



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

Reza Arista Pratama¹⁾, Eko Hari Rachmawanto¹⁾, Candra Irawan²⁾ Lalang Erawan²⁾, Deddy Award Widya Laksana³⁾, Rabei Raad Ali⁴⁾

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 becomes 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, including avalanche effect tests, encryption and decryption execution times, and entropy values 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 for securing documents with stronger security, while maintaining standard processing times.

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

1 Introduction

Security risks to critical information within a system necessitate strict access control, ensuring that such information can only be accessed by authorized individuals or parties. These risks include acts such as theft, unauthorized disclosure, or deliberate destruction of data, all of which can cause significant harm or loss to individuals, organizations, or communities. The presence of such threats has driven the development of information security methods that integrate multiple scientific disciplines, such as cryptography, network security, and data protection, to safeguard information during storage and when it is transmitted across communication networks [1], [2], [3].

One method that can be used to maintain the confidentiality of important information is the use of cryptography or data disguising. Cryptography is a method to conceal logical messages or data from

individuals or groups who are not authorized 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 form, 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 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 by changing it into something that cannot be recognized or into a new character form with an algorithm. Decryption is the process of converting disguised or encoded information back into its original form from encrypted data. There is a term in the data before encryption, called plaintext, which then becomes ciphertext after undergoing the encoding process [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 document file's contents, an algorithm is needed that is difficult for unauthorized parties to decipher. Cryptographic algorithms generally [10], [11], [12] have three types of key differences, namely symmetric key cryptography, 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, and 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 of 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 encryption or decryption process of the algorithm is carried out in ten rounds or iterations to secure or open it, making it 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 that transforms data of a certain length by incorporating it, making it difficult to recover, even though it examines the hash's structure.

In this case, several parameters are used to determine that the algorithm used is strong and efficient, including the execution times of encryption and decryption for various file sizes, 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 the efficiency of the combined algorithm's encryption and decryption times. Avalanche effect and entropy testing are particularly suitable for testing document files because they reveal the randomness and

uncertainty inherent in the results of encrypted document files [19], [20]. Based on the background information provided, this research conducted a combination analysis of AES-128 bit with a key that was previously encrypted using MD-5 and truncated to 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 Encryption Standard (AES)

Advanced Encryption 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 was designed by Vincent Rijmen and John Daemen (also known as Rijndael) in 2001. It was selected as the winner of a cryptography competition in the United States, held by NIST, after several rounds of AES underwent 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. 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 illustrates the types of AES distinguished by their key length, block length, and number of rounds. This study uses the AES-128 with 10 rounds, as in Figure 1. Transformations in AES rounds for encryption include:

1. SubBytes. The SubBytes function 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 the 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 by the matrix in AES.
4. AddRoundKey. Function performs the XOR state array with the 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.

