IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 26 December 2023, accepted 7 January 2024, date of publication 12 January 2024, date of current version 23 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3353289

RESEARCH ARTICLE

Experimental Validation of the Attack-Detection Capability of Encrypted Control Systems Using Man-in-the-Middle Attacks

AKANE KOSUGI[1], (Graduate Student Member, IEEE), KAORU TERANISHI[1,2], (Graduate Student Member, IEEE), AND KIMINAO KOGISO[1], (Member, IEEE)

[1]Department of Mechanical and Intelligent Systems Engineering, The University of Electro-Communications, Chofu, Tokyo 182-8585, Japan
[2]Japan Society for the Promotion of Science, Chiyoda, Tokyo 102-0083, Japan

Corresponding author: Akane Kosugi (kosugi@uec.ac.jp)

This work was supported by the Japan Society for the Promotion of Science (JSPS) Grants-in-Aid for Scientific Research (B) Grant Number JP22H01509.

ABSTRACT In this study, the effectiveness of encrypted control systems in detecting attacks is experimentally demonstrated using a networked control system testbed that allows for man-in-the-middle (MITM) attacks. The developed testbed is a networked position control system for an industrial-use linear stage. Generally, an attacker can reroute and modify packet data via a wireless router, harnessing the address-resolution-protocol-spoofing technique, which allows for the execution of MITM attacks, such as falsification and replay attacks. The deployed MITM-attack-detection method is grounded on a threshold-based method that monitors control inputs. The demonstration examines falsification- and replay-attack scenarios across unencrypted, static-key, and key-updatable encrypted control systems. The results confirm that encrypted control systems are both effective and apt in detecting attacks in real time. Furthermore, the potential for developing alternative attack-detection schemes based on variations in processing times is discussed.

INDEX TERMS Encrypted control, man-in-the-middle attack, attack detection, experimental validation.

I. INTRODUCTION

The networked control system (NCS) has emerged as a pivotal technology, bolstering computer scalability and enhancing management flexibility [1]. Advancements in communication technologies have facilitated the integration of NCS into an array of systems. These range from small-scale systems, such as unmanned aerial vehicles [2], and remote control over mobile networks [3], to expansive industrial control systems integral to various sectors, including oil and gas production, power, transportation, and manufacturing [4], [5], [6]. This proliferation in networking has amplified the capabilities of control systems, enabling the incorporation of a relatively broad spectrum of software components. These components are adept at intricate computations, encompassing equipment configuration, manufacturing-process optimization, cloud-based data processing, and robot-operation orchestration from distant locations.

Concurrently, there are some concerns regarding the vulnerability of systems that are essential for equipment provisioning, service delivery, and safety maintenance. These systems are susceptible to potential cyberattacks, given their partial integration with cyberspace. Notable instances of cyberattacks on control systems include the Stuxnet malware that targeted uranium-enrichment centrifuges at an Iranian nuclear facility [7], [8]. It demonstrated that the manipulation of information in cyberspace leads to the physical destruction of the controlled plant; the Industroyer malware, which disrupted a Ukrainian power facility, leading to an extensive power blackout [9]; the Triton malware that attacked a safety instrumented system based on reverse-engineered protocols,

The associate editor coordinating the review of this manuscript and approving it for publication was Hosam El-Ocla.

ultimately shutting down the plant [10]; and the Maroochy Water Services attack, where remote manipulation of sewage pumping stations caused the release of untreated sewage into local waterways for a three-month period [11].

Several defensive measures against various attack schemes, such as denial-of-service attacks [12], [13], [14], false data injection attacks [15], [16], [17], [18], zero-dynamics attacks [19], [20], [21], [22], covert attacks [23], [24], and replay attacks [25], [26], [27], [28], [29], have been actively proposed. Some of these studies consider the situation where the attacker knows about the target control system prior to conducting a sophisticated attack on it. Moreover, in exploring reactive measures against cyberattacks, attack detection plays a crucial role in bolstering security. The detection method presented in [15] uses the χ^2-detector based on Kalman filters; however, the detection performance is limited by the model's uncertainties, and there is a detection delay. The threshold-based detection method in [26] can detect replay attacks; however, it does not operate in real time. These studies focusing on approaches based on state estimation and statistics have drawbacks that require solutions.

Meanwhile, as a preventive and reactive measure against cyberattacks, encrypted control is a promising and innovative methodology that ensures the secure implementation of controllers [30], [31], [32], [33]. This approach encrypts control parameters and/or communications using homomorphic encryption [34], [35], [36], [37], which allows operations on encrypted data to be performed. Further, it has been applied to enhance the security of a linear control [38], [39], a model predictive control [40], [41], a cooperative control and consensus [42], [43], [44], [45], and a nonlinear control [46], [47]. Moreover, encrypted control systems are apt for integrating threshold-based attack detection. This suitability arises from the fact that when encrypted data and parameters are inappropriately overwritten due to a cyberattack, the correctness is compromised. This typically results in a conspicuous noise leakage during the decryption process. Indeed, several studies [48], [49], [50] have assessed the efficacy of encrypted control systems equipped with attack-detection mechanisms.

However, these studies were conducted from a theoretical perspective, typically involving the simulation of idealized cyberattacks programmatically within control system simulations or testbeds. Additionally, a demo abstract [51] conveys that the encrypted control framework significantly reduces false detections in an industrial control-system testbed, although it does not provide specifics. Based on a theoretical property of control systems regarding attack detection, the practical demonstration of an actual cyberattack on control devices and communication systems, which is capable of overwriting parameters and packet data, is crucial for developing secure control technologies. Such demonstrations can help clarify both the challenges and feasibility of launching these attacks, shedding light on the technical intricacies. Furthermore, these demonstrations will provide insights that could facilitate the discovery of novel detection methods.

The objective of this study is to demonstrate the effectiveness of encrypted control systems in attack detection using an NCS testbed that allows for man-in-the-middle (MITM) attacks. The encrypted control is grounded in multiplicatively homomorphic encryption schemes, such as static-key and key-updatable ElGamal-based encryption [30], [52]. These schemes preserve the confidentiality of control parameters during operations and ensure a relatively lightweight computational load. The testbed developed for this study is a networked PID position control system for an industrial-use linear stage. In the system, wireless communications between the plant and controller sides are achieved via a wireless router. An attacker can reroute and modify packet data from the router, leveraging the address resolution protocol (ARP)-spoofing technique, which allows for the execution of falsification and replay attacks. For attack detection, this study uses the threshold-based method presented in [49], which monitors the control inputs, and also reveals that the employed detector results in detecting the attacks as a theorem. The theorem provides a novel result, which is the main difference from the previous studies using the threshold-based detector [48], [49]. The demonstration examines three attack scenarios. In the first, involving a falsification attack, a portion of the packet data is modified with inappropriate values, whereas in the second, also involving a falsification attack, the packet data are overwritten using proper ciphertexts through a public key. In the third scenario, involving a replay attack, the current packet data are replaced with previously recorded data. This study confirms that the key-updatable encrypted control system is effective and appropriate in the sense that the attack can be detected, in contrast with the static-key encrypted and unencrypted control systems, which support the theorem. Moreover, using the experimental results, we discuss the potential of developing a different attack-detection scheme, considering variations in processing time.

A. CONTRIBUTIONS

This study offers both practical and theoretical contributions. The practical contribution involves the development of key-updatable encrypted control systems involving the threshold-based detector, which address MITM attacks by leveraging practical network protocols to enhance cybersecurity. Implementing real-world attack scenarios provides insights into practical attack detection and defense strategies. Although similar results have been presented in previous studies, e.g., [48] and [49], which explored simulated attacks within control algorithms, this study marks the beginning of experimental demonstrations. The theoretical contribution involves revealing analytical features of encrypted control systems with the detector as a theorem. This theorem discusses the probability of detecting falsification or replay attacks, supported by experimental results on the attack detection of encrypted control systems.

B. ORGANIZATION OF THIS PAPER

This paper is organized as follows. Section II provides preliminary information on the encrypted control methodology. In Section III, the positioning-control-system testbed developed using the industrial-use linear stage is detailed. Section IV describes the cyberattack environment and elaborates on the execution of falsification and replay attacks. Section V demonstrates the effectiveness of the attack-detection mechanism in the encrypted control systems. Section VI discusses the potential of developing a different attack-detection method based on variations in the processing times. Finally, Section VII concludes this paper.

II. PRELIMINARIES

A. NOTATION

The sets of real numbers, rational numbers, integers, primes, security parameters, key pairs, public keys, secret keys, plaintexts, and ciphertexts are denoted by $\mathbb{R}$, $\mathbb{Q}$, $\mathbb{Z}$, $\mathcal{S}$, $\mathcal{K}$, $\mathcal{K}_\mathsf{p}$, $\mathcal{K}_\mathsf{s}$, $\mathcal{M}$, and $\mathcal{C}$, respectively. We define sets $\mathbb{R}^+ := \{x \in \mathbb{R} \mid 0 \leq x\}$, $\mathbb{Z}^+ := \{z \in \mathbb{Z} \mid 0 \leq z\}$, and $\mathbb{Z}_n := \{z \in \mathbb{Z} \mid 0 \leq z < n\}$. The set of vectors whose sizes are n is denoted by $\mathbb{R}^n$, and the set of matrices whose sizes are $m \times n$ is denoted by $\mathbb{R}^{m\times n}$. The ith element of vector v and the (i, j)th entry of matrix M are denoted by v_i and M_{ij}, respectively.

