Login : Jurnal Teknologi Komputer
ISSN : 2302-9692 (print) | 2723-8695 (online)
Vol. 18, No. 02, 2024, pp. 114-119
LOGIN http://login.seaninstitute.org/index.php/Login

Application of Diffie-Hellman Algorithm Method (Key-Exchange) in Cryptography

Sarah Sagala
Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Katolik Santo Thomas, Medan

Article Info

Keywords:

Diffie-Hellman Key Exchange,
Secret Key, Computation Time.

ABSTRACT

In cryptography, key exchange security is essential. Because the message code can only be a message that is analyzed with a key offer that matches this message. Secret messages can be delivered through symmetric cryptographic algorithms with coded keys or with key exchange algorithms. One of the key exchange methods that has been developed is Diffie-Hellman Key-Exchange (DHKE). Diffie-Hellman is a simple method used to generate a key between two parties in order to communicate securely. The two parties running this algorithm have their own secret variables, by exchanging two public variable values and performing calculations using the received variables and secret variables obtained randomly shared secret. The security strength in this algorithm can be improved by combining an asymmetric key algorithm, namely RSA. This algorithm is not only in generating RSA keys, but also in calculating discrete logarithms in RSA and Diffie-Hellman. The results of this algorithm are key strength and confidentiality that can be used in symmetric cryptographic algorithms.

Corresponding Author:

Sarah Sagala
Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Katolik Santo Thomas, Medan
E-mail: sarahsagala246@gmail.com

1. INTRODUCTION

An algorithm is a computer work system that has brainware, hardware and software. (Ng, 2017). While cryptography is an encoding technique to provide extra security in sending data or special messages based on mathematical calculation techniques.(Romindo & Ferawaty, 2021). In the world of Cryptography, the key to encrypt and decrypt a secret message is the most important element.(Jainudin et al., 2015)The key determines whether a ciphertext can be read or not. The secrecy of the key is actually a crucial thing that is important than the secrecy of the ciphertext itself. In the sense that the message (ciphertext) may be leaked but not the key. Various methods are used to maintain the secrecy of the key. Public key cryptography techniques or algorithms are one method developed to overcome this. (key secrecy). Public key cryptography requires two keys, namely a public key that is freely informed and used without secrecy, and a private key that is only used specifically by one person and is never informed to anyone, so that its secrecy is very well maintained. Although public key cryptography algorithms are popular because of this, symmetric key cryptography algorithms are still widely used because they are easier to use, but it is necessary to think about how to maintain the secrecy of the key.(Wahyuni, 2011)The algorithm discovered by Diffie and Hellman is a technique for maintaining the secrecy of symmetric keys.

Previous research Implementation of Diffie-Hellman Algorithm in Least Significant Bit Steganography combines the Diffie-Hellman algorithm with the Least Significant Bit (LSB) steganography technique to hide secret messages in digital images. The Diffie-Hellman algorithm is used for secure key exchange, while LSB steganography is used to insert encrypted data into the

image. Testing is carried out to evaluate the image quality after the steganography process using the PSNR and MSE parameters. The results show that this method can maintain data security with good image quality.(Lie & Alamsyah, nd).

Theoretical Basis and Implementation of Diffie-Hellman Algorithm This research explains the basic theory of Diffie-Hellman algorithm as the first public key exchange protocol that allows two parties to share a secret key over an insecure channel. The security of this algorithm is based on the difficulty of calculating the discrete logarithm of large numbers, making it suitable for use in various symmetric cryptography applications such as AES or DES(2023-Diffie-Hellman-Algorithm, nd).

(Gunawan H et al., 2021)The idea of performing key exchange using the diffie-hellman algorithm is an interesting key algorithm. In theory, this will certainly make the solution more complex, so that the use of symmetric key cryptography can be done more freely.(Hendarsyah & Wardoyo, 2015) There are 3 basic elements that need to be considered when developing and discussing security systems, including: confidentiality, integrity and availability. In this paper, the public key cryptography algorithm used to strengthen the Diffie-Hellman key exchange algorithm is RSA. RSA is used for reasons of its very high level of security.

