Proposal of a Secure and Highly Reliable Edge System that Integrates DRAM/NVMM and FPGA

WENKANG HUANG[†1] YANZHI LI[†1] YOICHI ISHIWATA[†2]
MIDORI SUGAYA[†1]

Abstract: Currently, FPGAs are expected to be applied to the edge computing systems as the hardware accelerators with low power consumption and high performance. Since the edge computers are close to users, they are expected to handle the privacy-related information, but the information security has not been sufficiently discussed. Therefore, in this study, we have proposed a system that integrates data responsiveness and security by encrypting the data aggregated in edge server which is based on fogcached (DRAM/NVMM), computing with a FPGA of low power consumption and high performance, and storing it in the non-volatile memory (NVMM). By evaluating the encryption performance of FPGA, we find that it is faster and consumes less power than a processor with the same level of power consumption.

Keywords: FPGA, Edge Computing, Encrypt, NVMM

1. Introduction

In Society 5.0, the technologies such as IoT, robotics and artificial intelligence are integrated, which will provide advanced services to humans [1]. It is assumed that the edge computing technology can realize these. Edge computing, which is closely located to the user, can be applied to high-performance and highly responsive processing [2], [3], [4]. However, there is not enough discussion about the suitable hardware and software for the edge computing.

In this research, we discussed the necessary edge configurations and requirements, examined the necessary mechanisms, and assumed that a "human-friendly robot" could be used as an application and multiple operations are performed.

Furthermore, we examined the practical application of innovative robots that flexibly respond to human feelings, as a potential application for the Society 5.0. The "Talking Robot" proposed in 2017, which is attentive to human feelings [5], could model the unconscious personal feelings from the human biological signal (heart rate variability value) and behave according to the feelings of the person [5]. In addition, the robot can communicate naturally according to the feelings of the person and provide services for the hospitalized patients. Such a technology has become more widespread in recent years. As shown by research, smooth communication is possible when the facial expressions of robots are changed based on biological data [7] and the personal spaces are adjusted [6]. For example, Suzuki et al. proposed a method for constructing human emotional models and mental models from such biological signals and utilizing them for pathological diagnosis [8].

The data handled by these services has significant characteristics. It contains not only personal data related to personal property (e.g. credit cards and personal information), but also private information related to personal life (e.g. biometric data and mental state). These data are used in large quantities to support human lives. Although many studies have been conducted on security issues related to personal information, sufficient studies have not been made on the protection and safe use of such privacy information. In the future, it is expected that the service of Society 5.0 will collect information related to human privacy by various sensors and realize a service that is more in line with the intentions, feelings, and mental states of the person by utilizing AI. Therefore, while protecting these data, we need to consider the advanced fusion of technologies to provide effective services.

In order to protect privacy-related information and handle it effectively for services, it is necessary to consider the security of this information and guarantee the responsiveness and reliability required for conventional servers. Indeed, information security aims to protect digitized information in a system, and protects the CIA of that information (i.e. Confidentiality, Integrity, and Availability). Generally speaking, these are supported by user access authentication, data encryption, and so on using encryption technology. However, these encryptions have a large overhead. Thus, it is necessary to reduce the overhead and maintain the responsiveness of the service.

In this research, we proposed a secure and highly reliable edge system that integrates DRAM / NVMM and FPGA as a new system so as to improve the data responsiveness and reliability and protect the privacy-related information. In the edge computing, a design that emphasizes responsiveness to multiple sensor nodes is proposed, to respond in real time over a short distance to sensor nodes that exist on the edge side close to the device [9]. By arranging an edge server with abundant computing resources between the cloud and the robot (i.e. the client machine), it is possible to distribute the load concentrated on the cloud server and eliminate delays and instability caused by network communication. [9]. As shown in the previous example, the main service that utilizes the internal state of a person is a personal service. Moreover, it is a rational choice to place advanced resources used by robots and other devices in a short distance. In order to ensure the responsiveness and reliability of the edge servers, we first configured a system using fogcached [11], which adapts [10] to NVMM and improves responsiveness and reliability. In order to ensure the security of data confidentiality, we focused on FPGAs. Currently, FPGAs are expected to be applied in edge computing systems as the hardware accelerators with low power consumption and high performance. In this research, we consider using FPGAs to realize the security of information that requires a high degree of confidentiality. FPGAs can freely create security-aware circuits and ensure confidentiality in information security. Since FPGA is good at one-way processing, it is considered that encryption can achieve the desired information security and utilize the advantage of the performance of FPGA.

