Performance Evaluation of Fully Homomorphic Encryption for End-to-End Cryptographic Communication in Multihop Networks

Hye-Yeon Shim*, Tae-Rim Park**, Il-Gu Lee*

*Department of Future Convergence Technology Engineering, Sungshin University, South Korea

**Department of Future Convergence Technology Engineering, Sungshin University, South Korea

{20180922, 20180917, iglee} @sungshin.ac.kr

***Abstract*—With the advent of a hyperconnected society, network services that connect the cloud and user terminals are emerging. Accordingly, the security technology that guarantees security and speed in end-to-end communication is becoming more important. Homomorphic encryption is useful in environments that require security in the end-to-end communication that can be operated without decryption. However, it is difficult to apply in an actual communication environment because the speed is slower than other encryption methods. In this study, we used fully homomorphic encryption and advanced encryption standards. And we built an end-to-end encryption communication network simulation environment that transmits data. Based on this, this study compares the transmission time according to the transmission environment. According to the experimental results of this study, a more effective encryption method can be selected and transmitted according to the length of the transmitted message, number of intermediate nodes, and encryption setting.**

***Keywords*—Homomorphic encryption, network, communication, Advanced Encryption Standard, fully homomorphic encryption**

I. INTRODUCTION

With the development of the network environment and technology, the era of the Internet of everything has arrived, where people and people, people and devices, and devices and devices are connected. As cloud technology enables real-time interaction, personal and digital information security is attracting attention [1]. Encryption algorithms, such as the Advanced Encryption Standard (AES), Rivest, Shamir, and Adleman (RSA) algorithm, and Data Encryption Standard, are used to ensure confidentiality and integrity in the expanding network environment.

In communication using encrypted traffic, encrypted data must be decrypted at an intermediate node to detect malware, attacks, and data forgery [2]. Users are at risk of data leakage if end-to-end encryption is not guaranteed when distributed network services are used for file storing, sharing, and collaborating [3]. Homomorphic encryption that can directly assess arithmetic operation ciphertext while maintaining the characteristics of the format and function of the encrypted data without decrypting the data has been proposed to solve this problem [4, 5]. Homomorphic encryption has the advantage of reducing the risk of data leakage because it enables searching, statistical processing, and machine learning without decrypting the ciphertext and does not require decryption in the middle of data processing [6]. However, homomorphic encryption has a very high rate of increase in ciphertext compared to plaintext and requires reboot time for noise reduction; thus, a limitation is that the speed is slowed in proportion to the message length [7].

This study compares and analyzes the performance of the existing encryption algorithm, AES, and TenSEAL, a fully homomorphic encryption library, in an end-to-end cryptography communication environment for multihop networks. This study experimentally demonstrates that the encryption method can be selected according to the network transmission parameters (message length and number of hops between the sender and destination) in terms of speed.

The structure of this paper is organized as follows. Section 2 describes the previous studies on homomorphic encryption in a network environment. Section 3 details the end-to-end encryption method, multihop network concept, and operation principle. Section 4 presents the experimental environment, process, results, and assumptions in this paper. Section 5 concludes the work.

II. RELATED WORK

Homomorphic encryption has the advantage that statistical processing is possible after encryption and can be used for machine learning. Unlike existing encryption algorithms, such as RSA and AES, data forgery and malware can be detected without decryption. Research to apply the advantages of homomorphic encryption for communication is continuously being conducted. In this section, we examine previous studies conducted to assess homomorphic encryption in a network environment.

Using a general cryptographic in the Amazon Public Cloud (AWS, Amazon Web Service) environment, Potey proposed overcoming the decryption problem during data processing and using fully homomorphic encryption (FHE) to ensure confidentiality [8]. However, because FHE is used, the size of the ciphertext increases, and thus, inefficiency arises due to the overhead generated during data processing. Peralta proposed the combined use of network coding and homomorphic encryption on the Internet of Things (IoT) environment [9]. The

idea proposed in this study is to provide data privacy, maintain the confidentiality of sensitive data, and reduce latency in an end-to-end environment as part of cloud-based IoT architecture. However, although the method proposed in this study has the advantage of increasing the system reliability, it increases computational cost and lacks interoperability. In [10], Microsoft Simple Encrypted Arithmetic Library (SEAL) is used to demonstrate security flaws in terms of the practical implementation and application of FHE. Moreover, SEAL demonstrates the security problems that can occur when homomorphic encryption is applied in protocols and applications.

