Randwick International of Social Sciences (RISS) Journal
Vol. 2, No.3, July 2021 | Page: 326-334
ISSN Online: 2722-5674 - ISSN Print: 2722-5666
http://www.randwickresearch.com/index.php/rissj

Data Security Analysis Against Chosen Ciphertext Secure Public Key Attack Using Threshold Encryption Scheme

DOI: https://doi.org/10.47175/rissj.v2i3.275

| Khairunas[1,*] | Muhammad Zarlis[2] | Sawaluddin[2] |

[1] Master of Informatics Engineering Study Program, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Indonesia

[2,3]Departement of Computer Science, Universitas Sumatera Utara, Indonesia

[*]khairunnas422@gmail.com

ABSTRACT

A public key encryption cryptography system can be utilized to generate ciphertext of a message using a public key. However, this public key encryption cryptography system cannot be utilized if you want to generate ciphertext using several different keys. Solving the problems above can use the Chosen Ciphertext Secure Public Key Threshold Encryption scheme but are the securities from Threshold Encryption really strong in securing messages, therefore the above problems can be analyzed for Data Security Against Chosen Ciphertext Secure Public Key Attacks Using Threshold Encryption Schemes. The work process starts from Setup which functions to generate the server's private key and public key. Then, the process is continued with ShareKeyGen which functions to generate private keys based on the user's identity. After that, the process continues with ShareVerify which serves to verify the key generated from the ShareKeyGen process. The process will be continued again with Combine which serves to generate a private key that will be used in the decryption process. After that, the process will continue with the encryption process of the secret message. The ciphertext obtained will be sent to the recipient. The receiver verifies the ciphertext by running ValidateCT. Finally, the ciphertext is decrypted by running Decrypt. The software created can be used to display the workflow process of the Threshold schema. In addition, it makes it easier to test intercepts of ciphertext messages to other users so that generic securities analysis is carried out in testing the resulting ciphertext. The results of the implementation of Threshold Encryption algorithm scheme can protect important personal data, because it involves human rights, namely the right to access, the right to delete, the right to correct, the right to be corrected and the right to transfer personal data safely from attacks.

KEYWORDS

Chosen Ciphertext Secure Public Key Threshold Encryption Without Random Oracle; ciphertext; encryption; decryption

INTRODUCTION

In a *threshold public key encryption* system, the private key is conveyed among n decoding servers so that at slightest k servers are required for the decoding prepare. In a edge encryption framework, an substance called a combiner incorporates a C ciphertext that it needs to unscramble. The combiner sends C to the unscrambling server and gets the unscrambling parcel from at slightest k of n decoding servers, where $k \leq n$. Then, the *combiner* will combine these k decryption parts in a complete decryption of C. Ideally, no other interaction is required on the system, as the servers do not need to communicate with

each other during the decryption process. This *threshold* system is called non-interactive. Sometimes, it is necessary for the decryption threshold to be robust, where if the *decryption threshold* of a substantial ciphertext comes up short, the combiner can recognize the decoding server that provided the invalid decryption part.

This construction consists of two stages. First, the CCA construction from Canetti and partners was developed into a *threshold* system. Second, a *robust threshold* version of the Identity Based Encryption (IBE) scheme is provided by Boneh and Boyen. The *robustness* of this scheme is obtained by including a number of inner checks to the system. In cryptography, a irregular prophet is an prophet (hypothetical dark box) that reacts to each inquiry with a reaction that's chosen arbitrarily and consistently from its output domain. Or in other words, a irregular oracle may be a scientific work that maps a conceivable inquiry with a irregular reaction from its output domain.

Based on the depiction of the foundation above, the problems encountered can be formulated as follows:

1. The ordinary public key encryption scheme cannot use multiple keys to generate the *ciphertext* of a message for distribution and the *threshold encryption* scheme, makes the combiner unable to identify the decryption *server* used to decrypt the message.
2. Proof of generic security by implementing a threshold encryption scheme against attacks from *Chosen Ciphertext Secure (CCA).*

The purpose of this study is to apply a public key encryption threshold scheme to produce ciphertext that can be returned using several keys and can verify the ciphertext that will be used to decrypt the message. The combiner then performs a generic security proof analysis of the threshold encryption algorithm whether it is really safe and strong against attacks from *Chosen Ciphertext Secure* (CCA). Thus, data security in the network system can be avoided from threats such as access, theft, changes to data destruction caused by viruses, sniffing or attacks (Nasution, A.M., et al. 2021).

