Confidentiality Attacks against Encrypted Control Systems

Amir Mohammad Naseri, Walter Lucia and Amr Youssef

Concordia Institute for Information Systems Engineering (CIISE), Concordia University, Montreal, QC, H3G-1M8, Canada.

ABSTRACT
Encrypted control systems were introduced to enhance the security of cyber-physical systems which outsource their computation of the control actions to a third-party platform. To protect the confidentiality of the transmitted data (i.e., sensor measurements and control inputs), homomorphic encryption schemes are particularly appealing, given their capability of allowing computation of the control inputs directly on the encrypted measurement data. This paper shows that encrypted control systems based on homomorphic encryptions are vulnerable to attackers leveraging the inherently small domains of the plaintext data in control systems and the randomization process required to make the utilized ciphers semantically secure. In particular, by considering the popular ElGamal and Paillier encryption schemes, we investigate different attacks that enable a malware, which compromises the random number generator used by the randomized encryption schemes, to covertly leak the private decryption key and/or the measurements to an eavesdropper who has access to the measurement channel. Finally, we present some countermeasures to defend against these attacks.

KEYWORDS
Encrypted Control Systems, Cyber-Physical Systems Security, Homomorphic Encryption.

1. Introduction

Cyber-Physical Systems (CPSs) are anticipated to have a rapid diffusion in safety critical domains such as intelligent transportation, energy distribution, and industry 4.0. Therefore, their security against cyber-attacks is a primary concern. Since CPSs can be often modeled as networked control systems (NCSs), their security has been intensively investigated from the control theory community, and different control-theoretic inspired solutions have been developed to detect/mitigate a variety of cyber-attacks against the integrity and confidentiality of such systems [1–7]. Recently, a new field of research, namely "encrypted-control" has emerged as a new paradigm to ensure the confidentiality of CPSs when the controller's operations are performed by a third-party platform (e.g., cloud or edge node), see, e.g., [8] for an introduction to the field of encrypted control.

Cloud services can provide a high computation power typically not available on site. Moreover, cloud services can significantly decrease maintenance costs while providing more availability for geographically distributed systems. In the literature, the idea of outsourcing the control computation to a cloud is believed to be beneficial to the

Contact Author: Walter Lucia. Email: walter.lucia@concordia.ca

development of the concept of control-as-a-service. As discussed in [8,9], the use of a cloud-based controller is expected to improve the economic efficiency of industrial control systems as well as their security and safety in many application areas such as smart grid, building automation, robot swarm and intelligent transportation systems. On the other hand, some concerns raise about the privacy of the outsourced data if the data are not encrypted or decryption is required on the cloud. To solve this problem, the authors of [10] proposed the idea of using Homomorphic Encrypted (HE) [11] to perform arithmetic operations on the cloud directly on encrypted data. In particular, by leveraging a multiplicatively homomorphic scheme (which allows multiplications between two encrypted variables) based on the RSA and ElGamal algorithms [12], implemented a linear state-feedback controller. Since then, different homomorphic encryption schemes supporting a different set and number of mathematical operations have been explored to improve the performance of encrypted control systems, e.g., by utilizing the additively homomorphic encryption scheme (allowing only encrypted additions), based on the Paillier encryption [13], developed in [14] or fully homomorphic encryption (allowing both encrypted additions and multiplications) [15].

Ensuring the confidentiality of processed data at the cloud via encrypted control systems introduces computation and communication overheads with respect to traditional non-encrypted networked control systems. However, the extra computational load is mainly related to the operations performed inside the cloud, which, in such architectures, is assumed to have high computation capabilities. Therefore, in the presence of high-performance cloud and communication infrastructures, the resulting transmission and execution delays introduced by encrypted control systems can be minimized in order to satisfy the delay constrains ts of the underlying control system [16]. An interesting study about the efficiency of four different homomorphic encryption schemes can be found in [17]. Moreover, experimental engineering studies about the feasibility of homomorphically encrypted control systems can be found in, e.g., [18,19] where the authors have proved the feasibility of such architecture to control a DC motor [18] and an inverse pendulum [19].

Although encrypted control systems can, in principle, solve the security and privacy problems of NCSs, different deception attacks against these control architectures have been proposed in [20–23]. In [20], by exploiting the encrypted control system sensitivity to signal and parameter falsifications, an attack detector based on a low-pass filter is proposed to detect falsified control signals and parameters. In [21], stealthy replay attacks are investigated, and a switching private/public keys management system is proposed to prevent and detect such attacks. In [22], first, the authors show that any encrypted control system based on homomorphic encryption can be subject to attacks exploiting the inherent malleability of the encryption scheme (i.e., the attacker can manipulate encrypted data without the need to decrypt them). In particular, if the adversary is aware of the used homomorphic scheme, then it can change sensor measurements, control parameters and even re-assign the poles of the closed-loop system. Then, the authors propose a QR decomposition technique to prevent malleability-based pole-assignment attacks. In [23], it is shown that if the encrypted controller is implemented resorting to an additively homomorphic encryption system, then it is possible to exploit the malleability property to launch undetectable zero-dynamics attacks.

1.1. Paper's Contribution

While the main objective of encrypted control systems is to improve the confidentiality of such systems, to the best of the authors' knowledge, no attacks against the confidentiality of encrypted control systems (e.g., sensor measurements, control inputs, controller parameters) have been reported in the literature. However, some of the inherent characteristics of control systems and encryption schemes can provide an opportunity to adversaries to attack the confidentiality of encrypted control systems. In this paper, we show that exploiting the small domain of the plaintext data (e.g., sensor measurements) and the randomization process used by semantically secure encryption schemes, a malware located on the plant side of the NCS is able to covertly leak sensitive information to an eavesdropper located on the measurement channel. Such information can be plaintext sensor measurements, secret encryption keys or any other confidential data, illegitimately obtained by the malware about the operations of the control system. In simpler words, we demonstrate that the attacker is able to establish an illegitimate communication channel, also known in the literature as a covert-channel [24,25]. The existence of such covert channels is a relevant security concern as specified by the "Orange Book" [26] of the U.S. Department of Defence. The feasibility of our proposed attacks is illustrated by considering three different attack scenarios against the popular ElGamal and Pallier encryption schemes. Then, a countermeasure capable of preventing such disclosure attacks is proposed.

