Journal of Computer Science, Information Technology and Telecommunication Engineering (JCoSITTE)
Vol. 5, No. 2, September 2024, pp. 595~606
ISSN: 2721-3838, DOI: 10.30596/jcositte.v5i2.20834

Message Security Application Using Mobile-Based AES Algorithm

Mhd Ikhsan Rifki[1]**, Nanda Syamia**[2]
[1,2]Department of Computer Science, Universitas Islam Negeri Sumatera Utara, Medan, Indonesia

ABSTRACT

All information in data exchange transactions on communication networks travels through the network infrastructure. The network infrastructure sends various types of data, including text, images, and documents with various extensions. These documents may be private and confidential. Therefore, it is crucial to have a message security option that is not only user-friendly but also has a high level of security system complexity. The purpose of this research is to build a text message security system using the Advanced Encryption Standard (AES) algorithm on mobile devices. We can utilize it as a security solution to safeguard text messages received through mobile devices. The AES security algorithm is known to have reliability in processing data encryption and decryption. The research will describe components such as the design, implementation, and analysis of application requirements. Important features include encryption, decryption, and key management in addition to an easy-to-use UI. The research findings show that this program successfully secures text messages while maintaining the confidentiality and integrity of the data sent. Additionally, we adjusted the application's parameters to align with mobile device standards, ensuring consistent encryption and decryption procedures that users can easily operate. For application testing, we use the black box testing method to ensure that the design and application function as intended, adhere to security regulations, and provide a positive user experience. This study's results suggest that AES-based mobile messaging security applications can serve as a tool to fulfill the secure communication expectations of mobile users.

Keyword: AES, Applications, Mobile, Text Data, Security

Corresponding Author:
Mhd Ikhsan Rifki,
Department of Computer Science
Universitas Islam Negeri Sumatera Utara
Jl. Lapangan Golf, Durian Jangak, Tuntungan
Email: rifki.mhdikhsan@uinsu.ac.id

Article history:
Received Aug 8, 2024
Revised Aug 11, 2024
Accepted Aug 15, 2024

1. INTRODUCTION

Data transmission in the digital world occurs through a communication network infrastructure. When the sender transmits data to the recipient, it becomes vulnerable to hacking, allowing irresponsible parties to use it for their own personal gain. This can certainly harm the data owner if there are actions that have the potential to violate the law. Some descriptions of the information sent have the potential to relate to personal data, including identity data, personal images, audio, video, and various other types of data and information. To prevent irresponsible parties from using personal data, data security plays a crucial role during the data transaction process in the communication network. Additionally, the data security mechanism incorporates a key that only the sender and receiver can understand.

We can achieve data security by encrypting the information we send over the communication network. The encryption process transforms plaintext into coded text. The communication network receives the results of the plaintext conversion during the information transmission process. The composition of chipertext differs significantly from plaintext, which is the original data form. This can disguise the original information structure that will be sent over the communication network. The implementation of using keys to open messages is another addition (Nababan & Setyadi, 2022). Only the sender and receiver will be able to understand the key. So that the protection applied in data security will get an additional guarantee on the aspect of data confidentiality that can protect data from being read or seen by parties outside the sender and receiver of the data. The DES (Data Encryption Standard) algorithm, which has special characteristics and uses keys with a dimension length of 56 bits, can implement various types of algorithms in data security systems. However, the DES key length presents unique challenges due to its relatively small key size, leading to its perceived weaknesses (Rahman et al., 2020) (Mukwevho & Chibaya, 2020). To enhance its performance, researchers transformed DES by increasing the encryption round cycle to three, a technique known as Triple DES. We believe that adding more encryption cycles can enhance the security complexity of the generated key. This will also lengthen the computation process by requiring three security cycles to complete. The transformation of plaintext into chipertext takes a considerable amount of time.

The AES algorithm is known to have a high level of security in the process of using it (Mohammed et al., 2023). The speed of the encryption process demonstrates the efficiency of using AES as an encryption algorithm. Additionally, AES offers flexibility in the application of keys with varying dimensions. The key length can range from 128 to 192 bits, and it can be extended up to 256 bits (Verma & Sharma, 2020) (Balli & Banik, 2019). (Liu et al., 2023) describe AES as a type of symmetric encryption, utilizing keys in the same ordo at both the sender and the receiver of the data. Additionally, AES is universally applicable across all platforms and devices. Therefore, you can utilize AES as an encryption that adheres to international standards. The security concept can be combined in the form of supporting applications, such as web-based or even mobile applications. This ensures that the implementation review offers sufficient examples of the interface elements for every user. The application's advantages also enable users to experience the application firsthand. Therefore, by combining applications with the concept of data security, users can operate the security system more easily. Another thing is that message security applications can add sharing features to various communication media platforms.