B. DYNAMIC ELGAMAL ENCRYPTION

The ElGamal encryption is a public-key cryptosystem with multiplicative homomorphism. A key-updatable ElGamal cryptosystem is defined as $\mathcal{E} := (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, T_\mathcal{K}, T_\mathcal{C})$. These transition maps are defined as follows [52]:

$$
\begin{aligned}
\mathsf{Gen} &: \mathcal{S} \to \mathcal{K} = \mathcal{K}_\mathsf{p} \times \mathcal{K}_\mathsf{s},\\
&: \lambda \mapsto (\mathsf{pk}, \mathsf{sk}) = ((\mathbb{G}, q, g, h), s),\\
\mathsf{Enc} &: \mathcal{M} \times \mathcal{K}_\mathsf{p} \to \mathcal{C},\\
&: (m, \mathsf{pk}) \mapsto c = (c_1, c_2) = (g^r \bmod p, mh^r \bmod p),\\
\mathsf{Dec} &: \mathcal{C} \times \mathcal{K}_\mathsf{s} \to \mathcal{M},\\
&: ((c_1, c_2), \mathsf{sk}) \mapsto {c_1}^{-s} c_2 \bmod p,\\
T_\mathcal{K} &: \mathcal{K} \to \mathcal{K}, \qquad (1)\\
&: (\mathsf{pk}, \mathsf{sk}) \mapsto ((p, q, g, hg^{s'} \bmod p), s + s' \bmod q),\\
T_\mathcal{C} &: \mathcal{C} \to \mathcal{C}, \qquad (2)\\
&: (c_1, c_2) \mapsto (c_1 g^{r'} \bmod p, (c_1 g^{r'})^{s'} c_2 h^{r'} \bmod p),
\end{aligned}
$$

where Gen is a key-generation algorithm, Enc is an encryption algorithm, Dec is a decryption algorithm, $T_\mathcal{K}$ is the mapping that updates the key, and $T_\mathcal{C}$ is the mapping that updates the ciphertext of the control parameter. pk is a public key, sk is a secret key, λ is a security parameter, q is a λ-bit prime, and $p = 2q+1$ is a safe prime. Parameter g represents a generator of a cyclic group $\mathbb{G} := \{g^i \bmod p \mid i \in \mathbb{Z}_q\}$ such that $g^q \bmod p = 1$, $h = g^s \bmod p$, and $\mathcal{C} = \mathbb{G}^2$. r and s are random numbers in $\mathbb{Z}_q$, s' and r' are random numbers in $\mathbb{Z}_q$ generated by $T_\mathcal{K}$ and $T_\mathcal{C}$ respectively; and h is pk before updating by $T_\mathcal{K}$. For $m, m' \in \mathcal{M}$, the ElGamal encryption satisfies the following homomorphism:

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m) * \mathsf{Enc}(\mathsf{pk}, m') \bmod p) = mm' \bmod p,$$

where $*$ is the Hadamard product. In a dynamic key scheme, the key-update operation is added after each f operation. For any $k \in \mathcal{K}$, $m \in \mathcal{M}$ and $c = \mathsf{Enc}(k, m) \in \mathcal{C}$, the encryption scheme $\mathcal{E}$ and the mapping $T_\mathcal{K}$, $T_\mathcal{C}$ at time step $t \in \mathbb{Z}^+$ satisfy the following homomorphism:

$$
\begin{aligned}
\mathsf{Dec}(T_\mathcal{K}(t), \mathsf{Enc}(T_\mathcal{K}(t), m)) &= m,\\
\mathsf{Dec}(T_\mathcal{K}(t), T_\mathcal{C}(c)) &= \mathsf{Dec}(k, c).
\end{aligned}
$$

In addition, if s' and r' are set to zero in the encryption scheme $\mathcal{E}$, the keys and ciphertexts do not change over time. In this case, $\mathcal{E}$ is identical to the static-key encryption scheme $\mathcal{E}_s := (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ used in [30].

TABLE 1. Specifications of the experimental apparatuses.

Plant	
Slide screw:	LX3010CP-MX
Length	1250 mm
Lead	10 mm
AC servo motor:	MITSUBISHI HK-KT13W
Rated power	100 W
Rated torque	0.32 N·m
Rated speed	3000 r/min
Rated current	1.2 A
Moment of inertia	6.86×10^{-6} kg·m^2
Resolution	67108864 ppr
Servo amplifier:	MITSUBISHI MR-J5-10A
Main circuit power supply	1/3-phase 200 to 240 VAC 50/60 Hz
Wireless router:	Apple AirMac Extreme ME918J A
PC	
CPU & Memory	Intel Core i7-10700K 3.80 GHz
OS	CentOS Linux 8
DA / AD board:	Interface PEX-340216 (Resolution 16bit)
Counter board:	Interface PEX-632104 (Resolution 32bit)
Controller	
PC:	MacBook Pro (13-inch, M1, 2021)
CPU & Memory	Apple M1 16 GB
OS	MacOS version 11.5.2
Attacker's device	
PC:	Raspberry Pi 4B
CPU & Memory	Broadcom BCM2711 1.5GHz 8 GB
OS	Kali Linux 4.19.118

III. NETWORKED CONTROL SYSTEM

This section introduces the developed NCS testbed and its specifications.

A. THE DEVELOPED TESTBED

The testbed system developed in this study embodies a position control system for an industrial-use linear stage. The system features wireless communication between the stage and the controller via a router. Fig. 1 provides the whole view of the developed NCS, where the red and white arrows represent wired and wireless communications, respectively. The control system comprises a motorized linear stage, a plant-side PC, a controller-side PC, a wireless router, and

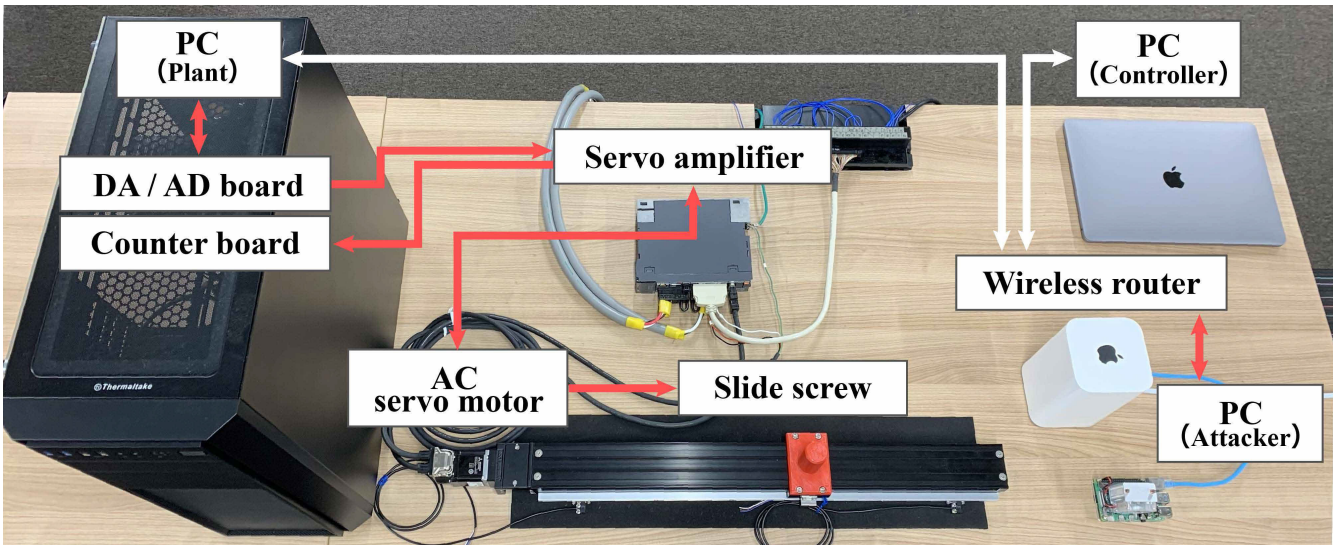


FIGURE 1. Whole view of the developed networked control system that allows the execution of man-in-the-middle attacks.

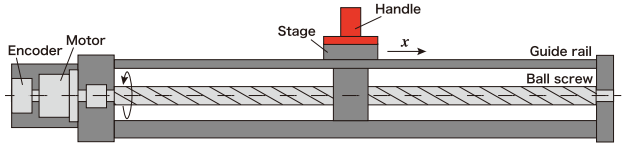


FIGURE 2. Side view of the linear stage schematic diagram.

the attacker's computer. Their respective specifications are detailed in Table 1.

The motorized stage, illustrated in Fig. 2, consists of a handle fixed on a stage, a ball screw, an AC servo motor as an actuator, and an encoder attached to the motor as a sensor. With power supplied by a servo amplifier, the motor generates torque to rotate the ball screw. The ball screw mechanism then converts between rotational and linear motion, and the stage moves in the linear direction. The encoder measures the stage position and communicates this information to the plant-side PC via the amplifier.

The plant-side PC, in turn, transmits the received control input from the controller-side PC to the amplifier and communicates the received stage position from the amplifier to the controller-side PC. Due to the functionality of the ARCS6 C++ library,[1] these processes on the plant-side PC can be executed in real time. For the controller-side PC, it runs a control algorithm written in C++ that calculates a control input using the control parameters and the received stage position. The computed control input is subsequently dispatched to the plant-side PC. Furthermore, signal communication between the two PCs occurs wirelessly via the router. TCP/IP was chosen as the protocol for the wireless communication to ensure reliable signal exchange, given its widely recognized functions, such as data interpolation and data-order maintenance.

The third PC is for the attacker and is wired to the router. The attacker uses the Scapy library[2] in Python, a packet manipulation tool for computer networks, to perform the cyberattacks on the position control system. The details of how the cyberattacks are performed will be explained in Section IV.

In this study, the control period, denoted as T_s, was set to 0.5 s to establish a real-time NCS, considering the transmission time between the plant-side and controller-side PCs via the attacker's PC, as well as the computation time required to perform the cyberattacks. These computation times will be discussed further in Section V.

The discussion in this subsection is limited to an unencrypted configuration of the NCS and does not include explanations of any encryption and decryption processes related to the encrypted control system. In subsequent subsections, we will introduce encrypted control and revise the explanation to accommodate the configuration in the context of an encrypted control system.

