Hindawi
Security and Communication Networks
Volume 2023, Article ID 6473497, 12 pages
https://doi.org/10.1155/2023/6473497

Research Article

Secure Control Using Homomorphic Encryption and Efficiency Analysis

Jingshan Pan,[1,2,3,4,5] Tongtong Sui,[4] Wen Liu,[3,5] Jizhi Wang,[2,4,5] Lingrui Kong,[4] and Yue Zhao[4]

[1]*College of Information Science and Engineering, Ocean University of China, Qingdao 266100, China*
[2]*Shandong Provincial Key Laboratory of Computer Networks, Shandong Computer Science Center (National Supercomputing Center in Jinan), Jinan 250014, China*
[3]*Jinan Institute of Supercomputing Technology, Jinan 250301, China*
[4]*Qilu University of Technology (Shandong Academy of Science), Jinan 250102, China*
[5]*Quancheng Laboratory, Jinan 250100, China*

Correspondence should be addressed to Jizhi Wang; wangjzh@sdas.org

Received 9 October 2022; Revised 20 January 2023; Accepted 21 January 2023; Published 14 April 2023

Academic Editor: Vincenzo Conti

In order to enhance the cyber-security of networked control systems, Kogiso and Fujita (2015) proposed a concept of controller encryption using homomorphic encryption for the first time. Encrypted linear controllers using a homomorphic encryption scheme could conceal the information processed inside the controller device and maintain the original functions of controllers. In this paper, we propose a scheme to encrypt the linear controller using the BGN encryption, which supports homomorphic addition and multiplication. We also compare the efficiency of this scheme with the scheme with an encrypted controller using RSA encryption and Paillier encryption.

1. Introduction

A networked control system (NCS) is an emerging technology developed gradually with the development of control technology, network communication technology, and computer technology [1]. NCS is different from traditional control technology, employing shared communication channels to close the control loop. This provides many benefits, including easier installation and maintenance, as well as lower cost, weight, and volume; so NCS is used in water, transportation, electricity, and other industries and critical infrastructure. However, while being widely used, NCS also exposes some security problems [2–4], among which network security and privacy is an important issue.

An eavesdropping attack is a kind of important attack in cyber-security. A malicious adversary may eavesdrop on the communication channels between sensors, controllers, and actuators to extract valuable information about the model and controller based on the transmitted data. A typical control loop with encryption-decryption units is shown in Figure 1. It encrypts to protect the data transmitted through the communication channel, thus significantly enhancing the security of the data in the communication channel. However, a disadvantage of traditional encryption is that the received information must be decrypted in the controller in order to complete the initial operation. That is, the key must be stored inside the controller which increases the burden of the controller.

In addition, the key data in the controller are still at risk of being eavesdropped, which can easily cause serious damage to the industrial control system [5–7].

Aiming at the deficiency of traditional encryption as shown in Figure 1, Kogiso and Fujita [8] first proposed the concept of controller encryption using partially homomorphic encryption and proposed the encrypted controller scheme of the encrypted linear controller. Homomorphic encryption (HE) [9] is a scheme that allows homomorphic operations (such as addition and multiplication) on

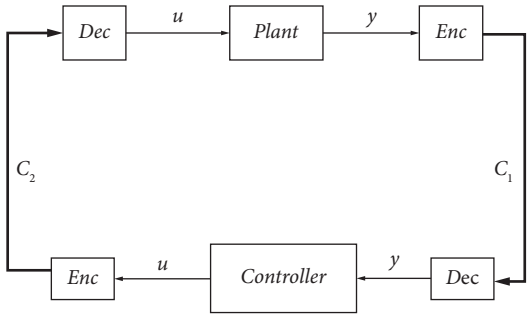

Figure 1: The NCS with traditional encryption.

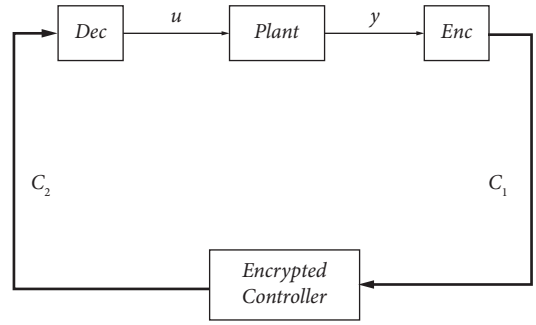

Figure 2: The security-enhanced NCS.

encrypted data without decryption. Partially homomorphic encryption usually only satisfies one of addition and multiplication in homomorphic operations. Therefore, if a homomorphic encryption scheme is used to encrypt the controller, it is not necessary to decrypt the input data of the controller first, which may avoid the problem of the traditional encryption scheme mentioned above. As shown in Figure 2, the controller can directly evaluate the encrypted data without using the private key to decrypt it. Specifically, they modified RSA encryption [10] and ElGamal encryption [11] to solve the problem that such partially homomorphic encryption does not support additive homomorphism and then encrypted the linear controller based on the modified scheme. Paillier encryption [12] is a partial homomorphic encryption that only supports homomorphic addition. Farokhi et al. [13] proposed an encrypted static feedback controller scheme based on Paillier encryption. Based on this, a series of studies on the cyber-security of NCSs have been made [14–16].

