Development and Examination of Fog Computing-Based Encrypted Control System

Kaoru Teranishi, Naoki Shimada, and Kiminao Kogiso

***Abstract*—This letter develops a fog computing-based encrypted control system in a practical industrial setting. The developed system conceals controller gains and signals over communication links using multiplicative homomorphic encryption to prevent eavesdropping attacks. Experimental validation confirms the feasibility of position servo control for the motor-driven stage with the developed system in terms of performance degradation, parameter variation, and processing time. The developed system inherits its stability regardless of whether plant parameters fluctuate or not even after the controller gains and signals are encrypted. Furthermore, although processing time becomes longer by increasing a key length of encryption, degradation of control performance is improved simultaneously.**

***Index Terms*—Networked robots, robot safety, motion control, encrypted control, fog computing.**

I. Introduction

CLOUD-BASED control systems [1], in which controlled devices are connected to a communication network to be monitored and controlled in the cloud, are gaining popularity. Control as a Service (CaaS) for automotive control, a cloud-based control concept, was proposed in [2]. The authors of [3] introduced RobotControl as a Service. This concept also realizes higher-layer control (e.g., motion planning) for industrial robots. Rapyuta [4] cooperating with RoboEarth [5] is Platform as a Service (PaaS) for cloud robotics applications. The main advantage of these architectures lies in their improved flexibility, scalability, and efficiency over conventional networked systems [6].

On the other hand, lower-layer control (e.g., servo control of actuators) still needs local execution, and a cloud architecture is not suitable for such control because of latencies between controlled devices connected to the cloud [7], [8]. This issue can be solved by fog computing [9], which is a decentralized computing architecture with an intermediate layer called fog. Fog computing-based control systems reduce communication delay and retain the advantages of cloud-based control systems, that is, the controller does not need to be installed locally, and operators can remotely monitor the plant condition and easily change the control law. Additionally, the fog aggregates and cleans dirty data to support analytics in the cloud [10]. Fog computing offers many potential benefits, especially for real-time applications, although security and privacy issues in the fog persist similar to the case of the cloud [11]–[13].

Attacks on cyber-physical systems, such as networked control systems, are more damaging than attacks on information systems because physical systems can directly affect real environments [14], [15]. Adversaries can eavesdrop, invade, and falsify the system if security measures have not been implemented sufficiently. The authors of [16] verified the risks of manipulators by actual attacks, which tamper with controller gains. It is critical to obfuscate controller gains and to conceal signals from the attacks.

Encrypted control [17], a fusion of cryptography and control theory, is a promising methodology to improve the security of control systems by reducing risks of eavesdropping attacks. Eavesdropping attacks aim to steal information of control systems in order to execute more severe attacks, such as zero dynamics attacks, in the future [15]. In encrypted control systems using ElGamal encryption [18], which is multiplicative homomorphic encryption, control inputs are calculated in ciphertext from encrypted controller parameters, encrypted sensor data, and an encrypted reference without decryption. Additionally, encrypted control can be applied for the detection of replay attacks and controller or signal falsification attacks [19].

The encrypted control system with Paillier encryption [20], which is additive homomorphic encryption was proposed in [21], [22]. The authors of [23] provided the signal concealment method with fully homomorphic encryption. Homomorphic encryption is utilized as a security measure in control systems, as noted above. However, it is not straightforward to obfuscate the controller parameters with additive homomorphic encryption because multiplication between two data cannot be executed in ciphertext. Furthermore, additive and fully homomorphic encryptions require a large number of computational resources for homomorphic operation. Thus, these encryption schemes are not suitable for lower-layer control of mechanical systems.

Another approach to security enhancement of fog computing-based control systems was proposed in [24]. In this method, an artificial noise is added to sensor data, and a controller in the fog determines the control input required to achieve

Manuscript received February 4, 2020; accepted June 1, 2020. Date of publication June 15, 2020; date of current version June 29, 2020. This letter was recommended for publication by Associate Editor Kiju Lee and Editor Nak Young Chong upon evaluation of the reviewers' comments. *(Corresponding author: Kaoru Teranishi.)*

Kaoru Teranishi and Kiminao Kogiso are with the Department of Mechanical and Intelligent Systems Engineering, The University of Electro-Communications, Tokyo 182-8585, Japan (e-mail: teranishi@uec.ac.jp; kogiso@uec.ac.jp).

Naoki Shimada is with the Department of Electronics and Information Engineering, National Institute of Technology, Ishikawa College, Ishikawa 929-0392, Japan (e-mail: n_shimada@ishikawa-nct.ac.jp).