In order to configure this middleware, fogcached [11] and ROS [12] [13] (the architecture for realizing robot services) are implemented. However, FPGA integrations for security and encryption performance have not been fully evaluated. Therefore, in this research, we first put forward an overall architecture of integration including ROS and FPGA server for next-generation services. Next, we proposed and evaluated a FPGA encryption system. Thus, it is confirmed that the proposed architecture can

†1 Shibaura Institute of Technology
†2 VA Linux Systems Japan K K

encrypt the data in 53% shorter time than a processor with the same level of power consumption.

The structure of this paper is as follows. Specifically, Section 2 describes related research, and Section 3 proposes a system that realizes data responsiveness and security. In Section 4, we present the preliminary experiment as well as its results, and propose a method to accelerate encryption with FPGA. In Section 5, an experiment is conducted to evaluate the proposed system. According to the results of experiments, this system can encrypt the data within a short time. Furthermore, based on this result, Section 6 summarizes and discusses future issues.

2. RELATED WORK

ROS (Robot Operating System) is a middleware for robot applications [12] [13], which provides a communication framework called Publisher/Subscriber, and enables unified communication. In recent years, many robots and other devices using the ROS system have been studied [9].

Ozawa et al. proposed a fogcached that achieves both responsiveness and reliability in edge servers using DCPM, a non-volatile main memory provided by Intel Corporation [11][21]. Besides, fogcached extends memcached [15] and deploys an In-Memory Key-Value Store on a hybrid main memory consisting of RAM and NVMM (DCPM) to improve the responsiveness of the edge server.

FPGA (Field Programmable Gate Array) is a kind of PLD. In recent years, various proposals for safety and security have been made, since FPGA is also an integrated circuit that can define and change the internal ethics circuit after manufacturing [16] [17]. However, its use as an encryption application is still insufficient.

There are several ways to utilize the computational power of FPGA, in which one is to assist the CPU in its arithmetic processing. This reduces the load on the CPU. In order to achieve this, the FPGA needs to be connected to the CPU, which is commonly done via PCIe or MMIO. XILINX and ALTERA provide FPGAs that support these two interfaces. In addition, a method called FaaS (FPGA-as-a-Service) that uses FPGA on the cloud has also been proposed. Amazon provides EC2 F1 instance [14]. However, the application methods of FPGA in the cloud and edge server have not been sufficiently proposed. As pointed out by Monamed (Mohamed Khalil-Hani) et al., the FPGA can be implemented as an accelerator dedicated to OpenSSL [18]. However, the implementation of MonaMed (Mohamed Khalil-Hani) et al. is an implementation of embedded FPGA. Beyond that, Ohkawa et al. proposed to use FPGA as a network-only accelerator [9] [19] [20]. The proposal by Ohkawa et al. has an advantage that FPGA can accelerate ROS network traffic and improve responsiveness. However, these studies are mainly for the development of FPGA itself, and there are still not enough mechanisms and researches to link the edge and the cloud with applications for specific use.

3. PROPOSAL

The objective of this study is to put forward a secure and highly reliable edge system that integrates DRAM / NVMM and FPGA. In order to achieve our goal, we first defined ROS as a unified middleware and envisioned a general-purpose system based on its communication. In this system, in order to improve responsiveness and reliability of this system as a short-range server, fogcached was used for data transmission / reception and NVMM was adopted to improve the reliability [9]. Furthermore, we examined the design and implementation of the middleware that performs encryption. The middleware consists of the encryption part using FPGA. Before the private data (e.g. biometric data) is saved, it is encrypted by this middleware.

3.1 Overview

Regarding the overall architecture, in this research, we extended the ROS mechanism for service integration in middleware. The premises or issues of the study are (A), (B), and (C) as follows.

(A) The effect of encrypting private data is unknown.

(B) The comparison of encryption performance between high-performance FPGA and CPU is not sufficient.

(C) How to save the private data on the edge.

In order to address these three points, we investigated a method for high-speed encryption and storage of biometric data, so as to achieve the fastness and reliability of edge servers.

Regarding a system that encrypts and stores biometric data using FPGA, firstly, as a proposal (1), we evaluated the effect of the system throughput when there is an encryption application, for the problem (A). As a proposal (2), in response to issues (B) and (C), we compared the encryption performance of FPGA and CPU, and design, implement and evaluate the system for encrypting and storing biological data.

3.2 Design

This system uses ROS [15] as a basis of middleware. Thus, it is a secure and highly reliable edge middleware that integrates FPGA. FPGA is used to realize data security, high performance, and low power consumption. Figure 1 shows the overall view of the proposed system, and Figure 2 displays the data communication diagram.

