ComEX

Special Cluster in Conjunction with IEICE General Conference 2023

LETTER

Response time of cloud-based facial recognition system utilizing homomorphic encryption

Sei Nakanishi[1, 2, 3, a)], **Yoshiaki Narusue**[1], **and Hiroyuki Morikawa**[1]

Abstract We are developing a system that augments user accessibility by integrating various facial recognition engines across multiple enterprises, thereby facilitating the widespread adoption of facial recognition technology within societal structures. In this system, homomorphic encryption is employed to mitigate the risk of data leakage during facial recognition. However, the use of homomorphic encryption in facial recognition significantly increases latency, making it challenging to meet practical response time requirements. We experimentally evaluated the impact of adopting homomorphic encryption on the response time. The evaluation revealed that the number of registrants per facial recognition engine should be "< 120". Additionally, we evaluated a clustering strategy for reducing the response time to the level of practical application.

Keywords: homomorphic encryption, facial recognition, secure computation, cloud, clustering

Classification: Network system

1. Introduction

In recent years, facial recognition technology has been introduced in various aspects of daily life. However, because various facial recognition techniques exist, facial recognition engines are often developed differently by individual vendors [1]. As a result, user registration is required for each service, which reduces the degree of usability. For facial recognition technology to be further incorporated into society as a fundamental infrastructure, individual facial recognition engines should not be isolated; rather, they should be interconnected to enhance user convenience.

To address this issue, we implemented a Facial Recognition Integration Platform (FRIP) [2] that facilitates a cross-functional use of the previously separated facial recognition engines. Users can utilize various facial recognition engines for cross-functional purposes such as office access, residential entry, amusement-park admission, and payment transactions, which significantly enhances usability.

However, the problem of information leakage is inherent in this type of system, which necessitates data linkage with other facial recognition engines. In this system, facial features extracted from facial images are stored on the servers of cloud service providers. As the data are transmitted via the internet, they may be vulnerable to unauthorized access by third parties. If facial features are leaked, reconstructing the original facial image from facial features is technically feasible [3]. Furthermore, several studies state that impersonation attacks utilizing the reconstructed facial images are plausible [4, 5].

Homomorphic encryption [6], which is among the secure computation technologies for cloud-based operations, has been used to reduce the risk of data leakage. Employing homomorphic encryption, the authentication completes without decryption. The risk of plaintext facial features being leaked is minimized, eliminating concerns over personal-information exposure to cloud service providers, the owners of facial recognition engines, or third parties. Related works such as that of Drozdowski et al. [7] presented an architecture for using homomorphic encryption in facial recognition.

However, homomorphic encryption has its own set of complexities. The latency for facial recognition employing homomorphic encryption significantly exceeds that in the case of plaintext usage [8, 9]. Consequently, fulfilling the required response time for practical use may be challenging [10, 11].

In this study, to address these issues, we propose the implementation of homomorphic encryption for a real service based on the FRIP. We quantitatively measured the impact of employing homomorphic encryption in our system on the response time. The response time can be evaluated for actual business usage by evaluating the response time using real services. Additionally, we evaluated the clustering method to reduce the response time, and found that the cluster size necessary to achieve the response time required for societal implementation was 120.

2. Structure and functions of FRIP

In this section, we describe the structural design and the functions of the FRIP. This system can associate a facial image with an array of facial recognition engines. Users no longer need to repetitively register their facial images with multiple facial recognition engines.

A schematic of this configuration is presented in Fig. 1. The architecture comprises multiple facial recognition engines, an API linkage server that bridges the FRIP and the facial recognition engines, and a storage server that safeguards

[1] University of Tokyo, Faculty of Engineering, Building 2, 12th Floor, Room 121C1, 7–3–1 Hongo, Bunkyo-ku, Tokyo 113–8656 Japan

[2] Property Agent Co., Ltd., Shinjuku Island Tower, 41st Floor, 6–5–1 Nishi-Shinjuku, Shinjuku-ku, Tokyo 163–1341 Japan

[3] DXYZ Co., Ltd., Shinjuku Island Tower, 41st Floor, 6–5–1 Nishi-Shinjuku, Shinjuku-ku, Tokyo 163–1341 Japan