LITERATURE REVIEW

Threshold Public Key Encryption

A threshold public key encryption system may be a open key framework in which the private key is disseminated among n decoding servers such that a least of k servers is required for the decoding handle. The combiner sends C to the unscrambling server and gets the decoding parcel from at slightest k of n decoding servers,where $k \leq n$. Then, the *combiner* will combine these k pieces of unscrambling into a total decoding of C.

Threshold Identity Based Encryption System

In 2005, Dan Boneh, Xavier Boyen and Shai Halevi introduced a *threshold encryption* system that is secure against *chosen ciphertext attacks* (CCA) without arbitrary oracle. The following describes a concrete *Threshold* IBE (TIBE) system. The operating principle of the TIBE system is as follows:

1. ***Setup***(n, k,Λ): By running GG(Λ) to generate a *bilinear* group G of order prime p > n. Choose a random generator g, g_2, h_1 on G, and a random polynomial of degree k – 1, $f \in Z_p[X]$. Set $\alpha = f(0)$ Z_p and $g_1 = g^{\alpha}$.
   The PK system parameters comprise of PK = (G, g, g_1, g_2, h_1). For i = 1, …, n, the *master key share* (i, SK_i) of *server* i can be characterized as $SK_i = g_2^{f(i)}$. The VK public verification key consists of n-*tuples* ($g^{f(1)}$, …, $g^{f(n)}$).

2. ***ShareKeyGen***(PK, i, SKi, ID). Let PK = (G, g, g_1, g_2, h_1) and take a arbitrary number r ∈ Z_p.. This algorithm generates the private key θ_i = (i, ($w_{i,0}$, $w_{i,1}$)) using the following formula:

$$w_{i,0} = \mathsf{SK}_i \cdot (g_1^{\mathsf{ID}} h_1)^r \qquad w_{i,1} = g^r \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

3. ***ShareVerify***(PK, VK, ID, θi). To confirm that θ_i is a valid private key subset for identity ID, assume VK = (u_1, …, u_n) where $u_i = g^{f(i)}$ and θ_i = (i, ($w_{i,0}$, $w_{i,1}$)).. This algorithm returns valid or invalid based on the taking after conditions:

$$e(u_i, g_2) \cdot e(g_1^{\mathsf{ID}} h_1, w_{i,1}) = e(g, w_{i,0}) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

4. ***Combine***(PK, VK, ID, (θ_1, …, θ_k). If any of θ_1, …, θ_k is invalid, or if two parts θ_i and θ_j produce the same *server* index , then return to ⊥ and exit. If not, then assume θ_i = (i, ($w_{i,0}$, $w_{i,1}$)). Accept that decryption *server* i = 1, …, k is used to generate θ_1, …, θ_k. To determine the private key for ID, suppose λ_1, …, λ_k∈ Z_p is the Lagrange coefficient so that α = f(0) = $\sum_{i=1}^{k} \lambda_i f(i)$. This algorithm generates the private key d_{ID} = (w_0, w_1) using the following equation:

$$w_0 = \prod_{i=1}^{k} w_{i,0}^{\lambda_i} \quad , \quad w_1 = \prod_{i=1}^{k} w_{i,1}^{\lambda_i} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

5. ***Encrypt***(PK, ID, M): Encrypt M ∈ G_1 as ID identity, select a arbitrary number s ∈ Z_p and generate:

$$C = \left(e(g_1, g_2)^s \cdot M, \quad g^s, \quad g_1^{s \cdot \mathsf{ID}} h_1^s \right) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

6. ***ValidateCT***(PK, ID, C). To approve a *ciphertext* C = (A, B, C_1) for an ID identity, this algorithm returns substantial or invalid depending on the following equation:

$$e(B, g_1^{\mathsf{ID}} h_1) = e(C_1, g) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