Digital Object Identifier 10.1109/LRA.2020.3002195

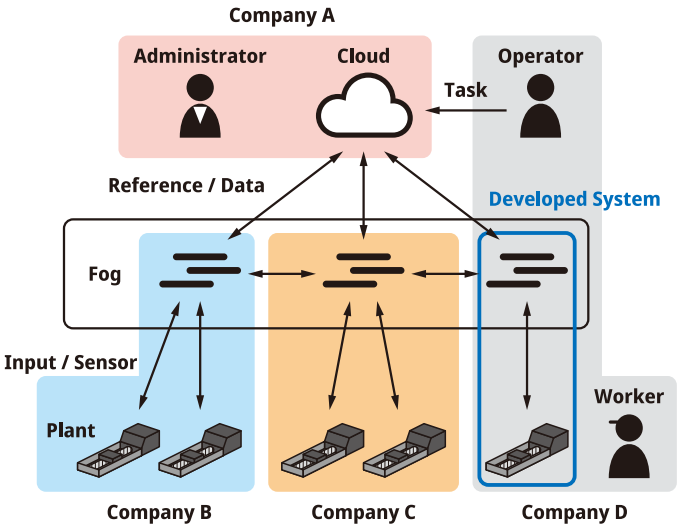


Fig. 1. Concept of the fog computing-based control system with the public cloud.

mean-square asymptotic stability. However, unlike the method of [17], the controller parameters and control inputs are not concealed.

This letter focuses on the development of a fog computing-based encrypted control system with the aim to realize secure modern control systems, e.g., Fig. 1. The developed system uses a basic PID controller encrypted by ElGamal encryption for position control of a linear stage. In the previous studies [25], [26], although the feasibility and property of the encrypted control systems have been evaluated through implementations on Raspberry Pi, validity has not been investigated in realistic settings such as an environment using industrial equipment and networks. This letter demonstrates the first implementation of the encrypted control system that is more representative of a real environment in factories. The effects of the load fluctuation and real-time property are validated. The PID gains and stage position, as well as a reference signal, are encrypted in the developed system. Additionally, control inputs in ciphertext are determined by using the relevant ciphertext without decryption in the fog. The experimental results confirm that the proposed control system retains the stability and control performance of the original unencrypted control system even when the controller encryption method is applied.

The remainder of this letter is organized as follows. Section II summarizes the controller encryption method and provides a state-space representation of a discrete-time PID controller. Section III describes the concept of a fog computing-based control system. The network architecture and the specifications of the developed system, and C language library to implement an encrypted controller are explained. Section IV presents the experimental results of confirming that controller parameters and signals are concealed by encryption. Performance degradation due to controller encryption and the effect of load fluctuation for the stability of the encrypted control system are also examined. Furthermore, the processing time of the developed system is investigated. Section V describes conclusions and future works.

II. Encrypted PID Controller

This section summarizes the controller encryption method using ElGamal encryption in [17] and introduces a state-space representation of PID controller for the method. Since the ElGamal cryptosystem uses a randomized parameter for the encryption, it is a more secure public-key encryption scheme compared with the RSA cryptosystem [27].

A. Controller Encryption

A plant P and a controller f are given as follows:

$$P : \begin{cases} x_{k+1} = Ax_k + Bu_k, \\ y_k = Cx_k, \end{cases}$$
$$f : \begin{cases} z_{k+1} = A_c z_k + B_c v_k, \\ u_k = C_c z_k + D_c v_k, \end{cases} \tag{1}$$

where x is a state, u is an input, y is an output, A, B, and C are plant parameters, z is a controller state, v is a controller input, and A_c, B_c, C_c, and D_c are controller parameters. (1) can be rewritten as follows:

$$\psi_k = \Phi\xi_k =: f(\Phi, \xi_k), \tag{2}$$
$$\psi_k := \begin{bmatrix} z_{k+1} \\ u_k \end{bmatrix}, \ \Phi := \begin{bmatrix} A_c & B_c \\ C_c & D_c \end{bmatrix}, \ \xi_k := \begin{bmatrix} z_k \\ v_k \end{bmatrix}.$$

A map Q for translation to a plaintext space $\mathcal{M}$ is defined as follows:

$$Q : \mathbb{R} \ni x \mapsto \bar{x} = \lceil \gamma x + a \rfloor \in \mathcal{M}, \ a := \begin{cases} 0, & x \geq 0, \\ p, & x < 0, \end{cases}$$

where a scaling parameter $\gamma \in \mathbb{R}$ is given by key length λ, $\lceil \cdot \rfloor$ is a function that rounds to the nearest element in $\mathcal{M}$, and p is a modulo parameter used in operations of ElGamal encryption. Since Q operates as a quantizer, it causes quantization errors, but if λ increases, the quantization errors decrease [28].