[a)] nakanishi@mlab.t.u-tokyo.ac.jp

DOI: 10.23919/comex.2023COL0010
Received June 30, 2023
Accepted July 31, 2023
Publicized November 21, 2023
Copyedited December 1, 2023

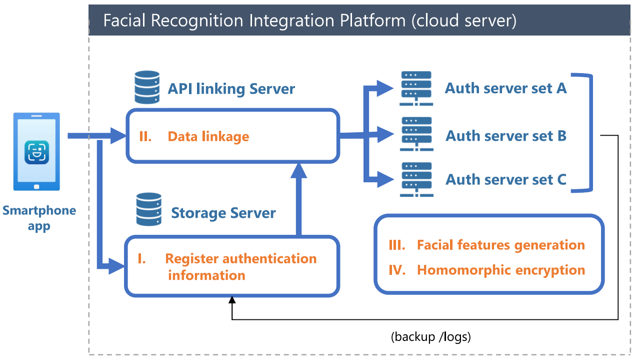
Facial Recognition Integration Platform (cloud server)
Smartphone app
API linking Server
II. Data linkage
Auth server set A
Auth server set B
Auth server set C
Storage Server
I. Register authentication information
III. Facial features generation
IV. Homomorphic encryption
(backup /logs)

Fig. 1 Structure of FRIP

authentication information. The FRIP is implemented in a cloud-native environment, considering scalability. It has the following four principal functions.

I. Registration of authentication information

The user registers facial images and attribute information, such as age and gender, from a smartphone and stores it on the storage server. This eliminates the need for users to register multiple times with different facial recognition engines.

II. Data linkage with other authentication servers

When the user opts in from the smartphone screen, the authentication information registered in the storage server is sent to each facial recognition engine through the API linking server. In addition, information such as authentication logs and backup data can be exchanged and stored in the storage server.

III. Facial features generation for multiple engines

Facial images provided by the user are converted into facial features in a format that can be analyzed by various facial recognition engines. Thus, the FRIP supports a wide variety of facial recognition engines.

IV. Homomorphic encryption

Facial features generated in the previous phase are subjected to homomorphic encryption using the CKKS [12] method. The encrypted data are subsequently employed in the authentication procedure. We explain this in detail in Section 3.

Next, we explain the key management methodology for homomorphic encryption. Key management is crucial for preventing information leakage. In this system, we utilized a hardware security module (HSM) and implemented a process of retaining keys for the minimal necessary duration, followed by their immediate destruction [13].

The HSM is secure computer hardware that amalgamates physical and logical protection functions and is adept at executing key generation, management, and storage [14]. Upon request, the management of decryption keys and data decryption can be executed within the HSM. This strategy precludes unnecessary private key information from persisting on the authentication server and minimizes the risk of information leakage due to key leakage.

3. Evaluation of response time

The latency for facial recognition employing homomorphic encryption significantly exceeds that in the case of plaintext facial features used. The response time is one of the crucial factors for FRIP to provide practical service. Therefore, we conducted an evaluation to minimize the response time of authentication server sets in Fig. 1, assuming implementation in FRIP.

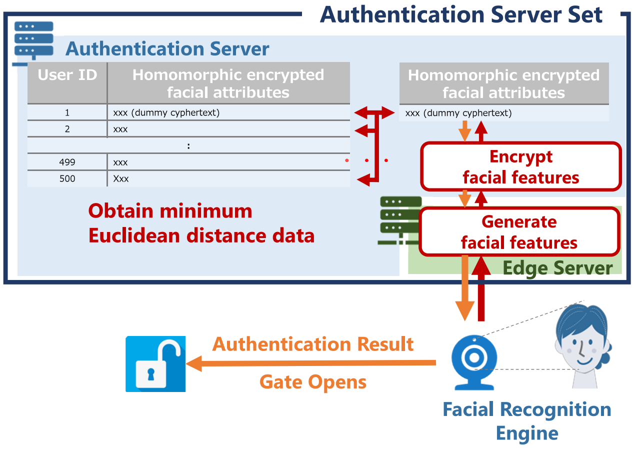

Fig. 2 Authentication process using homomorphic encryption

The detailed authentication procedure utilizing homomorphic encryption is presented in Fig. 2.