1.2. Notation

We denote with $\mathbb{R}$, $\mathbb{Z}$ and $\mathbb{Z}_+$ the sets of real, integer and non-negative integer numbers, respectively. $\mathbb{Z}_n := \{0, \dots, n-1\}$ defines the complete residue system modulo $n \in \mathbb{Z}_+$, while $\mathbb{Z}_n^\times$ is the reduced system modulo n obtained from $\mathbb{Z}_n$ by removing all integers not relatively prime to n. The set of real-valued $n_r \times n_c$ matrices is denoted by $\mathbb{R}^{n_r \times n_c}$, while the real-valued $n_r \times 1$ column vector is denoted with $\mathbb{R}^{n_r}$. Moreover, given a matrix $M \in \mathbb{R}^{n_r \times n_c}$ and a vector $v \in \mathbb{R}^{n_r}$, M_{ij} denotes the (i,j) entry of M, while v_i denotes the $i-th$ element of v. Given a plaintext message m, $Enc[m]$ defines the corresponding ciphertext (encrypted) message according to a given encryption algorithm. Moreover, the decryption operator, namely $Dec[\cdot]$, is such that $Dec[Enc[m]] = m$. The sets of all possible plaintext (m) and ciphertexts ($Enc[m]$) messages are denoted with $\mathcal{M}$ and $\mathcal{C}$, respectively. Given two positive integer numbers $v_1, v_2 \in \mathbb{Z}_+$, then $gcd(v_1, v_2)$ and $lcm(v_1, v_2)$ and $v_1 \mod v_2$ denote the largest common positive integer divisor, the smallest positive integer common multiple and the remainder of the Euclidean division, respectively. Given an integer $m \in \mathbb{Z}_+$, the functions $|m|$ denotes the length of the binary string representing m. Given a variable v, $v(t)$ denotes the $t-$th, $t \in \mathbb{Z}_+$, sample of v obtained by sampling v with a constant sampling time $T_s > 0$. Given a set $\mathcal{S}$, $|\mathcal{S}|$ denotes the number of elements in $\mathcal{S}$. Let $r \in \mathbb{Z}_+$ be a integer number generated by a random number generator (RG), then the set of all possible values of r is denoted with $\mathcal{R}_{rg} \subset \mathbb{Z}_+$.

This paper is organized as follows. In section 2, we first provide a brief background on encrypted control systems and HE, then the considered threat model and the problem of interest are presented. In section 3, the existence of three different attack scenarios against the confidentiality of encrypted control systems (equipped with either El-Gamal or Paillier cryptosystems) is proved. A re-randomization solution capable of preventing the existence of confidentiality attacks is proposed and proved in section 4. In section 5, a numerical example is shown as a proof of concept to verify the

effectiveness of the considered attacks and countermeasure. Finally, our conclusion is presented in section 6.

2. Preliminaries and Problem Formulation

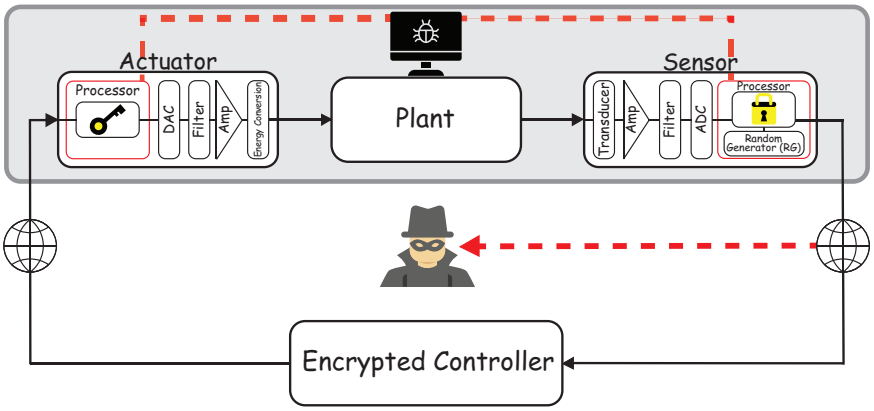

Figure 1.: Encrypted control system using HE.

2.1. Encrypted Sampled-Data Networked Control System

Consider the encrypted NCS architecture shown in Fig. 1. In such a scheme, the plant is regulated by a networked controller implemented on a third party platform (e.g., cloud), and accessible through a communication channel. To guarantee the confidentiality of the closed-loop control system (e.g., sensor measurements $y(t) \in \mathbb{R}^{n_p}$, $n_p \geq 1$, and control inputs $u(t) \in \mathbb{R}^{n_m}$, $n_m \geq 1$), the control-loop operates as follows. The sensor measurements $y(t)$ are encrypted into $Enc[y(t)]$ and then transmitted to the controller. The controller executes its logic directly on the received encrypted measurements $Enc[y(t)]$, producing in output the encrypted control input vector $Enc[u(t)]$ (see the subsection 2.3 for more details) which is then sent to the actuators. A decryptor module, local to the actuator, recovers the plaintext control vector $u(t) = Dec[Enc[u(t)]]$, so allowing the actuator (by means of a digital-to-analog converter (DAC)) to apply $u(t)$ to the plant. Note that the "sensor" and "actuator" boxes in Fig. 1 are assumed to contain the operations performed by the sensor and actuator as well as the operations needed to support the used cryptosystem. For instance, since the encryption algorithms work on integer numbers, we assume that the sensor and actuator processor units are able to implement the required mapping function from fixed point numbers to integers and vice-versa, see e.g., [8].

In what follows, for simplicity, we assume that the encryption operations of the $i-th$ component of $y(t)$, namely $Enc[y_i(t)]$, are performed on an implicit integer representation of $y_i(k)$. The fact that storing the encryption key on the controller side can endanger privacy of the system (against either internal cloud adversary or external adversaries who may compromise the cloud service) implies that the encryption scheme used in Fig. 1 is not arbitrary, but it must belong to a class of homomorphic encryption schemes supporting an suitable set and number of mathematical operations on the encrypted data [13,27,28] (e.g., to implement the control logic). Hereafter, two popular

schemes (used to implement encrypted control), namely "ElGamal" and "Paillier", will be considered. See Section 2.2 for a brief overview.

The control architecture in Fig. 1 defines a sampled-data control system where the signals transmitted over the network are digital while the ones applied to the plant are analog. This aspect is particularly relevant in an encrypted setups because it provides prior information on the maximum number of bits used to present/transmit digital data over the network. Also, all homomorphic encryption (HE) schemes proposed for use in cloud-based control systems work on integer messages space as required by the used modulo arithmetic. Hence, to encrypt the $i-th$ component of the sensor measurement $y_i(t) \in \mathbb{R}^{n_p}$, $y_i(t)$ needs to be mapped onto the integer message space $\mathcal{M}$. The first step of the mapping is usually an element-wise approximation of the real-valued measurements with fixed-point numbers from the set $\mathbb{Q}_{\beta,\gamma,\delta} = \{-\beta^\gamma, -\beta^\gamma + \beta^{-\delta}, \ldots, \beta^\gamma - 2\beta^{-\delta}, \beta^\gamma - \beta^{-\delta}\}$, where $\beta \geq 1 \in \mathbb{N}$ is the basis and γ and $\delta \in \mathbb{N}$ are known as the magnitude and the resolution of the set, respectively. Such operations is described by the following mapping function:

$$g : \mathbb{R} \to \mathbb{Q}_{\beta,\gamma,\delta}$$

$$|g(y_i(t)) - y_i(t)| \leq \beta^{-\delta} \; \forall \; y_i(t) \in [-\beta^\gamma, \beta^\gamma]$$

Typically, the analog signals produced by the sensors are sampled and quantized into fixed-point numbers by the analog-to-digital converter (ADC) using, for each sensor measurement y_i, $i = 1, \ldots, n_p$ of $y(t)$, a finite small number of binary digits d. This procedure can be performed inside the sensor, by its processing unit. Given the fixed-point approximation $g(y_i(t))$ of $y_i(t)$, the next step prescribes a suitable mapping from $\mathbb{Q}_{\beta,\gamma,\delta}$ to the integer message space $\mathcal{M}$. This operation can be easily done by scaling $\mathbb{Q}_{\beta,\gamma,\delta}$ with the factor of β^δ in modulo φ where φ is a user-defined parameters for the used cryptosystem [8], i.e.,

$$(\beta^\delta \mathbb{Q}_{\beta,\gamma,\delta} \mod \varphi) \in \mathcal{M}$$

The processing unit of the sensor typically performs the above operation.