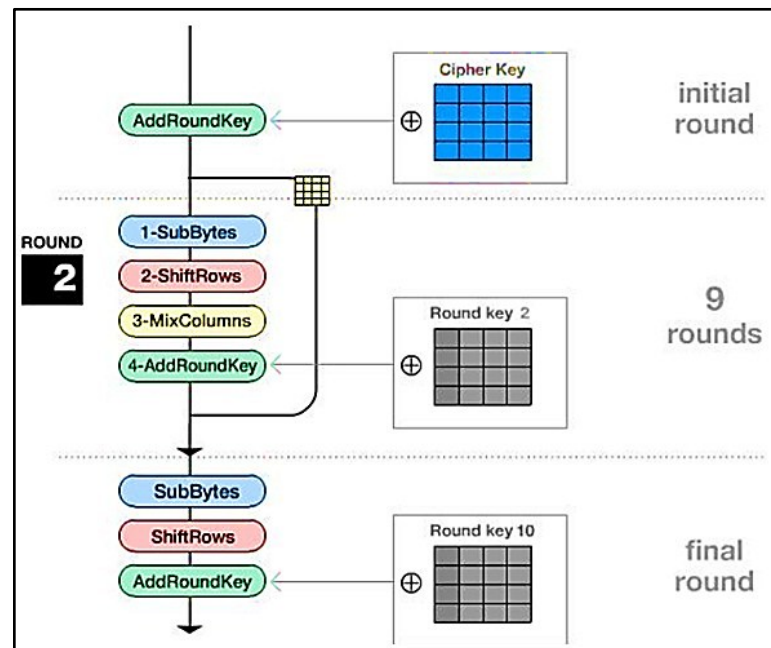


Figure 1 AES Encryption Stages

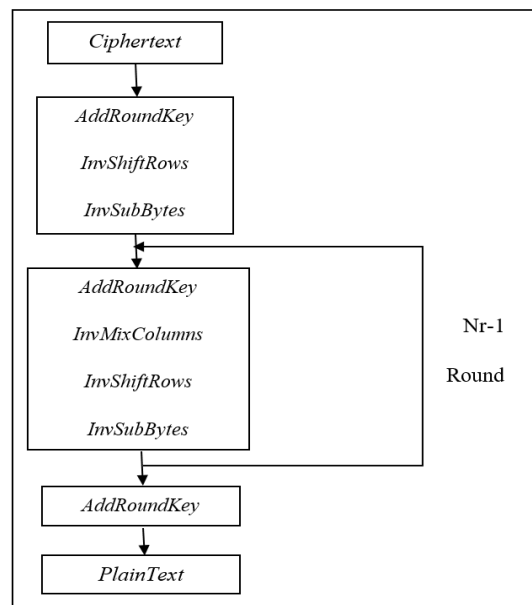


Figure 2 AES decryption

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 shown 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