Some research that has been done by (Kaffah et al., 2020) provides the concept of AES security for data transmission via email. E-mail that does not have message security can allow parties outside the recipient and sender to intercept and manipulate data, causing information leakage. AES and Huffman cryptographic methods can achieve data security in e-mail transmission. The results illustrate that both methods can encrypt up to 32,200 characters with an accuracy value of 90.62%. Meanwhile, research conducted by (Su et al., 2019) optimized AES in the IoT environment. The analysis shows that DESI (Data Encryption Standard in IoT) has better security when encrypting data in an IoT environment than AES. In addition, research conducted by (Kumar et al., 2020) focuses on the hardware implementation of one of the AES algorithms. The Vivado 2014.2 ISE Design Suite applies the AES algorithm and observes it using a 28-nanometer (nm) Artix-7 FPGA. The study presents the results of implementing the AES algorithm on FPGAs, utilizing various sources such as slide registers (SR), look-up tables (LUT), input/output (I/O), and global buffers (BUFG).

In addition, research conducted by (Lytvyn et al., 2019) focuses on a neural network system combined with AES. The encryption system utilizes a diagonalized matrix of neural network-like synapse connection weight coefficients, which are based on the input image vector in the system, and incorporates key features for each image. The proposed approach alters the keys continuously during the information encryption process. This aims to increase the stability of the cryptographic algorithm. We will propose a message security system using AES to secure information, based on our research. The application features that will be designed and built include the encryption process and decryption with the use of keys oriented to the 128-bit dimension. As well as the addition of features that can facilitate users sending encrypted messages to various communication media platforms.

2. RESEARCH METHODS

The study concentrates on the R&D (Research and Development) research approach, utilizing the waterfall development method. We use the waterfall method to develop mobile-based software applications. This is based on the fact that the waterfall method is one of the models in the Software Development Life Cycle (SDLC), which outlines a series of processes, including design, implementation, verification, and maintenance. Additionally, mobile application development and interactive system design in learning systems can employ the waterfall method (Wongkhamdi et al., 2020). Requirement, design, implementation, verification, and maintenance are part of the description of the waterfall method.

According to (Dadkhah et al., 2019), the waterfall diagram is considered capable of delivering detailed information, particularly in the form of a software development model, which can provide a clear representation for each individual user and potential user to understand. In addition, the waterfall method has a comprehensive documentation aspect. This documentation can be used to describe requirements specifications, system design, and other documentation as needed. Figure 1 illustrates the process description for the waterfall method, which is a system development method.

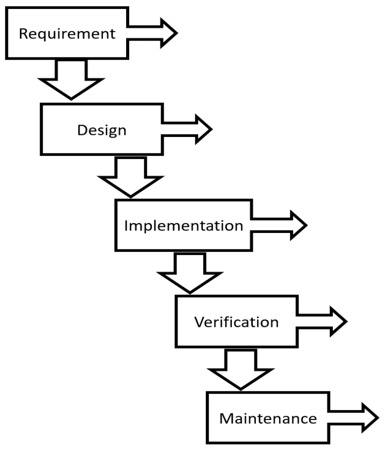


Fig 1. Stages of the Waterfall Method

Preparation begins with examining journals, scientific articles, and other literature sources related to the research title. This analysis yields several research-related variables and parameters, which the research plan then arranges in a system modeling block diagram. The following stages illustrate how the AES-128 method's encryption process works:

1. 128-bit plaintext block
   The initial form of plaintext is strings or text, which requires conversion to hexadecimal numbers. The results of the conversion to hexadecimal numbers will be converted into ASCII form, to be used as a 128-bit plaintext matrix. Then in the AES-128 algorithm, the set of characters in the plaintext used are uppercase letters, lowercase letters, numbers, and the symbols "+", "/", and "=" as padding.
2. Converting plain text to ASCII format
   The conversion of plaintext into ASCII table format is necessary, as it corresponds to ASCII code values. Each character conversion process in a series of plaintexts must be converted one by one, in stages based on the ASCII code value. So that the result of the hexadecimal conversion is obtained in the form of ASCII code for each character.
3. Plaintext matrix generation
   The encryption process implements the plaintext matrix as an input sequence. The matrix form has an ordo size of 4x4.
4. The formation of the key matrix.
   The encryption process uses the key matrix as a security key. The conversion of the key from a string to a hexadecimal number code generates the matrix size. The matrix on the key undergoes a conversion into ASCII code, just like it does in plaintext. We obtain the matrix in ASCII code numbers, which have an ordo dimension of 4x4.
5. Formation of Addroundkey
   In AES operations, the addroundkey function can be applied in integrating the key matrix with a plaintext matrix with an ordo size of 4x4. Both matrices will be used as input to the XOR (Exclusive OR) operation (Manullang & Allwine, 2023). The addroundkey result is illustrated as a combination of the message matrix with the key matrix formed at each round. In detail, each round of the AES method will generate keys that are performed on a scheduled basis. The process continues by performing a bitwise XOR operation. Bitwise is defined as a special mathematical operation implemented at the individual bit level operation (Gao et al., 2021). AES can be described through equation (1).