Remark 1. As described in [29, Ch. 5, Sec. 5.2.5], d is typically between 12 and 16. As a consequence, the size of the plaintext message's space $\mathcal{M}$, namely $|\mathcal{M}| = 2^d$ is relatively small. Besides, another limitation to the message space is imposed by the desire to compute the control logic directly on the encrypted variables, particularly in dynamic controllers [30]. As explained in, e.g., [31,32], according to the kind and number of mathematical operations required to compute the control action, the size of the plaintext variables should be sufficiently small to avoid overflow with the used modulo space. □

2.2. El-Gamal and Paillier Homomorphic Cryptosystems

In this subsection, some definitions used to describe the properties of homomorphic encryption schemes are given, and the relevant details of El-Gamal and Paillier cryptosystems are briefly reviewed. Then, we show how a simple static feedback controller can be implemented in an encrypted control system.

Definition 1. An encryption scheme is said homomorphic if it allows some compu-

tations on the encrypted data without access to the secret encryption key (i.e., there exists an homomorphism between the plaintext $\mathcal{M}$ and ciphertext $\mathcal{C}$ spaces) [33].

Definition 2. A cryptosystem is called multiplicatively homomorphic if $\forall\, m_1, m_2 \in \mathcal{M}$

$$m_1 m_2 = Dec[Enc[m_1] \otimes Enc[m_2]] \tag{1}$$

where $\otimes$ denotes the multiplicative operator between two encrypted variables. □

Definition 3. A cryptosystem is called additively homomorphic if $\forall\, m_1, m_2 \in \mathcal{M}$

$$m_1 + m_2 = Dec[Enc[m_1] \oplus Enc[m_2]] \tag{2}$$

where $\oplus$ denotes the addition operation between two encrypted variables. □

2.2.1. El-Gamal Cryptosystem

El-Gamal is an asymmetric-key multiplicative homomorphic encryption scheme based on the difficulty of the discrete-logarithm problem [12]. The cryptosystem is characterized by the following operations:
- *Public ($\mathcal{K}_{pu}$) and private ($\mathcal{K}_{pr}$) keys generation*:

$$\mathcal{K}_{pr} = k, \quad \mathcal{K}_{pu} = \{\mathbb{G}, p, q, g, h\} \tag{3}$$

where q and $p \in \mathbb{Z}_+$ are two large randomly selected prime number satisfying $((p-1) \mod q = 0)$, $k \in \mathbb{Z}_q$ and $g \in \mathbb{G}$, and $h = g^{\mathcal{K}_{pr}}$. $\mathbb{G} \subset \mathbb{Z}_p^{\times}$ is a cyclic group of the order q modulo p.
- *Encryption*: A message $m \in \mathcal{M}$ is encrypted into a pair $(c_1, c_2) \in \mathcal{C}$ using $\mathcal{K}_{pu}$ and a random number $r \in \mathcal{R}_{rg} = \{1, \ldots, q-1\}$, i.e.,

$$\begin{gathered} Enc[m] = (c_1, c_2) \\ c_1 = g^r \bmod p, \quad c_2 = (m \times h^r) \bmod p \end{gathered} \tag{4}$$

- *Decryption*: An encrypted message $(c_1, c_2) = Enc[m]$ is decrypted using $\mathcal{K}_{pr}$ and $\mathcal{K}_{pu}$ as follows:

$$m = Dec[(c_1, c_2)] = (c_1^{-\mathcal{K}_{pr}} \bmod p)(c_2 \bmod p) \tag{5}$$

2.2.2. Paillier Cryptosystem

Paillier is an asymmetric-key additive homomorphic encryption scheme based on the difficulty of the integer factorization problem [13]. It is characterized by the following operations:
- *Public ($\mathcal{K}_{pu}$) and private ($\mathcal{K}_{pr}$) keys generation*:

$$\mathcal{K}_{pr} = ((p-1)(q-1),\ \eta), \quad \mathcal{K}_{pu} = (n, g) \tag{6}$$

where $p \in \mathbb{Z}_+$ and $q \in \mathbb{Z}_+$ are two large and randomly selected integer prime numbers, $n = pq$ and $\eta = ((p-1)(q-1))^{-1} \bmod n^2$. A random integer number g should be

selected, where $g \in \mathbb{Z}_{n^2}^{\times}$. In what follows, we assume that $g = n + 1$ [13].
- *Encryption*: A message $m \in \mathcal{M}$ is encrypted into $c \in \mathcal{C}$ using $\mathcal{K}_{pu}$ and a random generted number $r \in \mathcal{R}_{rg} := \mathbb{Z}_{n^2}^{\times}$ such that $gcd(r, n) = 1$, i.e.,

$$Enc[m] = c = g^m r^n \bmod n^2 = (n+1)^m r^n \bmod n^2 \tag{7}$$

- *Decryption*: An encrypted message $c = Enc[m]$ is decrypted using $\mathcal{K}_{pr}$ as follow:

$$m = Dec[c] = \left(\frac{(c^{\mathcal{K}_{pr}} \bmod n^2) - 1}{n} \eta \right) \bmod n \tag{8}$$

Although Paillier cryptosystem is only additively homomorphic, it is also possible, exploiting the malleability of the cryptosystem, to compute multiplications between an encrypted message $Enc[m_1]$, $m_1 \in \mathcal{M}$ and a plaintext message $m_2 \in \mathcal{M}$, i.e.,

$$m_1 m_2 = Dec[Enc[m_1]^{m_2} \bmod n^2] = Dec[Enc[m_1] \odot m_2] \tag{9}$$

with $\odot$ denoting the multiplicative operator between one encrypted variable and one plaintext variable.