The components of the system are a facial recognition engine, an edge server, and an authentication server. The authentication server retains pre-registered user facial images as facial features encrypted via homomorphic encryption. The plaintext facial features utilized during registration are deleted after the completion of the registration procedure.

In the authentication process, the facial image captured by the authentication device is initially transmitted to the edge server as input data. The edge server transforms these data into facial features, which are sent to the authentication server. The authentication server then applies homomorphic encryption to the received facial features. Subsequently, calculations are performed on a round-robin basis across all the encrypted facial features registered within the authentication server. This process is executed without the decryption of facial features. The comparison is executed by calculating the Euclidean distance employed to quantitatively evaluate the similarity of the facial feature vectors. The authentication process retrieves the user data with the shortest Euclidean distance. If the Euclidean distance is within a certain threshold, the authentication result is approved and relayed to the facial recognition engine.

The Euclidean distance between two encrypted facial feature vectors $m = (m_1, m_2, \ldots, m_n)$, $m' = (m'_1, m'_2, \ldots, m'_n)$, can be derived as follows [15]:

$$\begin{aligned} d = \sqrt{m^2 - m'^2} &= \sqrt{\sum_{i=1}^{n}(m_i^2 - m_i'^2)} \\ &= \sqrt{(m_1^2 - m_1'^2) + \ldots + (m_n^2 - m_n'^2)} \end{aligned} \tag{1}$$

3.1 Evaluation method

The experimental environment is presented in Table I. The facial recognition engine established a connection with the edge server via HTTPS to transmit facial im-

Table I Experimental environment

Language	Server	Facial recognition engine Edge/Auth. server	Face detection library	Facial features Extraction library	Homomorphic encryption Library	Facial image dataset
Python	Azure	Standard F2s v2 (2 vcpu/4 Gib)	MTCNN	FaceNet OpenCV	TenSEAL	LFW dataset

Table II Breakdown of response time (unit: s)

Number of Registrants	100	110	120	130	140
Homomorphic encryption of facial features	0.004	0.005	0.005	0.006	0.005
Derivation of Euclidean distance	1.612	1.741	1.962	2.055	2.320
Procurement of minimal Euclidean distance data	0.025	0.026	0.028	0.030	0.032
Others	1.103	1.109	1.100	1.099	1.123
RTT	2.744	2.881	3.095	3.190	3.480

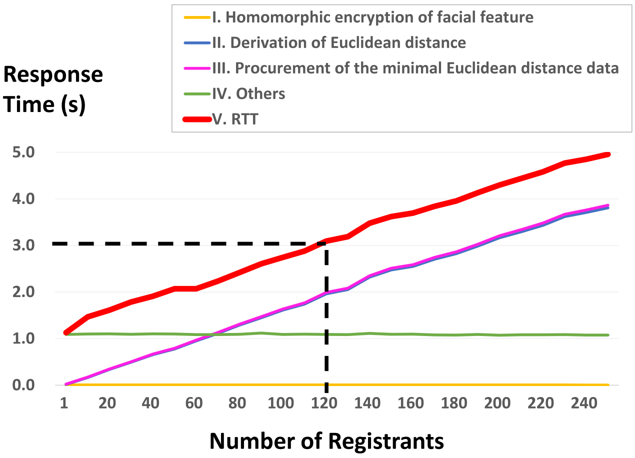

Fig. 3 Experimental result overview

ages. TenSEAL [16] was used for homomorphic encryption. TenSEAL is an open-source library designed to facilitate the use of homomorphic encryption. The facial images were obtained from the Labeled Faces in the Wild (LFW) dataset [17].

We measured the processing time for the following procedures and analyzed the effect of homomorphic encryption on the latency of each procedure:

- Homomorphic encryption of facial features
- Derivation of Euclidean distance between homomorphic ciphertexts
- Procurement of minimal Euclidean distance data
- Others (Network communication delays, etc.)

The round-trip time (RTT) is defined as the time taken to complete authentication after the face image captured by the facial recognition engine is input. The benchmark for the RTT was set as 3 s. This is based on a previous report indicating that more than half of website users leave the website if loading takes more than 3 s [18]. Despite the distinction between facial recognition engines and websites, we assume that user tolerance to latency exhibits similar characteristics between them. If the authentication of facial recognition exceeds 3 s, users may perceive inconvenience and the degree of usability reduced.