The proposed system consists of the client application and edge server. A database and servers for encoding processing are placed on the edge server. These servers communicate with each other through ROS interface within the edge and operate in cooperation as components. The client-side application is configured using the ROS application. Beyond that, the client and the edge provide services using SLAM (Simultaneous Localization and Mapping) and information acquired from biosensors as the ROS applications. SLAM is a technology that creates a map of the surrounding space from sensor data and identifies the location, which is used by many robots. For this reason, SLAM is assumed to be used as a general-purpose application. The edge gate processes the ROS data. The edge server consists of an encoding server (FPGA), an Edge Server with KVS for highly reliable data storage and high-speed response, and a Database server. Furthermore, edge server uses fogcached-ros [9], which supports ROS communication, and fogcached is the interface. By using DRAM / NVMM (DCPM), high responsiveness and data reliability can be guaranteed. Among the data in the NVMM data, the data suitable for long-term storage is transferred to the database.

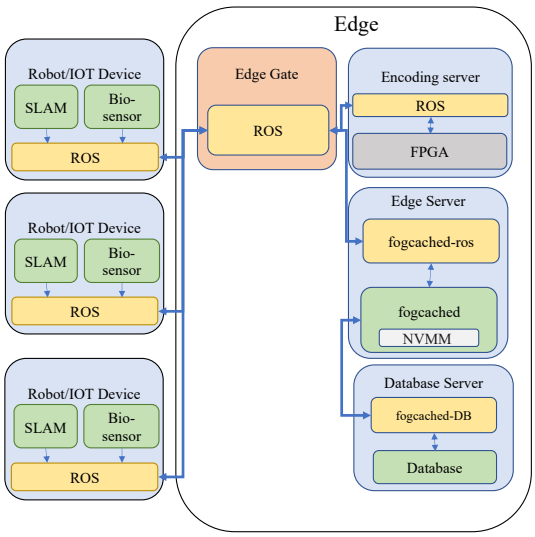

Figure 1. Overall view of the system

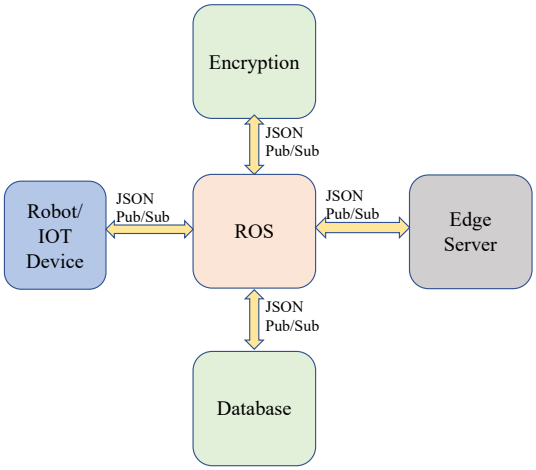

Figure 2. Data communication diagram

3.3 ROS (Robot Operating System)

ROS (Robot Operating System) [12] [13] is not only a middleware that provides distributed communication frames, but also an architecture that uses communication modules and implements P2P network connection between modules. Although communication support for multiple nodes is provided, the real-time performance is not taken into consideration. In this research, for the communication system of ROS, we designed a method for transmitting biometric data (the private information).

In ROS, there are (a) node, (b) master, (c) message, (d) topic, (e) publish, and (f) subscribe. (a) Node is a process that uses the ROS client library to communicate with other nodes. In general, applications such as SLAM consist of many nodes. (b) Master plays a central role in the execution of the entire ROS, registers nodes, services and topic names, and maintains parameter servers. (c) Message is communication data, and (d) topic is a shared data. Meanwhile, each of them is independent. As for (e) and (f), ROS performs Pub/Sub communication.

3.4 FPGA (Field Programmable Gate Array)

In this research, FPGA (Field Programmable Gate Array) is used. FPGAs, as the programmable devices, have a different structure than (from) the conventional logic circuits and gate arrays. The FPGA used in this research is PYNQ-Z1. PYNQ is an open-source framework. And PYNQ-Z1 work with the 50 MHz input clock[24]. By using PYNQ, the functions installed in PYNQ-Z1 can be fully utilized without designing a logic circuit. High-level synthesis (HLS) is a process of converting C / C ++ code to a FPGA circuit. In this research, we used Zynq to integrate programmable logic and microprocessor so as to build a FPGA encryption server. Since FPGA is suitable for high-speed calculation by encryption algorithm, we constructed an encryption system using FPGA.