Remark 2. The encryption algorithms of both El-Gamal (4) and Paillier (7) require that the random variable $r \in \mathcal{R}_{rg}$ to be freshly generated for every encryption operation by a cryptographically secure pseudorandom number generator [34]. Such a requirement is necessary to ensure that these cryptosystems are semantically secure [35]. This raises the challenge of dealing with the lack of randomness needed by a real time CPS process. For example, in modern Unix-variants and Linux, /dev/random interface blocks until the operating system generates more entropy. However, such blocking option is not acceptable in CPS applications that require real time response. In our work, however, we focus on the case where the attacker can maliciously tamper with the RG, i.e., scenarios where the encryption protocols is vulnerable to attacks known as "random number generator attacks," see, e.g., [36]. □

2.3. Encrypted Controller

In this section, we recall how a simple static feedback controller in the form

$$u(t) = Ky(t), \quad K \in \mathbb{R}^{n_m \times n_p} \tag{10}$$

can be implemented in an encrypted fashion using El-Gamal and Paillier cryptosystems, see the survey paper [8] and references therein for a more detailed discussion.
- *Encrypted control computation with El-Gamal*: Since El-Gamal is multiplicative homomorphic, the control law (10) can be computed in the encrypted domain if each sensor measurement $y_i(t)$ and each element K_{ij} of K are separately encrypted. Indeed, the controller can compute the following encrypted matrix $\Gamma(t)$,

$$\Gamma(t) = \begin{pmatrix} Enc[K_{11}] \otimes Enc[y_1] & \cdots & Enc[K_{1n_p}] \otimes Enc[y_{n_p}] \\ \vdots & \ddots & \vdots \\ Enc[K_{n_m 1}] \otimes Enc[y_1] & \cdots & Enc[K_{n_m n_p}] \otimes Enc[y_{n_p}] \end{pmatrix} \tag{11}$$

If $\Gamma(t)$ is transmitted to the actuator, then it can compute each component $u_i(t)$ of $u(t)$ as

$$u_i(t) = \sum_{j=1}^{n_p} Dec[\Gamma_{ij}(t)], \quad i = 1, \ldots, n_m \tag{12}$$

- *Encrypted control computation with Paillier*: Since Paillier cryptosystem is additively homomorphic, it is not possible to compute the matrix $\Gamma(t)$ as in (11). However, $\Gamma(t)$ can still be computed if each entry K_{ij} of K is in plaintext. Moreover, differently from El-Gamal, there is no need to transmit the entire matrix $\Gamma(t)$ to the actuator, because the summation required by (12) can be performed encrypted on the controller's side. Therefore, each $i-$th component of $u(t)$, $i = 1, \ldots, n_m$, can be computed as:

$$Enc[u_i(t)] = (K_{i1} \odot Enc[y_1(t)]) \oplus \cdots \oplus (K_{in_p} \odot Enc[y_{n_p}(t)]) \tag{13}$$

2.4. Problem Formulation

Assumption 1. (*Threat Model*) - The adversary model consists of two coordinated entities: (i) a malware capable of tampering the RG module of the sensor's processing unit and, in one scenario, capable of accessing the private key stored in the actuator's processing unit on the plant's side of the NCS (e.g., by means of a supply chain attack [37]) and (ii) a passive eavesdrop capable of reading the encrypted sensor measurements $Enc[y(t)], \forall\, t$. It is important to note that we assume that no dedicated communication channels exists between the malware and the eavesdropper.

The problem considered in this paper can be summarized as follows:

Given the encrypted control architecture described in the Sections 2.1 - 2.3, show that under Assumption 1, an attacker is able to compromise the confidentiality of encrypted control systems by covertly revealing private information (e.g., secret encryption key or plaintext sensor measurements) to an eavesdropper intercepting the encrypted measurement channel.

3. Proposed Attacks

In this section, under Assumption 1, three different attacks against the confidentiality of encrypted NCS are presented. In all these scenarios, the objective of the malware (the sender) is to covertly tamper the encryption operations to encode sensitive private information in the transmitted encrypted measurements $Enc[y(t)]$. On the other hand, the eavesdropper (the receiver), given the prior knowledge of the sender operations, has the objective to extract the embedded information from $Enc[y(t)]$ and reconstruct private data such as the plaintext sensor measurements $y(t)$ or the secret key $\mathcal{K}_{pr}$.

The considered attacks leverages two potential vulnerabilities of encrypted control systems, namely the *small size of the plaintext message space* $\mathcal{M}$ (see Remark 1) and the *randomness of the cryptosystems* (see Remark 2).

3.1. Attack Scenarios

According to the privileges that the malware can obtain, the following scenarios can be analyzed (see Fig. 1 for a better understating of the description below):

- SC_1- The malware is able to read the private key stored in the actuator's processing unit, and repeat the calls for the random number generator and encryption operation, without outputting the ciphertext, until it satisfies a specific condition.
- SC_2- The malware is able to compromise the initial seed of the RG module.
- SC_3- The malware is able to map the output of the RG module into a restricted space (e.g., by setting some of the output bits of the RG module to zeroes or any pre-specified values).

In what follows, we explain the details of these attack scenarios.

3.1.1. Attack Scenario SC_1

Proposition 1. *Consider the encrypted NCS in Fig. 1. Under the scenario SC_1, the malware can covertly disclose the private key $\mathcal{K}_{pr}$ to the eavesdropper using the encrypted measurement channel.*

Proof - Given the assumed capabilities of the malware, at each time t, it can encode the $j-th$ bit of $\mathcal{K}_{pr}$, namely $\mathcal{K}_{pr}[j]$ into the parity bits of $Enc[y_i(t)]$, $i \in [1, \ldots, n_p]$. More precisely, the malware can re-compute the encrypted sensor measurement $Enc[y_i(t)]$ (with a different random number r) until the encrypted binary vector $Enc[y_i(t)]$ has a parity bit equals to $\mathcal{K}_{pr}[j]$. Such encoding operations are summarized in Algorithm 1.
On the other hand, the eavesdropper on the measurement channel can recover the

Algorithm 1: Encoding the binary secret $\mathcal{K}_{pr}$ in the encrypted sensor measurements

```
Initialization: length_of_secret = |K_pr|, j = 0;
— ∀ t : —
 if j < length_of_secret then
    for i = 1 : n_p do
       while (parity bit of Enc[y_i(t)] ≠ K_pr[j]) do
          r ← generate a new random number ∈ R_rg;
          Enc[y_i(t)] ← compute the encrypted sensor measurement y_i(t);
       end
       j = j + 1;
    end
end
```

transmitted secret key by simply sequentially storing the parity bit of the received encrypted sensor measurements. □

Remark 3. Using Algorithm 1, the attacker is able to transmit, at each sampling time t, n_p bits of $\mathcal{K}_{pr}$ using a tampered but legitimate ciphertext that is indistinguishable from a normal ciphertext. For example, consider a case where the plant has two sensor

measurements, i.e., $n_p = 2$, the sampling time is $T_s = 1$ ms and the secret key $\mathcal{K}_{pr}$ is 1024 bits. In this setup, the attacker can embed 2 bits of $\mathcal{K}_{pr}$ at each sampling time t in the parity bit of $Enc[y_1(t)]$ and $Enc[y_2(t)]$. Therefore, after 512 ms, the entire key is transmitted.