Definition 1: It is assumed that f of (2) is given as $f = f^+ \circ f^\times$ [17]. Additionally, ξ is encrypted by a modified ElGamal encryption scheme $\mathcal{E}^+ = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec}^+)$ with $\mathrm{Dec}^+ = f^+ \circ \mathrm{Dec}$. Under this assumption, an encrypted controller is defined as the following map:

$$f^\times_{\mathcal{E}^+} : \left(\mathrm{Enc}\left(\bar{\Phi}\right), \mathrm{Enc}\left(\bar{\xi}\right)\right) \mapsto \mathrm{Enc}\left(\bar{\Psi}\right),$$

where $f^\times(\Phi, \xi) = \Psi$ and $\mathrm{Dec}^+(\mathrm{Enc}(\bar{\Psi})) = \bar{\psi}$.

In the conventional cloud-based control systems, only the signals over networks are encrypted, and the controller gains and structure are not concealed. Therefore, if the adversaries invade the cloud, they can steal the controller's recipes and easily estimate the dynamics of the plant. In contrast, in the encrypted control system, the control input in ciphertext is calculated from encrypted gains and encrypted signals without decryption, and thus, the information is protected.

B. State-Space PID Controller

The state-space representation of a discrete-time PID controller is written as follows:

$$A_c = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix},\ B_c = \begin{bmatrix} 1 & -1 \\ T_s & -T_s \end{bmatrix},\ C_c = \begin{bmatrix} -\frac{K_D}{T_s} & K_I \end{bmatrix},$$

$$D_c = \begin{bmatrix} K_P + K_I T_s + \frac{K_D}{T_s} & -\left(K_P + K_I T_s + \frac{K_D}{T_s} \right) \end{bmatrix},$$

$$z_k = \begin{bmatrix} e_{k-1} \\ w_{k-1} \end{bmatrix},\ v_k = \begin{bmatrix} r_k \\ y_k \end{bmatrix},$$

where K_P is a proportional gain, K_I is an integral gain, K_D is a derivative gain, and T_s is a sampling period, r is a reference, e is an error between r and y, and w is an integrated value of e. In the proposed system, A_c, B_c, C_c, D_c, z, and v are encrypted. Encrypting of the controller parameters can only be realized by using homomorphic encryption, which allows multiplication in ciphertext.

III. Developed System

This section introduces the architecture of the cloud and fog computing-based control system as well as its specifications. Furthermore, a C language library for the encrypted control is described.

A. Concept

Fig. 1 illustrates a concept of the fog computing-based control system with a *Public cloud* [29]. *Company A* administrates a cloud infrastructure and provides a platform to operate the higher-layer control. *Company B, C, and D* manage fog connected to the cloud and each other. *Company B and C* may be branches of *Company D*, and they aim to control devices, which include some actuators and are owned by each company. An operator sends tasks for the higher-layer control to an application in the cloud. The application generates reference signals to implement the tasks and transfers them to the fog. The fog decides the input signals from the reference signals and sensor data of the devices in real time. Additionally, the fog handles operating data and transfers them to the cloud. The cloud storages the data and visualizes them with a web interface for the operator.

B. Architecture

This letter focuses on developing the fog computing-based control system within the blue frame seen in Fig. 1. Fig. 2 illustrates the network architecture of the developed system. We use personal computers for a fog-computing environment and the interface between a controlled device and the network. The computers are connected to L2 switches, which in turn are connected to an L3 switch via an Ethernet cable. Additionally, as per the requirements of a logical network, both computers are installed in the same VLAN.

The computers communicate with each other by TCP/IP socket communication, and the L3 switch addresses routing decisions. Scientific Linux, which is compatible with RedHat Enterprise Linux (RHEL), is used as the operating system for the computers. The computers are also equipped with the Advanced Robot Control System (ARCS) [30], a framework for real-time computation of the lower-layer control system.

