Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th June 2024. Vol. 3. No. 3; e-ISSN: 2808-4519

Enhancing AES Key Generation Using Diffie-Hellman Method for Image Security

Rifdahtul Ghinaa Sinambela[1*], Achmad Fauzi[2], Husnul Khair[3]

[1, 2, 3] Informatic Engineering, STMIK Kaputama
rgs011219@gmail.com[1], fauzyrivai88@gmail.com[2], Husnul.khair@gmail.com[3]*

Abstract

The increasing need for secure image transmission has led to the development of cryptographic techniques that combine encryption algorithms and secure key exchange methods. This paper proposes a system for enhancing the security of digital images by using the Advanced Encryption Standard (AES) combined with the Diffie-Hellman method for key generation. AES is applied for encryption and decryption, while Diffie-Hellman ensures the secure exchange and generation of secret keys. The system, developed in Microsoft Visual Basic .NET 2010, was tested on various image formats and demonstrated effective protection against brute force attacks. This approach ensures secure digital image transmission and confidentiality.

Keywords: *AES; Diffie-Hellman; Encryption; Image Security; Key Generation*

1. Introduction

Image files often contain confidential or sensitive information. Efforts to protect digital image files are essential to prevent misuse, information theft, and alteration or manipulation of identity, ensuring that unauthorized parties cannot access the files for malicious purposes. Therefore, effective methods are needed to secure digital image files. One such method is using cryptographic algorithms, specifically the AES algorithm with Diffie-Hellman. The AES algorithm can be applied in various encryption applications, data communication, and data storage by encrypting information so that only authorized parties with the correct key can read or decrypt the data. In the AES encryption process, the algorithm goes through several complex mathematical operations. However, if the encryption key is too short and follows a predictable pattern, it can become vulnerable to brute force attacks, where the attacker tries all possible keys until the correct one is found, thereby allowing the attacker to solve the entire AES algorithm system. To maintain the confidentiality and authenticity of digital image security, a key generation process using the Diffie-Hellman method is necessary. The Diffie-Hellman method presents challenges in calculating discrete logarithms (key agreement scheme) and can create a shared secret key between the sender and the receiver. As a result, the AES algorithm is used in the encryption and decryption processes of the image file, while the Diffie-Hellman method is applied for secure key exchange, generating a key that secures the AES algorithm.

2. Theoretical Foundation

2.1. Cryptography

Cryptography is the study of mathematical techniques that deal with aspects of information security such as confidentiality, data integrity, and authentication. Cryptography is the process of using various techniques or sciences to keep messages secure. Cryptographic algorithm is a mathematical function used for encryption and decryption. There are two interconnected functions, one for encryption and one for decryption [1].

To be able to run well in the cryptographic process there must be four main elements in it, which are most related to each other, namely:

1. Plain Text; Is the initial message or the original message sent in the communication process. This Plain Text is then encrypted and decrypted.
2. Cipher Text; Is a hidden message, namely the original message (Plain Text) that has been encrypted in the cryptography process. This Cipher Text can be changed back to its original form (Plain Text) utilizing the Key that has been provided.
3. Cryptography Key; This is the key used to perform encryption and description in the cryptography process. Without the same key, the encryption and description process cannot be done properly. The key is solid information that controls the cryptography process.
4. Encryption Decryption Algorithm; The last component that is equally important in the cryptography process is the algorithm used for encryption and decryption [2].

2.2. Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES) is a cryptographic algorithm that can be used to secure data. The AES algorithm is a block symmetric chipertext that can encrypt (encipher) and decrypt (decipher) information. Encryption changes data that can no longer be read called ciphertext, on the other hand decryption is changing ciphertext data into its original form which we know as plaintext. The AES algorithm uses cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data [2].