   $$\text{State}\,[r] = \text{State}\,[r] \oplus \text{RoundKey}\,[r] \tag{1}$$

   Description:
   a. *State [r]* represents the encrypted message block for a given number of rounds (r). Each round's operation transforms the message block, storing it in the State [r] variable.

b. *RoundKey [r]* denotes a key that is utilized as frequently as the number of round operations. We will schedule and use the round key in each round until the round process completes. The master key generates the round key, resulting in a unique set of key components for each round.
c. The $\oplus$ symbol is used to represent the XOR operation on the input bit that is operated bitwise.

6. SubBytes
The AES method uses SubBytes as a mathematical operation to replace each byte in the text block with bytes corresponding to the row in the S-Box (Substitution Box) table. The S-box is a substitution table with a matrix ordo size of 16x16. The output of the subbytes operation will show the new bytes that the input text block will replace (Endrayanto et al., 2019). S-Box will be used for SubByte mathematically, which can be described using Equation (2).

$$\text{State}[r][c] = \text{S-box}\ [\text{State}[r][c]] \qquad (2)$$

7. ShiftRows
In AES, shift row operation is defined as the process of moving each row in the data matrix to the left by a certain amount (Mulud Muchamad et al., 2023) (Freyre et al., 2020). In the initial stages of its operation, the first row in the data matrix is fixed and unchanged. The second row shifts one position to the left, while the third row shifts three positions to the left. Let's assume that we have a data matrix that represents the size of the ordo. The result of shifting rows on matrix *X* will then produce matrix *X*', which will contain the following details:

$$X = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \\ b_0 & b_1 & b_2 & b_3 \\ c_0 & c_1 & c_2 & c_3 \\ d_0 & d_1 & d_2 & d_3 \end{bmatrix} X' = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 & b_0 \\ c_2 & c_3 & c_0 & c_1 \\ d_3 & d_0 & d_1 & d_2 \end{bmatrix}$$

The purpose of applying ShiftRows is to combine data horizontally in the data matrix. This will result in the creation of a new matrix featuring a randomly ordered byte sequence.

8. MixColoums
MixColoums represents a linear transformation operation on each column of the matrix separately (Nino, 2023). During the implementation stage of MixColoums, we assume each column to be a 4-byte polynomial and then multiply it with a matrix of ordo 4x4. If ShiftRows illustrates the combination of bits horizontally, then MixColoums represents the transformation of position changes vertically in the text matrix. Figure 2 illustrates the encryption modeling diagram for the AES-128 method.

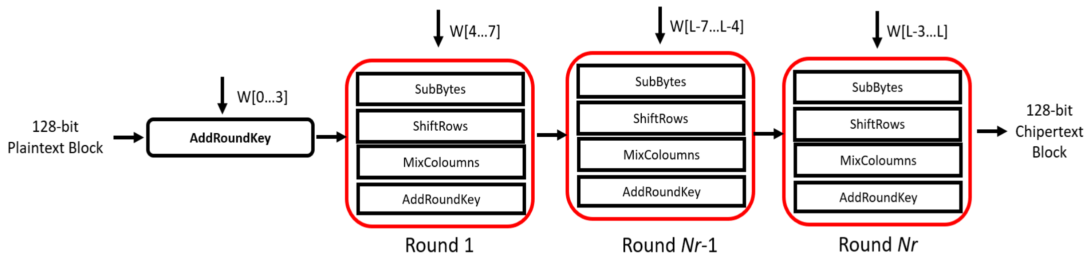


Fig 2. The diagram illustrates the encryption mechanism of the AES-128 Method.

3. RESULTS AND DISCUSSION

The research encrypts the message transformation process with AES. Encryption transforms plaintext into ciphertext. The encryption process involves transforming text messages into ASCII format at various stages. An illustration of message transformation in text format (string) and key to hex number code can be described through the following stages:

Plaintext : ilmukomputerjaya
Key : KriptografiAESku
Tables 1 and 2 show the transformation process of the plaintext and key into hexadecimal number format.