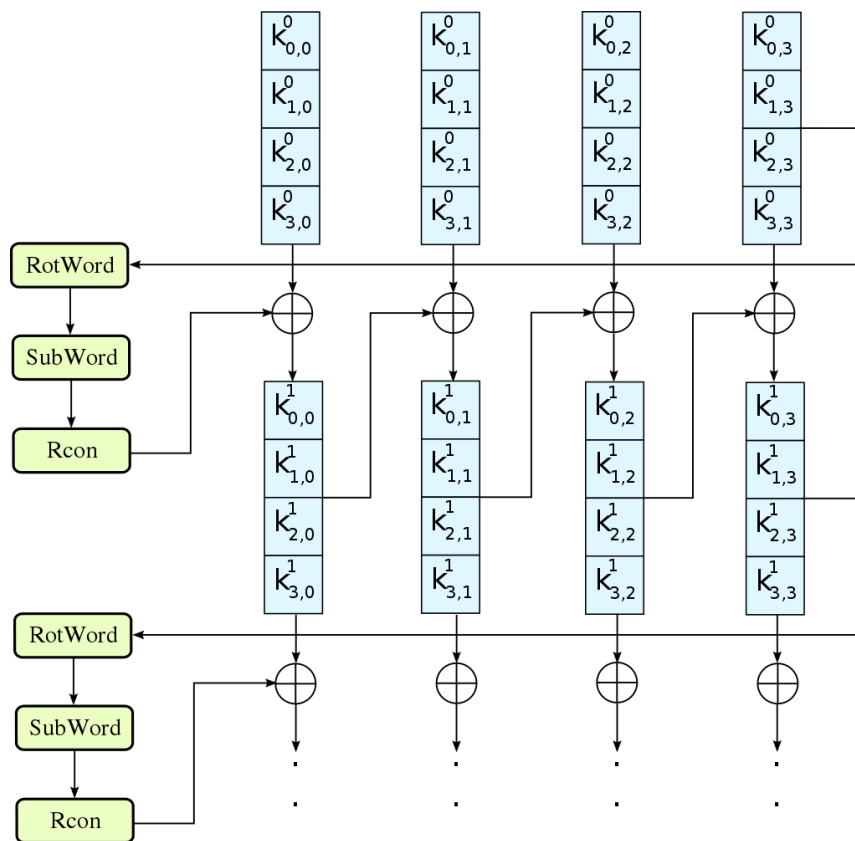


Figure 4 AES Key Expansion

2.3 Message Digest-5 (MD5)

In cryptography, MD5 is a cryptographic hash function that produces a 128-bit hash value. According to [16]. Message Digest algorithm 5 is a one-way hash encryption algorithm innovation of MD-4 by adding 1 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 sub-blocks of 32 bits each. The MD-5 output becomes 32 bits, which can be divided into four pieces, totaling 128 bits, also known as the hash value [3, 10]. In essence, MD-5 has a message block with a length of 512 bits that undergoes four rounds, and the output is 128 bits, comprising 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 and 512 bits, starting from bit 1, followed by the remaining bits, which are all 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 four 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 co-occurs with the buffer and produces an 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, such as 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 the author's quantitative data from their experiments to address problems in the study. The data received 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, see 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 the data in the study were avalanche effect value analysis, encryption and decryption time analysis, and entropy value analysis.

1. Avalanche Effect (AE)

This technique is used to test the quality of the AES-128 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 the character to 8 bits. In this process, the encryption characters are compared are taken 16 characters for each document file, and then the comparison is continued with the next 16 characters. An illustration of the avalanche effect is shown in Figure 5.

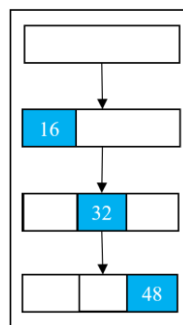


Figure 5 An Illustration of the calculation of the avalanche effect

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. In contrast, decryption time is obtained from the process of translating the encrypted file retrieval into binary form and retrieving the key from the database until the file is restored to its original form.