To ensure the precision of the authentication in the experimental environment, the verification across 500 registration data was conducted in advance, We confirmed that there were no false positive results, e.g., a user being authenticated as the wrong registrant.

3.2 Evaluation result

As shown in Fig. 3, the RTT in facial recognition was approximately 1.13 s for a single user, 3.10 s for 120 users, and 4.95 s for 250 users, increasing significantly with an increase in the number of registrants. Presuming an implementation standard of 3 s, the number of registrants in each facial recognition engine must be constrained to 120 for practical implementation.

Subsequently, we assessed the time required for each process. As shown in Fig. 3, the time required for deriving the Euclidean distance between homomorphic ciphers was proportional to the number of registrants, reaching 3 s for approximately 190 registrants and 3.8 s for 250 registrants. When the time required for procurement of minimal Euclidean distance data was added to the Euclidean distance acquisition time, the total duration was 3.03 s for 190 registrants and 3.86 s for 250, as shown in Fig. 3. These results also increased in proportion to the number of registrants.

Table II presents the RTT of each process for 100–140 registrants. The time required to derive the Euclidean distance between homomorphic ciphertexts accounted for a large proportion of the RTT. The RTT for the process of performing homomorphic encryption on the target facial features remained constant regardless of the number of registered users, thus having a minimal effect on the RTT. Although the processing time for procurement of the minimal Euclidean distance data increased proportionally with the number of registrants, it constituted only approximately 1% of the RTT for approximately 120 registrants; hence, it also had a minimal effect on the RTT.

Factors significantly influencing the RTT of the Euclidean distance acquisition process included the computational cost of the distance acquisition process, which increases in proportion to the number of registrants, and the substantial increase in the data size of the facial features due to homomorphic encryption [9]. The average size of plaintext facial features was 4,160 bytes, whereas the average size of homomorphic ciphertext was approximately 334,300 bytes. Compared with plaintext, the data size of homomorphic ciphertext was approximately 80 times larger, which significantly impacted the processing time for Euclidean distance acquisition.

Finally, we consider measures for reducing the RTT. One approach is to restrict the targets of the Euclidean distance calculation. The time required for Euclidean distance calculation is directly proportional to the number of registrants because calculations are performed on a round-robin basis across all the data registered within the authentication server. To limit the calculation targets while ensuring authentication accuracy, users can be grouped via methods such as k-means clustering, and the Euclidean distance calculations can be performed on a group-by-group basis, eliminating unnecessary calculations.

Executing the Euclidean distance calculations in parallel for multiple clusters is considered to be effective for reducing the RTT, although the computational load remains unchanged. When this method is applied to the experimental results above, the size of each cluster should be maintained below 120, so that the RTT does not exceed the implementation standard. Assuming that the upper limit on the number of registrants for the conventional facial recognition engine in practical application is 500, five clusters are required: ($5 = \lceil 500/120 \rceil$). Each cluster executes the Euclidean distance acquisition process in parallel to reduce the RTT.

4. Conclusion

We evaluated the response time of facial recognition engines in the FRIP when homomorphic encryption was applied in the authentication process.

Under the current implementation, the response time exceeds the 3 s benchmark with a total of approximately 120 registrants. Therefore, the maximum cluster size should be less than 120. An evaluation of the duration for each process revealed that the computation of the Euclidean distance between encrypted facial features accounted for a large proportion of the response time. Furthermore, we examined parallel processing with clustering to reduce the response time. When the upper limit of registrants for facial recognition engines in practical application is set as 500, five clusters are required for parallel processing. According to the evaluation, we obtained insight into the FRIP design using homomorphic encryption that applies to practical applications.

In future research, we aim to investigate the operational expenses, which increase proportionally with the number of prepared clusters, along with more efficient clustering methodologies, such as the k-means algorithm. From the perspectives of response time and operating cost, we will investigate the configuration and authentication process of a FRIP using homomorphic encryption.