Research on utilizing homomorphic encryption in cloud and IoT end-to-end communication environments and machine learning frameworks has been continuously conducted. TenSEAL was presented as an open-source library that protects personal information by applying homomorphic encryption within a machine learning framework [11]. They benchmarked using the MNIST(Modified National Institute of Standards and Technology) dataset and demonstrated that an encrypted convolutional neural network could be evaluated in less than 1 s using less than 0.5 MB of communication.

Homomorphic encryption guarantees data confidentiality during data communication and can overcome the limitations of data decryption. However, a limitation in speed degradation exists due to overhead generation. Due to these problems, it is difficult to apply homomorphic encryption in an actual network communication environment. Therefore, in this study, the communication speed of FHE and AES encryption is compared by configuring an end-to-end network communication environment composed of multihop networks.

III. End-to-End Cryptographic Communication

End-to-end encryption applies encryption in all processes between sending and receiving nodes when delivering messages through shared encryption keys between targets to communicate. In the end-to-end encryption process, the sender and receiver keep the encryption key; therefore, a third party, such as a transmission platform provider, cannot decrypt and read the message. Due to this feature, many communication programs, such as Apple's iMessage and WhatsApp, apply end-to-end encryption technology [12].

A multihop network communicates through one or more intermediate nodes rather than directly transmitting messages from a sending node to a receiving node without passing through anywhere. Generally, when transmission is performed using a multihop network in an environment where routing is performed rather than a fixed network, the scalability of the network increases, and the transmission rate is higher than that of the sending node and receiving node [13].

In this paper, a multihop network is configured, and end-to-end encryption is applied to the network. In this case, the intermediate node of the multihop network includes an operation process for confirming the packet validity.

IV. Performance Evaluation

This experiment reveals how the client applies encryption to the plaintext and transmits it to the server. At this time, the transmission is performed virtually in a local environment. The difference in speed between the fully homomorphic encryption and AES encryption, which are encryption methods, is compared through simulation according to the length of the transmitted plaintext, number of intermediate nodes passing through the transmission process, and encryption key length and context setting.

A. Experimental Environment

The transmission performance experiment for each encryption type was conducted by establishing a Python-based network simulation environment in one system. Table 1 summarizes the components and versions in the experimental environment.

TABLE 1. Computer environment in the experiment

Component	Version
CPU	Intel (R) Core (TM) i7-8550U CPU at 1.80 GHz 1.99 GHz
RAM	16.0 GB
Python	Python 3.6.12

The laptop CPU in the experiment is Intel i7-8550U, with 16 GB RAM. Python v. 3.6.12 was used in the network simulation environment. Among the context of the FHE library TenSEAL used for encryption, the scheme is CKKS(Cheon, Kim, Kim and Song). Moreover, the poly_modulus_degrees were set to 4096 and 8192, respectively, considering the message length in the experiment and the provided security. In addition, coeff_mod_bit_sizes were set to [30, 20, 20, 30] for the poly_modulus_degrees of 4096 and to [60, 40, 40, 60] for 8192. The key lengths of the AES were set to 128 and 256 bits.

In this experiment, the network simulation commonly used in FHE and AES processes consists of sending, receiving, and intermediate nodes.

- A receiving node generates a random plaintext and applies encryption to transmit to an intermediate node.
- The sending node decrypts a ciphertext received from an intermediate node.
- An intermediate node operates on a ciphertext received from a receiving node and transmits the same to a sending node. The operation consists of a simple addition and subtraction operation for each character in the plain text. The intermediate node is configured in different ways to proceed with the operation in each encryption method.

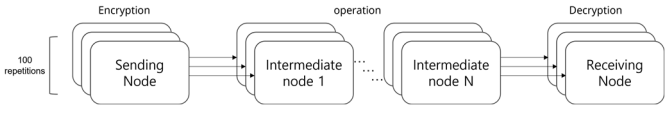

Figure 1. Network simulation flow chart

For FHE, due to the characteristics of the homomorphic encryption, the ciphertext can be calculated without decryption. Therefore, the intermediate node of the FHE calculates the ciphertext and immediately transmits it to the server.

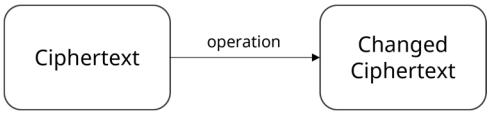

Figure 2. Intermediate node of the fully homomorphic encryption network simulation

The AES cannot operate without decryption. Therefore, after decrypting the ciphertext, the operation is performed on each character in the character string decrypted using the repetition statement. After the operation, the string is encrypted again and transmitted to the server.