3. Entropy

Entropy testing is used as a method for determining the best attribute, measuring the uncertainty in a data set. The higher the entropy result, the better the quality of the attribute uncertainty in the data, particularly in cryptographic encryption. The entropy value is calculated by averaging the probability of a codeword character's existence.

3 Results

The initial phase of the study involved collecting data to evaluate the performance of AES-128 with the MD5 key, using several document files mentioned earlier, and preparing the database needed for encryption and decryption. These documents, obtained from the author's experiments, were resized to meet the required dimensions and included text, images, and diagrams with file sizes ranging from 500KB to 3000KB, as shown in Figure 6.





















 tes_doc1.docx	15/11/2022 21:46	Microsoft Word Doc...	500 KB
 tes_doc2.docx	15/11/2022 21:57	Microsoft Word Doc...	1.000 KB
 tes_doc3.docx	15/11/2022 22:21	Microsoft Word Doc...	2.000 KB
 tes_doc4.docx	07/12/2022 12:45	Microsoft Word Doc...	3.000 KB
 tes_pdf1.pdf	17/03/2022 13:15	Chrome HTML Docu...	503 KB
 tes_pdf2.pdf	19/10/2021 19:11	Chrome HTML Docu...	1.000 KB
 tes_pdf3.pdf	16/11/2022 0:59	Chrome HTML Docu...	2.033 KB
 tes_pdf4.pdf	16/11/2022 1:04	Chrome HTML Docu...	3.000 KB
 tes_ppt1.pptx	15/11/2022 22:55	Microsoft PowerPoint...	500 KB
 tes_ppt2.pptx	15/11/2022 23:01	Microsoft PowerPoint...	1.000 KB
 tes_ppt3.pptx	15/11/2022 23:12	Microsoft PowerPoint...	2.000 KB
 tes_ppt4.pptx	15/11/2022 23:05	Microsoft PowerPoint...	3.000 KB
 tes_txt1.txt	16/11/2022 0:18	Text Document	500 KB
 tes_txt2.txt	09/01/2023 1:46	Text Document	1.000 KB
 tes_txt3.txt	16/11/2022 0:30	Text Document	2.000 KB
 tes_txt4.txt	16/11/2022 0:26	Text Document	3.000 KB
 tes_xls1.xlsx	15/11/2022 23:57	Microsoft Excel Work...	500 KB
 tes_xls2.xlsx	16/11/2022 0:05	Microsoft Excel Work...	1.000 KB
 tes_xls3.xlsx	16/11/2022 0:02	Microsoft Excel Work...	2.000 KB
 tes_xls4.xlsx	16/11/2022 0:12	Microsoft Excel Work...	3.000 KB