Compared with partially homomorphic encryption, fully homomorphic encryption supports homomorphic addition and multiplication on the encrypted data. Kim et al. [17] proposed the use of FPHE (floating-point homomorphic encryption) to encrypt linear controllers. In order to solve the problem of limited ciphertext lifetime, they proposed a solution to run multiple controllers at the same time and complete bootstrapping by scheduling controllers.

In this paper, we present a scheme of the encrypted controller using BGN encryption [18] which is a somewhat homomorphic encryption. It allows homomorphic addition and one multiplication during homomorphic evaluation. Different from the previous encryption controller scheme, it can not only complete the operation of the controller but also complete the encryption protection of data in the controller and in the channel at the same time. Then, we describe the attack scenario [19–21] in this study, considering the security of the scheme. Finally, we give the simulation of the iteration time of NCS with an encrypted controller using RSA encryption, Paillier encryption, and BGN encryption and then compare their efficiency. The advantages and disadvantages of the three schemes are compared in detail.

The rest of this paper is organized as follows. We introduce the related work about the encrypted controller in Section 2. We recall some notions, definitions, and facts in Section 3. Then, we present the two encrypted controller schemes using RSA and Paillier encryption and propose the encrypted controller using optimized BGN encryption in Section 4. And then, we consider the attack scenario in Section 5. In Section 6, we show the numerical example to verify that the encrypted controller can achieve normal control and compare the efficiency of the three encrypted controllers. We analyze the advantages and disadvantages of the above-given schemes in Section 7. In the end, we conclude in Section 8.

2. Related Work

Many scholars have studied the encrypted controller based on homomorphic encryption and proposed a variety of encrypted controller schemes. For the encrypted controller, the relevant research status at home and abroad is as follows.

The idea of an encrypted controller was first proposed in [8] based on RSA and ElGamal cryptosystems. Subsequently, an encrypted control system using the Paillier encryption is considered in [13], and conditions on the parameters of the encryption technique are given to ensure the closure of the closed-loop system and the boundedness of the closed-loop performance. Kim et al. [17] studied the encrypted controller design based on fully homomorphic encryption, which alleviated the extra overhead and quantization error caused by quantization recovery.

With the increase of system complexity, more and more systems adopt distributed control schemes, and the application of distributed controllers requires communication among agents. Farokhi et al. [13] implemented a stochastic control scheme for encrypted distributed control. Schulze Darup et al. [22] used Paillier encryption to design encryption cooperative control based on structural feedback. Alexandru et al. [23] used homomorphic encryption and private and aggregation schemes to strengthen distributed control security. To solve the optimization problem of distributed control, Lu et al. [24] proposed a privacy protection scheme based on homomorphic encryption, which ensured that the state of each agent and the coefficient of the holding component function were private, and the algorithm had high accuracy and fast computational efficiency.

Schulze Darup et al. [25] show that homomorphic encryption can be used to implement model predictive control schemes for linear systems with state and input constraints. Alexandru et al. [26] explored the confidentiality of

predictive control of the cloud outsourcing model for linear systems with input constraints and proposed two methods to achieve private computation by fast gradient method and using additive homomorphic cryptography. The cryptographic model predictive control proposed by Schulze Darup [27] allows the inclusion of state and input constraints without the need for an explicit solution of the optimal control problem (OCP).

3. Preliminaries

The following notions will be used throughout the paper. R is the set of real numbers, Z_n is a reduced residue system modulo n, i.e., $Z_n: = \{0, 1, \ldots, n-1\}$ and Z_n^* is the set of integers coprime to n which belongs to Z_n (or is a multiplicative group of residues modulo n). 1^n is a binary string of length n. For a positive number n, the Euler function $\varphi\{n\}$ returns the order of the group $|Z_n^*|$. For a real number r, bre denotes the nearest integer to r.

3.1. Basic Notions

3.1.1. Public-Key Encryption. A public-key encryption scheme consists of three probabilistic polynomial time(PPT) algorithms $E = (\text{Gen, Enc, Dec})$ such that:

(i) Gen: this is the key-generation algorithm which takes as input the security parameter 1^n and outputs a pair of keys (pk, sk). We refer to the former as the public key and the latter as the secret key. The public key pk determines the message space M.