2. RESEARCH METHODS

The Diffie-Hellman key exchange algorithm is useful for exchanging secret keys when communicating using symmetric cryptography.(Kasim, 2010).Symmetric key cryptography is a method that uses a single key in both the encryption and decryption processes of messages. This algorithm is a little difficult to perform discrete logarithm calculations.(Nisa et al., 2020). Diffie and Hellman are referred to as public key cryptography and are commonly referred to as Diffie Hellman key exchange or Diffie Hellman protocol.(Permana & St, 2013) The purpose of this algorithm is to allow two users to exchange keys securely and can be used to encrypt and decrypt subsequent messages. The algorithm itself is limited to the exchange of secret values.(Hendarsyah & Wardoyo, 2011). The steps are as follows:

1. Alice and Bob are the communicating parties. First, Alice and Bob agree on two large (preferably prime) numbers P and Q, such that P < Q. The values of P and Q do not need to be secret, and Alice and Bob can even discuss them over an insecure channel.
2. Alice generates a large random integer x and sends the following result to Bob: X = Px mod Q.
3. Bob generates a large random integer y and sends the result of the calculation, as follows, to Alice:
   Y = P y mod Q
4. Alice calculates K = Y x mod Q. 5. Bob calculates K' = X y mod Q. If the calculation is done correctly then K = K'. Thus Alice and Bob have the same key without the other party knowing.

DHKE formula

$$A^b \bmod p = g^{ab} \bmod p = g^{ba} \bmod p = B^a \bmod p$$

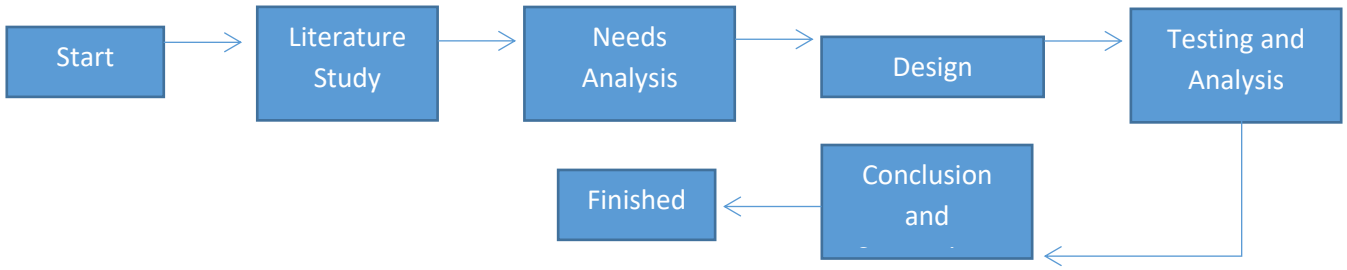

Figure 1 Research method flow

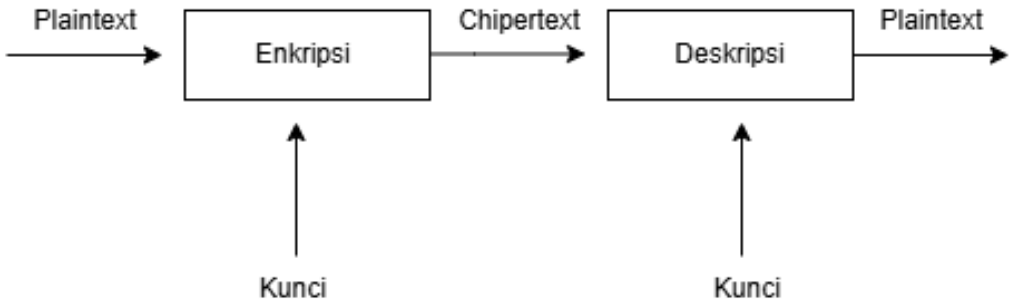

Figure 2 Encryption and Key Description

3. RESULTS AND DISCUSSION

In the manual calculation stage of Diffie-Hellman Key Exchange (DHKE), the first step is to enter a public number agreed upon by both parties, Joko and Winda. They choose a prime number p = 23 and a base g = 5. This value is public and can be known by anyone. Next, each party chooses their private key secretly. Joko chooses private key a = 6, while Winda chooses private key b = 15. Using these private keys, they calculate each other's public keys using the following formula:

a. Joko's public key: A = g^a mod p = 5^6 mod 23 = 8
b. Winda public key: B = g^b mod p = 5^15 mod 23 = 19

After exchanging public keys, Joko and Winda calculate the shared secret key using the following formula:

a. Common key by Joko: K = B^a mod p = 19^6 mod 23 = 2
b. Shared key by Winda: K = A^b mod p = 8^15 mod 23 = 2

The secret key obtained is K = 2.

Next, the encryption process is performed on the secret message "HASUGIAN". Each character in the message is converted to its ASCII value, then encrypted using the formula:

Encrypted ASCII = (ASCII Original + K) mod 256

Table 1 The encryption results can be seen in the following table:

Plaintext	ASCII	Binary	Encryption (M+K) mod 256	Binary Results and Ciphertext
H	72	01001000	(72+2) mod 256 = 74	01001010 (J)
A	65	01000001	(65+2) mod 256 = 67	01000011 (C)
S	83	01010011	(83+2) mod 256 = 85	01010101 (U)
U	85	01010101	(85+2) mod 256 = 87	01010111 (W)
G	71	01000111	(71+2) mod 256 = 73	01001001 (I)
I	73	01001001	(73+2) mod 256 = 75	01001011 (K)
A	65	01000001	(65+2) mod 256 = 67	01000011 (C)
N	78	01001110	(78+2) mod 256 = 80	01010000 (P)

The encryption result is "JCUWIKCP".

For decryption, the formula is used:

Decrypted ASCII = (Encrypted ASCII - K) mod 256

Table 2 The decryption results can be seen in the following table:

Ciphertext	ASCII	Binary	Decryption (CK) mod 256	Binary Results and Plaintext
J	74	01001010	(74-2) mod 256 = 72	01001000 (H)
C	67	01000011	(67-2) mod 256 = 65	01000001 (A)
U	85	01010101	(85-2) mod 256 = 83	

Ciphertext	ASCII	Binary	Decryption (CK) mod 256	Binary Results
W	87	01010111	(87-2) mod 256 = 85	01010011 (S)
I	73	01001001	(73-2) mod 256 = 71	01010101 (U)
K	75	01001011	(75-2) mod 256 = 73	01000111 (G)
C	67	01000011	(67-2) mod 256 = 65	01001001 (I)
P	80	01010000	(80-2) mod 256 = 78	01000001 (A)

The decryption result is "HASUGIAN". Testing using Python was also carried out to verify the results of manual calculations. The test results prove that the Diffie-Hellman algorithm successfully produces identical secret keys on both sides, allowing the encryption and decryption processes to run smoothly and safely.