Figure 6 Testing File

The results of this research test include the encryption and decryption times for the test file, the percentage of the avalanche effect of the encrypted file, and the entropy value of the encrypted file. The test was conducted by measuring the process time from when the program was started until it completed. The goal is to assess the algorithm's performance speed 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 shortest processing time was recorded for the document with a .txt extension and a size of 500 KB, while the longest was for the document with a .docx extension and a size of 3000 KB. The .txt file type achieved the fastest processing time compared to other types because its document parsing was quicker, as shown in Figure 7. Figure 8 displays a graph illustrating the differences in encryption time for each extension type. The graph indicates that encryption time differences increase with

document size and type, except for the .pdf and .pptx extensions, which show a decrease at sizes between 1000 KB and 2000 KB, and at 2000 KB and 3000 KB. This decrease occurs because the parsing of these file types is faster.

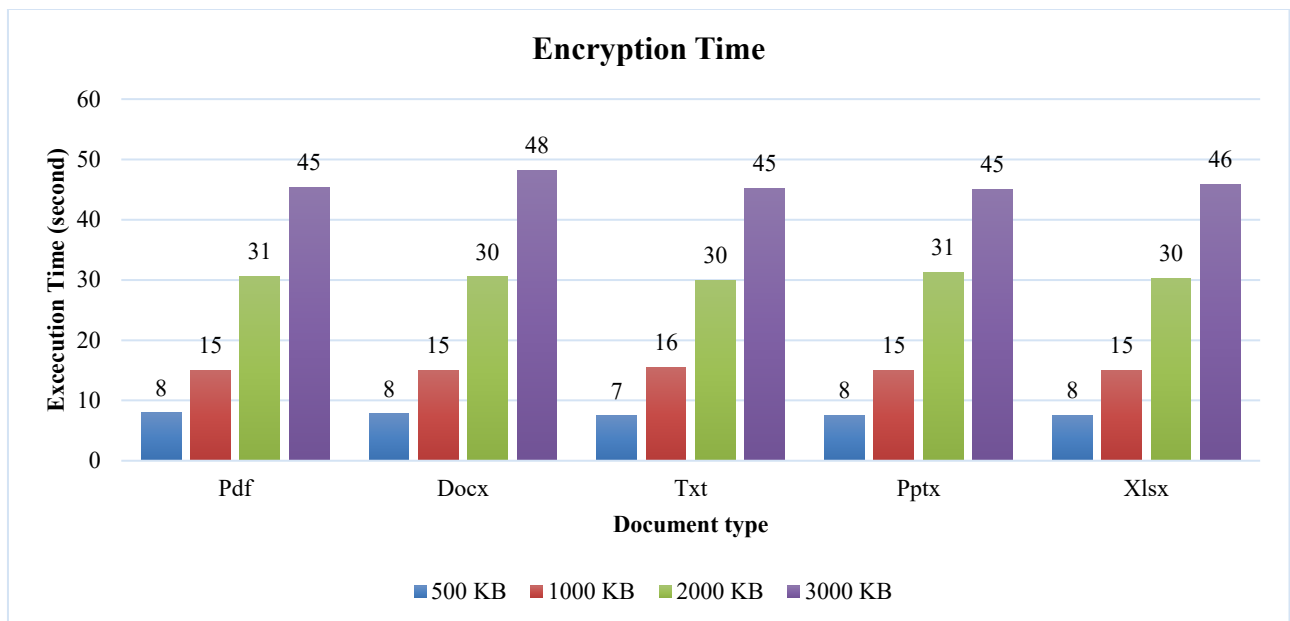


Figure 7 Encryption Time

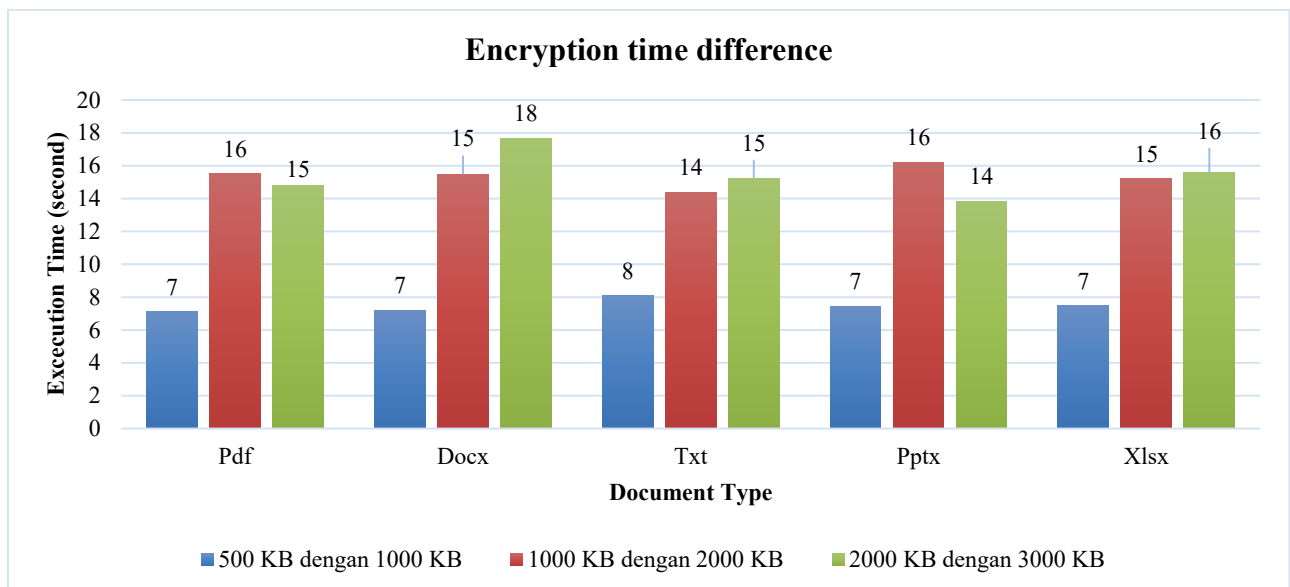


Figure 8 Encryption Time Difference

Figure 9 shows that the fastest time was achieved with the document processed as a .txt file of 500 KB, while the longest time was recorded for the document file with a .xlsx extension of 3000 KB. The .xlsx file took longer to process compared to other types because its parsing required more time. Testing involves comparing each bit of the encrypted file with the encryption results obtained using a different key, either partially or fully. The purpose is to gauge the level of binary change. The results of the file testing used in this study can be seen in Table 4.