3.5 System Design

In the system proposed in this study, ROS is extended, data is encrypted through the encoding server, the encrypted data is cached in the edge server via fogcached-ros, and the database server and the encrypted data are stored. The details of the basic configuration are described as follows.

Client application: It is a robot connected to the ROS system. It is assumed that the robot is equipped with a heart rate / brain wave sensor that collects personal biometric data (the private information), and LIDAR used in SLAM should also be equipped. The client application communicates with the edge server on which ROS is installed through pub / sub communication and transfers data.

Edge server: NVMM (DCPM) can be used with fogcached-ros [9] as an interface. Data caching and data processing services are provided on this server. Besides, the data is saved and processed by placing the edge server near the IoT.

In this system, communication between ROS and Node is done by publisher/subscriber basis. Topic segments the data and communicates with each other asynchronously when it sends data to the shared memory. Fogcached-ros is used for the connection between fogcached [11] and ROS, which improves the response performance and reliability of the entire ROS system. Apart from that, the biometric data and SLAM data are cached in fogcached (DRAM / DCPM) through fogcached-ros on the edge server. The data is saved in fogcached.

At the same time, the privacy data in Topic is transferred to the encoding server and encrypted. Furthermore, the encrypted data is transferred to ROS again and sent to the data server through Edge Gate and Edge Server. After the data is encrypted, the unencrypted data in fogcached-ros is deleted. Figure 3 shows the data sample, and Figure 4 displays the ROS Topic and the data flow.


```
[{"Time": "2021-2-14 15:22:17"},
  {"Real altitude": "-21.85"},
  {"Pressure": "100930"},
  {"Humidity": "37.00"},
  {"Temperature": "20.13"},
  {"Luminance": "473"},
  {"MQ3": "453"},
  {"db": "41"},
  {"heart": " 486"}]
```


Figure 3. data sample of Data flow

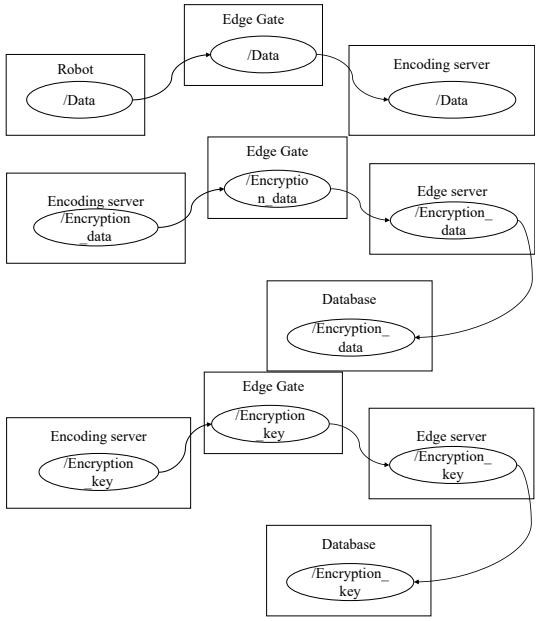

Figure 4. Data flow of encryption

4. PROPOSAL

As shown in Section 2, the purpose of this research is to encrypt the biometric data (i.e. the private information) before saving it [22]. As displayed in the proposed system, the encrypted data is stored in DCPM through ROS to ensure the security of the data.

In this preliminary experiment, we compared the throughput of the system with encryption with that of the system without encryption. In this way, we investigated the decrease in the throughput of the system and discussed its effectiveness. Meanwhile, we designed an experimental system for conducting preliminary experiments. The system used is Raspberry Pi 3B.

4.1 Overview

Table 1 shows the equipment and computer resources used in the experiment, as well as the ROS application executed. Raspberry Pi 3B contains a quad-core Cortex-A53 processor but it does not contain AES encryption processor.

Table 1. Computer resources

OS	Memory	PC	Environment
Ubuntu 18.04	1GB	Raspberry Pi 3B	High CPU load
Ubuntu 18.04	1GB	Raspberry Pi 3B	Low CPU load

4.2 Evaluation

The results of the preliminary experiment are shown in Table 2 and Figure 6. We compared the throughput of high CPU load with that of low CPU load. High CPU load means that the system needs to process encryption tasks at the same time while transmitting data. Moreover, low CPU load means that the system does not need to process encryption tasks at the same time while transmitting data.

According to the result, the high CPU load runs the encryption application, while the low CPU load is idle and does not run the encryption applications. As shown in Table 2, the throughput of the high CPU load is 170% lower than that of the low CPU load.