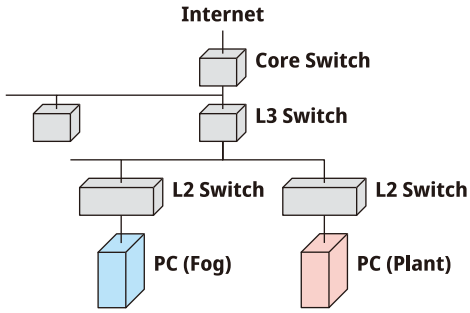


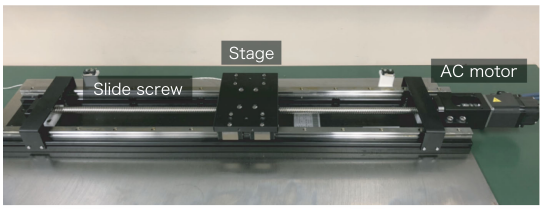
Fig. 2. Network architecture of the developed system.

TABLE I
Specifications of the Developed System

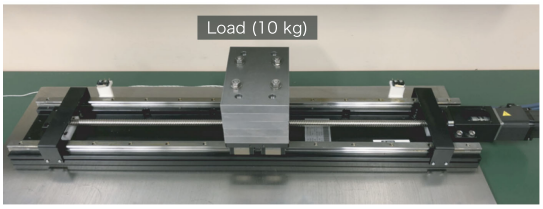
Servo amplifier	MITSUBISHI MR-J3-10A
Main circuit power supply	1/3-phase 200 to 230 VAC 50/60 Hz
AC servo motor	MITSUBISHI HF-KP13
Rated power	100 W
Rated torque	0.32 N/m
Rated speed	3000 r/min
Rated current	0.8 A
Moment of inertia	0.88 kg·m^2
Rotary encoder	
Pulse per rotation	131072
Slide screw	
Length	700 mm
Lead	10 mm
PC (Fog)	
CPU	Intel Core i7-7700K (4.20 GHz)
Memory	16 GB
OS	Scientific Linux 7.6
Kernel	3.10.0-327.el7.x86_64
PC (Plant)	
CPU	Intel Core i7-7700K (4.20 GHz)
Memory	16 GB
OS	Scientific Linux 7.4
Kernel	3.10.0-327.el7.x86_64
D/A board	Interface PCI-3340
Resolution	16 bit
Counter board	Interface PCI-6205C
Resolution	24 bit

C. Specifications

The developed system consists of a motor-driven stage, the plant-side computer, and the fog-side computer. Their specifications are described in Table I. Motor-driven stages or linear actuators are major components for factory automation. The stage is moved by an AC servo motor through a slide screw, as shown in Fig. 3(a), and the AC motor and the slide screw are connected via a coupling. The AC motor with the attached rotary encoder is driven by the servo amplifier, and we use the servo amplifier as a current controller as it has sufficient control bandwidth. Note that although the developed system uses a slide screw, the security enhancement method in Section II can also be applied to a system that uses a ball screw or other linear actuators.



(a) View of the whole plant.



(b) View of the whole plant with a 10 kg load.

Fig. 3. Plant in the developed system. (a) View of the whole plant. (b) View of the whole plant with a 10 kg load.

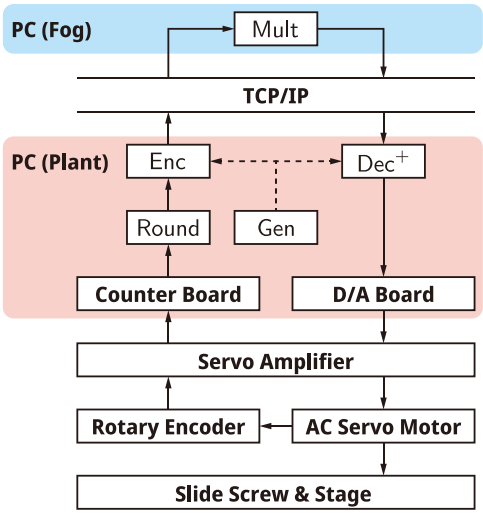


Fig. 4. Control flow of the developed system.

D. C Language Library

This letter also develops a C language library to execute encrypted control. This library contains five functions, Gen, Enc, Dec$^+$, Round and Mult. These functions operate based on BIGNUM library in OpenSSL to address multiple-precision integers. Gen searches a safe prime from a given key length, and then, it generates a public key pk and a secret key sk based on the safe prime. Enc encrypts each element of a matrix or vector by using pk, and Dec$^+$ restores a plaintext vector from an encrypted matrix by using sk. The matrix or vector used in Enc is rounded into a plaintext space by Round, a function that corresponds to Q. Mult calculates the Hadamard product of an encrypted matrix and an encrypted vector.