(ii) Enc: the encryption algorithm Enc takes as input a public key pk and message (or we called plaintext) $m \in M$ and then outputs a ciphertext c. We denote this by $c = \text{Enc(pk, m)}$. The ciphertext space is C.

(iii) Dec: the decryption algorithm Dec takes as input a secret key sk and ciphertext $c \in C$ and outputs a message m or a special symbol $\perp$ denoting failure. We denote this by $m = \text{Dec(sk, } c)$.

Here, except with negligible probability over the randomness of Enc and Dec, it holds that $\text{Dec(sk, (Enc(pk, } (m)) = m$ for any message $m \in M$.

3.1.2. Homomorphic Encryption. A public-key scheme (Gen, Enc, Dec) is homomorphic if for all n and (pk, sk) outputs by $\text{Gen}(1^n)$, the following conditions are fulfilled:

(1) The set M together with operation $\bullet$ and the set C together with operation $*$ from a group, respectively

(2) Any plaintext $m \in M$ is mapped into C, i.e.

$$\text{Enc}(\text{pk}, \text{m}) \in \text{C}, \forall \text{m} \in \text{M}. \tag{1}$$

(3) If for any plaintexts, m_1 and $m_2 \in M$, the corresponding ciphertexts are written as

$$\begin{aligned} \text{Enc}(\text{pk}, \text{m}_1) &= \text{c}_1 \in \text{C}, \\ \text{Enc}(\text{pk}, \text{m2}) &= \text{c2} \in \text{C}. \end{aligned} \tag{2}$$

Then, the following equation:

$$\text{Enc}(\text{pk}, \text{m}_1 \bullet \text{m}_2) = \text{c}_1 * \text{c}_2, \tag{3}$$

is held.

3.2. Encryption Schemes

3.2.1. RSA Encryption. The RSA encryption scheme consists of three PPT algorithms (Gen, Enc, Dec) as follows:

(i) Gen: on input 1^n to obtain N, e, and d, the public key is $\text{pk} = (N, e)$ and the secret key is $\text{sk} = (N, d)$, where n-bit binary $N = pq$ with different n-bit prime integers p and q, as well as integers $e, d > 0$ with $\gcd(e, \varphi(N)) = 1$ and $\text{ed} = 1 \bmod \varphi(N)$. Here, the message space is $M = Z_N$.

(ii) Enc: on input public key $\text{sk} = (N, e)$ and a message $m \in M$, compute the ciphertext

$$\text{c} = [\text{m}^{\text{e}} \bmod \text{N}]. \tag{4}$$

(iii) Dec: on input secret key $\text{sk} = (N, d)$ and the ciphertext c, compute the following message:

$$\text{m} = \left[\text{c}^{\text{d}} \bmod \text{N}\right]. \tag{5}$$

The RSA encryption allows homomorphic evaluation of the multiplication on the ciphertexts.

With two messages $m_1, m_2 \in M$ and the corresponding ciphertexts $c_1, c_2 \in \text{C}$:

$$\text{c}_1 = \text{Enc}(\text{pk}, \text{m}_1) = \text{m}_1^{\text{e}} \bmod \text{N}, \text{c}_2 = \text{Enc}(\text{pk}, \text{m}_2) = \text{m}_2^{\text{e}} \bmod \text{N}, \tag{6}$$

multiplication of m_1 and m_2, results in,

$$\begin{aligned} \text{Enc}(\text{pk}, m_1 m_2) &= (m_1 m_2)^e \bmod \text{N} = (m_1^e \bmod \text{N})(m_2^e \bmod \text{N}) \bmod \text{N} \\ &= \text{Enc}(\text{pk}, m_1)\text{Enc}(\text{pk}, m_2) \bmod \text{N} = c_1 c_2 \bmod \text{N}. \end{aligned} \tag{7}$$

This means that in equation (1), the operation $\bullet$ over the set M is multiplication and the operation $*$ over the set C is modular multiplication.

3.2.2. Paillier Encryption. Let GenModulus be a PPT algorithm that, on input 1^n, outputs (N, p, q) where $n = pq\ N = pq$ and p and q are n-bit primes (except with probability

negligible in n). Define the following Paillier encryption scheme:

(i) Gen: on input 1^n run GenModulus to obtain (N, p, q), the public key is $\mathrm{pk} = N$, the secret key is $\mathrm{sk} = (N, \varphi(N))$. The message space is $M = Z_N$

(ii) Enc: on input public key $\mathrm{pk} = N$ and a message $m \in M$, choose a uniform $r \leftarrow Z_N^*$ and output the ciphertext $c = [(1+N)^m \cdot r^N \bmod N^2]$

(iii) Dec: on input secret key $\mathrm{sk} = (N, \varphi(N))$ and a ciphertext c, compute

$$\mathrm{m} = \left[\frac{\left[\mathrm{c}^{\varphi(\mathrm{N})} \bmod \mathrm{N}^2 \right] - 1}{\mathrm{N}} \bullet \varphi(\mathrm{N})^{-1} \bmod \mathrm{N} \right]. \tag{8}$$