Testing Using Python

Input

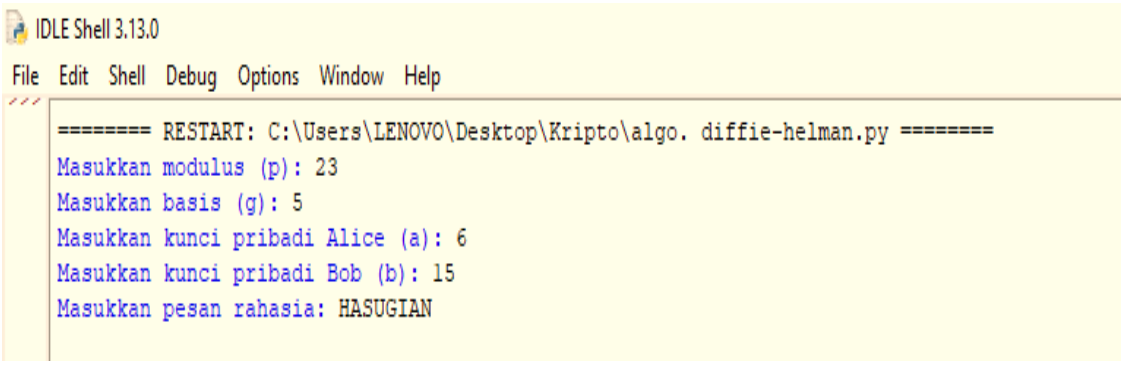

Figure 1 . Public Number input, Private Key input

Python program view running in IDLE Shell version 3.13.0 to implement the Diffie-Hellman Key Exchange (DHKE) algorithm. The program asks the user to enter several inputs, namely modulus p = 23p = 23p = 23, basis g = 5g = 5g = 5, Alice's private key a = 6a = 6a = 6, Bob's private key b = 15b = 15b = 15, and the secret message "HASUGIAN". These inputs are used to calculate each party's public key and generate a shared secret key that will be used for the message encryption and decryption process. These steps show how the Diffie-Hellman algorithm allows two parties to share a secret key securely over an insecure communication channel.

Process

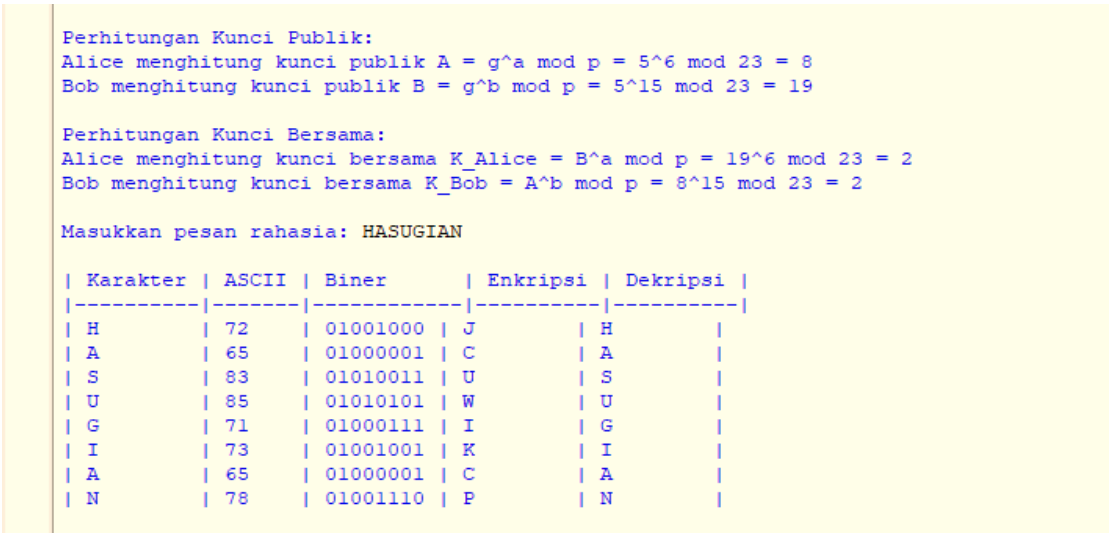

Figure 2. Public Key Calculation, shared key

The figure shows the process of calculating the public key and shared key using the Diffie-Hellman Key Exchange (DHKE) algorithm, as well as the encryption and decryption process of the