Fig. 4 illustrates the process in the developed system using the developed C library. The plant-side computer obtains a current position from the rotary encoder through the counter board and the servo amplifier. Then, the plant-side computer converts the current position, reference input, and controller states, which are double-precision floating-point data, into multiple-precision integers by using Round. The converted data are encrypted by Enc, and they are sent to the fog-side computer. The fog-side computer decides a control input in ciphertext from the encrypted data and encrypted controller parameters by using Mult. Additionally, the fog-side computer returns the ciphertext of the control input to the plant-side computer. The plant-side computer decrypts the ciphertext by using Dec$^+$, and then, inputs a command voltage into the servo amplifier through the D/A board. Note that Gen should be executed to obtain a key pair before the abovementioned periodic control process, and the encrypted controller parameters should be set in advance.

TABLE II
EXPERIMENTAL CONDITIONS FOR ENCRYPTION

Key length	λ	32 bit	64 bit
Scaling parameter for Φ	γ_c	10^3	10^3
Scaling parameter for ξ	γ_p	10^8	10^8

IV. EXPERIMENTAL RESULTS

This section presents the results of some experiments for validating the developed system. Effect of load fluctuation and real-time computation in the proposed system are also indicated.

A. Performance Degradation and System Concealment

It is well known that the addition of a quantizer in a control loop decreases the performance of the control system. Thus, the control performance of the encrypted PID control system is expected to be worse than that of a normal PID control system because Q acts as a quantizer. This study includes investigation of the control performance deterioration caused by encryption in order to confirm the practicality of the proposed system. The validations of the gain and signal concealment are also included, and the conditions of this experiment are listed in Table II.

A model of the plant shown in Fig. 3(a) is obtained by system identification as follows:

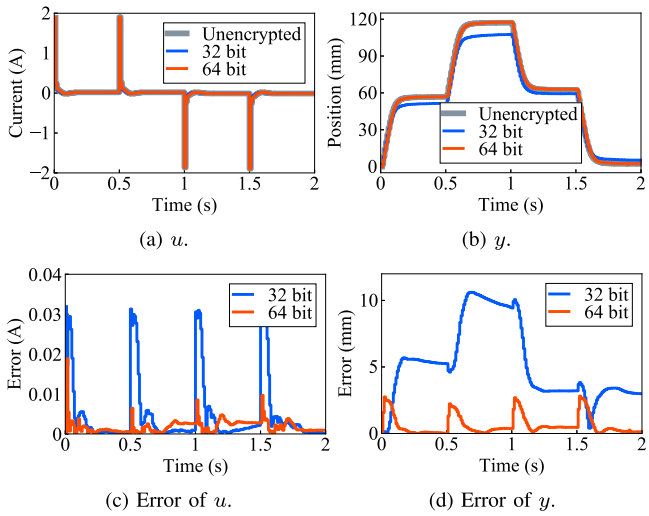
$$A = \begin{bmatrix} 1 & 0.0002 \\ 0 & 0.9959 \end{bmatrix}, \; B = \begin{bmatrix} 0.0009 \\ 8.9008 \end{bmatrix}, \; C = \begin{bmatrix} 1 & 0 \end{bmatrix},$$

where T_s is set to 10 ms. In addition, PID controller gains are tuned manually as $K_P = 0.01$, $K_I = 0.001$, and $K_D = 0.0005$, and gain Φ is represented as follows:

$$\Phi = \begin{bmatrix} 0 & 0 & 1 & -1 \\ 0 & 1 & 0.01 & -0.01 \\ -0.05 & 0.001 & 0.06001 & -0.06001 \end{bmatrix}.$$

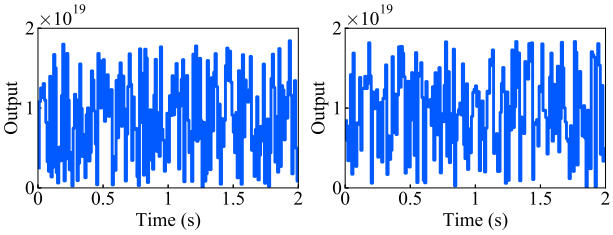
Thus, the encrypted controller gain with a 64 bit key is given as bottom of the next page, where the elements of Enc($\bar{\Phi}$) are displayed as hexadecimal numbers.

Fig. 5 shows the results of the tracking control whose reference input is a stair wave. The position of the stage is detected by a motion capture camera every 10 ms. Figs. 5(a)(b) show the results of the encrypted and unencrypted PID control. It can be seen that the stability of closed-loop systems is retained even if



(a) u. (b) y. (c) Error of u. (d) Error of y.