B. NETWORKED CONTROL SYSTEM

The control objective of the NCS is to track the stage position to a given reference; therefore, we designed a discrete-time PID controller in a state-space representation,

$$f : \psi(t) = \Phi\xi(t), \tag{3}$$

with

$$\psi(t) := \begin{bmatrix} z(t+1) \\ u(t) \end{bmatrix}, \quad \Phi := \begin{bmatrix} A_c & B_c \\ C_c & D_c \end{bmatrix}, \quad \xi := \begin{bmatrix} z \\ v \end{bmatrix},$$

where $t \in \mathbb{Z}^+$ is a step; $z := [\, e \;\; w \,]^{\mathrm{T}} \in \mathbb{R}^n$ is a state; $v := [\, r \;\; y \,]^{\mathrm{T}} \in \mathbb{R}^l$ is an input; $u \in \mathbb{R}^m$ is an output (a control input); $\Phi \in \mathbb{R}^{\alpha\times\beta}$ is a constant coefficient with $\alpha := n + m$ and $\beta := n+l$; r is a reference to a stage position; y is a measured

[1] https://github.com/Sidewarehouse/ARCS6

[2] https://github.com/secdev/scapy

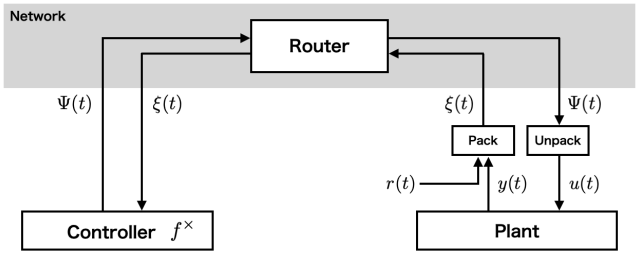


FIGURE 3. Block diagram of the unencrypted networked control system.

position; e is a feedback (tracking) error between r and y, i.e., $e := r - y$; and w is a state of the integral compensator, i.e., $w(t+1) := \sum_{k=0}^{t} T_s e(\tau) = w(t) + T_s e(t)$. Now, we consider the PID controller with $n = 2$, $m = 1$, and $l = 2$. The system coefficient $\Phi \in \mathbb{R}^{3\times 4}$ consists of

$$A_c = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix},\ B_c = \begin{bmatrix} 1 & -1 \\ T_s & -T_s \end{bmatrix},\ C_c = \begin{bmatrix} -\dfrac{K_D}{T_s} K_I \end{bmatrix},$$
$$D_c = \begin{bmatrix} K_P + K_I T_s + \dfrac{K_D}{T_s} & -K_P - K_I T_s - \dfrac{K_D}{T_s} \end{bmatrix},$$

where K_P, K_I, and K_D are proportional, integral, and derivative gains, respectively. By trial and error, we determined the gains: $K_P = 1.2 \times 10^{-3}$, $K_I = 6.0 \times 10^{-3}$, and $K_D = 1.0 \times 10^{-4}$, that result in

$$\Phi = \begin{bmatrix} 0 & 0 & 1 & -1 \\ 0 & 1 & 0.5 & -0.5 \\ -0.0002 & 0.0007 & 0.00175 & -0.00175 \end{bmatrix}.$$

The block diagram of the unencrypted NCS is shown in Fig. 3. In this configuration, the communication signals are transmitted as plain text, and the operation (3) is performed over plain text. In the following subsection, we describe the reconstruction of the configuration to incorporate a key-updatable encryption scheme to enhance the control system's cybersecurity.

C. KEY-UPDATABLE ENCRYPTED CONTROL SYSTEM

This section introduces the key-updatable encrypted controller presented in [52]. Based on the controller encryption technique [30], the linear operation (3) is divided into multiplication and addition. That is, $f = f^+ \circ f^\times$, where $f^\times(\Phi, \xi) := \left[\Phi_1\xi_1\ \Phi_2\xi_2\ \cdots\ \Phi_\beta\xi_\beta\right] =: \Psi$ and $f^+(\Psi) := \sum_{i=1}^{\beta} \Psi_i = \sum_{i=1}^{\beta} \Phi_i\xi_i$. The division enables us to incorporate the multiplicative homomorphism of the ElGamal encryption into the control system to conceal the coefficient and signals.

Definition 1 [52]: Let us assume that an unencrypted controller, f, is in (3), and that $\mathcal{E}$ is modified to $\mathcal{E}^\star = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}^+, T_\mathcal{K}, T_\mathcal{C}, \mathscr{E}_\gamma, \mathscr{D}_\gamma)$, where $\mathscr{E}_\gamma$ and $\mathscr{D}_\gamma$ are an encoder and a decoder, respectively:

$$\mathscr{E}_\gamma : \mathbb{R} \ni x \mapsto \bar{x} = \lceil \gamma x + a(\gamma x) \rfloor \in \mathcal{M},$$
$$\mathscr{D}_\gamma : \mathcal{M} \ni \bar{x} \mapsto \check{x} = \frac{\bar{x} - b(\bar{x})}{\gamma} \in \mathbb{Q},$$

$$a(\gamma x) := \begin{cases} p, & \gamma x < 0, \\ 0, & \gamma x \geq 0, \end{cases} \qquad b(\bar{x}) := \begin{cases} p, & \bar{x} > q, \\ 0, & \bar{x} \leq q, \end{cases}$$

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 $\mathcal{M}$, p is a modulo parameter used in operations of ElGamal encryption, and $\mathsf{Dec}^+ := f^+ \circ \mathsf{Dec}$. Let C_Φ, C_ξ, and C_Ψ be ciphertexts corresponding to Φ, ξ, and Ψ, respectively. Thus, an encrypted controller $f^\times_{\mathcal{E}^\star}$ with dynamic keys is defined as follows:

$$f^\times_{\mathcal{E}^\star} : (C_\Phi(t), C_\xi(t)) \mapsto C_\Psi(t),\ \forall t \in \mathbb{Z}^+,$$

with the following update rules using (1) and (2):

$$(\mathsf{pk}(t+1), \mathsf{sk}(t+1)) = T_\mathcal{K}(\mathsf{pk}(t), \mathsf{sk}(t)), \tag{4a}$$
$$C_\Phi(t+1) = T_\mathcal{C}(C_\Phi(t)), \tag{4b}$$
$$C_\xi(t+1) = T_\mathcal{C}(C_\xi(t)), \tag{4c}$$

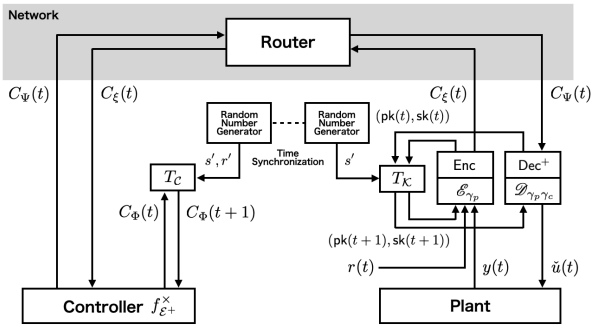
where the initial keys are given by $(\mathsf{pk}(0), \mathsf{sk}(0)) = \mathsf{Gen}(k)$, *and* $C_\Psi =: (C^1_\Psi, C^2_\Psi)$ *with* $C^l_{\Psi_{ij}}(t) = C^l_{\Phi_{ij}}(t) C^l_{\xi_j}(t) \bmod p, \forall i \in \mathbb{Z}^+_{\alpha+1}, \forall j \in \mathbb{Z}^+_{\beta+1}, \forall l \in \{1, 2\}$. *The encrypted controller sends computed ciphertext $C_\Psi(t)$ to the plant side. Additionally, $\mathscr{E}_\gamma$ and $\mathscr{D}_\gamma$ operate as quantizers; accordingly, the operation causes quantization errors, which decrease as the λ increases.*

From **Definition 1**, it is confirmed that under (4a), the equations $T_\mathcal{C}(C_\Phi(t)) = \mathsf{Enc}(\mathscr{E}_{\gamma_c}(\Phi), \mathsf{pk}(t+1))$ and $T_\mathcal{C}(C_\xi(t)) = \mathsf{Enc}(\mathscr{E}_{\gamma_p}(\xi(t+1)), \mathsf{pk}(t+1))$ hold, where γ_c and γ_p are scaling parameters for Φ and ξ, respectively. Thereby, the control input is extracted on the plant side by

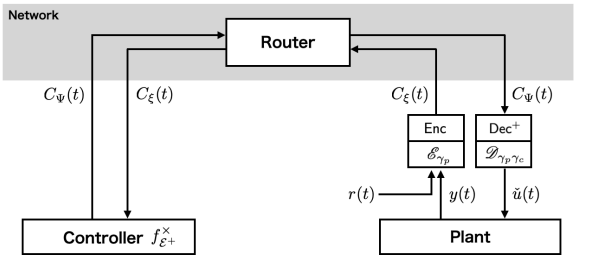
$$\check{u}(t) = f^+(\bar{\Psi}(t)) = \mathscr{D}_{\gamma_c\gamma_p}(\mathsf{Dec}^+(C_\Psi(t), \mathsf{sk}(t))),$$

where there exists $\delta \in \mathbb{R}^m$ at step t such that $u(t) = \check{u}(t) + \delta(t)$ holds, corresponding to the total quantization errors. The errors impact the stability of the encrypted control system, which will be discussed **Remark 1**. Moreover, **Definition 1** does not use the property of the multiplication of the controller $f^\times$, defined in Section III-B, for constructing the encrypted control system. Therefore, the (unencrypted) controller can be designed independently of the controller encryption process.