7. ***Decrypt***(PK, ID, dID, C). To decrypt C = (A, B, C_1) using a private key d_{ID} = (w_0, w_1), to begin with check ValidateCT (PK, ID, C) = valid e(g_1, g_2) . e(g_1^{ID} h_1, w_1) = e(g, w_0).. If any of the checks fail, output ⊥ and exit. Otherwise, generate *plaintext* using the following formula: [1]

$$A \cdot e(C_1, w_1) / e(B, w_0) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6)$$

RESEARCH METHODS

The working procedure of a secure *threshold encryption* system against a *chosen ciphertext attack* (CCA) can be detailed as shown in the following *activity diagram*:

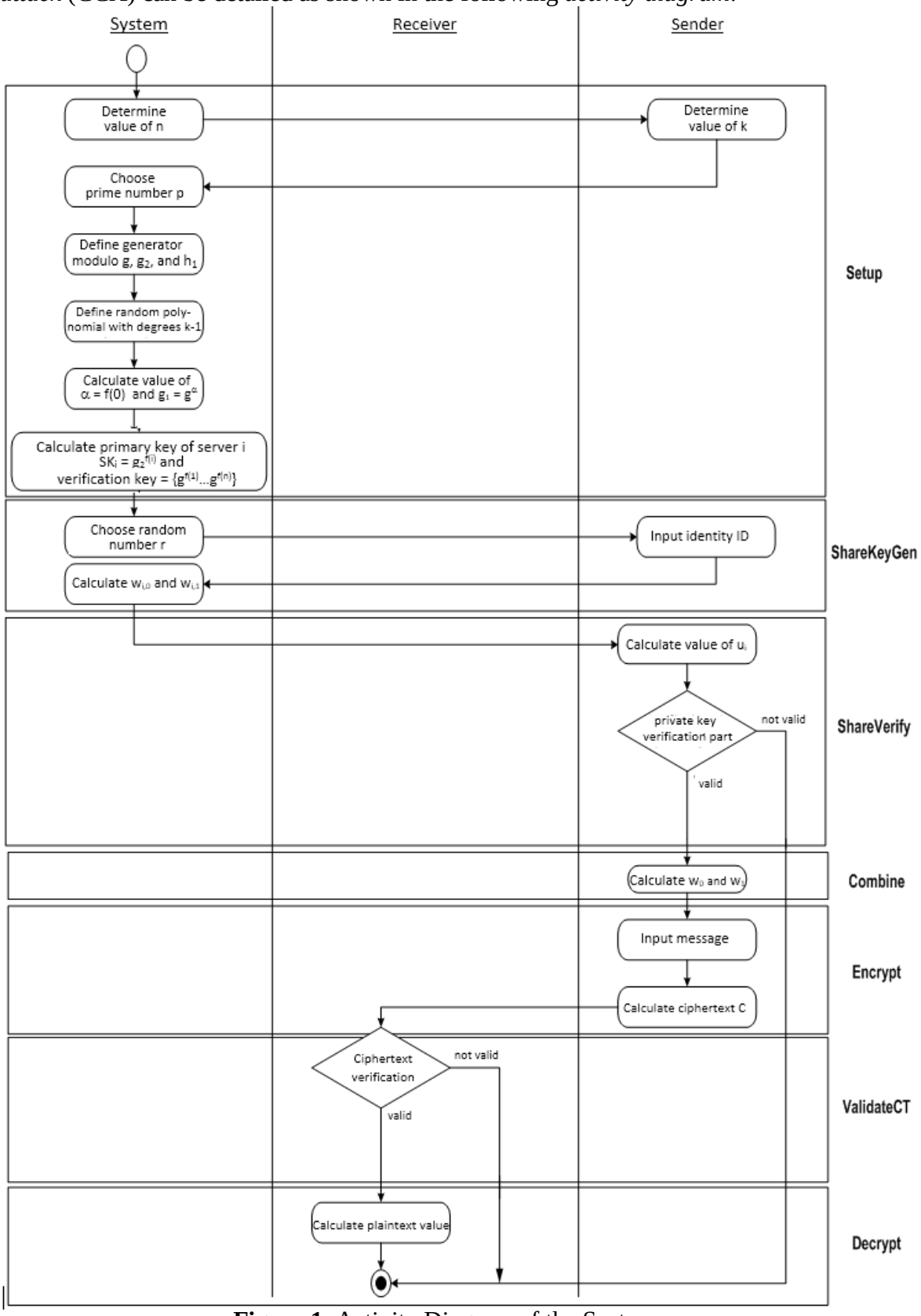

Figure 1. Activity Diagram of the System

System Modeling

The tool used to analyze and model the system is a use case. The following figure shows the use case of the system:

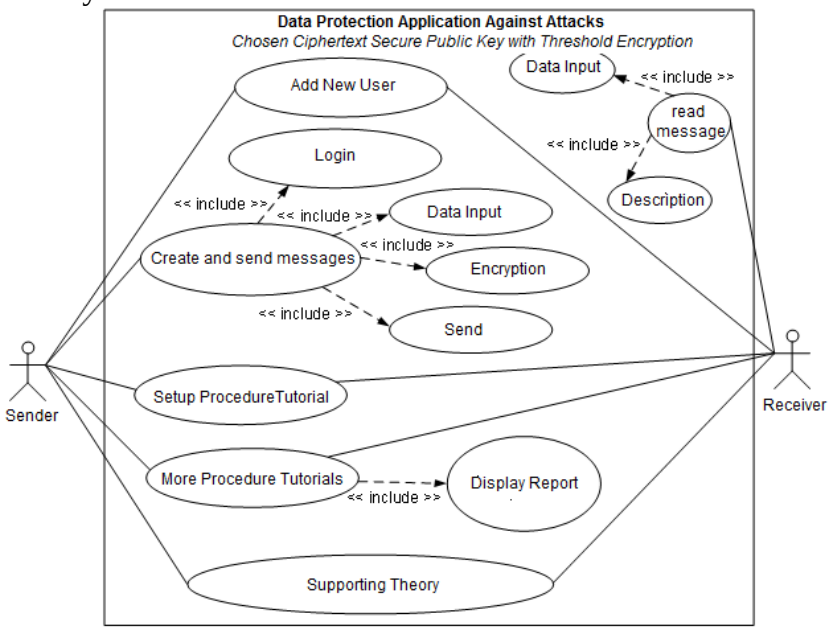

Figure 2. Use Case Diagram of the System

RESULTS AND DISCUSSION

The results of the analysis of attacks from chosen ciphertext attacks (CCA) can be categorized as follows:

1. Complex calculation process and uses several other algorithms to help generic securities from Threshold Encryption

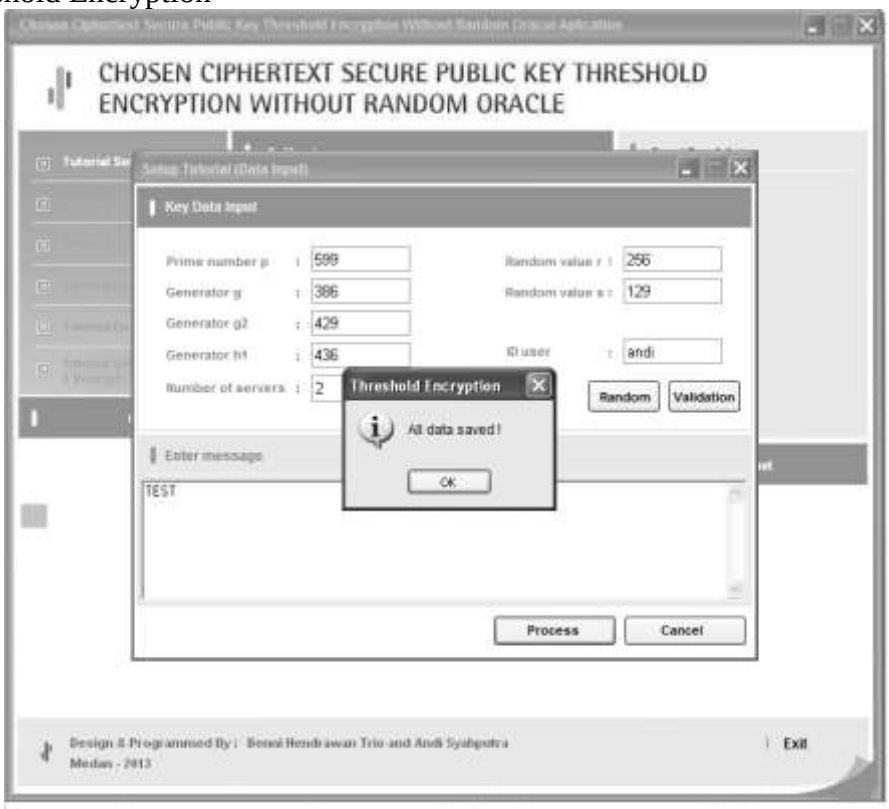

Figure 3. Generic display of numbers in setup

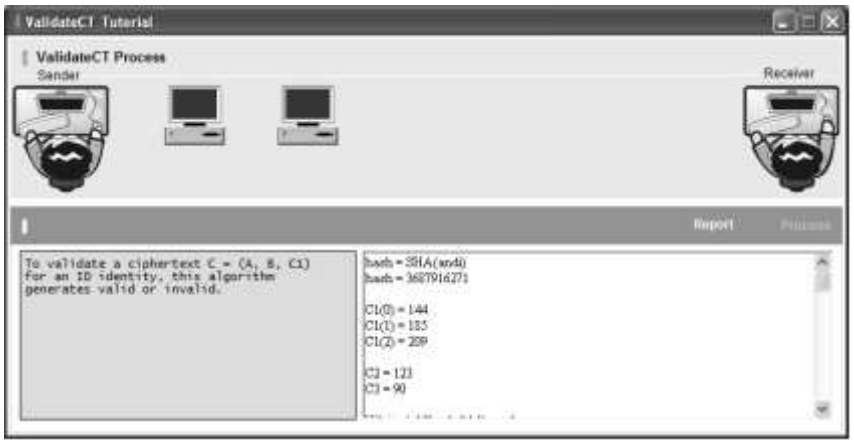

Figure 4. Results of complex generic calculations on validate

Wiretapping Test

1. Different *Server* Options

The encryption process is carried out on the input message 'Khairunas, USU's master student of informatics engineering', if the message interceptor made an error in selecting the server, even though it succeeded in validating the encryption threshold scheme, it will not be able to find out that the message did not come from the real sender, it can be proven by the display following:

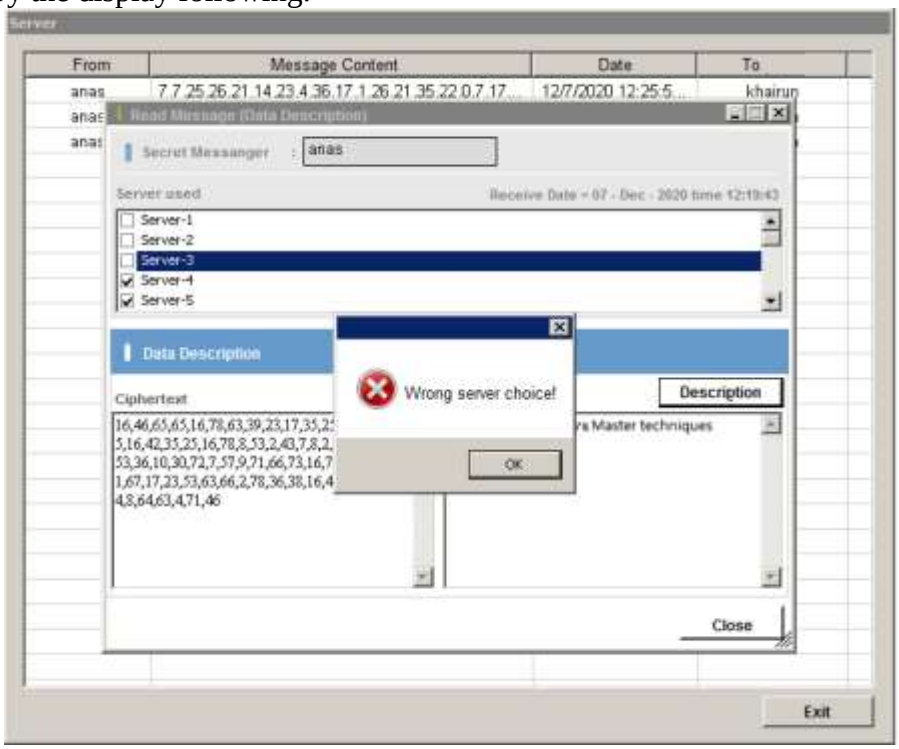

Figure 5. Encryption Process Display

If the selected *server* does not follow the encryption process, then the decryption process will fail, and here there will be provisions of the encryption threshold algorithm for the minimum server used in every message delivery given to the recipient as shown in the following figure:

Figure 6. Process Display, Wrong Server Choice

2. Ciphertext Replacement

If the ciphertext is replaced, the text to be decrypted will experience an oddity in the form of the decrypted words will become an incomprehensible language so that it can be seen that the message does not come from the actual sender and can be proven by the following picture:

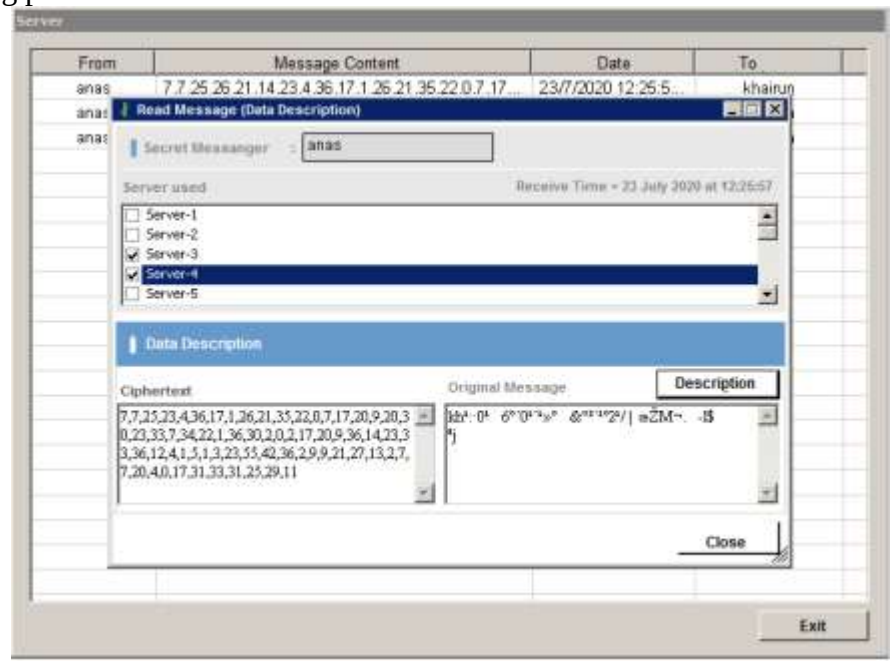

Figure 7. Display of the Ciphertext Replacement Process

Discussion

Based on the results of experiments carried out on testing the Threshold Encryption algorithm using the SHA-1 hash function that and at least using 2 servers as part of the decryption, the resulting m value is 320 bits even though the message lengths are different, thus increasing the security generated against the strength of the message security alert. In the verification and validation of the created message, the values generated using the weil pairing function e are values to produce in the form of signatures or points such as coordinates of curve points which are linear and the level of security is relatively complicated because it uses an elliptic curve discrete logarithm problem other than It is also a prime number which is relatively prime helping the level of security that produces only 1 inverse number for each message sent with different key formations.

Based on the testing process, it can be obtained from the analysis results, namely the encryption and decryption time of the message m into a value in binary form using the hash function SHA-1 does not experience a time difference with the same amount but is more effective because the results issued by SHA-1 are greater, namely 320 bits. that the algorithm that has been tested produces a better level of security because it produces output to 320 bits from the previous scheme. And from the calculation results, the complexity of the inverse value is more to return the point value based on the elliptic bend discrete logarithm issue because the values are taken randomly. This value makes it more complex and produces a safer level of security even though with the same character but the difference in the output of the point value generated from different signatures.