Remark 4. Note that the disclosure attack described in Propostion 1 leverages the randomness of the cryptosystem to launch the attack. As a consequence, this attack can be performed in both Paillier and El-Gamal. Moreover, exploiting the same idea, the attacker can transmit any other sensitive information that the malware might have access to. For El-Gamal, it is implicitly assumed that each bit of $\mathcal{K}_{pr}$ is encoded in either the parity bit of c_1 or c_2. □

3.1.2. Attack Scenario SC_2

Proposition 2. *Consider the encrypted NCS in Fig. 1. Under the scenario SC_2, if the malware and eavesdropped have offline shared a seed number ζ, then the malware can covertly enable the eavesdropper to correctly decode $Enc[y(t)]$.*

Proof - In SC_2, the malware can set the initial seed of the RG. As a consequence, the eavesdropper (who also knows ζ) can predict the entire sequence of random numbers r generated by RG. According to the used cryptosystem, the eavesdropper operations to recover y_i are as follows:
El-Gamal: According to (5), each scalar variable $y_i, i = 1, \dots, n_p$ must be decrypted from $E[y_i(t)]$ as

$$y_i(t) = (c_1^{-\mathcal{K}_{pr}} \bmod p)(c_2 \bmod p)$$

However, since $c_1 = g^r \bmod p$ (see (4)) and $h = g^{\mathcal{K}_{pr}}$ (see (3)), we can re-write the above as

$$y_i(t) = \left(h^{-r} \bmod p\right)(c_2 \bmod p) \tag{14}$$

Therefore, since all the variables on the right hand side of (14) are known, i.e., r, p, h with p, h part of the public key, to the eavesdropper, then $Enc[y_i(t)], i = 1, \dots, n_p$ can be successfully recovered.
Paillier: According to (7),

$$Enc[y_i(t)] = (n+1)^{y_i(t)} r^n \bmod n^2$$

and, by exploiting the knowledge of r and of the public key n, we can multiply both sides by $(r^{-n} \bmod n^2)$, obtaining

$$Enc[y_i(t)]r^{-n} \bmod n^2 = (n+1)^{y_i(t)} \bmod n^2 \tag{15}$$

Then, by resorting to the binomial theorem and exploiting the mod operator (which makes zero all the terms of the binomial multiple of n^2), we can simplify the right hand side of (15) and obtain

$$Enc[y_i(t)]r^{-n} \bmod n^2 = (1 + ny_i(t)) \bmod n^2$$

from which

$$y_i(t) = \frac{(Enc[y_i(t)]r^{-n} - 1)}{n} \bmod n^2 \tag{16}$$

concluding the proof. □

Note that in (16), the notation $\frac{a}{b}$ does not denote the modular multiplication of a times multiplicative inverse of b.; it denotes the quotient of a divided by b.

3.1.3. Attack Scenario SC_3

Proposition 3. *Consider the encrypted NCS in Fig. 1. Under the scenario SC_3, if the malware and the eavesdropper agree on a restricted random space $\mathcal{R}_{small} \subset \mathcal{R}$, then the malware can covertly enable the eavesdropper to correctly decode $Enc[y(t)]$.*

Proof - Given the knowledge of $\mathcal{K}_{pu}$, $Enc[y(t)]$ and $\mathcal{R}_{small}$, the eavesdropper, can perform the following actions:
El-Gamal: by taking advantage of the restricted the random space $\mathcal{R}_{small}$ imposed by the malware, the eavesdropper can offline build a lookup table LT containing the following pairs:

$$\left\{ \left(r, \underbrace{g^r \bmod p}_{=c_1} \right) : r \in \mathcal{R}_{small} \right\}$$

As a consequence, given $Enc[y_i(t)] = (c_1, c_2)$, it is possible to use c_1 and LT to obtain r.
Paillier: by taking advantage of the restricted random space $\mathcal{R}_{small}$, given $Enc[y_i(t)]$, the eavesdropper can compute the set of admissible plaintext messages:

$$\mathcal{Y}(R_{small}) = \{ y_i(t) \in \mathcal{M} : r \in \mathcal{R}_{small},\, gcd(r, n) = 1,\, y_i(k) \text{ as in (16) } \} \tag{17}$$

Since the message space is restricted to $\mathcal{M}$, with $|\mathcal{M}| \ll |n|$ (see Remark 1), then the probability ρ of obtaining a random valid message $y_i \in \mathcal{M}$, given a randomly chosen $r \in \mathcal{R}_{small}$, is negligible, i.e., $\rho = \frac{1}{2^{|n|-|\mathcal{M}|}} \approx 0$. As a consequence, almost surely $\mathcal{Y}(\mathcal{R}_{small}) = y_i(t)$ (in a practical encrypted control setup, using Paillier, $\mathbb{Z}_n = \mathbb{Z}_{2^{1024}}$ and $\mathcal{M} = \mathbb{Z}_{2^{16}}$. Therefore, $\rho = \frac{1}{2^{1008}}$, and for a restricted random space, e.g., $\mathcal{R}_{small} = \mathbb{Z}_{2^{32}}$, the cumulative probability that $\mathcal{Y}(\mathcal{R}_{small})$ contains two or more valid messages is practically zero). This concludes the proof.

From the above discussion, it follows that the time required by the attacker to recover y_i from its ciphertext in the case of El-Gamal is independent of $|R_{small}|$ since the eavesdropper is using a lookup table. However, for Paillier, this time grows exponentially with $|R_{small}|$. The important point, however, is that eavesdropper finds only one admissible value for $y_i(t)$ in both cryptosystems, independent of $|R_{small}|$. □

Remark 5. In Paillier, if the more general form of the cryptosystem is used, i.e., $g \neq (n + 1)$, then the attack in SC_2 and SC_3 can still be performed with some modifications:
- SC_2: Since r is known and the message space is restricted, the eavesdropper can

offline build a lookup table $LT(\mathcal{M})$, containing the following pairs:

$$LT(\mathcal{M}) := \{(y_i, \underbrace{g^{y_i} r^n \mod n^2}_{=Enc(y_i)}) : y_i \in \mathcal{M}\}$$

Therefore, given $Enc[y_i(t)]$, the eavesdropper can obtain $y_i(t)$ from $LT(\mathcal{M})$.
- SC_3: By taking advantage of the restricted random and message spaces, the eavesdropper can offline build a lookup table $LT(\mathcal{M})$, containing the following pairs:

$$LT(\mathcal{M}) := \left\{\left(y_i, g^{y_i} \mod n^2\right) : y_i \in \mathcal{M}\right\}$$

Moreover, given $Enc[y_i(t)]$, the attacker can perform a search over the admissible random space ($r \in \mathcal{R}_{small}$, $gcd(r, n) = 1$) and compute $g^{y_i(t)} \mod n^2 = Enc[y_i(t)]r^{-n} \mod n^2$ until a value contained in $LT(\mathcal{M})$ is found. □

Remark 6. In this section, we have developed the attacks in $SC1 - SC3$, assuming that the popular El-Gamal or Paillier cryptosystems are used. However, the proposed attacks leverage the inherently small domain of the message space $\mathcal{M}$ in control systems as well as the randomization process used in HE schemes. Therefore, the proposed attacks are valid for a more general class of encrypted control systems where the cryptosystem utilizes a randomization process for encryption. □

4. Countermeasures

Since the considered attacks exploit intrinsic vulnerabilities related to the random generator (RG) and small message space ($\mathcal{M}$)), existing anomaly/attack detectors for encrypted control systems (see e.g., [20–22] and references therein) are not effective. Moreover, this class of random generator attacks cannot be detected by analyzing the ciphertext (e.g., see [38]). Consequently, instead of proposing an attack detection