The definition implies that Φ and ξ on the controller side are encrypted during the operation of the control system, as shown in Fig. 4. On examining the plant side in the figure, we see that a measured position by the sensor, y, and a reference to the stage position, r, are encrypted and sent to the controller via the router. On the controller side, the encrypted signal, C_ξ, and the encrypted coefficients, C_Φ, perform the homomorphic operation. The resulting C_Ψ, corresponding to the control input, is sent to the plant side, and the decrypted signal, $\check{u}$, is inputted to the plant. Furthermore, there is a key-updatable mechanism in the control system. A random-number generator for s' is required on both the controller and plant sides, which assumes the same seed and time



(a) Key-updatable encrypted control system



(b) Static-key encrypted control system

FIGURE 4. Configurations of two types of encrypted control systems in terms of key management.

synchronization. To update (4a), a random number s' is required, which is also used in (4b) and (4c). Moreover, another random-number generator for r' is required on the controller side to update the encrypted system matrix C_Φ with (4b). These generators provide random numbers periodically and offer them to $T_\mathcal{C}$ and $T_\mathcal{K}$. Additionally, for example, the encrypted coefficients with $\lambda = 64$ and $\gamma_c = 10^8$ are as shown at the bottom of the page, where the elements are displayed as hexadecimal numbers. Meanwhile, when considering a static-key encrypted controller, r' and s' are set to zero for any step, and in this case, the configuration is simplified, as shown in Fig. 4(b).

Remark 1: The presented encrypted control system incorporates the ElGamal-encryption scheme through the encoder and decoder, introducing quantization errors induced by these components. These errors may compromise the system's stability. To ensure stability in the encrypted control system, it is necessary to implement methods such as dynamic quantizers [53] and the design of controllers with integer coefficients [54], [55], [56], [57]. Moreover, when the impact of quantization errors is significant on tracking performance, increasing the key length becomes necessary.

IV. MITM ATTACKS

This section provides a detailed methodology for executing the MITM attacks, such as falsification and replay attacks, against the developed control system.

A. ARP SPOOFING

The ARP is a vital tool for mapping an IP address to its corresponding physical machine address within a local area network, such as a media access control (MAC) address. ARP identifies and registers MAC addresses associated with specific IP addresses. Once an IP-MAC address pair is established, this mapping is cached within the ARP table to expedite future communications. However, a critical area of concern is that this table, which maintains MAC-IP correspondences, lacks security measures, such as authentication or encryption during updates. Attackers can exploit this vulnerability by deploying ARP requests to extract the MAC address of a node associated with a specific target IP address. Thereafter, they utilize ARP replies to masquerade as a MAC address in the ARP response directed back to the attacker. This action overwrites the cached ARP information. Consequently, when the poisoned cache is employed for data transmission, the packets are misdirected, compromising the network's integrity.

This study uses ARP cache poisoning to alter the MAC address, as demonstrated in Table 2. This manipulation reroutes the communication path between the controller and the plant, directing it toward the attacker. Initially, the attacker connects a computer directly to the router. Leveraging ARP, the attacker can transmit the overwritten cache to every controller, router, and plant-side computer, thereby replacing their original ARP caches with the contaminated ones. Consequently, even though the plant-side computer intends to transmit the packet to the controller, the information inadvertently passes through the attacker's computer. Considering the poisoned ARP cache, the NCS is depicted schematically in Fig. 6.

Throughout the ARP-spoofing process, the source and destination MAC addresses of the received packet are overwritten. This alteration allows for the interception and forgery of the packet without alerting legitimate system operators. Consequently, the attacker can monitor and modify the packet data, as illustrated in Fig. 5, using the Python Scapy module. The subsequent section explains the methodology and specifics concerning the overwriting of the packet data.

$$\mathsf{Enc}(\mathscr{E}_{\gamma_c}(\Phi), \mathsf{pk}(0)) = \left(\begin{bmatrix} \texttt{84b4344d059038a4176af9bb7ca2c703bfc389a9f2b1fe7fb8cdcff478a69853} \\ \texttt{1956f8c7170d8a5d35fc1a84fde52d61826e4cc6a3daf7b332fbf769a572896d} \\ \texttt{1fa265237225484be90908f1581d2887ef47cc615b78227af211c65b20480eb6} \end{bmatrix}, \right.$$
$$\left. \begin{bmatrix} \texttt{82a91aa3d9d9a7203658d40c390e9bc73e8d6793178508b9f001f702bf11ffb9} \\ \texttt{e91e6661ddd7095b9c05f3f1f28624fe5155a344ae39e929dcb8af517842075} \\ \texttt{c99ecf2cb841c132165c05ce8f566d8996a442f79bea994d97e88da0d5f72ca6} \end{bmatrix} \right),$$

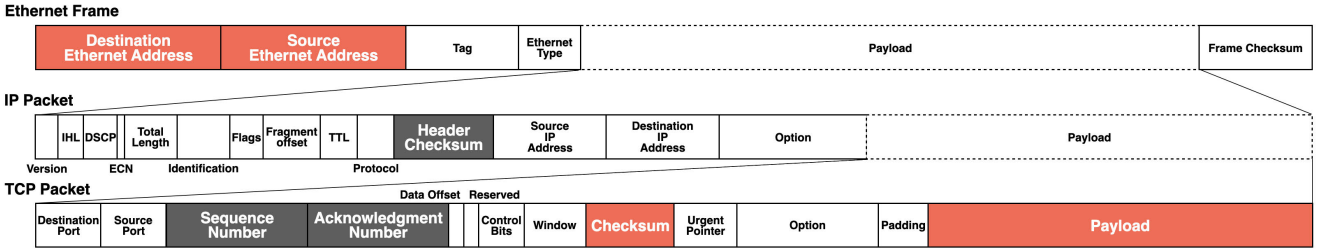


FIGURE 5. Structure of the Ethernet frame with the red sections indicating areas targeted for tampering in the experiments.

TABLE 2. List of normal and forged MAC addresses for ARP spoofing (masked by "x").

Component	IP address	MAC address	
		normal	under attack
Router	10.0.1.1	dc:a9:xx:xx:xx:a6	dc:a9:xx:xx:xx:a6
Controller	10.0.1.2	a0:78:xx:xx:xx:c2	dc:a6:xx:xx:xx:c2
Plant	10.0.1.5	4:33:xx:xx:xx:4e	dc:a6:xx:xx:xx:c2
Attacker	10.0.1.13	dc:a6:xx:xx:xx:c2	dc:a6:xx:xx:xx:c2

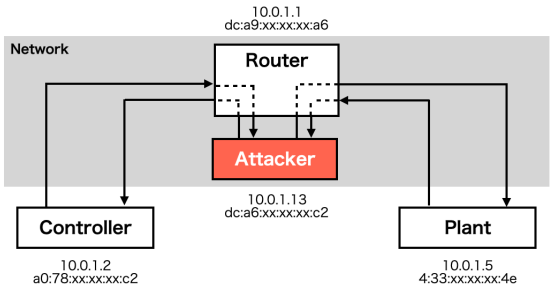


FIGURE 6. Rerouting communication paths in the networked control system using poisoned ARP cache.

B. FALSIFICATION ATTACK

In the falsification attack scenario, the attackers overwrite the communication data sent from the controller to the plant for N steps from t_a:

$$u(t) = \begin{cases} u_a(t), & t \in [t_a, t_a + N), \\ u(t), & \text{otherwise}, \end{cases} \tag{5}$$

where u_a is the overwritten signal, t_a is the step when overwriting begins, and $N > 0$ is the duration of the overwriting process. The determination of u_a refers to the following demonstration section. Furthermore, while the communication data sent from the plant to the controller is not falsified, the attackers overwrite certain packet information to ensure the packet passes through the attacker's computer. The parts of the data packet to be overwritten are indicated in the red areas in Fig. 5. If the data size changes when overwriting the data, the acknowledgment (ACK) and sequential (SEQ) numbers on the TCP packet layer in the gray areas in Fig. 5 must be overwritten to maintain communications during payload replacement. These are the total sequence numbers of the bidirectional communication data.

At the attacker's computer, packets such as Ψ or C_Ψ are overwritten at every step based on the following procedure: i) the source and destination Ethernet addresses (MAC addresses) on the Ethernet frame are overwritten; ii) the payload data on the TCP packet layer are overwritten; iii) the checksum and header checksum on the TCP and IP packet layers are recalculated and overwritten. The details of overwriting or replacing the payload data will be explained in Section V-A. Packets such as ξ or C_ξ are allowed to pass through without falsification of the payload data based on the following procedure: i) the source and destination Ethernet addresses on the Ethernet frame are overwritten; ii) the header checksum on the IP packet layer is recalculated and overwritten. Additionally, checksums are required to be deleted and recalculated before data transmission because any changes in the packet would be detected by the structural corruption or fraud-detection mechanism, and the packet would not be received correctly.

C. REPLAY ATTACK

In the replay attack scenario, the attackers record the communication data sent from the controller to the plant and replace the current data with the recorded ones. The recording starts at step t_r and ends at step $t_r + N$, where N is the duration of the recording process, and the replacement persists for N steps from t_a:

$$u(t) = \begin{cases} u(t - t_a + t_r), & t \in [t_a, t_a + N), \\ u(t), & \text{otherwise}, \end{cases} \tag{6}$$

where t_a is the step when the replacement begins. The payload size of encrypted signals varies depending on the step, as a multiple-precision integer library is used to express a ciphertext as an integer. To maintain communications during payload replacement, the attacker must overwrite the ACK and SEQ numbers on the TCP packet layer for all packets.

During the recording phase, the attacker's computer saves the payload data, such as Ψ and C_Ψ, based on the following procedure: i) the source and destination Ethernet addresses on the Ethernet frame are overwritten; ii) the payload data on the TCP packet layer are saved; iii) the header checksum on the IP packet layer is recalculated and overwritten. During the replacement phase, the attacker's computer replaces the payload data, such as Ψ and C_Ψ, with the previously saved data based on the following procedure: i) the source and destination Ethernet addresses on the Ethernet frame are overwritten; ii) the payload data are overwritten by the

recorded data; iii) the ACK and SEQ numbers on the TCP packet layer are overwritten; iv) The checksum and header checksum on the IP and TCP packet layers are recalculated and overwritten. Packets such as ξ or C_ξ are allowed to pass through without replacement based on the following procedure: i) the source and destination Ethernet addresses on the Ethernet frame are overwritten; ii) the ACK and SEQ numbers on the TCP packet layer are overwritten; iii) the checksum on the IP packet layer is recalculated and overwritten.