message "HASUGIAN". In calculating the public key, Alice calculates the public key A with the formula A=gamod p=56mod 23=8A = g^a \mod p = 5^6 \mod 23 = 8A=gamodp=56mod23=8, while Bob calculates the public key B with the formula B=gbmod p=515mod 23=19B = g^b \mod p = 5^{15} \mod 23 = 19B=gbmodp=515mod23=19. Next, the shared key calculation is done by Alice with the formula KAlice=Bamod p=196mod 23=2K_{Alice} = B^a \mod p = 19^6 \mod 23 = 2KAlice=Bamodp=196mod23=2 and by Bob with the formula KBob=Abmod p=815mod 23=2K_{Bob} = A^b \mod p = 8^{15} \mod 23 = 2KBob=Abmodp=815mod23=2, which produces the same secret key, which is 2. After that, the message "HASUGIAN" is encrypted by adding the key value 2 to the ASCII value of each character, resulting in the ciphertext "JCUWIKCP". In the decryption process, the same key is used to return the ciphertext to the original message by subtracting the key value 2 from the encrypted ASCII value. The table in the figure shows the conversion of each character from ASCII to binary, the encryption process, and the decryption results that return to the original message. This proves the success of the Diffie-Hellman algorithm in producing a secure secret key for the encryption and decryption process.

```
Proses Enkripsi:
Karakter 'H': ASCII = 72, Biner = 01001000 -> Enkripsi dengan kunci 2 menghasilkan 'J'
Karakter 'A': ASCII = 65, Biner = 01000001 -> Enkripsi dengan kunci 2 menghasilkan 'C'
Karakter 'S': ASCII = 83, Biner = 01010011 -> Enkripsi dengan kunci 2 menghasilkan 'U'
Karakter 'U': ASCII = 85, Biner = 01010101 -> Enkripsi dengan kunci 2 menghasilkan 'W'
Karakter 'G': ASCII = 71, Biner = 01000111 -> Enkripsi dengan kunci 2 menghasilkan 'I'
Karakter 'I': ASCII = 73, Biner = 01001001 -> Enkripsi dengan kunci 2 menghasilkan 'K'
Karakter 'A': ASCII = 65, Biner = 01000001 -> Enkripsi dengan kunci 2 menghasilkan 'C'
Karakter 'N': ASCII = 78, Biner = 01001110 -> Enkripsi dengan kunci 2 menghasilkan 'P'

Proses Dekripsi:
Karakter 'J': ASCII = 74, Biner = 01001010 -> Dekripsi dengan kunci 2 menghasilkan 'H'
Karakter 'C': ASCII = 67, Biner = 01000011 -> Dekripsi dengan kunci 2 menghasilkan 'A'
Karakter 'U': ASCII = 85, Biner = 01010101 -> Dekripsi dengan kunci 2 menghasilkan 'S'
Karakter 'W': ASCII = 87, Biner = 01010111 -> Dekripsi dengan kunci 2 menghasilkan 'U'
Karakter 'I': ASCII = 73, Biner = 01001001 -> Dekripsi dengan kunci 2 menghasilkan 'G'
Karakter 'K': ASCII = 75, Biner = 01001011 -> Dekripsi dengan kunci 2 menghasilkan 'I'
Karakter 'C': ASCII = 67, Biner = 01000011 -> Dekripsi dengan kunci 2 menghasilkan 'A'
Karakter 'P': ASCII = 80, Biner = 01010000 -> Dekripsi dengan kunci 2 menghasilkan 'N'
>>>
```

Figure 3. After obtaining the secret key K = 2, both parties can use it to encrypt and decrypt the messages they send.

The figure shows the encryption and decryption process using the Diffie-Hellman algorithm with a shared secret key (K) of 2. In the encryption process, each character in the original message "HASUGIAN" is converted into ASCII and binary values, then encrypted by adding a key value of 2 to each ASCII value, resulting in the ciphertext "JCUWIKCP". For example, the character 'H' which has an ASCII value of 72 is encrypted into 'J' with an ASCII value of 74. The same process is carried out for each character in the message. Furthermore, in the decryption process, the ciphertext "JCUWIKCP" is returned to the original message by subtracting the key value of 2 from each encrypted ASCII value. For example, the character 'J' with an ASCII value of 74 is returned to 'H' with an ASCII value of 72. This process proves that the use of the same secret key allows the encryption and decryption processes to run perfectly, so that the original message can be recovered.