In addition, the maximum value of the throughput of high CPU load is smaller than the minimum value of the throughput of low CPU load. Thus, it is necessary to install a corresponding encryption module to help the system complete the encryption work. The load on the system will be reduced by introducing a dedicated encryption module.

Table 2. Preliminary experimental results

	High CPU load throughput (KB/S)	Low CPU load throughput (KB/S)
Average	0.52713	1.42458
Standard Deviation	0.011514633	0.011566792
Minimum	0.5151	1.4067
Maximum	0.5498	1.4431

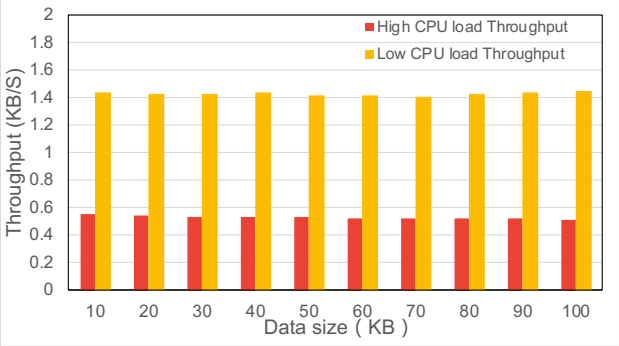

Figure 6. Preliminary experimental results

5. EVALUATION OF ENCRYPTION PERFORMANCE IN FPGA

5.1 Overview

In the preliminary test, the system throughput was evaluated under high CPU load and low CPU load. It was confirmed that the system throughput decreased in both cases. For this reason, an encryption module should be introduced to reduce the load on the CPU.

5.2 AES Encryption Method

Encryption can be roughly classified into two types. One is a symmetric cipher that anyone with the same key can decrypt, and the other is an asymmetric cipher in which the encryption key and decryption key are different. Our proposal is a basic system that is assumed as a "human-friendly robot". In order to realize this, sharing information among multiple nodes should be well considered.

Since the transmitting side and the receiving side are equal, the common key cryptosystem is suitable when multiple edges transmit to each other. In addition, AES encrypted data is saved in fogcached. The AES encryption method allows data to be shared with other devices by sharing the key. In terms of the edge characteristics, since data is expected to travel between edges, it is also necessary to send the saved data to another edge without decoding it.

```
static void SubBytes(state_t* state)
{
  uint8_t i, j;
  for (i = 0; i < 4; ++i)
  {
    for (j = 0; j < 4; ++j)
    {
      (*state)[j][i] = getSBoxValue((*state)[j][i]);
    }
  }
}
static void ShiftRows(state_t* state)
{
  uint8_t temp;
  temp           = (*state)[0][1];
  (*state)[0][1] = (*state)[1][1];
  (*state)[1][1] = (*state)[2][1];
  (*state)[2][1] = (*state)[3][1];
  (*state)[3][1] = temp;
  temp           = (*state)[0][2];
  (*state)[0][2] = (*state)[2][2];
  (*state)[2][2] = temp;
  temp           = (*state)[1][2];
  (*state)[1][2] = (*state)[3][2];
  (*state)[3][2] = temp;
  temp           = (*state)[0][3];
  (*state)[0][3] = (*state)[3][3];
  (*state)[3][3] = (*state)[2][3];
  (*state)[2][3] = (*state)[1][3];
  (*state)[1][3] = temp;
}
static void MixColumns(state_t* state)
{
  uint8_t i;
  uint8_t Tmp, Tm, t;
  for (i = 0; i < 4; ++i)
  {
    t   = (*state)[i][0];
    Tmp = (*state)[i][0] ^ (*state)[i][1] ^ (*state)[i][2] ^ (*state)[i][
3] ;
    Tm  = (*state)[i][0] ^ (*state)[i][1] ; Tm = xtime(Tm);  (*state)[i]
[0] ^= Tm ^ Tmp ;
    Tm  = (*state)[i][1] ^ (*state)[i][2] ; Tm = xtime(Tm);  (*state)[i]
[1] ^= Tm ^ Tmp ;
    Tm  = (*state)[i][2] ^ (*state)[i][3] ; Tm = xtime(Tm);  (*state)[i]
[2] ^= Tm ^ Tmp ;
    Tm  = (*state)[i][3] ^ t ;             Tm = xtime(Tm);  (*state)[i][
3] ^= Tm ^ Tmp ;
  }
}

static void Cipher(state_t* state, const uint8_t* RoundKey)
{
  uint8_t round = 0;
  AddRoundKey(0, state, RoundKey);
  for (round = 1; ; ++round)
  {
    SubBytes(state);
    ShiftRows(state);
    if (round == Nr) {
      break;
    }
    MixColumns(state);
    AddRoundKey(round, state, RoundKey);
  }
  AddRoundKey(Nr, state, RoundKey);
}
```