V. DEMONSTRATION

This section demonstrates the effectiveness of the encrypted control system in detecting MITM attacks, such as falsification and replay attacks, through experiments using our developed testbed control system. For detecting the attacks, the threshold-based method presented in [49] was employed to run on the plant side, as follows:

$$u(t) = \begin{cases} \check{u}(t), & |\check{u}(t)| < \delta, \\ 0, & |\check{u}(t)| \geq \delta, \end{cases} \tag{7}$$

where u and $\check{u}$ are actual and decoded control inputs, respectively. If $|\check{u}(t)| \geq \delta$ is satisfied, we say that a cyberattack has been detected. This study reveals that the threshold-based detector (7) theoretically results in detecting the attacks.

Theorem 1: For key-updatable encrypted control systems as described in ***Definition 1****, if a threshold δ is chosen such that $\mathscr{E}_\gamma(\delta) \ll p$, then the probability of the detector (7) detecting falsification attack (5) or replay attack (6) during N steps from step t_a is approximately one. Furthermore, if the attacker continues the attack indefinitely, i.e., $N \to \infty$, then the detector (7) can detect the attack with probability one.*

Proof: The order of the plaintext space $\mathbb{G}$ is q, i.e., $|\mathbb{G}| = q$, and the maximum component of $\mathbb{G}$ is approximately p. Let $\mathbb{G}_\delta$ be a subset of the plaintext space $\mathbb{G}$, consisting of elements up to $\mathscr{E}_\gamma(\delta)$. Because the attacker does not know the current encryption keys, the probability of the plaintext, corresponding to the injected signal, being included in $\mathbb{G}_\delta$ at a step is $|\mathbb{G}_\delta|/q$, which represents the probability of a false negative for the detector. The probability of N consecutive false negatives is given by $(|\mathbb{G}_\delta|/q)^N$. Therefore, the probability of detecting the attack during N steps is expressed as $1-(|\mathbb{G}_\delta|/q)^N =: P_{\delta,N}$. Since $\mathscr{E}_\gamma(\delta) \ll p$ implies $|\mathbb{G}_\delta| \ll q$, $|\mathbb{G}_\delta|/q$ is close to zero. Consequently, under several steps N, $(|\mathbb{G}_\delta|/q)^N$ approaches zero, so $P_{\delta,N} \approx 1$. As $N \to \infty$, it follows that $P_{\delta,N} \to 1$. ■

Corollary 1: For static-key encrypted control systems as described in ***Definition 1****, the falsification attack (5) or replay attack (6) can be stealthy for the detector (7). Furthermore, if the attacker does not know a public key and a threshold δ such that $\mathscr{E}_\gamma(\delta) \ll p$, then the probability of the detector (7) detecting the falsification attack (5) during N steps is approximately one.*

TABLE 3. Summary of attack-detection experiments using the threshold-based detector.

Attack	Encryption Scheme		
	Unencrypted	Static	Updatable
Falsification (1st scenario)	✓	✓	✓
Falsification (2nd scenario)			✓
Replay			✓

Proof: For (5), an attacker with access to the public key can prepare a specific ciphertext corresponding to a plaintext belonging to $\mathbb{G}_\delta$. For (6), the use of the recorded ciphertext results in normal operation due to the time-invariant keys. In these cases, $P_{\delta,N} = 1 - (|\mathbb{G}_\delta|/|\mathbb{G}_\delta|)^N = 0$. When the attacker cannot access the public key, the proof follows similarly to that of **Theorem 1**. ■

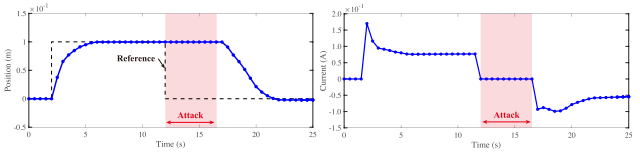
Throughout the experiments, threshold δ was set to 6 A, five times the rated current of the AC servo motor. Additionally, when considering the unencrypted control system, $\check{u}$ was replaced by u because of the lack of encryption and decryption processes. The parameters related to the encryption scheme were set to $\lambda = 64$ and $\gamma_p = \gamma_c = 10^8$, and q changed every experiment. The detection results are summarized in Table 3, which support **Theorem 1** and **Corollary 1** and will be discussed in detail in the following section.

Remark 2: The determination of threshold δ can be discussed as follows. The threshold must be set to a larger value than the upper limit of an input constraint to maintain the original control performance. Meanwhile, the change in the threshold does not impact the detection rate because q is sufficiently large. In the experimental setup, the probability introduced in the proof of ***Theorem 1*** *is computed as $|\mathbb{G}_\delta|/q = 1.3 \times 10^{-9}$, using δ, γ_p, and $q = 9223372036854777359 \approx 9.2 \times 10^{18}$. If the threshold is multiplied by 10, i.e., $\delta' = 10\delta$, then the probability is updated to $|\mathbb{G}_{\delta'}|/q = 1.3 \times 10^{-8}$, which is approximately equal to $|\mathbb{G}_\delta|/q$, i.e., $P_{\delta,N} \approx P_{\delta',N}$.*

Remark 3: Because the proofs of ***Theorem 1*** *and* ***Corollary 1*** *do not rely on the information about the controller and plant dynamics, we can generalize the considered control system in* ***Definition 1****. The presented ElGamal-based controller encryption can be applied to linear and polynomial-type controllers. Identifying a class of admissible nonlinear controllers needs more rigorous analysis, so it will be in future work.*

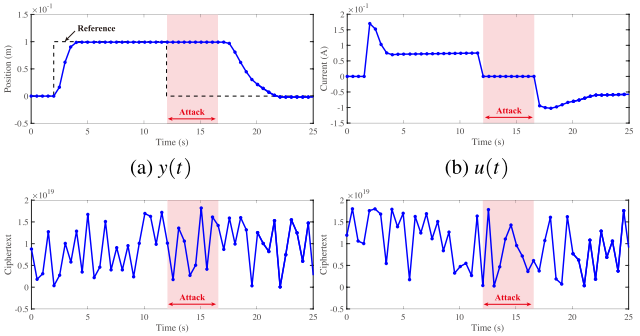
A. BEHAVIOR IN THE FALSIFICATION-ATTACK SCENARIO

This study considers two falsification attack scenarios. In the first scenario, the attacker understands the data format in the payload and the dimensions of input and output in the plant. The last two digits of the payload of packets related to control inputs between 12 s and 17 s. This implies that t_a and N are set to 24 and 10, respectively. For example, the encrypted payload data at step 15, $C^1_{\Psi_{31}}(15) =$ `179ba73be04fae1fc`, would be overwritten by `179ba73be04fae101`. In the



(a) $y(t)$ (b) $u(t)$

FIGURE 7. Time responses of the unencrypted control system in the first case of the falsification attack between 12 s and 17 s.



(a) $y(t)$ (b) $u(t)$

(c) $\mathcal{C}^1_{\Psi_{31}}(t)$ (d) $\mathcal{C}^2_{\Psi_{31}}(t)$

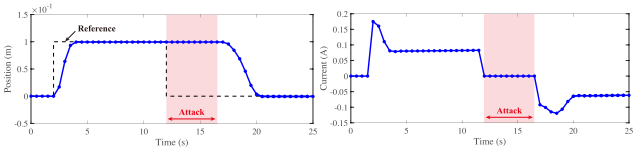
(e) $\check{u}(t)$ (f) Whole view of (e)

FIGURE 8. Experimental time responses of the static-key encrypted control system in the first case of the falsification attack between 12 s and 17 s.

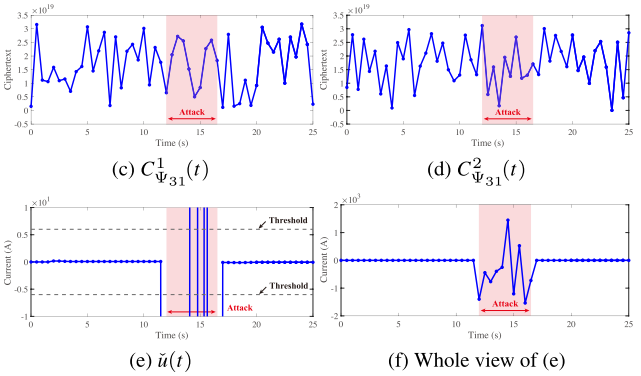
second scenario, the attacker understands the data format in the payload and the input and output dimensions in the plant. It also presumes that the attacker is aware of the employed encryption scheme and its public key at the initial step, $\mathsf{pk}(0)$. They overwrite the packets associated with control inputs with either -0.12 A or their corresponding ciphertexts (`556869e0024e6b5d`, `3c94b5f38f7b030f`).

The experimental results of the unencrypted, static-key, and key-updatable encrypted control systems with the detector (7) for the first falsification-attack scenario are shown in Figs. 7, 8, and 9, respectively. The results for the second falsification-attack scenario are illustrated in Figs. 10, 11, and 12, respectively. Subfigures (a) and (b) in these figures show the time responses of the measured stage position and the control input to the motor, respectively. Subfigures (c) and (d) in Figs. 8, 9, 11, and 12 depict the time responses of the encrypted component Ψ_{31}. Subfigures (e) and (f) in Figs. 8, 9, and 12 show the time responses of the decoded control input and a comprehensive view of (e), respectively. The red area indicates the duration of the falsification attack.

Figs. 7, 8, and 9 confirm that the control input zeros out and that the stage remains motionless during the attacks. This occurs because the decoded control input exceeds the detector's threshold, as defined in (7). Once the attacks end, the control system operation routinely resumes.