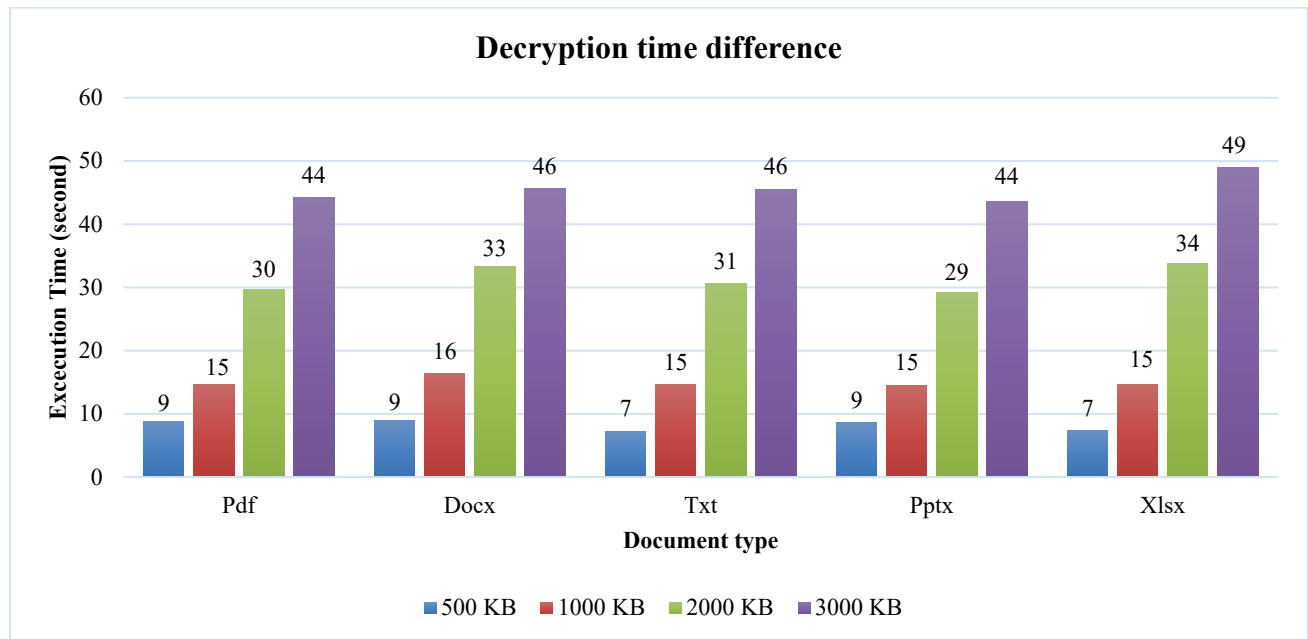


Figure 9 Decryption Time Difference

Table 4 Avalanche effect test results

No	Filename	Size	Extention	Avalanche effect result (%)
1.	tes_pdf1.pdf	500 Kb	.pdf	98.826
2.	tes_pdf2.pdf	1000 Kb	.pdf	98.867
3.	tes_pdf3.pdf	2000 Kb	.pdf	98.844
4.	tes_pdf4.pdf	3000 Kb	.pdf	98.873
5.	tes_doc1.docx	500 Kb	.docx	98.893
6.	tes_doc2.docx	1000 Kb	.docx	99.439
7.	tes_doc3.docx	2000 Kb	.docx	98.842
8.	tes_doc4.docx	3000 Kb	.docx	98.889
9.	tes_txt1.txt	500 Kb	.txt	98.856
10.	tes_txt2.txt	1000 Kb	.txt	98.851
11.	tes_txt3.txt	2000 Kb	.txt	98.846
12.	tes_txt4.txt	3000 Kb	.txt	98.851
13.	tes_ppt1.pptx	500 Kb	.pptx	98.875
14.	tes_ppt2.pptx	1000 Kb	.pptx	98.868
15.	tes_ppt3.pptx	2000 Kb	.pptx	98.869
16.	tes_ppt4.pptx	3000 Kb	.pptx	98.863
17.	tes_xls1.xlsx	500 Kb	.xlsx	99.022
18.	tes_xls2.xlsx	1000 Kb	.xlsx	99.021
19.	tes_xls3.xlsx	2000 Kb	.xlsx	99.005
20.	tes_xls4.xlsx	3000 Kb	.xlsx	99.007

Testing was conducted by changing the original key "Cryptography" to "Kripto123" by modifying five characters. The highest result was obtained with a .docx file of 1000 KB size, achieving a percentage of 99.44%. The results for this file are due to the extension type .docx, which contains larger image content, causing more varied bit parsing in the file. The test involved calculating the average probability of codeword characters. The results of the document file tests used in this study are shown in Table 4. Through entropy testing, the lowest average number of calculations was observed with files having the .pdf extension, indicating that PDF files have a lower average probability of containing the same codeword characters codeword.