The AES algorithm encryption process consists of 4 types of bytes transformation, namely SubBytes, ShiftRows, Mixcolumns, and AddRoundKey. At the beginning of the encryption process, the input that has been copied into the state will undergo AddRoundKey byte transformation. After that, the state will undergo the transformation of SubBytes, ShiftRows, MixColumns, and AddRoundKey repeatedly as many as Nr. This process in the AES algorithm is called a round function. The last round is somewhat different from the previous rounds where in the last round, the state does not undergo the MixColumns transformation. Illustration of the AES encryption process can be described as in Figure 1.

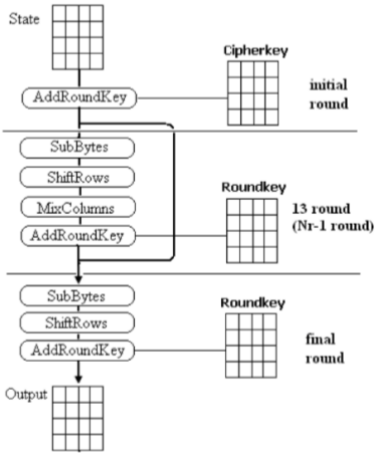

Fig. 1: AES Encryption Process Diagram

Decryption Process of AES Algorithm the cipher transformations can be reversed and implemented in the opposite direction to produce an easy-to-understand inverse cipher for the AES algorithm. The byte transformations used in the inverse cipher are InvShiftRows, InvSubBytes, InvMixColumns, and AddRoundKey. The decryption algorithm can be seen in Figure 2.

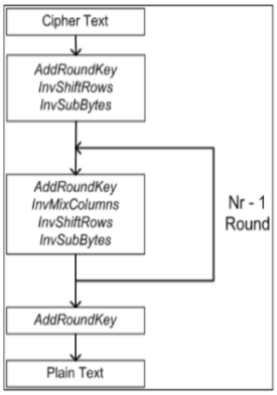

Fig. 2: AES Decryption Process

2.3. Diffie-Hellman Method

This algorithm was first introduced by Whitfield Diffie and Martin Hellman in 1975. They were both researchers at Stanford University. They introduced this algorithm to provide a solution for the confidential exchange of information. This algorithm is not based on the encryption and decryption process, but rather on the mathematical process carried out to generate a secret key that can be shared freely without having to worry because the secret key can only be decrypted only by the sender and recipient of the message. The basis of this algorithm is the basic math of exponent algebra and modulus arithmetic [3], [4].

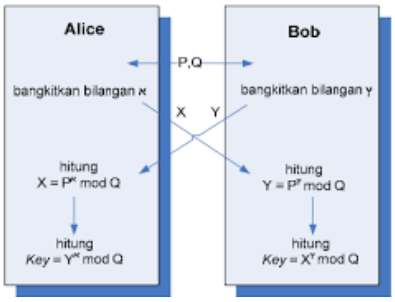

Fig. 3: Diffie-Hellman Key Exchange Protocol

3. Research Methods

The steps taken by the author to solve the design problem of improving the key generation of the AES algorithm with the Diffie-Hellman method in the image security encryption and decryption process consist of several stages [5], [6].

3.1. Analysis of Diffie Hellman Method and AES Algorithm

Before performing the process of securing image files in manual calculations, image files are first converted into hexadecimal numbers using the binary viewer in the HxD software, the results of image file conversion. Some hexadecimal numbers are taken to be used as plaintext in the AES algorithm encryption process, namely: 01 5E 00 00 FF E1 12 12 45 78 69 66 00 00 4D 4D.

3.2. Diffie-Hellmen Method Key Generation Calculation

The Diffie-Hellman algorithm is used for key generation which will later be used in the encryption and decryption process using the AES algorithm. The key exchange process stage using the Diffie-Hellman algorithm, the process begins when the sender and receiver of the message exchange keys by agreeing on the value of the prime number p and the same number g. The sender chooses a random number (a) and then calculates it with the formula: A=ga mod p and the receiver chooses a random number (b) then calculates with the formula: B = gb mod p [7].
p = 337, g = 10
a = 51 (public): sender, b = 55 (public): receiver
$A=g^a \bmod p = 10^{51} \bmod 337 = 97$, $B = g^b \bmod p = 10^{55} \bmod 337 = 114$
$Xa = B^a \bmod p = 114^{51} \bmod 337 = 66$, $Xb = A^b \bmod p = 97^{55} \bmod 337 = 66$
Calculating the private key value Y: $Y = g^x \bmod p = 10^{66} \bmod 337 = 25$