Table 1. Illustrates the conversion of a plaintext in text format to a Hexadecimal number format

Plaintext	i	l	m	u	k	o	m	p	u	t	e	r	j	a	y	a
Hexadecimal	69	6C	6D	75	6B	6F	6D	70	75	74	65	72	6A	61	79	61

Table 2. Conversion of key in text format to hexadecimal number format

Key	K	r	i	p	t	o	g	r	a	f	i	A	E	S	k	u
Hexadecimal	4B	72	69	70	74	6F	67	72	61	66	69	41	45	53	6B	75

The next step is to write the plaintext and key in the form of a matrix with an ordo of 4x4. Matrix Table The results of writing plaintext and keys can be explained in Table 3 as follows:

Table 3. Matrix Table of plaintext and key writing results

***Plaintext* (input)**			
69	6C	6D	75
6B	6F	6D	70
75	74	65	72
6A	61	79	61

***Key* (input)**			
4B	74	61	45
72	6F	66	53
69	67	69	6B
70	72	41	75

The plaintext and key matrices share the same order dimension, enabling the execution of bitwise arithmetic operations. To obtain the AddroundKey matrix, we perform the bitwise operation using the XOR function. Equation 3 provides the AddroundKey matrix.

$$\text{RoundKey}\ (i,j) = plaintext\,(i,j) \oplus Key\ (i,j) \tag{3}$$

Therefore, the XOR operation process will initialize the plaintext matrix [1,1] and key [1,1] based on the implementation of Equation (3). Bits with the same row and column position are the focus of the XOR operation. The XOR operation will illustrate the result of bit 0 if the plaintext and key binary inputs have the same value. In addition, the XOR operation will produce a value of 1 if there is a difference in input between the plaintext and key input bits. The Addroundkey matrix table at the initial stage of computation (Round 0) can be described in Table 4 as follows:

Table 4. Addroundkey matrix table at the initial stage of computation (Round 0)

AddRoundKey Round 0			
22	1F	14	2F
1E	00	12	32
04	0A	0C	12
05	02	33	14

In the formation of the key schedule before the first-round process, the key schedule will be initialized by using the key as the main key. This ensures that the key schedule containing the primary key prior to the first round shares the same structure. The next step involves the substitution process, which uses s-box and the available AddroundKey components. Each Addroundkey component will serve as a substitution reference in the implementation of the substitution operation, drawing from the s-box table. The addroundkey matrix (1,1) with the value *02h* will adjust the value *0* to the row side of the s-box table component, while the value 2 represents the column position in the s-box table. Table 5 describes the overall results of the s-box substitution in round 1.

Table 5. Round 1 Matrix of SubBytes Operation Result

	SubBytes			
Round 1	93	C0	FA	15
	72	63	C9	23
	F2	67	FE	C9
	6B	77	C3	FA

The process continues by implementing the shiftrows process, which will provide matrix position transformations based on the results of the subBytes process. The first row in the data matrix remains fixed and unchanged, allowing the composition to alter in accordance with the provisions. The second row is shifted one position to the left, and the third row is shifted three positions to the left. Table 6 describes the outcomes of the matrix component's position transformation.

Table 6. Round 1 Matrix Result of ShiftRows Operation

	SubBytes				ShiftRows			
Round 1	93	C0	FA	15	93	C0	FA	15
	72	63	C9	23	63	C9	23	72
	F2	67	FE	C9	FE	C9	F2	67
	6B	77	C3	FA	FA	6B	77	C3

The next step is to apply the multiplication of the MixColoums matrix with the ShiftRows matrix. So, the matrix operation is done bitwise between the Mixcoloums matrix and the ShiftRows matrix. Furthermore, the concept of row and column multiplication guides the matrix arithmetic operation, or the multiplication process. The XOR process will intersperse each multiplication result. Table 7 provides a description of the multiplication results between the MixColoums and ShiftRows matrices.

Table 7. Results of Mixcoloums

Mix Coloums			
9C	79	0F	18
B6	62	C6	9B
02	3D	BF	F7
DC	8D	2A	B7

The addrounkey process, which describes the XOR operation between the processed data block and the round key, is the next step in the process. Table 8 provides an illustration of the addrounkey process.

Table 8. Addroundkey Matrix Result

Addrounkey			
3B	AA	BD	EF
BB	00	C2	CC
F6	AE	45	66
C2	E1	07	EF

The roundkey process operates in a ten-round cycle, generating 10 rounds until it achieves AES 128 encryption outcomes. The final round of AES 128 encryption yields a chipertext, which users can utilize to transmit data to their intended recipients. The roundkey results are displayed in Table 8. The process continues until the tenth iteration. The following description of Table 9 provides a detailed illustration of the process and computation of AES-128.

Table 9. AddRoundKey Computation Results on Each AES-128 Round Cycle

Round	AddRoundKey			
	22	1F	14	2F
Round 0	1E	00	12	32
	04	0A	0C	12
	05	02	33	14
	3B	AA	BD	EF
Round 1	BB	00	C2	CC
	F6	AE	45	66
	C2	E1	07	EF
	...	...	....	...
Round ...	...	...	...	...
	...	...	...	...
	...	...	...	...
	F0	23	E8	DF
Round 10	B3	8E	B4	20
	D6	F7	87	14
	88	3F	46	D1

Based on the computational results of AES-128 encryption using ten computational cycles, a description of the transformation of plaintext into chipertext is produced, as illustrated in Table 10.