CONCLUSION

After completing this research, the author draws several conclusions:

1. The process flow section of the Threshold Encryption Scheme can be used to help understand how it works for users who have studied and understood the basic concepts of cryptography.
2. The algorithm scheme of Threshold Encryption can be used to secure text-based document file data, to protect important personal data, because it involves human rights, namely the right to access, delete, corrected and transfer personal data securely from attacks.
3. From the results of the tests carried out, information is obtained that in the analysis of Data Security Against Selected Ciphertext Secure Public Key Attacks Using Threshold Encryption Scheme, if there is a ciphertext replacement, the original secret message data cannot be recovered.

REFERENCES

Aryasa, Komang, Yeyasa Tommy Paulus (2014). Implementasi Secure Hash Algorithm-1 Untuk Pengaman Data Library Pada Pemrograman Java. *Jurnal Citec Jurnal* 1 (1), 57-66

Boneh, D., X. Boyen dan S. Halevi (2014). Chosen Ciphertext Secure Public Key Threshold Encryption Without Random Oracles

Boneh. D., B. Lynn dan H. Shacham. (2001). Short Signatures from the Weil Pairing, In Asiacrypt, volume 2248 of Lecture Notes in Computer Science.

Canetti, R., S. Halevi dan J. Katz (2014). Chosen-Ciphertext Security from Identity-based Encryption, In Proceedings of Eurocrypt 2004, volume 3027 of LNCS, Springer-Verlag

Dan Boneh and Mark Zhandry. (2013). Secure Signatures and Chosen Ciphertext Security In a Quantum Computing World. In Advances in Cryptology - CRYPTO 2013, volume 8043 of LNCS, pages 461 - 478. Springer.

Dan Boneh and Xavier Boyen. (2011). Eficient Selective Identity-Based Encryption Without Random Oracles. In Proceedings Stanford University volume 3027 of LNCS, pages 223–38. Springer-Verlag, USA.

Fangguo Zhang, Reihaneh Safavi Naini dan Willy Susilo. (2004). An Efficient Signature Scheme from Bilinear Pairings and Its Applications, School of Information Technology and Computer Science University of Wollongong, NSW 2522 Australia.

Hovav Shacham. (2008). Implementing pairing-based signature schemes. Presentation at the Pairings in Cryptography workshop—PiC 2008. Dublin, Ireland.

Jee Hea An, Yevgeniy Dodis, and Tal Rabin. (2002). On The Security of Joint Signature and Encryption. In Proceedings of Eurocrypt 2002, volume 2332 of LNCS. Springer-Verlag.

Kumari Sarita, (2017). A research Paper on Cryptography Encryption and Compression Techniques, *International Journal Of Engineering And Computer Science* ISSN:2319-7242 Volume 6 Issue 4 April 2017, Page No. 20915-20919, India

Markel Amit dan Nemirovskiy Leonid. (2014). Pairing-based Short Signatures, Project in Computer Security Elliptic Curve Cryptography, Israel.

Munir, R., (2005). Matematika Diskrit, Penerbit Informatika Bandung.

Nalini C. Lyer and Sagarika Mandal. (2013). Implementation of Secure Hash Algorithm -1 using FPGA. *International Journal of Information and Computation Technology*. ISSN 0974-2239 Volume 3, Number 8 (2013), pp. 757-764.

Nasution, A. M., Zarlis, M., & Suherman, S. (2021). Analysis and Implementation of Honeyd as a Low-Interaction Honeypot in Enhancing Security Systems. *Randwick International of Social Science Journal*, *2*(1), 124-135. https://doi.org/10.47175/rissj.v2i1.209

PwC (2019). Data Privacy Handbook: A starter guide to data privacy compliance. p. 14, Accessed at https://www.pwc.com/m1/en/services/assurance/risk-assurance/documents/data-privacy-handbook.pdf

William Stallings. (1999). Cryptography and Network Security: Principle and Practice, Second Edition, Prentice Hall.

Zhou Hua and Liu Qiao. (2011). Hardware Design for SHA-1 Based on FPGA. IEEE International Conference Publications on Electronics, Communication and Control (ICECC), pp. 2076-2078.