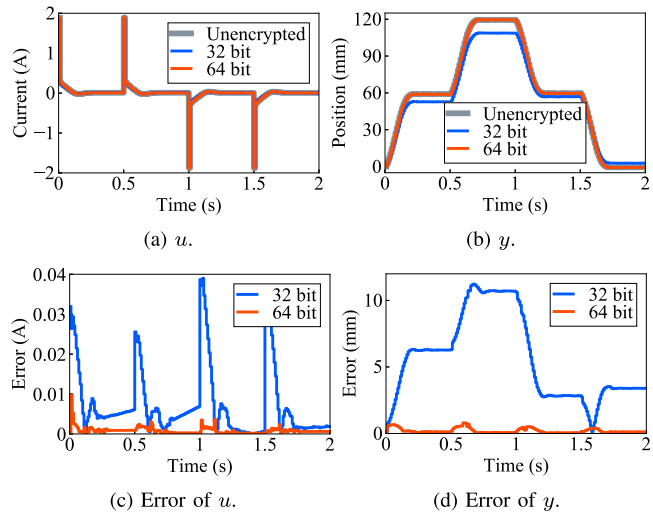
Fig. 5. Comparison between the signals of the encrypted and unencrypted PID controllers.



(a) First element of Enc $(\bar{y})$. (b) Second element of Enc $(\bar{y})$.

Fig. 6. Enc $(\bar{y})$ with the 64 bit key. (a) First element of Enc $(\bar{y})$. (b) Second element of Enc $(\bar{y})$.

the PID controller is encrypted. Figs. 5(c)(d) show the absolute value of errors between the input/output of the encrypted PID control system and that of the unencrypted PID control system, which are denoted as $|\bar{u}_k - u_k|$ and $|\bar{y}_k - y_k|$, respectively. These figures show that the errors with the 64 bit key are smaller than those with the 32 bit key. Thus, it is practically possible to ignore the deterioration in the control performance due to encryption when using a sufficient length key. Furthermore, Fig. 6 shows the elements of Enc$(\bar{y})$ with the 64 bit key flowing in the communication link. The encrypted controller conceals the original signal.



(a) u. (b) y. (c) Error of u. (d) Error of y.

Fig. 7. Comparison between the signals of the encrypted and unencrypted PID controllers with the load.

B. *Effect of Load Fluctuation*

Industrial robots process various tasks by changing their end effector according to the work content. Their model parameters depend on their posture and hand mass. This parameter fluctuation affects the performance of the tracking control and stability of the control systems. Therefore, in order to apply the controller encryption method to the control systems, it is necessary for the encrypted control system to maintain stability under uncertain conditions. This study examines the behavior of the developed system with a 10 kg load on the stage, as shown in Fig. 3(b), by conducting the same experiment as Section IV-A. This load fluctuation affects the moment of inertia and the viscous friction of the stage, leading to a change in the time constant.

Fig. 7 shows the results of the tracking control with the load. These results are similar to those in Fig. 5, which means that the performance of the encrypted controller is independent of uncertainty due to load fluctuations. In other words, the stability of the encrypted control system with sufficient length key is determined only by the original controller properties.

C. *Processing Time*

A key of longer length makes the ciphertext stronger. On the other hand, the encrypted controller increases processing time as the key length increases. Hence, the key length cannot be increased indiscriminately because real-time computation is

$$\mathrm{Enc}\left(\bar{\Phi}\right) = \left(\begin{bmatrix} \texttt{586A58C24B3AA933} & \texttt{7EC78FF4FE079B6A} & \texttt{0335DB4BE5C2638B} & \texttt{29D977C6743E4DE8} \\ \texttt{82F57F31DCBC5782} & \texttt{95EBC813B201A385} & \texttt{888C054705C9D2A1} & \texttt{4D98C5AD7E70FE18} \\ \texttt{EEF2190C8730E276} & \texttt{7AA6409B900A96B9} & \texttt{F0550B325FF6EE13} & \texttt{9FB1FB28014B73FD} \end{bmatrix}, \right.$$