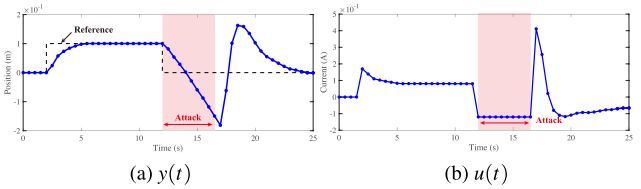
(a) $y(t)$ (b) $u(t)$



(c) $\mathcal{C}^1_{\Psi_{31}}(t)$ (d) $\mathcal{C}^2_{\Psi_{31}}(t)$

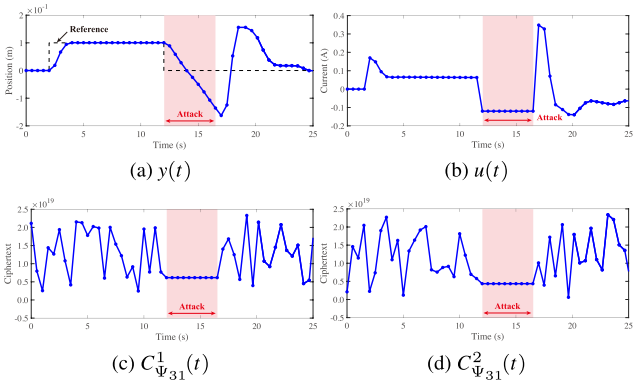
(e) $\check{u}(t)$ (f) Whole view of (e)

FIGURE 9. Time responses of the key-updatable encrypted control system in the first case of the falsification attack between 12 and 17 s.



(a) $y(t)$ (b) $u(t)$

FIGURE 10. Time responses of the unencrypted control system in the second case of the falsification attack between 12 s and 17 s.

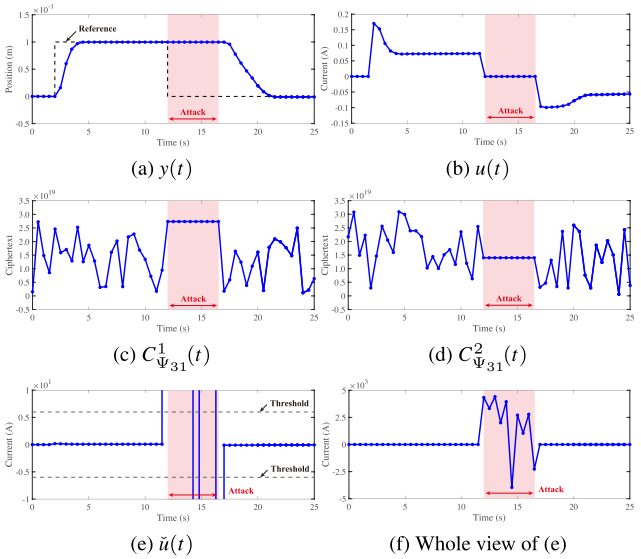


(a) $y(t)$ (b) $u(t)$

(c) $\mathcal{C}^1_{\Psi_{31}}(t)$ (d) $\mathcal{C}^2_{\Psi_{31}}(t)$

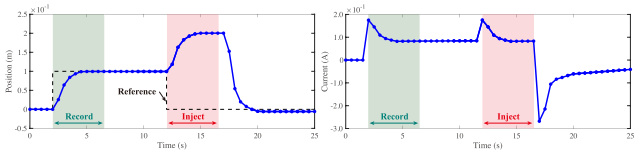
FIGURE 11. Experimental time responses of the static-key encrypted control system in the second case of the falsification attack between 12 s and 17 s. In this case, $\check{u}(t) = u(t)$.

Figs. 10 and 11 reveal that the attacker could identify the control input in the payload and replace the corresponding data with plaintext or their ciphertexts. In these cases, the detections failed because the decoded control input did not exceed the detector's threshold in Figs. 10(b) and 11(b), although the controlled outputs in Figs. 10(a) and 11(a) were affected by the attacks. Meanwhile, Fig. 12 confirms that the falsification attack could be detected. The attackers never know the updated keys and the correctness of the encryption scheme is not compromised. Thus, the decryption at the current step fails unless the attackers can identify the latest private key within the sampling period.



(a) $y(t)$ (b) $u(t)$ (c) $C^1_{\Psi_{31}}(t)$ (d) $C^2_{\Psi_{31}}(t)$ (e) $\check{u}(t)$ (f) Whole view of (e)

FIGURE 12. Time responses of the key-updatable encrypted control system in the second case of the falsification attack between 12 s and 17 s.



(a) $y(t)$ (b) $u(t)$

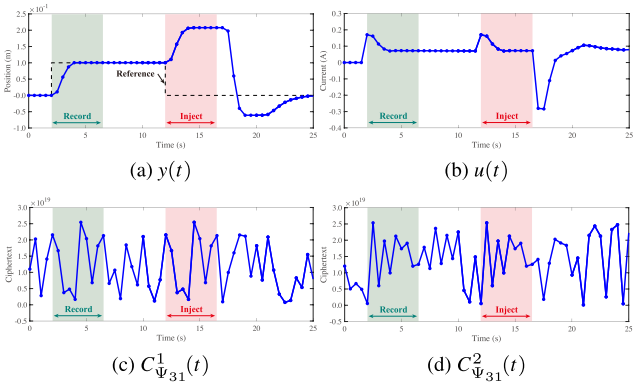
FIGURE 13. Time responses of the unencrypted control system in the case of the replay attack.

Through the demonstration, consequently, we confirmed that the key-updatable encrypted control system serves as a cybersecurity measure to detect falsification attacks as long as the attackers never know the latest public keys.

B. BEHAVIOR IN THE REPLAY-ATTACK SCENARIO

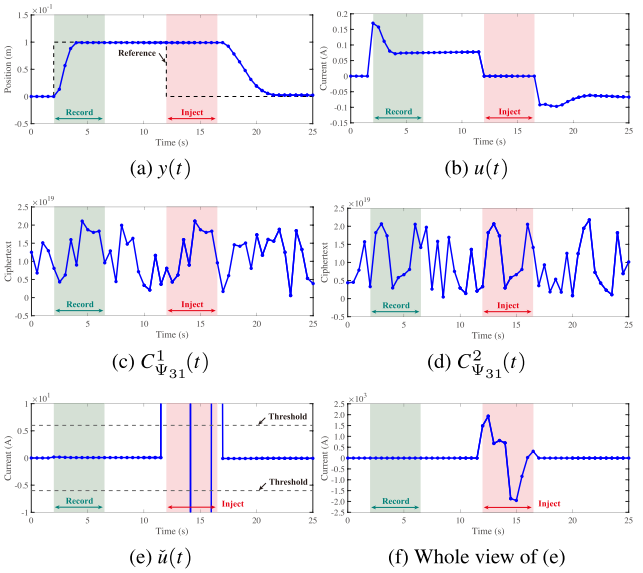
This study considers a scenario of replay attacks to validate the attack-detection capability of the key-updatable encrypted control system. The attacker does not have any information about the plant and controller and are assumed to record packets regarding control inputs between 2 s and 7 s and inject the recorded packets between 12 s and 17 s. Accordingly, t_r, t_a, and N were set to 4, 24, and 10, respectively. The resulting time responses of the unencrypted, static-key, and key-updatable encrypted control systems with the detector are shown in Figs. 13, 14, and 15, respectively. The meanings of the subfigures are the same as those in Figs. 7, 8, and 9.

Fig. 13(a) confirms that the stage position deviates significantly from the reference during the injection when the injected control input shown in Fig. 13(b) is not adequate to control the current stage position. In this case, cyberattack detection becomes difficult to achieve because the injected control input is an actual signal generated in the controller and, thus, remains within the detection threshold. Moreover, Fig. 14(a) also reveals the significant deviation in the stage



(a) $y(t)$ (b) $u(t)$ (c) $C^1_{\Psi_{31}}(t)$ (d) $C^2_{\Psi_{31}}(t)$

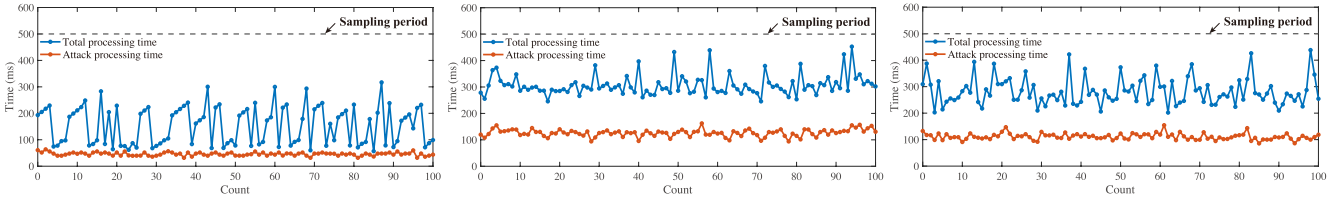
FIGURE 14. Time responses of the static-key encrypted control system for the replay attack.