Figure 4 illustrates the key exchange process using the Diffie-Hellman algorithm. Determine prime numbers p and g. The sending party determines a random number a and calculates A=ga mod p. The receiving party determines a random number b and calculates B=gb mod p. Then the sending and receiving parties exchange keys and will generate a shared private key. The key will be called to be used in the encryption and decryption process using the AES algorithm.

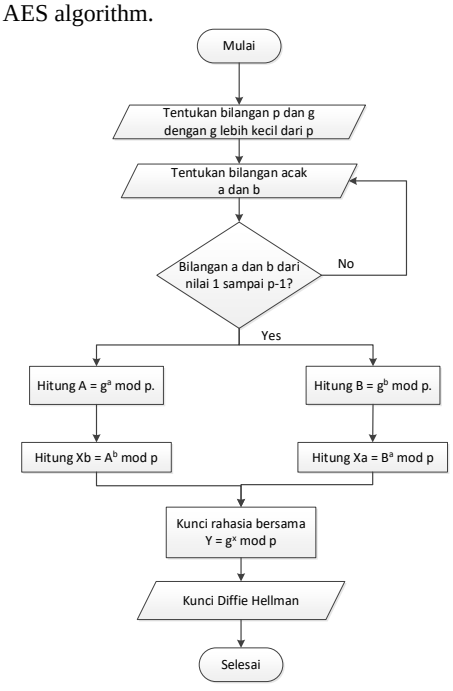

Fig. 4: Diffie-Hellman Key Generation Flowchart

3.3. Encryption and Decryption Process of AES Algorithm

The key results from Diffie-Hellman will be used for the key in the AES encryption and decryption process. The AES key is 25
The outline of the AES Rijndael algorithm that operates on 128-bit blocks with 128-bit keys (excluding the round key generation process) is as follows:

1. AddRoundKey, XORs the initial plaintext with the cipher key.
2. Nr-1 rounds. The processes performed in each round are:
    a. SubBytes is byte substitution using the substitution table (S-Box).
    b. ShiftRows is shifting the rows of the state array by wrapping.
    c. MixColumns is randomizing the data in each column of the state array.
    d. AddRoundKey is to XOR between the current state round key.
3. Final round, the process for the last round:
    a. SubBytes
    b. ShiftRows
    c. AddRoundKey

Figure 5 illustrates the encryption and decryption process using the AES algorithm.