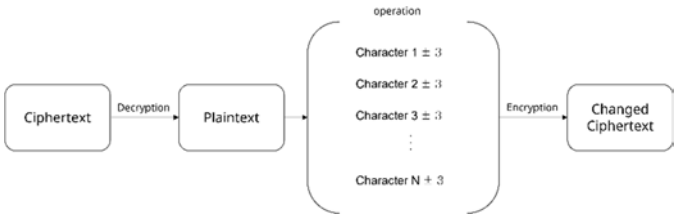

Figure 3. Intermediate node of Advanced Encryption Standard network simulation

In the configured network simulation, the time for each step was calculated by increasing the plaintext length by 10 from 10 to 1000 through repeated statements. The experiment was conducted by setting the number of intermediate nodes to 1 and 50, respectively. The average value was selected by repeating all processes 100 times to reduce the variation over time.

B. Evaluation Results

Figures 4 and 5 present the average transmission time according to the encryption settings in the case of one intermediate node and 50 when TenSEAL and AES encryption increase by 10 from 10 to 1000 in message length.

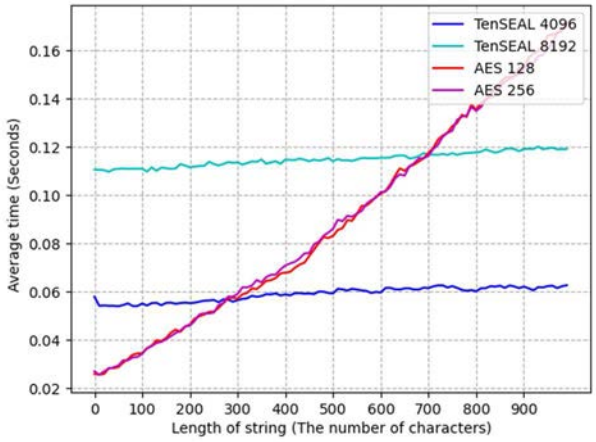

Figure 4. Average transmission time of the Advanced Encryption Standard and TenSEAL according to the message length increase with one intermediate node

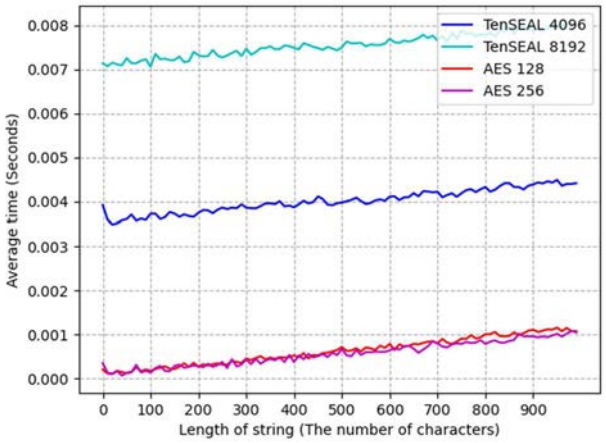

Figure 5. Average transmission time of the Advanced Encryption Standard and TenSEAL according to the message length increase with 50 intermediate nodes

When there is one intermediate node, AES has uniformly less transmission time than TenSEAL regardless of the message length. For AES, regardless of the number of intermediate nodes, the difference in the average transmission speed according to the length of the encryption key is not large.

For 50 intermediate nodes, the average transmission time taken increases linearly as the message length increases in the case of AES. In contrast, for TenSEAL, the average transmission time does not change significantly even if the message length increases. When TenSEAL's context is 4096, if the message length exceeds 300, faster transmission than AES may be performed. If the message length exceeds 700 when the context is 8192, it performs better than AES.

In conclusion, when the number of intermediate nodes is 50, it is appropriate to use AES if the message length is less than 300 or 700 and to use TenSEAL if the message length is more than that. Therefore, it suggests that an efficient transmission method can be selected for a more suitable environment according to the number of intermediate nodes at transmission time, message length, and encryption settings.

V. CONCLUSIONS

In this study, for FHE and AES encryption, the difference was compared for the transmission time according to the change in the number of intermediate nodes and message lengths. Through this analysis, the conditions were found under which the transmission times of FHE and the AES intersect, and the environment, in which the number of intermediate nodes and message length was long, was suitable for transmission using FHE.

In this study, we identified the appropriate environment for each encryption by comparing the two encryption methods through network simulation. In a follow-up study, we plan to apply the results of this study and conduct a study that proposes a model that automatically proceeds with decision-making when transmitting data.