Table 5 Entropy testing

No	Filename	Size	Extention	Entropy
1.	tes_pdf1.pdf	500 Kb	.pdf	4.807
2.	tes_pdf2.pdf	1000 Kb	.pdf	5.170
3.	tes_pdf3.pdf	2000 Kb	.pdf	4.000
4.	tes_pdf4.pdf	3000 Kb	.pdf	3.750
5.	tes_doc1.docx	500 Kb	.docx	5.629
6.	tes_doc2.docx	1000 Kb	.docx	4.744
7.	tes_doc3.docx	2000 Kb	.docx	4.709
8.	tes_doc4.docx	3000 Kb	.docx	6.309
9.	tes_txt1.txt	500 Kb	.txt	6.616
10.	tes_txt2.txt	1000 Kb	.txt	7.696
11.	tes_txt3.txt	2000 Kb	.txt	7.307
12.	tes_txt4.txt	3000 Kb	.txt	6.978
13.	tes_ppt1.pptx	500 Kb	.pptx	6.459
14.	tes_ppt2.pptx	1000 Kb	.pptx	5.658
15.	tes_ppt3.pptx	2000 Kb	.pptx	5.231
16.	tes_ppt4.pptx	3000 Kb	.pptx	5.403
17.	tes_xls1.xlsx	500 Kb	.xlsx	4.750
18.	tes_xls2.xlsx	1000 Kb	.xlsx	4.752
19.	tes_xls3.xlsx	2000 Kb	.xlsx	5.571
20.	tes_xls4.xlsx	3000 Kb	.xlsx	6.417

4 Conclusion

The results of the research on the encryption and decryption of document files using a combination of the AES-128 algorithm with an encrypted key have been obtained. 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, and then .pdf. For a size of 1000 KB, the quickest process is obtained by the .pptx extension file, followed by .docx, .xlsx, .pdf, and then .txt. Furthermore, the size of 2000 KB yields the quickest process, followed by the .txt extension file, then .xlsx, .docx, and finally .pptx. Then, the size of 3000 KB, the fastest process is obtained by the .pptx extension file, followed by .txt, .pdf, .xlsx, and 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, and then .docx. For a size of 1000 KB, the quickest process is obtained by the .pptx extension file, followed by .pdf, .txt, .xlsx, and then .docx. Then, the size of 2000 KB, the fastest process is obtained by the .pptx extension file, followed by .pdf, .txt, .docx, and then .xlsx. Then, the size of 3000 KB yields the quickest process, followed by the .pptx extension file, then .pdf, .txt, and finally .docx and .xlsx. The results of the processing time are not affected by the device's performance, so it cannot display 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 indicates that the .xlsx extension file with the best performance has the highest average value compared to other types, except for the .docx type, which has a file size of 1000 KB with large image inserts. The entropy test results show that files with the .txt extension have the highest average entropy value compared to other types of test files, as the value is close to 8. It can be inferred that the results of the .txt file encryption are challenging to detect visually or by calculation. From these results, it is evident that the combination of the AES-128 bit algorithm with the MD-5 encrypted key can be used as an alternative for securing document files, offering a strong level of security, relatively fast processing time, a high avalanche effect value, and an entropy value close to 8, as indicated by several test result files.

For future research, the author offers suggestions to enhance the following study's research, making it more relevant and improving the use of document files in testing that are more varied in terms of 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. On the other hand, improve the device used, as its condition is very influential during the process. Researchers 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](https://doi.org/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](https://doi.org/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](https://doi.org/10.21123/bsj.2023.8361).
- [5] R.K. Salih, M.S. Yousif, "Hybrid encryption using Playfair and RSA cryptosystems," *International Journal of Nonlinear Analysis and Applications*, July 2021, doi: [10.22075/ijnaa.2021.5379](https://doi.org/10.22075/ijnaa.2021.5379).
- [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](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/10.58496/MJCS/2025/012).