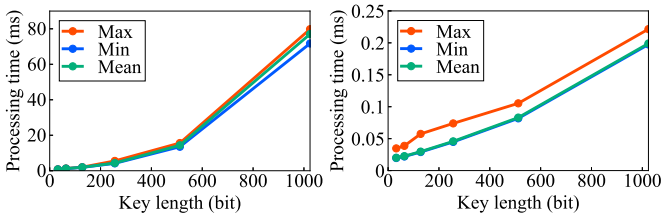
$$\left. \begin{bmatrix} \texttt{BA7F03BA5E19484C} & \texttt{6ACB8DB11E60EB50} & \texttt{5C96D96061699DBE} & \texttt{67813112DFE9C04E} \\ \texttt{5AE01791C824BD5D} & \texttt{97CF692792618658} & \texttt{9163C3D88AFA1FCE} & \texttt{C6208087836E10EF} \\ \texttt{927E0383346D74CA} & \texttt{CFC1B173BDDD228D} & \texttt{147B114E897FF840} & \texttt{AE1D7F44870921C1} \end{bmatrix} \right),$$

TABLE III
PROCESSING TIME AT THE PLANT SIDE

λ (bit)	Max (ms)	Min (ms)	Mean (ms)
32	0.82	0.72	0.74
64	1.28	1.18	1.21
128	2.02	1.88	1.94
256	5.53	4.16	4.33
512	15.62	13.51	14.35
1024	79.79	71.73	77.02

TABLE IV
PROCESSING TIME AT THE FOG SIDE

λ (bit)	Max (ms)	Min (ms)	Mean (ms)
32	0.03	0.02	0.02
64	0.04	0.02	0.02
128	0.06	0.03	0.03
256	0.07	0.04	0.05
512	0.11	0.08	0.08
1024	0.22	0.20	0.20



(a) Plant side. (b) Fog side.

Fig. 8. Processing time excluding communication delay in the developed system.

critical for industrial control systems. This study measures the time taken to execute the encrypted control in the developed system with the library, and describes the relationship between the key length and processing time. The key length is changed from 32 bits to 1024 bits, and the processing time, excluding the communication time, is measured 100,000 times for each key length. Then, the maximum processing time, minimum processing time, and mean processing time are obtained.

Table III and Table IV list the results of the processing time, and Fig. 8 visualizes these results. The times for the plant side in Fig. 8(a) increase exponentially with the key length. In contrast, the times of fog side in Fig. 8(b) are almost proportional to it. These results indicate that the increase in processing time required for Enc and Dec^+ is large compared to Mult. Note that the results cannot be compared with the processing time of other methods because the proposed system is the first implementation of an encrypted control system in the practical setting. Although we need more examination, the results are useful to choose the appropriate key length.

V. CONCLUSION

This letter develops a secure fog computing-based control system, which serves as the first implementation of an encrypted control system in an actual industrial setting. The controller gain and signals are concealed against adversaries. The developed system is resilient to eavesdropping attacks and prevents zero dynamics attacks. Thus, the controller encryption method can be employed as a new component of defense in depth for industrial control systems.

The experiment results confirm the feasibility of tracking control under load fluctuation and indicate the relationship between the key length and processing time. The results in Section IV-A and IV-B suggest that the controller encryption method is sufficiently practical. From the viewpoint of security level and control performance degradation, the key length should be large. However, the results in Section IV-C suggest that the key length is restricted by the processing time, especially the time of encryption and decryption. Therefore, the processes of encryption and decryption need to be implemented in the hardware (e.g., via a field programmable gate array) so that the encrypted control systems can be put to practical use in a more resource-limited setting.

In future work, we will consider a fog computing-based control system with the cloud for higher-layer control. Additionally, we will implement an attack detection method [19] to prevent DoS attacks, gain falsifications, and replay attacks.

ACKNOWLEDGMENT

The authors would like to thank Mr. Masahiro Kusaka for technical assistance to develop the C language library.