Figure 7. AES code

The FPGA is single and is excellent in tasks that are duplicated many times. The encryption algorithm has many such duplicate

calculations so that it is considered rational to apply FPGA to AES encryption. In addition, the AES-ECB encryption method is suitable for parallel computing. Therefore, we decided to use the AES-ECB encryption method. Figure 7 shows an example of AES code description. Here, AES encryption processing is performed after the AES-ECB key is set up. The SubBytes function is a data encryption function, the ShiftRows function shifts the data to the left, and MixColumns function is a function that mixes the matrix of AES state matrices. Cipher is the main function that processes data. The flow is as follows: after the key is entered, the data is input to the SubBytes function, and the matrix is set, the ShiftRows function shifts the data to the left, the MixColumns function mixes the data into the matrix, and AES encrypts the data.

5.3 FPGA And CPU

In the preliminary test, the throughput of high CPU load and that of low CPU load were compared. Based on the results of the preliminary test, we found that the throughput of high CPU load was low. Therefore, we thought that it was necessary to introduce an encryption module. Thus, we proposed a system that encrypts and stores private data for the encryption module with FPGA. The encoding servers used are Raspberry Pi 3B and PYNQ-Z1. This time, we designed, implemented, and evaluated a method to realize secure processing by encryption with FPGA. Table 3 shows the computer resources used, and Figure 8 displays the experimental design.

Table 3. Computer resources

OS	Memory	Hardware
Ubuntu 18.04	1 GB	Raspberry Pi 3B
Ubuntu 18.04	512MB	PYNQ Z1

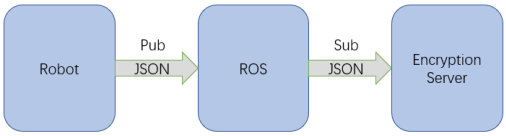

Figure 8. Design for Experiment

5.4 Method

In the evaluation, the data was created as a topic via the ROS system and published. Next, the data was subscribed to the encryption server via ROS. By testing the FPGA and CPU, we compared the encryption speed and the power consumption of the FPGA and CPU during encryption with the same amount of data.

5.5 Results

(1) Encryption speed

Figure 9 shows the encryption speed of each data amount. The FPGA encryption speed is 250.15 (KB / s), and the CPU of the Raspberry Pi 3B is 471.92 (KB / s). Table 4 displays the mean, standard deviation, maximum, and minimum.

Table 4. Results of encryption speed

	Raspberry Pi 3B	FPGA
Average (KB/s)	250.15	471.92
Standard Deviation	3.95	5.34
Minimum (KB/s)	242.93	460.27
Maximum (KB/s)	255.15	480.15

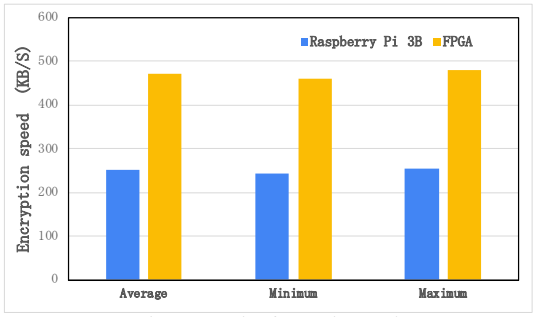

Figure 9. Results of encryption speed

According to the results in Table 4 and Figure 9, the FPGA encryption speed is 1.8 times faster than that of the Raspberry Pi 3B.

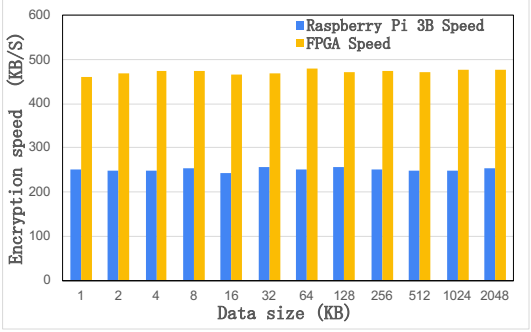

Figure 10. Encryption speed for each amount of data

Figure 10 summarizes the changes in encryption speed when comparing Raspberry Pi 3B and FPGA with different amounts of data. The results show that the encryption speed of the FPGA system is also 1.8 times faster than that of the Raspberry Pi 3B. The standard deviation suggests that the fluctuation of the encryption speed of the FPGA system is 1.35 times higher. That is to say, the data transfer between the CPU and FPGA is an I/O operation so that there is a delay in memory copying. Latency varies and gets stuck, and the standard deviation of the system increases. Although delays may occur, the worst speed of FPGA is found to be superior to that of Raspberry Pi 3B. Therefore, FPGA is more suitable for encryption than Raspberry Pi 3B.