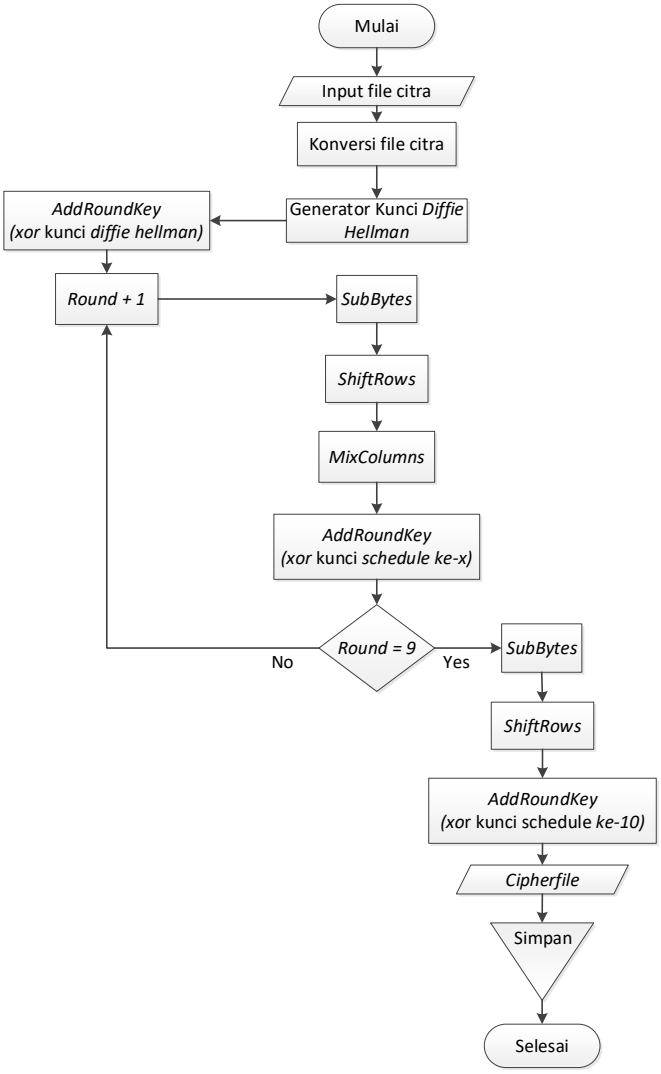

Fig. 5: AES Algorithm Encryption Flowchart

4. Testing

Digital image file security application which uses a key generator, namely the Diffie-Hellman method with encryption and decryption techniques in the AES algorithm, which is built with the aim of keeping files safer and preventing information and identity theft. This is done by encrypting the image file and can be decrypted as proof of ownership of the image file.
This trial is conducted to ensure the key generation process using the Diffie-Hellman method runs according to the design. The generated key is then used for file encryption and decryption. Previously, the sender and receiver input the numbers a, b, p, and q to perform the key generation process using the Diffie-Hellman algorithm. This interface is equipped with a Generate Key button that serves to generate public keys and private keys. After this button is pressed, the system automatically performs calculations to generate a Shared Key, which is the key that will be used in the encryption and decryption stages.

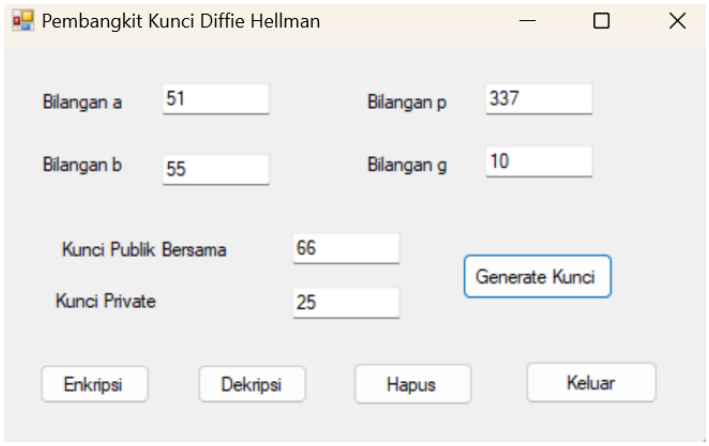

Fig. 6: Diffie-Hellman Key Generation Testing Process

This trial aims to ensure that the image file can be encrypted properly using the AES algorithm. Files that have been encrypted cannot be opened or accessed without the right key. This form comes with a Select File button that allows the user to browse and select the image file that they want to encrypt. Once the file is selected, the Encrypt button will be activated, and the user can start the encryption process. The key used for this process is the key generated from the Diffie-Hellman Key Generator form. Once the Encrypt button is pressed, the program will encrypt the file and display a confirmation message that the file has been successfully encrypted. Files that have been encrypted cannot be opened without going through the decryption process using the same key.