(a) $y(t)$ (b) $u(t)$ (c) $C^1_{\Psi_{31}}(t)$ (d) $C^2_{\Psi_{31}}(t)$ (e) $\check{u}(t)$ (f) Whole view of (e)

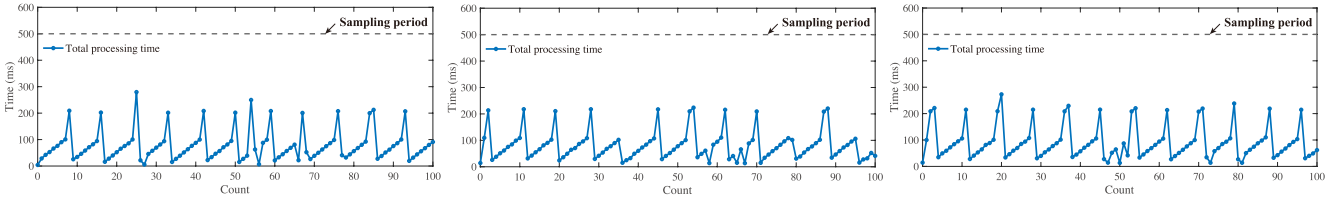
FIGURE 15. Time responses of the key-updatable encrypted control system for the replay attack.

position from the reference. As shown in Fig. 14(b), the resulting injected actual control input was the same as the recorded one, while the two signals $C^1_{\Psi_{31}}(t)$ and $C^2_{\Psi_{31}}(t)$ in Figs. 14(c) and 14(d), respectively, were encrypted. This is because the keys during recording and injecting are the same, which renders the effects of the replay attack invisible in the decoded signal, complicating the detection process. Meanwhile, Fig. 15(a) confirms that the stage remains stationary while injecting the recorded signals. In this case, the detector sets the control input to zero, as shown in Fig. 15(b), because the decoded control input exceeds the threshold, as shown in Figs. 15(e) and 15(f). This occurs because the key is updated at every sampling period, and the recorded keys differ from the ones relevant to the current control inputs, which causes the decryption to fail and enables replay-attack detection. Consequently, through the experimental demonstration, we conclude that the key-updatable encrypted control system more effectively serves as a cybersecurity countermeasure than unencrypted and static-key encrypted control systems.



(a) Unencrypted control system (b) Static-key encrypted control system (c) Key-updatable encrypted control system

FIGURE 16. Processing times in the replay-attack scenario.



(a) Unencrypted control system (b) Static-key encrypted control system (c) Key-updatable encrypted control system

FIGURE 17. Processing times in the no-attack scenario.

VI. DISCUSSION

This section examines the total processing times for each step to confirm the real-time capabilities of the control systems and discuss the potential of detecting attacks from the perspective of computational cost. The processing times for the unencrypted, static-key, and key-updatable encrypted control systems are listed in Table 4. The processing time spans the period from when the sensor output is measured to when the control command is sent to the servo amplifier, including the encryption, decryption, control computation, and transmission delays. In an attack scenario, the processing time also includes delays in communicating with the router from the attacker's PC and overwriting the data. Table 4 displays the minimum, maximum, and average processing times across 100 steps for each attack scenario and encryption scheme used. The far-right column presents the ratio of the computation time in an attack relative to that in a no-attack scenario. For example, the computation times of the three control systems in the replay attack and no-attack scenarios are plotted in Figs. 16 and 17, respectively. The blue and red lines represent the total processing time and the processing time for overwriting the data from the attacker's PC, respectively.

As can be observed in Table 4, all values are less than the 500 ms sampling period, which confirms the real-time nature of the control systems. Moreover, the processing time in each control system increases because of the attack. The attack-induced time increases in the encrypted control systems tend to be greater than those in the unencrypted control system. For example, when considering the replay attack scenario, the time increases in the static-key and key-updatable encrypted control systems are 380.6 % and 294.6 %, respectively, whereas the increase in the unencrypted control system is 198.4 %. The tendency is illustrated by comparing Figs. 16(b)(c) with Fig. 16(a), where Fig. 17 shows the similar computation load among the three control systems. The reasons for the time increases are as follows. One is that the reconstruction of the ciphertext data packet to be sent and received is time-consuming compared with the millisecond-order sampling period. The other is that the attacker has to falsify the packets two times to maintain continuous communication in a TCP protocol, which means that the ACK/SEQ numbers in the packet are overwritten in both directions of communication.

TABLE 4. Computation time of the encrypted controls with a sampling period of 500 ms under cyberattacks.

Attack	Key Scheme	Computation Time (ms)			
		min	max	average	(%)
Falsification (1st scenario)	Static	256.8	450.7	318.4	392.0
	Updatable	202.7	453.5	272.9	297.7
	Unencrypted	53.94	392.2	155.8	198.4
Falsification (2nd scenario)	Static	281.2	468.8	335.0	437.0
	Updatable	214.3	435.7	283.3	309.0
	Unencrypted	45.34	340.2	155.9	198.5
Replay	Static	243.1	451.4	309.1	380.6
	Updatable	191.0	404.0	270.1	294.6
	Unencrypted	59.0	388.7	155.8	198.4
N/A	Static	13.1	269.4	81.2	100
	Updatable	7.0	352.8	91.7	100
	Unencrypted	5.0	313.7	78.5	100

Consequently, the attack tests demonstrate that the use of an encrypted control system enhances cybersecurity since more processing time is required for the attackers to successfully implement MITM attacks. This knowledge would provide insights for the development of another attack-detection method involving the monitoring of any significant changes in processing time during operation. However, for this detection approach, control system designs that consider the occurrence of time-varying communication delays and losses must be explored. This area will be addressed in future works.

VII. CONCLUSION

This study experimentally demonstrated the effectiveness of key-updatable encrypted control systems against MITM attacks, such as falsification and replay attacks, using the industrial-grade linear stage. Encrypted and unencrypted control experimental environments were established, in which control and sensor signals were wirelessly communicated between the plant and the controller PCs. The attack processes, executed on an attacker's PC connected to the wireless router, could overwrite the communicated packets in real time using the ARP-spoofing technique. Furthermore, this study revealed the theoretical and practical features that the threshold-based detector monitoring the control input enables attack detection, and it confirmed that the experimental results support the features. Therefore, this study concludes that the key-updatable encrypted control systems outperform the static-key encrypted and unencrypted control systems in cyberattack detection. Additionally, this study offered guidance of how to determine the detector's threshold in Remark 2.

In future works, the authors will explore timestamp-based attack-detection methods to enhance cybersecurity against several cyberattacks, and they will develop a cybersecure industrial-use control technology enhancing real-time attack detection based on communication protocols such as UDP and EtherCAT, in the developed environment to ascertain the effectiveness of the encrypted control systems.

REFERENCES

[1] X.-M. Zhang, Q.-L. Han, X. Ge, D. Ding, L. Ding, D. Yue, and C. Peng, "Networked control systems: A survey of trends and techniques," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 1, pp. 1–17, Jan. 2020.

[2] J. Cui, Y. Liu, and A. Nallanathan, "Multi-agent reinforcement learning-based resource allocation for UAV networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 729–743, Feb. 2020.

[3] H. Laaki, Y. Miche, and K. Tammi, "Prototyping a digital twin for real time remote control over mobile networks: Application of remote surgery," *IEEE Access*, vol. 7, pp. 20325–20336, 2019.

[4] Y. Zhang, V. V. G. Krishnan, J. Pi, K. Kaur, A. Srivastava, A. Hahn, and S. Suresh, "Cyber physical security analytics for transactive energy systems," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 931–941, Mar. 2020.

[5] R. Ma, P. Cheng, Z. Zhang, W. Liu, Q. Wang, and Q. Wei, "Stealthy attack against redundant controller architecture of industrial cyber-physical system," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9783–9793, Dec. 2019.

[6] H. Pearce, S. Pinisetty, P. S. Roop, M. M. Y. Kuo, and A. Ukil, "Smart I/O modules for mitigating cyber-physical attacks on industrial control systems," *IEEE Trans. Ind. Informat.*, vol. 16, no. 7, pp. 4659–4669, Jul. 2020.

[7] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Secur. Privacy*, vol. 9, no. 3, pp. 49–51, May 2011.

[8] R. Langner, "To kill a centrifuge: A technical analysis of what Stuxnet's creators tried to achieve," Langner Group, Hamburg, Germany, Tech. Rep., 2013.

[9] A. Bindra, "Securing the power grid: Protecting smart grids and connected power systems from cyberattacks," *IEEE Power Electron. Mag.*, vol. 4, no. 3, pp. 20–27, Sep. 2017.

[10] R. Setola, L. Faramondi, E. Salzano, and V. Cozzani, "An overview of cyber attack to industrial control system," *Chem. Eng. Trans.*, vol. 77, pp. 907–912, Jan. 2019.

[11] J. Slay and M. Miller, "Lessons learned from the Maroochy water breach," in *Critical Infrastructure Protection*, E. Goetz and S. Shenoi, Eds. Cham, Switzerland: Springer, 2008, pp. 73–82.

[12] Y. Zhao and F. Zhu, "Security control of cyber-physical systems under denial-of-service sensor attack: A switching approach," in *Proc. IEEE 10th Data Driven Control Learn. Syst. Conf. (DDCLS)*, May 2021, pp. 1112–1117.

[13] A.-Y. Lu and G.-H. Yang, "Resilient observer-based control for cyber-physical systems with multiple transmission channels under denial-of-service," *IEEE Trans. Cybern.*, vol. 50, no. 11, pp. 4796–4807, Nov. 2020.

[14] W. Yu, R. Wang, X. Bu, Z. Hou, and Z. Wu, "Resilient model-free adaptive iterative learning control for nonlinear systems under periodic DoS attacks via a fading channel," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 7, pp. 4117–4128, Jul. 2022.

[15] K. Manandhar, X. Cao, F. Hu, and Y. Liu, "Detection of faults and attacks including false data injection attack in smart grid using Kalman filter," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 4, pp. 370–379, Dec. 2014.

[16] Y. Guan and X. Ge, "Distributed attack detection and secure estimation of networked cyber-physical systems against false data injection attacks and jamming attacks," *IEEE Trans. Signal Inf. Process. over Netw.*, vol. 4, no. 1, pp. 48–59, Mar. 2018.

[17] X.-M. Li, Q. Zhou, P. Li, H. Li, and R. Lu, "Event-triggered consensus control for multi-agent systems against false data-injection attacks," *IEEE Trans. Cybern.*, vol. 50, no. 5, pp. 1856–1866, May 2020.

[18] W. Qi, Y. Hou, G. Zong, and C. K. Ahn, "Finite-time event-triggered control for semi-Markovian switching cyber-physical systems with FDI attacks and applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 6, pp. 2665–2674, Jun. 2021.