(2) Comparison of power consumption

Figure 11 shows the results of comparing the power consumption of the Raspberry Pi 3B and FPGA during encryption. From the results, it can be observed that the FPGA system consumes 70.6% less power for encrypting. At the same time, when the FPGA is encrypting a large amount of data, the increase in power consumption is 380% lower than that of the Raspberry Pi 3B. Beyond that, FPGA has the advantage of being able to encrypt large amounts of data with low power consumption. Thus, it should meet these requirements.

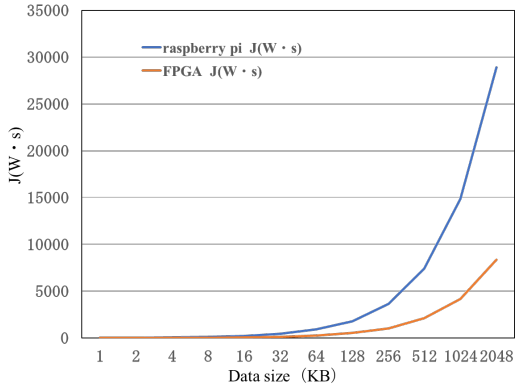

Figure 11. Power consumption of FPGA and Raspberry Pi 3B

(3) Comparison of Throughput

Figure 12 shows the data throughput under different conditions. Obviously, the throughput of the system using FPGA encryption module is increased by 1.8 times than that using no encryption module. Therefore, FPGA as an encryption module, can improve the data throughput.

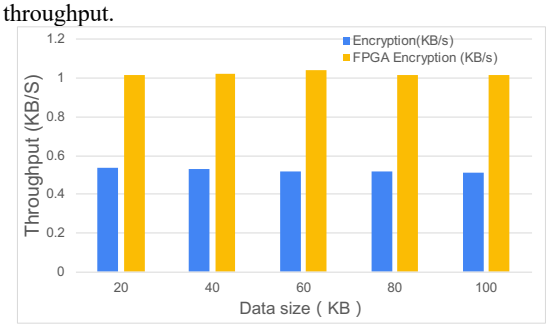

Figure 12. Data throughput of FPGA and Raspberry Pi 3B

6. CONCLUSIONS AND DISCUSSION

In this paper, we proposed a highly reliable edge middleware that integrates ROS-based FPGA/NVMM. Beyond that, we put forward a system to store biometric data at the edge, which emphasizes the safety of personal information and is a part of the middleware. In the proposal, we implemented and evaluated a method for encrypting and storing biometric data using FPGA. According to the evaluation, FPGA has better encryption performance than CPU. In terms of the power consumption, FPGA consumes less power.

In the future research, we will increase the encryption speed of the system and execute the system using FIC (Flow-in-Cloud) [23] of multiple FPGAs. FIC (Flow-in-Cloud) can realize multi-tenancy and make better use of FPGA resources by utilizing multiple FPGAs for direct serial linking. Apart from that, we will add the robot SLAM to the system, and incorporate deep learning into this system.

7. ACKNOWLEDGMENTS

This research was supported by Japan Science and Technology Agency (JST), CREST, JPMJCR19K1.

Reference

[1] Society5.0 : https://www8.cao.go.jp/cstp/society5_0/

[2] Y. Mao, C. You, J. Zhang, K. Huang and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," in *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322-2358, Fourthquarter 2017, doi: 10.1109/COMST.2017.2745201.

[3] W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, "Edge Computing: Vision and Challenges," in IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637-646, Oct. 2016, doi: 10.1109/JIOT.2016.2579198.

[4] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta and D. Sabella, "On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration," in *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657-1681, thirdquarter 2017, doi: 10.1109/COMST.2017.2705720.

[5] Teppei Ito, Reiji Yoshida, Yoshito Tobe, Midori Sugaya, Supportive Voice-Casting Robots using Bio-Estimated Emotion for Rehabilitation, The 15th International Conference on Intelligent Environments 2019, June 24-27, 2019, Rabat, Morocco

[6] Suzuki, Muhammad Nur Adilin Mohd Anuardi, Peeraya Sripian, Nobuto Matsuhira, Midori Sugaya, Multi-user Robot Impression with a Virtual Agent and Features Modification According to Real-time Emotion from Physiological Signals, The 29th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN 2020), Virtual Conference, Aug. 31st – Sep. 4th, 2020.