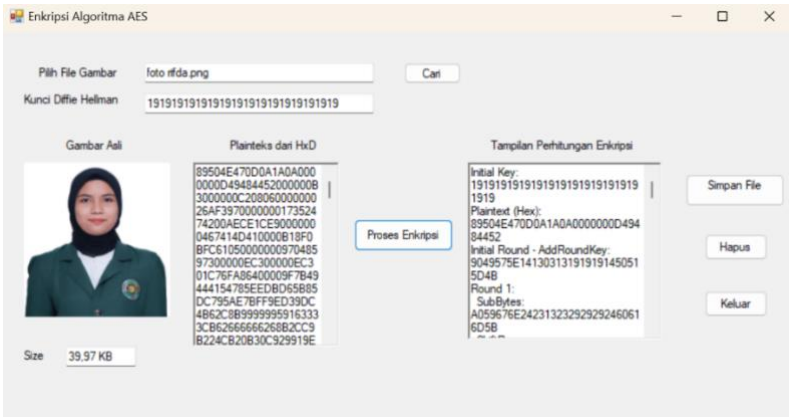

Fig. 7: Encryption Testing Process

This test ensures that the encrypted image file can be decrypted using the correct key. The user selects the encrypted file via the Select File button, then presses the Decrypt button. The system will verify whether the Shared Key used matches the key used during the encryption process. If the key is correct, the image file will be decrypted back to its original form, and the result will be saved in the specified folder.

Fig. 8: Decryption Testing Process

5. Conclusion

The AES algorithm is used in the encryption and decryption process, with the Diffie-Hellman method as the key generator. This implementation runs well and in accordance with the design. Testing of image files shows that the encryption results provide strong protection for the file. Image files that have been encrypted can only be accessed by parties who have the right key. The integration of

Diffie-Hellman as a key generator provides an additional layer of security due to the difficulty in solving the discrete logarithm used in this algorithm. However, there is a difference in file size between the original image and the decrypted image. Although the decryption successfully restores the image to the original file form, the file size of the decrypted result is slightly different due to the changes that occur during the encryption and decryption process. We recommend that in future research, it is better if the size of the decrypted image file remains the same as the original image file size.

References

[1] A. Fauzi, Y. Maulita, and N. Novriyenni, "Perancangan Aplikasi Keamanan Pesan Menggunakan Algoritma Elgamal Dengan Memanfaatkan Algoritma One Time Pad Sebagai Pembangkit Kunci," *JTIK (Jurnal Tek. Inform. Kaputama)*, vol. 1, no. 1, pp. 1–9, 2017, doi: 10.59697/jtik.v1i1.680.

[2] A. Aisiah Ibrahim, "Perancangan Pengamanan Data Menggunakan Algoritma AES (Advanced Encyption Standard)," *J. Tek. Inform. Stmik Antar Bangsa*, vol. III, no. 1, pp. 53–60, 2017.

[3] M. Mulyadi, "Aplikasi Kriptografi Pesan Teks Menggunakan Algoritma Advanced Encryption Standard 256 Bit (Aes-256) Dan Diffie Hellman," *Sisfo J. Ilm. Sist. Inf.*, vol. 3, no. 2, pp. 23–38, 2019, doi: 10.29103/sisfo.v3i2.6330.

[4] Muhammad Fadillah Azmi, Achmad Fauzi, and Husnul Khair, "Implementation Of Affine Cipher Combination And Merkle Hellman On The Process Digital Image Security", *j. of artif. intell. and eng. appl.*, vol. 2, no. 3, pp. 107–122, Jun. 2023.

[5] Irwansyah, Achmad Fauzi, and Siswan Syahputra, "A Combination Of A Rail Fence Cipher And Merkle Hellman Algorithm For Digital Image Security", *j. of artif. intell. and eng. appl.*, vol. 2, no. 3, pp. 135–143, Jun. 2023.

[6] Pappachan, P., Rahaman, M., Sreerakuvandana, S., Bansal, S. and Arya, V., 2024. Beyond Current Cryptography: Exploring New Frontiers. In *Innovations in Modern Cryptography* (pp. 1-30). IGI Global.

[7] R. . Prastya, A. M. . Pardede, and A. . Fauzi, "Teknik Pembangkit Kunci Algoritma RSA Menggunakan Algoritma Diffie Hellman pada Keamanan Citra", *KAKIFIKOM* , vol. 4, no. 1, pp. 16–22, Apr. 2022.