Acknowledgments

We express our gratitude for the invaluable guidance and collaboration of the members from Morikawa Narusue laboratory and the members from Property Agent Co., Ltd and DXYZ Co., Ltd. The expertise and insights provided by our colleagues contributed significantly to this study.

References

[1] S. Singh and S.V.A.V. Prasad, "Techniques and challenges of face recognition: A critical review," *Procedia Computer Science*, vol. 143, pp. 536–543, 2018. DOI: 10.1016/j.procs.2018.10.427

[2] DXYZ, "Freeid for face recognition systems (japanese)," https://freeid.dxyz.co.jp/, accessed Jun. 23, 2023.

[3] B.-W. Hwang, V. Blanz, T. Vetter, and S.-W. Lee, "Face reconstruction from a small number of feature points," Proceedings 15th International Conference on Pattern Recognition, pp. 842–845, Sept. 2000. DOI: 10.1109/icpr.2000.906205

[4] G. Mai, K. Cao, P.C. Yuen, and A.K. Jain, "On the reconstruction of face images from deep face templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 5, pp. 1188–1202, April 2018. DOI: 10.1109/tpami.2018.2827389

[5] H.O. Shahreza and S. Marcel, "Face reconstruction from facial templates by learning latent space of a generator network," The International Conference on Learning Representations (ICLR) 2023 Conference Program Chairs, pp. 1–13, Jan. 2023.

[6] C. Gentry, "Fully homomorphic encryption using ideal lattices," Proceedings of the forty-first annual ACM symposium on Theory of computing, New York, USA, pp. 169–178, May 2009. DOI: 10.1145/1536414.1536440

[7] P. Drozdowski, N. Buchmann, C. Rathgeb, M. Margraf, and C. Busch, "On the application of homomorphic encryption to face identification," International Conference of the Biometrics Special Interest Group, Darmstadt, Germany, pp. 1–5, Sept. 2019.

[8] M. Matsumoto and M. Koguchi, "Comparison of encryption algorithms using homomorphic encryption on iot devices," Multimedia, Distributed, Cooperative, and Mobile Symposium, pp. 156–162, June 2021.

[9] J. Liu, X.A. Wang, B. Chen, Z. Tu, and K. Zhao, "Outsourced secure face recognition based on CKKS homomorphic encryption in cloud computing," *International Journal of Mobile Computing and Multimedia Communications*, vol. 12, no. 3, pp. 27–43, Sept. 2021. DOI: 10.4018/ijmcmc.2021070103

[10] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?," Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop, New York, USA, pp. 113–124, Oct. 2011. DOI: 10.1145/2046660.2046682

[11] A. Acar, H. Aksu, A.S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: theory and implementation," *ACM Comput. Surv.*, vol. 51, no. 4, Article No. 79, July 2018. DOI: 10.1145/3214303

[12] J.H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, pp. 409–437, Dec. 2017. DOI: 10.1007/978-3-319-70694-8_15

[13] E. Barker, "Recommendation for key management: Part 1 – general," NIST special publication 800-57 Part 1 Rev. 5, May 2020. DOI: 10.6028/nist.sp.800-57pt1r5

[14] K. Foltz and W.R. Simpson, "Secure server key management designs for the public cloud," Mach. Learn. Artif. Intell., vol. 332, pp. 248–253, Dec. 2020. DOI: 10.3233/FAIA200789

[15] P.-E. Danielsson, "Euclidean distance mapping," *Computer Graphics and Image Processing*, vol. 14, no. 3, pp. 227–248, Nov. 1980. DOI: 10.1016/0146-664x(80)90054-4

[16] A. Benaissa, B. Retiat, B. Cebere, and A.E. Belfedhal, "Tenseal: a library for encrypted tensor operations using homomorphic encryption," The International Conference on Learning Representations (ICLR) Workshop on Distributed and Private Machine Learning, pp. 1–12, April 2021.

[17] G.B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: a database for studying face recognition in unconstrained environments," Tech. Rep. 07-49, University of Massachusetts, Amherst, Oct. 2007.

[18] Z. Nagy, "Improved speed on intelligent web sites," *Recent Advances in Computer Science*, WSEAS Press, pp. 215–220, Jan. 2013. ISBN: 978-960-474-311-7