Table 10. Illustration of Plaintext Transformation into Chipertext

Categories	Description
Plaintext	69 6c 6d 75 6b 6f 6d 70 75 74 65 72 6a 61 79 61
Key	4b 72 69 70 74 6f 67 72 61 66 69 41 45 53 6b 75
Chipertext	F0 B3 D6 88 23 8E F7 3F E8 B4 87 46 DF 20 14 D1

In designing a text data security system application using the AES-128 algorithm, a use case diagram is used to describe the interaction between the user and the developed application. This application has a menu that includes encryption, decryption, and exit. Therefore, Figure 3 provides a description of the developed application's use case diagram design.

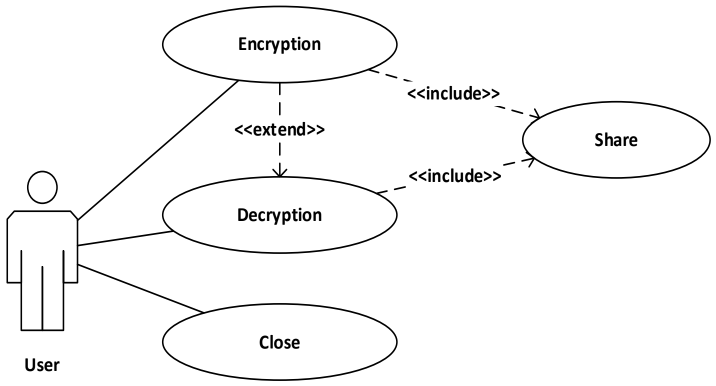
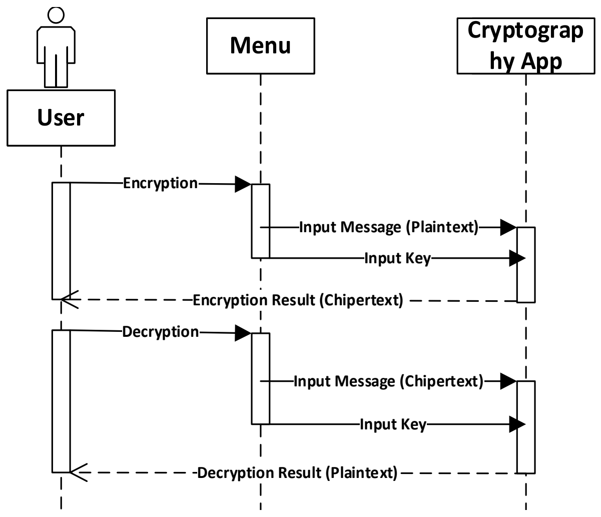
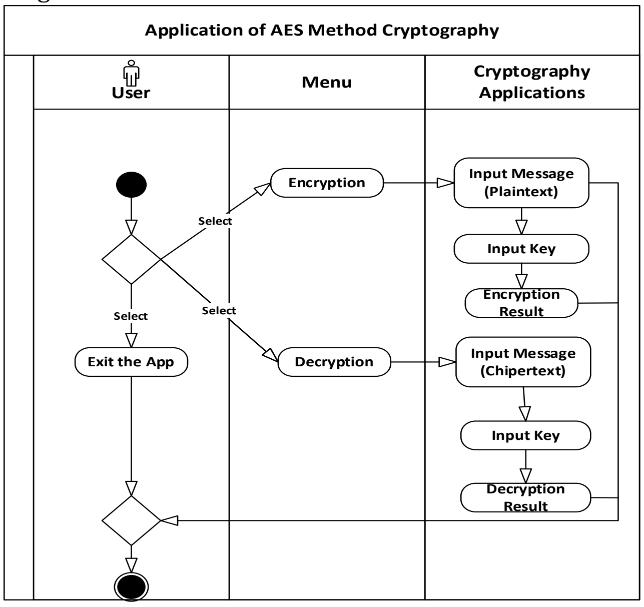