[7] Peeraya Sripian, Yuya Kurono, Reiji Yoshida, Midori Sugaya, "Study of Empathy on Robot Expression Based on Emotion Estimated from Facial Expression and Biological Signals", Proceedings of 2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN 2019), Le Meridien, Windsor Place, New Deli, India, Oct, 14th-18th, 2019.

[8] Kei Suzuki, Ryota Matsubara, Midori Sugaya, "Construction and Evaluation of an Emotion Estimation Model Using EEG and Heart Rate Variability Indices" Technical Committee on Biometrics / Technical Committee on Cloud Network Robotics, Online, March 2nd, 2021

[9] Koki Higashi, Yoichi Ishiwata, Takeshi Ohkawa and Midori Sugaya, fogcached-ros: Hybrid main memory KVS server, Asia Pacific Conference on Robot IoT System Development and Platform (APRIS2020), 202011, online

[10] "memcached" http://memcached.org/ (referred 2020-08-04)

[11] Kouki Ozawa, Takahiro Hirofuchi, Ryousei Takano, Midori Sugaya, "fogcached: DRAM-NVM Hybrid Memory-Based KVS Server for Edge Computing", Edge Computing – EDGE 2020 – 4th International Conference, Lecture Notes in Computer Science 12407, Springer, 2020, pp.50-62

[12] Lum,S.Utilizing Robot Operating System(ROS)in Robot Vision and Control.National Technical Reports Library U.S Department of Commerce. 2015.

[13] "ROS", https://www.ros.org/ (referred 2020-08-04)

[14] Amazon Web Services: "Amazon EC2 F1 Instance", https://aws. amazon.com/j p/ec2/instance-types/f1/.

[15] Y. Nitta, S. Tamura, H. Yugen and H. Takase, "ZytleBot: FPGA Integrated Development Platform for ROS Based Autonomous Mobile Robot," 2019 International Conference on Field-Programmable Technology (ICFPT), Tianjin, China, 2019, pp. 445-448, doi: 10.1109/ICFPT47387.2019.00089.

[16] S. K. R, S. R, M. A. M, P. K. M. S and R. M, "Design of High Speed AES System for Efficient Data Encryption and Decryption System using FPGA," 2018 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), Msyuru, India, 2018, pp. 1279-1282, doi: 10.1109/ICEECCOT43722.2018.9001535.

[17] K. Kumar, K. R. Ramkumar and A. Kaur, "A Design Implementation and Comparative Analysis of Advanced Encryption Standard (AES) Algorithm on FPGA," 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2020, pp. 182-185, doi: 10.1109/ICRITO48877.2020.9198033.

[18] M. Khalil-Hani, V. P. Nambiar and M. N. Marsono, "Hardware Acceleration of OpenSSL Cryptographic Functions for High-Performance Internet Security," 2010 International Conference on Intelligent Systems, Modelling and Simulation, 2010, pp. 374-379, doi: 10.1109/ISMS.2010.89.

[19] Takeshi Ohkawa, Kazushi Yamashina, Hitomi Kimura, Kanemitsu Ootsu, Takashi Yokota: FPGA components for integrating FPGAs into robot systems, IEICE Transactions on Information and Systems. E101.D. 363-375. 10.1587/transinf.2017RCP0011. (2018)

[20] Yuhei Sugata, Takeshi Ohkawa,Kanemitsu Ootsu Takashi Yokota: Acceleration of publish/subscribe messaging in ROS-compliant FPGA component, Proc. of the 8th International Symposium on Highly Efficient Accelerators and Reconfigurable Technologies. ACM, 2017.

[21]"Intel®OptaneTMPersistentMemory".https://www.intel.com/content/www/us/en/architecture-and-technology/optane-dc-persistent-memory.html (参照 2020-08-04)

[22] ISEC server report, 2014, Information-technology Promotion Agency,Japan(IPA)https://www.ipa.go.jp/security/fy26/reports/isec-survey/index.html

[23] Kazusa Musha, Tomohiro Kudoh and Hideharu Amano, "Deep Learning on High Performance FPGA Switching Boards: Flow-in-Cloud", Proc. of the International Simposium on Applied Reconfigurable Computing (ARC), 2018.

[24] PYNQ-Z1 Reference Manual https://digilent.com/reference/programmable-logic/pynq-z1/reference-manual