Output

```
Pesan yang dienkripsi: JCUWIKCP
Pesan yang didekripsi: HASUGIAN
Kunci Bersama (K): 2
```

Figure 4. Decrypted Message

The figure shows the results of the encryption and decryption process using the Diffie-Hellman Key Exchange (DHKE) algorithm. In this process, the original message "HASUGIAN" is encrypted using a shared secret key (K) obtained from the calculation of the Diffie-Hellman algorithm, namely K = 2. The result of the encryption process produces the ciphertext "JCUWIKCP". Furthermore, the

decryption process is carried out using the same key, so that the encrypted message is successfully returned to its original form, namely "HASUGIAN". The success of this process proves that the Diffie-Hellman algorithm is able to produce identical secret keys on both communicating parties, so that messages can be sent securely without being known by third parties.

4. CONCLUSION

Of the many possible combinations of algorithms, the combination of key exchange algorithms, the Diffie-Hellman algorithm can be an example of how security increases. The key exchange becomes more complex and more difficult to break by eavesdroppers. The result of this algorithm is multiple security on the message key. This algorithm scheme can be continued with other cryptographic algorithms.

REFERENCES

22-Algoritma-Diffie-Hellman-2023. (n.d.).

Gunawan H, Budi A. S, & Primananda R. (2021). Penerapan Algoritma Diffie Hellman Key Exchange dalam Komunikasi Data Antarnode pada Wireless Sensor Network. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *6 No 1*(1), 197–203.

Hendarsyah, D., & Wardoyo, R. (2011). Implementasi Protokol Diffie-Hellman Dan Algoritma RC4 Untuk Keamanan Pesan SMS. In *IJCCS* (Vol. 5, Issue 1).

Hendarsyah, D., & Wardoyo, R. (2015). Implementasi Protokol Diffie-Hellman Dan Algoritma RC4 Untuk Keamanan Pesan SMS. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, *5*(2), 14–25. https://doi.org/10.22146/ijccs.1997

Jainudin, A., Surya, N. M., & Tito, P. W. (2015). *Perancangan Dan Implementasi Secure Cloud Dengan Menggunakan Diffie-Hellman Key Exchange Dan Serpent Cryptography Algorithm*. *2*(2), 3808–3815.

Kasim, A. A. (2010). Implementasi Kriptosistem Kurva Eliptik Dengan Pertukaran Kunci Diffie-Hellman Pada Data Audio Digital. *Jimt*, *7*(2), 35–42.

Lie, I. R., & Alamsyah, D. (n.d.). *2 ND MDP STUDENT CONFERENCE (MSC) 2023 PENERAPAN ALGORTIMA DIFFIE-HELLMAN PADA STEGANOGRAFI LEAST SIGNIFICANT BIT*.

Ng, F. (2017). Pertukaran kunci Diffie-Hellman dengan Pembangkit Bilangan Acak Linear Congruential Generator (LCG). *Semantika (Seminar Nasional Teknik Informatika)*, *1*(1), 25–29.

Nisa, L., Indriyani, T., & Ruswiansari, M. (2020). Aplikasi Enkripsi Citra dan Teks Menggunakan Algoritma Diffie-Hellman dan ElGamal. In *Jurnal Teknologi dan Manajemen* (Vol. 1, Issue 1).

Permana, A. D., & St, S. (2013). *Pengamanan Sistem Login Aplikasi Menggunakan Protokol ID Based Diffie-Hellman Key Agreement*. *70*, 9–13.

Romindo, & Ferawaty. (2021). Penerapan Algoritma Hybrid RSA Terhadap Pembangkit Kunci Diffie Hellman untuk Sistem Keamanan. *SATIN - Sains Dan Teknologi Informasi*, *7*(2), 92–101. https://doi.org/10.33372/stn.v7i2.783

Wahyuni, A. (2011). Keamanan Pertukaran Kunci Kriptografi dengan Algoritma Hybrid : Diffie-Hellman dan RSA. *Majalah Ilmiah INFORMATiKA*, *2*(2), 15–23.