Fig 3. Use Case Diagram of Application Design

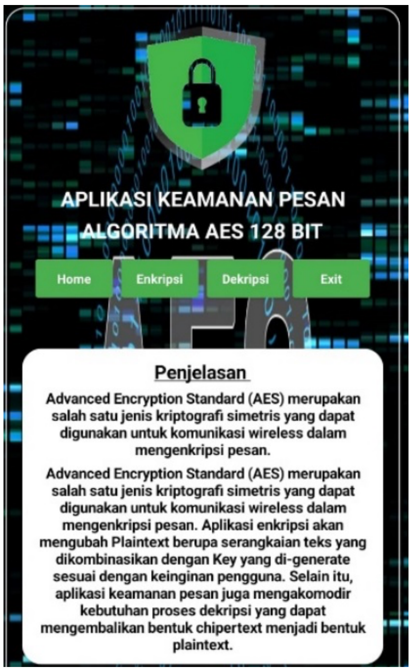
Meanwhile, sequence diagrams serve as illustrations in defining interactions between users and systems, which involve information parameters needed to carry out text data security application functions (Andri & Sitanggang, 2023). Users can choose the available menus according to their needs, both for the purposes of encrypting text data and for carrying out the process of converting ciphertext to plaintext by utilizing the decryption feature available in the text data security application. The application allows the user to enter text data as part of its encryption function. This text data can be a series of words or sentences composed of letters (alphanumeric), numbers (numerical), or a combination of both (alphanumeric). Next, within the decryption menu, the user can replicate the ciphertext they received from the sender through social media, the application's integration medium, and subsequently perform the decryption process on the ciphertext to retrieve the previously sent plaintext. The use of buttons is limited to numeric button types only. The sender and recipient must agree on this in order to decrypt the received or sent ciphertext. Figure 4 provides a detailed explanation of the sequence diagram in the application plan.


User
Menu
Cryptography App
Encryption
Input Message (Plaintext)
Input Key
Encryption Result (Chipertext)
Decryption
Input Message (Chipertext)
Input Key
Decryption Result (Plaintext)
Fig 4. Application Design Sequence Diagram

Activity diagrams can also illustrate user activity descriptions. The user's needs greatly influence the range of activity processes he can execute. The activity diagram work description is generally identical to the illustrations shown in the sequence diagram and use case diagram. This is due to the three model description functions in UML. Figure 5 illustrates the activity diagram model in detail within the application design.


Application of AES Method Cryptography
User
Menu
Cryptography Applications
Select
Select
Select
Exit the App
Encryption
Decryption
Input Message (Plaintext)
Input Key
Encryption Result
Input Message (Chipertext)
Input Key
Decryption Result
Fig 5. Activity Diagram Application

In the built application, when users enter the application, they will be faced with an initial display that directly represents the application's functions. Figure 6 displays the application view. On this page, users can encrypt text data that will be sent via social media that can be selected as needed. The encryption process begins by entering the data to be encrypted. Next, the user must enter the key, using a random key sequence and ensuring that the resulting key has a numeric component. The encryption page can be illustrated in Figure 7.

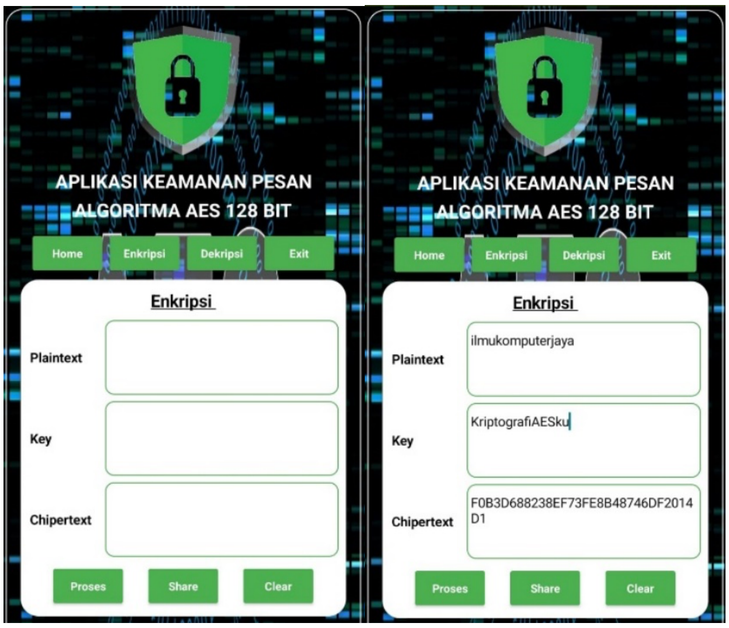


Penjelasan

Advanced Encryption Standard (AES) merupakan salah satu jenis kriptografi simetris yang dapat digunakan untuk komunikasi wireless dalam mengenkripsi pesan.

Advanced Encryption Standard (AES) merupakan salah satu jenis kriptografi simetris yang dapat digunakan untuk komunikasi wireless dalam mengenkripsi pesan. Aplikasi enkripsi akan mengubah Plaintext berupa serangkaian teks yang dikombinasikan dengan Key yang di-generate sesuai dengan keinginan pengguna. Selain itu, aplikasi keamanan pesan juga mengakomodir kebutuhan proses dekripsi yang dapat mengembalikan bentuk chipertext menjadi bentuk plaintext.