REFERENCES

[1] Y. Xia, "Cloud control systems," *IEEE/CAA J. Automatica Sinica*, vol. 2, no. 2, pp. 134–142, Apr. 2015.
[2] H. Esen, M. Adachi, D. Bernardini, A. Bemporad, D. Rost, and J. Knodel, "Control as a service (CaaS): Cloud-based software architecture for automotive control applications," in *Proc. Int. Workshop Swarm Edge Cloud*, Seattle, WA, USA, 2015, pp. 13–18.
[3] A. Vick, V. Vonásek, R. Pěnička, and J. Krüger, "Robot control as a service towards cloud-based motion planning and control for industrial robots," in *Proc. Int. Workshop Robot Motion Control*, Poznan, Poland, 2015, pp. 33–39.
[4] G. Mohanarajah, R. D'Andrea, and M. Waibel, "Rapyuta: A cloud robotics platform," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 481–493, Apr. 2015.
[5] M. Waibel *et al.*, "Roboearth," *IEEE Robot. Autom. Mag.*, vol. 18, no. 2, pp. 69–82, Jun. 2011.
[6] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg, "A survey of research on cloud robotics and automation," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 398–409, Apr. 2015.
[7] A. Botta, W. de Donato, V. Persico, and A. Pescape, "Integration of cloud computing and Internet of Things: A survey," *Future Gener. Comput. Syst.*, vol. 56, pp. 684–700, 2016.
[8] M. S. Mahmoud and M. M. Hamdan, "Fundamental issues in networked control systems," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 5, pp. 902–922, 2018.
[9] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. 1st Edition MCC Workshop Mobile Cloud Comput.*, Helsinki, Finland, 2012, pp. 13–16.
[10] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.
[11] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog computing for the Internet of Things: Security and privacy issues," *IEEE Internet Comput.*, vol. 21, no. 2, pp. 34–42, Mar./Apr. 2017.
[12] M. Mukherjee *et al.*, "Security and privacy in fog computing: Challenges," *IEEE Access*, vol. 5, pp. 19 293–19 304, 2017.
[13] P. Zhang, M. Zhou, and G. Fortino, "Security and trust issues in fog computing: A survey," *Future Gener. Comput. Syst.*, vol. 88, pp. 16–27, 2018.
[14] A. A. Cardenas, S. Amin, and S. Sastry, "Secure control: Towards survivable cyber-physical systems," in *Proc. Int. Conf. Distrib. Comput. Syst. Workshops*, Beijing, China, 2008, pp. 495–500.

[15] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, "A secure control framework for resource-limited adversaries," *Automatica*, vol. 51, pp. 135–148, 2015.
[16] D. Quarta, M. Pogliani, M. Polino, F. Maggi, A. M. Zanchettin, and S. Zanero, "An experimental security analysis of an industrial robot controller," in *Proc. IEEE Symp. Secur. Privacy*, San Jose, CA, USA, 2017, pp. 268–286.
[17] K. Kogiso and T. Fujita, "Cyber-security enhancement of networked control systems using homomorphic encryption," in *Proc. IEEE Conf. Decis. Control*, Osaka, Japan, 2015, pp. 6836–6843.
[18] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inf. Theory*, vol. 31, no. 4, pp. 469–472, Jul. 1985.
[19] K. Kogiso, "Attack detection and prevention for encrypted control systems by applicataion of switching-key management," in *Proc. IEEE Conf. Decis. Control*, Miami Beach, FL, USA, 2018, pp. 5032–5037.
[20] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. EUROCRYPT*, Prague, Czech Republic, 1999, pp. 223–238.
[21] F. Farokhi, I. Shames, and N. Batterham, "Secure and private cloud-based control using semi-homomorphic encryption," *IFAC-PapersOnLine*, vol. 49, no. 22, pp. 163–168, 2016.
[22] F. Farokhi, I. Shames, and N. Batterham, "Secure and private control using semi-homomorphic encryption," *Control Eng. Practice*, vol. 67, pp. 13–20, 2017.
[23] J. Kim *et al.*, "Encrypting controller using fully homomorphic encryption for security of cyber-physical systems," *IFAC-PapersOnLine*, vol. 49, no. 22, pp. 175–180, 2016.
[24] K. Sato and S. Azuma, "Secure real-time control through fog computation," *IEEE Trans. Ind. Informat.*, vol. 15, no. 2, pp. 1017–1026, Feb. 2019.
[25] K. Ishikawa, K. Nagasawa, K. Kogiso, and K. Sawada, "Experimental validation of encrypted controller implemented on Raspberry Pi," in *Proc. Int. Conf. Cyber-Physical Syst., Netw., Appl.*, Nagoya, Japan, 2016, pp. 1–6.
[26] K. Kogiso, R. Baba, and M. Kusaka, "Development and examination of encrypted control systems," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, Auckland, New Zealand, 2018, pp. 1338–1343.
[27] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystem," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.
[28] K. Kogiso, "Upper-bound analysis of performance degradation in encrypted control system," in *Proc. Amer. Control Conf.*, Milwaukee, WI, USA, 2018, pp. 1250–1255.
[29] P. Mell and T. Grance, "The NIST definition of cloud computing," NIST Special Publication, Gaithersburg, MD, USA, Rep. no. 800-145, 2011.
[30] Y. Yokokura, Side warehouse of laboratory. [Online]. Available: http://www.sidewarehouse.net/arcs51/index.html, Accessed: Jan. 23, 2020.