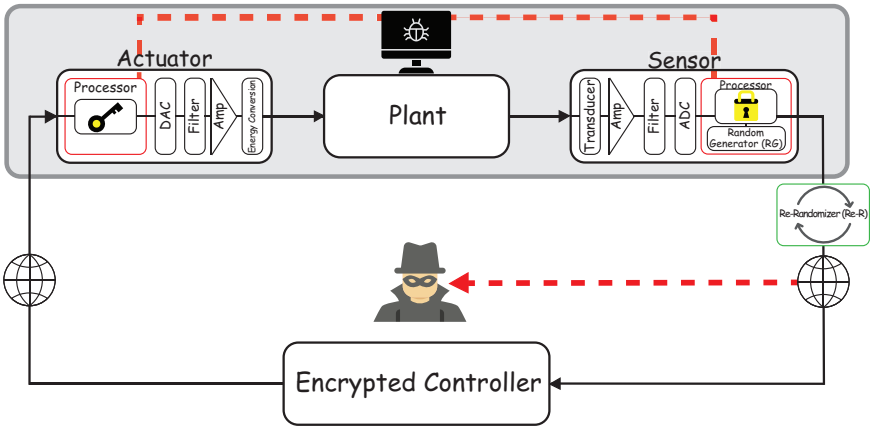

Figure 2.: Encrypted control system with Re-Randomizer.

strategy, we hereafter introduce a solution that prevents their existing. Specifically, we propose adding a new trusted subsystem, hereafter called "Re-Randomizer (Re-R)", between the sensor's subsystem and the communication channel (see Fig. 2), such that

(C_1) : Re-R takes in input $Enc[y_i]$, $\forall\, i$ and performs a re-randomizing of the ciphertext. By denoting with $\tilde{E}nc[y_i]$ the re-randomized version of $Enc[y_i]$, then y_i must be correctly decrypted from $\tilde{E}nc[y_i]$ by an entity possessing the private key $\mathcal{K}_{pr}$;
(C_2) : $\tilde{E}nc[y_i]$ prevents the attack scenarios SC_1-SC_3;
(C_3) : The Re-R's processor unit is completely independent from the actuator and sensor's units (i.e, if a malware has access to the plant, then it cannot compromise Re-R).

The following proposition proposes possible re-randomization solutions for the El-Gamal and Paillier cryptosystems.

Proposition 4. *Consider a single encrypted message $Enc[y_i]$. In El-Gamal ($Enc[y_i] = (c_1, c_2)$), the re-randomization*

$$\tilde{E}nc[y_i] = (c_1 g^{\tilde{r}}, c_2 g^{\tilde{r}}), \quad \tilde{r} \in \mathcal{R}_{rg} \tag{18}$$

and in Paillier, the re-randomization

$$\begin{gathered} \tilde{E}nc[y_i] = Enc[y_i] \times \tilde{r}^n \bmod n^2, \\ \tilde{r} \in \mathcal{R}_{rg},\ s.t.\ gcd(\tilde{r}, n) = 1 \end{gathered} \tag{19}$$

fulfill the conditions (C_1)-(C_2).

Proof - The proof that (C_1) and (C_2) hold true is here split in two parts:
(C_1) : In El-Gamal, by construction, the encrypted message (18) is equal to

$$\tilde{E}nc[y_i] = (g^{r+\tilde{r}} \bmod p,\ y_i h^{r+\tilde{r}} \bmod p)$$

that, using (5) can be correctly decrypted into y_i. In Paillier, the encrypted message (19) is equal to

$$\tilde{E}nc[y_i] = (n+1)^{y_i} (r\tilde{r})^n \mod n^2$$

Moreover, since $gcd(r, n) = 1$ and $gcd(\tilde{r}, n) = 1$, then also $gcd(r\tilde{r}, n) = 1$. As a consequence, using (8), $\tilde{E}nc[y_i]$ can be correctly decrypted into y_i.
(C_2) : The re-randomization process randomly changes the parity bit of the encrypted variable $\tilde{E}nc[y_i]$. The latter is sufficient to nullify the attacker attempt in SC_1 to embed each bit of $\mathcal{K}_{pr}$ in the parity bit of $Enc[y_i]$, i.e., the probability of successfully decoding each bit of the $\mathcal{K}_{pr}$ is 0.5; Since the re-randomization embeds into the encrypted message $\tilde{E}nc[y_i]$ a new random number $\bar{r} \in \mathcal{R}_{rg}$ ($\bar{r} = r + \tilde{r}$ in El-Gamal, $\bar{r} = r\tilde{r}$ in Paillier), then the attacker is not aware of the used random number as well as it cannot restrict the random space. The latter is sufficient to conclude that the attack scenarios SC_2 and SC_3 are prevented. □

Therefore, the operations performed by the Re-R module can be summarized as follows:

(1) At each time-step t, the Re-Randomizer unit generates a new full-range random number $\tilde{r} \in \mathcal{R}_{rg}$. Moreover, if the Paillier cryptosystem is used, then $\tilde{r}$ must satisfy the condition $gcd(n, \tilde{r}) = 1$.

(2) Given $Enc[y_i(t)]$ and generated new random number $\tilde{r}$, the Re-R entity computes the re-randomized encrypted message $\tilde{Enc}[y_i(t)]$ according to the used cryptosystem:
   - *El-Gamal:* $\tilde{Enc}[y_i(t)]$ is computed as in (18).
   - *Paillier:* $\tilde{Enc}[y_i(t)]$ is computed as in (19).

(3) The re-randomized-encrypted messages $\tilde{Enc}[y_i(t)]$ are transmitted instead of $Enc[y_i(t)]$ to the controller.

5. Simulation Results

In this section, by considering a simple encrypted control system setup, we show the effectiveness of the attacks scenarios described in section 3.1. The effectiveness of the proposed re-randomization technique is also verified. In the performed simulations, we considered a time-invariant discrete-time plant dynamical model whose state-space description is $x(t+1) = Ax(t) + Bu(t)$, $y(t) = Cx(t)$, and where

$$A = \begin{bmatrix} 1.01 & -0.01 \\ 0.00 & 1.02 \end{bmatrix}, B = \begin{bmatrix} 0.00 \\ 0.01 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$x(t) = y(t) \in \mathbb{R}^2$, $u(t) \in \mathbb{R}$. The plant is stabilized by a static feedback controller (10) where $K = [6.574, -6.201]$ and we assume that the ADC converter uses a sampling time period $T_s = 0.001\,s$ and $d = 16$ bits for the analog-to-digital conversion (see Remark 1) for each sensor measurement y_i, $i = 1, 2$. As a consequence, the considered message space is $|\mathcal{M}| = 2^{16}$. The El-Gamal and Paillier cryptosystems have been implemented with p, q such that $|p| = |q| = 1024$, and the encrypted control inputs are computed as in (12) and (13), accordingly. The encrypted control system operations have been simulated using the "eclib" python package [39].