Fig 6. Application Start View



Plaintext: ilmukomputerjaya

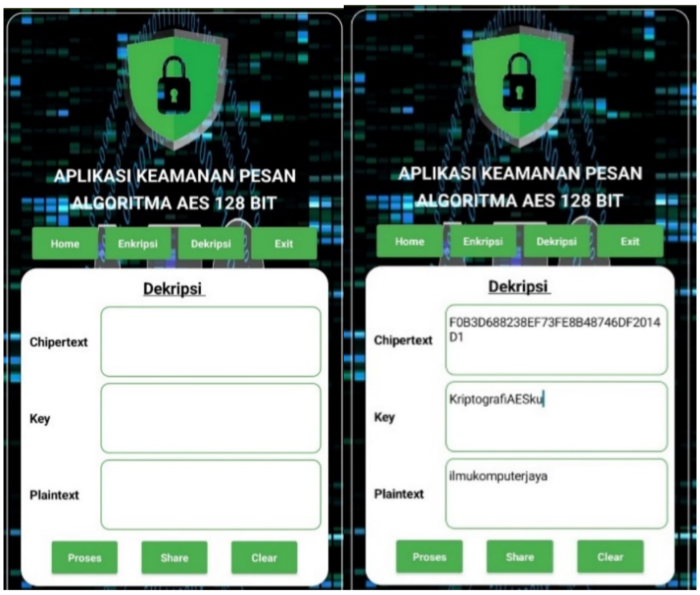
Key: KriptografiAESku

Chipertext: F0B3D688238EF73FE8B48746DF2014D1

Fig 7. Application Encryption Page Display

Users can select and encrypt text data on this page for social media transmission as needed. Entering the data for encryption initiates the encryption process. Next, the user must enter the key, using a random key sequence and ensuring that the resulting key has a numeric component. On the display of the encryption page there are several input columns. The "Enter Message" column is a column provided by the application for users to input text data to be sent. While the key column is a column that can be used by the user in setting the key that will be used in the encryption and decryption process, while the "Encryption Result" column is a column that functions as a viewer of the encryption results. The

"Encryption Result" column will have a value in the form of a row of numeric characters (numeric), letters (alphabet) or even a combination of the two, namely, letters and numbers (alphabetnumerik) with the possibility of inserting padding characters from the conversion of the inputted plaintext to form the chipertext written in the "Encryption Result" column. On this page the user can perform the encryption process by inputting the chipertext in the "Enter Message" column. Usually, chipertext input in the decryption process can be a series of input data consisting of alphabetic and numeric combinations and allows the application of padding in the chipertext in the form of the "=" symbol. The display of the decryption page and the decryption page with input can be illustrated through Figure 8.



F0B3D688238EF73FE8B48746DF2014D1

KriptografiAESku

ilmukomputerjaya

Fig 8. Application Decryption Page Display

Based on the application design and implementation that have been carried out, several tests are carried out with the aim of providing a representation of the application performance results in accordance with the plan. Each activity process and available feature, represented by success and failure conditions in the testing procedure, inform the description of application testing.

1. Success Condition: This represents the condition where the application runs as expected by fulfilling each component and stage required in the UML model.
2. Failure Condition: Illustrates a situation where the application cannot function according to its purpose due to a lack or non-fulfillment of the requirements specified in the model in the application design planning section. Table 9 displays the testing results for text data security applications using the AES algorithm.

Table 9. AES-128 Application Testing

Tested Module	Testing Procedure	Input	Output	Description
Encryption Process	1. Open the app 2. Select the encryption menu 3. Input a message “ilmukomputerjaya” 4. Input *key* “KriptografiAESku”, 5. Click Process	Plaintext: “ilmukomputerjaya” Key: “KriptografiAESku”	The encryption process was successful	Valid
Decryption Process	1. Open the app 2. Select the decryption menu 3. Input *Chipertext* “F0B3D688238EF73F	*Plaintext:* “ilmukompuuterjaya”, *Key*:	The decryption process was successful	Valid

Tested Module	Testing Procedure	Input	Output	Description
	E8B48746DF2014D1 ", 4. *Key:* "KriptografiAESku", 5. Click Process	"KriptografiAESku"		

4. CONCLUSION

A text data security application with the AES algorithm guarantees the security of text data sent via communication networks on mobile devices. The developed application utilizes the AES algorithm to convert plaintext into ciphertext, masking text data with a distinct code. It restricts access to individuals who possess the encrypted text data key, enabling the data to revert to its original form. This application is adapted to the specifications aThe application adapts to the specifications and compatibility of commonly used mobile devices, features a consistent and easy-to-use User Interface (UI) design, and simplifies the encryption and decryption process using the AES algorithm. uilt has features that function as expected. The developed application serves as a viable choice for safeguarding text data during transactions in communication networks.

REFERENCES

Andri, R. H., & Sitanggang, D. P. (2023). Sistem Penunjang Keputusan (SPK) Pemilihan Supplier Terbaik Dengan Metode MOORA. *Jurnal Sains Informatika Terapan*, *2*(3), 79–84.

