Utomo, W. S. Sari, D. Sinaga, and M. Doheir, "An Enhancement of DES, AES Based on Imperceptibility Along With LSB," in *2022 International Seminar on Application for Technology of Information and Communication (iSemantic)*, IEEE, Sep. 2022, pp. 150–155. doi: 10.1109/iSemantic55962.2022.9920444.

[19] M. M. A Meftah, H. H. Yusef Sa'ad, Y. Al-Ashmoery, A.-M. H. Y. Saad, A. H. Y. Sa'd, and K. Alwesabi, "A Comparative Analysis of Cryptography Algorithms in Information Security," in *2024 10th International Conference on Computing, Engineering and Design (ICCED)*, IEEE, Dec. 2024, pp. 1–6. doi: 10.1109/ICCED64257.2024.10983680.

[20] J. Ayad, N. Qaddoori, and H. Maytham, "Enhanced Audio Encryption Scheme: Integrating Blowfish, HMAC-SHA256, and MD5 for Secure Communication," *Mesopotamian Journal of CyberSecurity*, vol. 5, no. 1, pp. 178–186, Feb. 2025, doi: 10.58496/MJCS/2025/012.