By considering the attack scenario SC_1, Fig. 3, shows the number of key bits (of $\mathcal{K}_{pr}$) correctly recovered by the eavesdropper over time. The solid blue line depicts the result in the absence of the Re-R module, while the dashed red line in the case Re-R is used. In the absence of Re-R, the plot shows a slop equal to 1, denoting that all the bits are correctly decoded. On the other hand, the evolution of the red solid line shows that the eavesdropper can correctly decode (as expected) approximately 50% of the received key bits (see the proof of Proposition 4). Note that this does not provide the adversary with any useful information Since the adversary cannot know the positions of the correctly decoded bits.

By considering SC_2, and the Paillier cryptosystem, Fig. 4 shows, over a time interval of 2 second, the difference between the actual analog sensor measurements $y_i(t)$, $i = 1, 2$ and the decrypted value, namely $y_i^E(t)$, obtained by the eavesdropper using (16). The results show that the attacker can obtain $y_i(t)$ with an error that is limited only by the quantization error ($\frac{1}{2^{16}}$) in the considered ADC [40]. As a consequence, the attacker's estimation is identical to that obtained by the legitimate user using (8). Repeating the above experiment for SC3, produced an identical results to the one shown in Fig. 4.

Finally, Figs. 5 and 6 show the capability of the eavesdropper to correctly recover the sensor measurement data. In particular, for the time interval [0, 2] sec,, both figures depict the sensor measurement data produced by the sensor and the values recovered by the eavesdropper in $SC2$ and $SC3$. It can be observed that the data recovered by

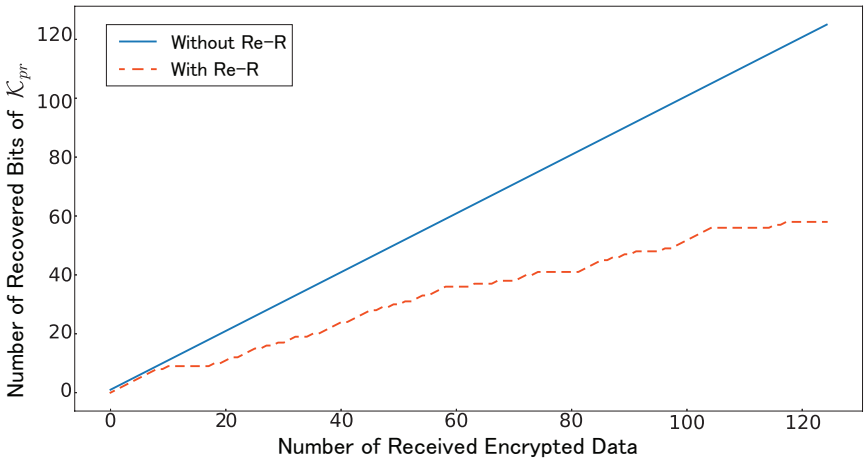

Figure 3.: Number of recovered key bits in SC_1, with and without the re-randomization (Re-R) module.

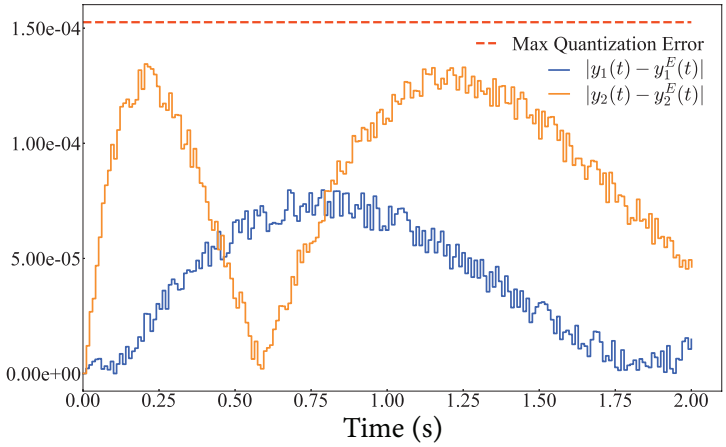

Figure 4.: Difference between what the adversary can recover and the actual sensor measurements.

the eavesdropper is equal to the quantized sensor measurements, which are encrypted and sent over the channel.

6. Conclusion and Future Work

In this paper, we have shown that different attacks can compromise the confidentially of encrypted control systems based on homomorphic cryptosystems. In particular, we have shown that if an attacker is capable of deploying a malware into the plant's side of the networked control system, then it can leverage intrinsic vulnerabilities (e.g., the limited message space and the randomness required to achieve semantic security of the encryption algorithms) to establish an illegitimate covert communication channel with an eavesdropper on the measurement channel. Then, we have proved that if a trusted re-randomization unit is used, these disclosure attacks are prevented.
For future work, one may investigate other attacks that do not require compromising

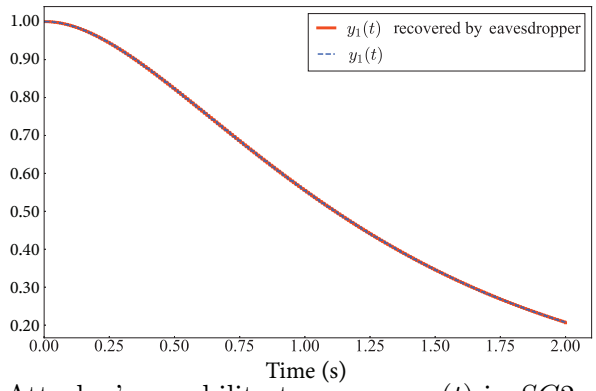

Figure 5.: Attacker's capability to recover $y_1(t)$ in $SC2$ and $SC3$.

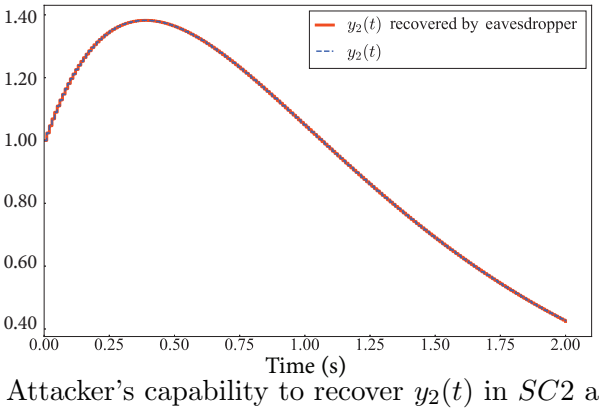

Figure 6.: Attacker's capability to recover $y_2(t)$ in $SC2$ and $SC3$.

the random number generator. For example, Boneh et. al. [41] showed that, under some conditions, when the length of the message is small, RSA and El-Gamal cryptosystems can be insecure. However, such attacks are probabilistic in nature and would only allow the recovery of a subset of the plaintext (measurements). Hence, it would be interesting to explore the effectiveness of such attacks in the context of encrypted control systems.

References

[1] S. M. Dibaji, M. Pirani, D. B. Flamholz, A. M. Annaswamy, K. H. Johansson, and A. Chakrabortty, "A systems and control perspective of cps security," *Annual Reviews in Control*, vol. 47, pp. 394–411, 2019.