[19] J. Lee, J. Kim, and H. Shim, "Zero-dynamics attack on homomorphically encrypted control system," in *Proc. 20th Int. Conf. Control, Autom. Syst. (ICCAS)*, Oct. 2020, pp. 385–390.

[20] D. Kim, K. Ryu, J. H. Kim, and J. Back, "Zero assignment via generalized sampler: A countermeasure against zero-dynamics attack," *IEEE Access*, vol. 9, pp. 109932–109942, 2021.

[21] J. Kim and H. Shim, "A countermeasure against zero-dynamics sensor attack via generalized hold feedback," in *Proc. 58th Annu. Conf. Soc. Instrum. Control Engineers Jpn. (SICE)*, Sep. 2019, pp. 663–668.

[22] A. Baniamerian, K. Khorasani, and N. Meskin, "Monitoring and detection of malicious adversarial zero dynamics attacks in cyber-physical systems," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Aug. 2020, pp. 726–731.

[23] L. Ma, Z. Chu, C. Yang, G. Wang, and W. Dai, "Recursive watermarking-based transient covert attack detection for the industrial CPS," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 1709–1719, 2023.

[24] F. Tedesco, D. Famularo, and G. Franzè, "Leader–follower multi-agent systems: A model predictive control scheme against covert attacks," in *Proc. IEEE Int. Conf. Auto. Syst. (ICAS)*, Aug. 2021, pp. 1–5.

[25] P. Ramanan, D. Li, and N. Gebraeel, "Blockchain-based decentralized replay attack detection for large-scale power systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 8, pp. 4727–4739, Aug. 2022.

[26] H. Guo, Z.-H. Pang, J. Sun, and J. Li, "An output-coding-based detection scheme against replay attacks in cyber-physical systems," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 68, no. 10, pp. 3306–3310, Oct. 2021.

[27] G. Franzè, F. Tedesco, and D. Famularo, "Resilience against replay attacks: A distributed model predictive control scheme for networked multi-agent systems," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 3, pp. 628–640, Mar. 2021.

[28] T. Irita and T. Namerikawa, "Detection of replay attack on smart grid with code signal and bargaining game," in *Proc. Amer. Control Conf. (ACC)*, May 2017, pp. 2112–2117.

[29] A. Hoehn and P. Zhang, "Detection of replay attacks in cyber-physical systems," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2016, pp. 290–295.

[30] K. Kogiso and T. Fujita, "Cyber-security enhancement of networked control systems using homomorphic encryption," in *Proc. 54th IEEE Conf. Decis. Control (CDC)*, Dec. 2015, pp. 6836–6843.

[31] J. Kim, C. Lee, H. Shim, J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Encrypting controller using fully homomorphic encryption for security of cyber-physical systems," *IFAC-PapersOnLine*, vol. 49, no. 22, pp. 175–180, 2016.

[32] F. Farokhi, I. Shames, and N. Batterham, "Secure and private control using semi-homomorphic encryption," *Control Eng. Pract.*, vol. 67, pp. 13–20, Oct. 2017.

[33] M. Schulze Darup, A. B. Alexandru, D. E. Quevedo, and G. J. Pappas, "Encrypted control for networked systems: An illustrative introduction and current challenges," *IEEE Control Syst. Mag.*, vol. 41, no. 3, pp. 58–78, Jun. 2021.

[34] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 26, no. 1, pp. 96–99, Jan. 1983.
[35] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inf. Theory*, vol. IT-31, no. 4, pp. 469–472, Jul. 1985.
[36] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology—EUROCRYPT 1999*, Lecture Notes in Computer Science, J. Stern, Ed. Berlin, Germany: Springer, 1999, pp. 223–238.
[37] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Proc. 23rd Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, in Lecture Notes in Computer Science, 2017, pp. 409–437.
[38] C. Murguia, F. Farokhi, and I. Shames, "Secure and private implementation of dynamic controllers using semihomomorphic encryption," *IEEE Trans. Autom. Control*, vol. 65, no. 9, pp. 3950–3957, Sep. 2020.
[39] N. Schlüter, M. Neuhaus, and M. S. Darup, "Encrypted dynamic control with unlimited operating time via FIR filters," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2021, pp. 952–957.
[40] A. B. Alexandru, M. Morari, and G. J. Pappas, "Cloud-based MPC with encrypted data," in *Proc. IEEE Conf. Decis. Control (CDC)*, Dec. 2018, pp. 5014–5019.
[41] M. S. Darup, "Encrypted MPC based on ADMM real-time iterations," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 3508–3514, 2020.
[42] A. B. Alexandru, M. Schulze Darup, and G. J. Pappas, "Encrypted cooperative control revisited," in *Proc. IEEE 58th Conf. Decis. Control (CDC)*, Dec. 2019, pp. 7196–7202.
[43] M. Ruan, H. Gao, and Y. Wang, "Secure and privacy-preserving consensus," *IEEE Trans. Autom. Control*, vol. 64, no. 10, pp. 4035–4049, Oct. 2019.
[44] M. Kishida, "Encrypted average consensus with quantized control law," in *Proc. IEEE Conf. Decis. Control (CDC)*, Dec. 2018, pp. 5850–5856.
[45] C. N. Hadjicostis and A. D. Domínguez-García, "Privacy-preserving distributed averaging via homomorphically encrypted ratio consensus," *IEEE Trans. Autom. Control*, vol. 65, no. 9, pp. 3887–3894, Sep. 2020.
[46] M. S. Darup, "Encrypted polynomial control based on tailored two-party computation," *Int. J. Robust Nonlinear Control*, vol. 30, pp. 4165–4448, Jan. 2020.
[47] K. Teranishi, K. Kogiso, and J. Ueda, "Encrypted feedback linearization and motion control for manipulator with somewhat homomorphic encryption," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jul. 2020, pp. 613–618.
[48] K. Kogiso, "Attack detection and prevention for encrypted control systems by application of switching-key management," in *Proc. IEEE Conf. Decis. Control (CDC)*, Dec. 2018, pp. 5032–5037.
[49] R. Baba, K. Kogiso, and M. Kishida, "Detection method of controller falsification attacks against encrypted control system," in *Proc. SICE Annu. Conf.*, 2018, pp. 244–248.
[50] M. Miyamoto, K. Teranishi, K. Emura, and K. Kogiso, "Cybersecurity-enhanced encrypted control system using keyed-homomorphic public key encryption," *IEEE Access*, vol. 11, pp. 45749–45760, 2023.
[51] X. Li, M. Liu, R. Zhang, P. Cheng, and J. Chen, "Demo abstract: An industrial control system testbed for the encrypted controller," in *Proc. ACM/IEEE 9th Int. Conf. Cyber-Phys. Syst. (ICCPS)*, Apr. 2018, pp. 343–344.
[52] K. Teranishi, T. Sadamoto, A. Chakrabortty, and K. Kogiso, "Designing optimal key lengths and control laws for encrypted control systems based on sample identifying complexity and deciphering time," *IEEE Trans. Autom. Control*, vol. 68, no. 4, pp. 2183–2198, Apr. 2023.
[53] K. Teranishi, N. Shimada, and K. Kogiso, "Stability-guaranteed dynamic ElGamal cryptosystem for encrypted control systems," *IET Control Theory Appl.*, vol. 14, no. 16, pp. 2242–2252, Nov. 2020.
[54] J. H. Cheon, K. Han, H. Kim, J. Kim, and H. Shim, "Need for controllers having integer coefficients in homomorphically encrypted dynamic system," in *Proc. IEEE Conf. Decis. Control (CDC)*, Dec. 2018, pp. 5020–5025.
[55] M. S. Tavazoei, "Pisot number-based discrete-time controllers with integer state matrices to ensure monotonic closed-loop step responses," *IEEE Trans. Autom. Control*, vol. 68, no. 12, pp. 8238–8243, Dec. 2023.
[56] M. S. Tavazoei, "Sufficient conditions for stabilizability by discrete-time controllers possessing monic characteristic polynomials with integer coefficients," *IEEE Control Syst. Lett.*, vol. 7, pp. 3337–3342, 2023.
[57] N. Schlüter and M. S. Darup, "On the stability of linear dynamic controllers with integer coefficients," *IEEE Trans. Autom. Control*, vol. 67, no. 10, pp. 5610–5613, Oct. 2022.

AKANE KOSUGI (Graduate Student Member, IEEE) received the B.E. degree in mechanical and intelligent systems engineering from The University of Electro-Communications, Tokyo, Japan, in 2022. She is currently a Graduate Student with The University of Electro-Communications. Her research interest includes the cybersecurity of control systems.

KAORU TERANISHI (Graduate Student Member, IEEE) received the B.S. degree in electromechanical engineering from the Ishikawa College, National Institute of Technology, Ishikawa, Japan, in 2019, and the M.S. degree in mechanical and intelligent systems engineering from The University of Electro-Communications, Tokyo, Japan, in 2021, where he is currently pursuing the Ph.D. degree. From October 2019 to September 2020, he was a Visiting Scholar with the Georgia Institute of Technology, GA, USA. Since April 2021, he has been a Research Fellow of the Japan Society for the Promotion of Science. His research interests include control theory and cryptography for the cybersecurity of control systems.

KIMINAO KOGISO (Member, IEEE) received the B.E., M.E., and Ph.D. degrees in mechanical engineering from Osaka University, Japan, in 1999, 2001, and 2004, respectively. He was appointed as a Postdoctoral Fellow with the 21st Century COE Program and an Assistant Professor with the Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan, in April 2004 and July 2005, respectively. From November 2010 to December 2011, he was a Visiting Scholar with the Georgia Institute of Technology, Atlanta, GA, USA. In March 2014, he was promoted to the position of an Associate Professor with the Department of Mechanical and Intelligent Systems Engineering, The University of Electro-Communications, Tokyo, Japan, where he has been a Professor, since April 2023. His research interests include cybersecurity of control systems, constrained control, control of decision-makers, and their applications.