Balli, F., & Banik, S. (2019). Six shades of AES. *Progress in Cryptology–AFRICACRYPT 2019: 11th International Conference on Cryptology in Africa, Rabat, Morocco, July 9–11, 2019, Proceedings 11*, 311–329.

Dadkhah, M., Lagzian, M., & Santoro, G. (2019). How can health professionals contribute to the internet of things body of knowledge: A phenomenography study. *VINE Journal of Information and Knowledge Management Systems*, *49*(2), 229–240. https://doi.org/10.1108/VJIKMS-10-2018-0091

Freyre, P., Cuellar, O., Díaz, N., & Alfonso, A. (2020). From AES to Dynamic AES. *Journal of Science and Technology on Information Security*, *1*(11), 11–22.

Kaffah, F. M., Gerhana, Y. A., Huda, I. M., Rahman, A., Manaf, K., & Subaeki, B. (2020). E-Mail message encryption using advanced encryption standard (AES) and huffman compression engineering. *Proceedings - 2020 6th International Conference on Wireless and Telematics, ICWT 2020*. https://doi.org/10.1109/ICWT50448.2020.9243651

Kumar, K., Ramkumar, K. R., & Kaur, A. (2020). A Design Implementation and Comparative Analysis of Advanced Encryption Standard (AES) Algorithm on FPGA. *ICRITO 2020 - IEEE 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)*, 182–185. https://doi.org/10.1109/ICRITO48877.2020.9198033

Liu, S., Li, Y., & Jin, Z. (2023). Research on Enhanced AES Algorithm Based on Key Operations. *2023 IEEE 5th International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, 318–322. https://doi.org/10.1109/ICCASIT58768.2023.10351719

Lytvyn, V., Peleshchak, I., Peleshchak, R., & Vysotska, V. (2019). Information Encryption Based on the Synthesis of a Neural Network and AES Algorithm. *2019 3rd International Conference on Advanced Information and Communications Technologies, AICT 2019 - Proceedings*, 447–450. https://doi.org/10.1109/AIACT.2019.8847896

Manullang, S. F., & Allwine, J. S. (2023). *Implementasi Kriptografi Pengamanan Data File Dokumen Menggunakan Algoritma Advanced Encryption Standard Mode Chiper Block Chaining*.

Mohammed, N. Q., Amir, A., Ahmad, B., Salih, M. H., Arrfou, H., Thalji, N., Matem, R., Abbas, J. K. K., Hussien, Q. M., & Abdulhassan, M. M. (2023). A Review on Implementation of AES Algorithm Using Parallelized Architecture on FPGA Platform. *2023 IEEE International Conference on Advanced Systems and Emergent Technologies (IC_ASET)*, 1–6. https://doi.org/10.1109/IC_ASET58101.2023.10150938

Mukwevho, N., & Chibaya, C. (2020). Dynamic vs Static Encryption Tables in DES Key Schedules. *2020 2nd International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, 1–5.

Mulud Muchamad, R., Asriyanik, A., & Pambudi, A. (2023). Implementasi Algoritma Advanced Encryption Standard (Aes) Untuk Mengenkripsi Datastore Pada Aplikasi Berbasis Android. *Jurnal Mnemonic*, *6*(1), 55–64. https://doi.org/10.36040/mnemonic.v6i1.5889

Nababan, S. A. M., & Setyadi, R. (2022). Pengimplementasian Algoritma RSA Untuk Mengamankan E-mail Menggunakan Outlook dan OpenPGP. *Resolusi: Rekayasa Teknik Informatika Dan Informasi*, *3*(2), 23–28.

Nino, B. E. (2023). Perbandingan performa algoritma aes dan twofish menggunakan metode strict avalanche criterion pada nomor induk kependudukan indonesia. *Jurnal Teknologi Informasi*, *9*(1), 19–29.

Rahman, Z., Hasanuddin, T., & Abdullah, S. M. (2020). Implementasi Metode Enkripsi dan Deskripsi File menggunakan Algoritma Twofish. *Buletin Sistem Informasi Dan Teknologi Islam (BUSITI)*, *1*(2), 66–70.

Su, N., Zhang, Y., & Li, M. (2019). Research on data encryption standard based on AES algorithm in internet of things environment. *Proceedings of 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2019*, *Itnec*, 2071–2075. https://doi.org/10.1109/ITNEC.2019.8729488

Verma, R., & Sharma, A. K. (2020). Cryptography: Avalanche effect of AES and RSA. *International Journal of Scientific and Research Publications*, *10*(4), 119–122.

Wongkhamdi, T., Cooharojananone, N., & Khlaisang, J. (2020). E-commerce competence assessment mobile application development for SMEs in Thailand. *International Journal of Interactive Mobile Technologies*, *14*(11), 48–75. https://doi.org/10.3991/ijim.v14i11.11358