[2] A. Cardenas, S. Amin, B. Sinopoli, A. Giani, A. Perrig, S. Sastry, *et al.*, "Challenges for securing cyber physical systems," in *Workshop on future directions in cyber-physical systems security*, vol. 5, no. 1, 2009.

[3] J. Giraldo, D. Urbina, A. Cardenas, J. Valente, M. Faisal, J. Ruths, N. O. Tippenhauer, H. Sandberg, and R. Candell, "A survey of physics-based attack detection in cyber-physical systems," *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–36, 2018.

[4] A. Sultangazin and P. Tabuada, "Symmetries and isomorphisms for privacy in control over the cloud," *IEEE Trans. on Automatic Control*, vol. 66, no. 2, pp. 538–549, 2020.

[5] A. M. Naseri, W. Lucia, M. Mannan, and A. Youssef, "On securing cloud-hosted cyber-physical systems using trusted execution environments," in *IEEE Int. Conference on Autonomous Systems*, 2021.

[6] Z. Ju, H. Zhang, and Y. Tan, "Distributed deception attack detection in platoon-based connected vehicle systems," *IEEE transactions on vehicular technology*, vol. 69, no. 5, pp. 4609–4620, 2020.

[7] Y. Xu, M. Fang, Z.-G. Wu, Y.-J. Pan, M. Chadli, and T. Huang, "Input-based event-triggering consensus of multiagent systems under denial-of-service attacks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 4, pp. 1455–1464, 2018.

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

[9] O. Givehchi, J. Imtiaz, H. Trsek, and J. Jasperneite, "Control-as-a-service from the cloud: A case study for using virtualized plcs," in *2014 10th IEEE Workshop on Factory Communication Systems (WFCS 2014)*. IEEE, 2014, pp. 1–4.

[10] K. Kogiso and T. Fujita, "Cyber-security enhancement of networked control systems using homomorphic encryption," in *IEEE Conf. on Decision and Control (CDC)*, 2015, pp. 6836–6843.

[11] C. Fontaine and F. Galand, "A survey of homomorphic encryption for nonspecialists," *EURASIP Journal on Information Security*, vol. 2007, pp. 1–10, 2007.

[12] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[13] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Int. Conf. on the Theory and Applications of Cryptographic Techniques*, 1999, pp. 223–238.

[14] 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.

[15] 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.

[16] K. Teranishi, N. Shimada, and K. Kogiso, "Stability analysis and dynamic quantizer for controller encryption," in *IEEE Conf. on Decision and Control (CDC)*. IEEE, 2019, pp. 7184–7189.

[17] Y. Geng *et al.*, "Homomorphic encryption technology for cloud computing," *Procedia Computer Science*, vol. 154, pp. 73–83, 2019.
[18] K. Ishikawa, K. Nagasawa, K. Kogiso, and K. Sawada, "Experimental validation of encrypted controller implemented on raspberry pi," in *2016 IEEE 4th International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA)*. IEEE, 2016, pp. 1–6.
[19] J. Tran, F. Farokhi, M. Cantoni, and I. Shames, "Digital implementation of homomorphically encrypted feedback control for cyber-physical systems," 2019.
[20] R. Baba, K. Kogiso, and M. Kishida, "Detection method of controller falsification attacks against encrypted control system," in *SICE Annual Conference*, 2018, pp. 244–248.
[21] K. Kogiso, "Attack detection and prevention for encrypted control systems by application of switching-key management," in *IEEE Conf. on Decision and Control (CDC)*, 2018, pp. 5032–5037.
[22] K. Teranishi and K. Kogiso, "Control-theoretic approach to malleability cancellation by attacked signal normalization," *IFAC-PapersOnLine*, vol. 52, no. 20, pp. 297–302, 2019.
[23] J. Lee, J. Kim, and H. Shim, "Zero-dynamics attack on homomorphically encrypted control system," in *Int Conf on Control, Automation and Systems*, 2020, pp. 385–390.
[24] B. W. Lampson, "A note on the confinement problem," *Communications of the ACM*, vol. 16, no. 10, pp. 613–615, 1973.
[25] A. Abdelwahab, W. Lucia, and A. Youssef, "Covert channels in cyber-physical systems," *IEEE Control Systems Letters*, vol. 5, no. 4, pp. 1273–1278, 2020.
[26] D. C. Latham, "US department of defense trusted computer system evaluation criteria," *Department of Defense*, 1986.
[27] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(leveled) fully homomorphic encryption without bootstrapping," *ACM Transactions on Computation Theory*, vol. 6, no. 3, pp. 1–36, 2014.
[28] C. Gentry, "Computing arbitrary functions of encrypted data," *Communications of the ACM*, vol. 53, no. 3, pp. 97–105, 2010.
[29] J. Park, A. John Park, and S. Mackay, *Practical data acquisition for instrumentation and control systems*. Newnes, 2003.
[30] C. Murguia, F. Farokhi, and I. Shames, "Secure and private implementation of dynamic controllers using semihomomorphic encryption," *IEEE Trans. on Automatic Control*, vol. 65, no. 9, pp. 3950–3957, 2020.
[31] J. H. Cheon, K. Han, H. Kim, J. Kim, and H. Shim, "Need for controllers having integer coefficients in homomorphically encrypted dynamic system," in *IEEE Conf. on Decision and Control (CDC)*, 2018, pp. 5020–5025.
[32] Y. Lin, F. Farokhi, I. Shames, and D. Nešić, "Secure control of nonlinear systems using semi-homomorphic encryption," in *IEEE Conf. on Decision and Control (CDC)*, 2018, pp. 5002–5007.
[33] X. Yi, R. Paulet, and E. Bertino, "Homomorphic encryption," in *Homomorphic Encryption and Applications*, 2014, pp. 27–46.
[34] B. D. Ripley, "Thoughts on pseudorandom number generators," *Journal of Computational and Applied Mathematics*, vol. 31, no. 1, pp. 153–163, 1990.
[35] M. Bellare, R. Dowsley, and S. Keelveedhi, "How secure is deterministic encryption?" in *IACR Int. Workshop on Public Key Cryptography*, 2015, pp. 52–73.
[36] I. Goldberg and D. Wagner, "Randomness and the netscape browser," Jan 1996.
[37] K. Yang, M. Hicks, Q. Dong, T. Austin, and D. Sylvester, "A2: Analog malicious hardware," in *IEEE Symposium on Security and Privacy*, 2016, pp. 18–37.
[38] P. Austrin, K.-M. Chung, M. Mahmoody, R. Pass, and K. Seth, "On the impossibility of cryptography with tamperable randomness," in *Annual Cryptology Conference*, 2014, pp. 462–479.
[39] Eclib python library. [Online]. Available: https://github.com/KaoruTeranishi/EncryptedControl
[40] S. Sokolov, V. Kamenskij, A. Novikov, and V. Ivetić, "How to increase the analog-to-

digital converter speed in optoelectronic systems of the seed quality rapid analyzer," *Inventions*, vol. 4, no. 4, p. 61, 2019.

[41] D. Boneh, A. Joux, and P. Q. Nguyen, "Why textbook elgamal and rsa encryption are insecure," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2000, pp. 30–43.