ACKNOWLEDGMENT

This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2020R1F1A1061107) and the Korea Institute for Advancement of Technology (KIAT) grant funded by the Korean government (MOTIE) (P0008703, The Competency Development Program for Industry Specialist).

REFERENCES

[1] X. Song and Y. Wang, "Homomorphic cloud COMPUTING scheme based on hybrid homomorphic encryption," *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, 2017.

[2] R. R. Salavi, M. M. Math, and U. P. Kulkarni, "A survey of various cryptographic techniques: From traditional cryptography to fully homomorphic encryption," *Innovations in Computer Science and Engineering*, pp. 295–305, 2019.

[3] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes," *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–35, 2018.

[4] S. S. Sathya, P. Vepakomma, R. Raskar, R. Ramachandra, and S. Bhattacharya, "A Review of Homomorphic Encryption Libraries for Secure Computation," arXiv preprint arXiv:1812.02428, 2018.

[5] Z. H. Mahmood and M. K. Ibrahem, "New fully homomorphic encryption scheme based on multistage partial homomorphic encryption applied in cloud computing," *2018 1st Annual International Conference on Information and Sciences (AiCIS)*, 2018.

[6] G. Bonnoron, C. Fontaine, G. Gogniat, V. Herbert, V. Lapôtre, V. Migliore, and A. Roux-Langlois, "Somewhat/fully homomorphic encryption: Implementation progresses and challenges," *Codes, Cryptology and Information Security*, pp. 68–82, 2017.

[7] F. Armknecht et al., ''A guide to fully homomorphic encryption,'' IACR Cryptol. ePrint Arch., vol. 2015, p. 1192, 2015.

[8] M. M. Potey, C. A. Dhote, and D. H. Sharma, "Homomorphic encryption for security of Cloud Data," *Procedia Computer Science*, vol. 79, pp. 175–181, 2016.

[9] G. Peralta, R. G. Cid-Fuentes, J. Bilbao, and P. M. Crespo, "Homomorphic encryption and network coding in IOT architectures: Advantages and future challenges," *Electronics*, vol. 8, no. 8, p. 827, 2019.

[10] Z. Peng, "Danger of using fully homomorphic encryption: A look at Microsoft SEAL," arXiv preprint arXiv:1906.07127, 2019.

[11] A. Benaissa, B. Retiat, B. Cebere, and A. E. Belfedhal, " TenSEAL: A Library for Encrypted Tensor Operations Using Homomorphic Encryption", arXiv preprint arXiv:2104.03152

[12] A. Greenberg, "Hacker lexicon: What is end-to-end encryption?," *Wired*, 25-Nov-2014. [Online]. Available: https://www.wired.com/2014/11/hacker-lexicon-end-to-end-encryption/. [Accessed: 29-Sep-2021].

[13] T. Braun, A. Kassler, M. Kihl, V. Rakocevic, V. Siris, and G. Heijenk, "Multihop Wireless Networks," *Lecture Notes in Electrical Engineering*, pp. 201–265, 2009.

Hye-Yeon Shim received her BS degree from the Department of convergence security engineering at Sungshin University, Seoul, Korea. She is a student in an MS course in the Department of Convergence Security Engineering at Sungshin University, Seoul, Korea. Her current research interests are artificial intelligence, deep learning, malware detection, and programming.

Tae-Rim Park received her BS degree from the Department of convergence security engineering at Sungshin University, Seoul, Korea. She is a student in an MS course in the Department of Convergence Security Engineering at Sungshin University, Seoul, Korea. Her current research interests are artificial intelligence, communication, and convergence security.

Il-Gu Lee received his BS degree in electrical engineering from Sogang University, Seoul, Korea, in 2003 and his MS degree from the Department of Information and Communications Engineering at the Korea Advanced Institute of Science and Technology (KAIST) in Daejeon, Korea in 2005. He also received his MA degree in intellectual property from KAIST in 2012. He received his Ph.D. degree from the Graduate School of Information Security at the Computer Science and Engineering Department at KAIST in 2016. He is a professor at the Department of Convergence Security Engineering at Sungshin University (SU), Seoul, Korea. Before joining SU in March 2017, he was with the Electronics and Telecommunications Research Institute as a senior researcher from 2005 to 2017 and served as a principal architect and project leader for Newratek (KR) and Newracom (US) from 2014 to 2017. His current research interests are wireless/mobile networks with an emphasis on information security, networks, wireless circuits, and systems. He has authored/coauthored more than 55 technical papers in the areas of information security, wireless networks, and communications and holds about 160 patents. He is also an active participant of and contributor to the IEEE 802.11 WLAN standardization committee.