The Paillier encryption allows homomorphic computation of the addition on the ciphertexts. We have two messages $m_1, m_2 \in M$, the corresponding ciphertexts c_1 and c_2:

$$\mathrm{c}_1 = \mathrm{Enc}(\mathrm{pk}, \mathrm{m}_1) = \left[(1+\mathrm{N})^{\mathrm{m1}} \mathrm{r}_1^{\mathrm{N}} \mathrm{mod} \mathrm{N}^2 \right], \mathrm{c}_2 = \mathrm{Enc}(\mathrm{pk}, \mathrm{m}_2) = \left[(1+\mathrm{N})^{\mathrm{m1}} \mathrm{r}_2^{\mathrm{N}} \mathrm{mod} \mathrm{N}^2 \right]. \tag{9}$$

The evaluation is

$$\begin{aligned} \mathrm{Enc}(\mathrm{pk}, \mathrm{m}_1 + \mathrm{m}_2) &= (1+\mathrm{N})^{\mathrm{m1+m2}} \mathrm{r}^{\mathrm{N}} \mathrm{mod} \mathrm{N}^2 \\ &\hat{=} ((1+\mathrm{N})\hat{}(\mathrm{m1}) \mathrm{r\_}(1)(\hat{\mathrm{N}}) \bmod \mathrm{N}(\hat{2}))((1+\mathrm{N})\hat{}(\mathrm{m1}) \mathrm{r\_}(2)(\hat{\mathrm{N}}) \bmod \mathrm{N}(\hat{2})) \\ &= \mathrm{Enc}(\mathrm{pk}, \mathrm{m\_}(1)) \mathrm{Enc}(\mathrm{pk}, \mathrm{m\_}(2)) \bmod \mathrm{N}(\hat{2}) \\ &= \mathrm{c\_}(1) \mathrm{c\_}(2) \bmod \mathrm{N}(\hat{2}), \end{aligned} \tag{10}$$

where $r, r_1, r_2 \leftarrow Z_N^*$ and $r = r_1 r_2$.

This means that in equation (1), the operation $\bullet$ over the set M is addition and the operation $*$ over the set C is modular multiplication.

3.2.3. BGN Encryption. BGN encryption is a homomorphic encryption scheme. Unlike traditional RSA and Paillier partially homomorphic encryption schemes, the BGN homomorphic encryption supports both addition and multiplication. However, since homomorphic multiplication is implemented based on bilinear pairs, only one homomorphic multiplication can be performed in the homomorphic evaluation.

Let G be a PPT algorithm that, on input security parameter $\tau \in Z+$, outputs a tuple q_1, q_2, G, G_1, e where G, G_1 are groups of order $n = q_1 q_2$ and e: $G \times G \longrightarrow G_1$ is a bilinear map. Define the following BGN encryption scheme:

(i) Gen: on input security parameter τ run G to obtain a tuple q_1, q_2, G, G_1, e. Let $n = q_1 q_2$. R pick two random generators g, $u \leftarrow -G$ and set $h = uq_2$. Then, h is a random generator of the subgroup of G of order q_1. The public key is $\mathrm{pk} = (n, G, G_1, e, g, h)$ and the secret key is $\mathrm{sk} = q_1$.

(ii) Enc: we assume the message space M consists of integers in the set $\{0, 1, \ldots, T\}$ with $T < q_2$. On input public key $\mathrm{pk} = (n, G, G_1, e, g, h)$ and a message $m \in M$, choose a random $r \longrightarrow^R Z_n$. Compute and output a ciphertext

$$\mathrm{c} = \mathrm{g}^{\mathrm{m}} \mathrm{h}^{\mathrm{r}} \in \mathrm{G}. \tag{11}$$

(iii) Dec: on input secret key $\mathrm{sk} = q_1$ and a ciphertext c, compute

$$\mathrm{c}^{\mathrm{q1}} = \left(\mathrm{g}^{\mathrm{m}} \mathrm{h}^{\mathrm{r}}\right)^{\mathrm{q1}} = \left(\mathrm{g}^{\mathrm{q1}}\right)^{\mathrm{m}}. \tag{12}$$

The message can be recovered by using Pollard's lambda method to complete the discrete log of $c^{q}{}_1$ base $g^{q}{}_1$.

In addition, the system is clearly additively homomorphic. We have two messages $m_1, m_2 \in M$, the corresponding ciphertexts $c1$, $c2$:

$$c_1 = \mathbf{Enc}(pk, m_1) = g^{m1} h^{r1} \in G_{c_2} = \mathbf{Enc}(pk, m_2) = g^{m2} h^{r2} \in \mathrm{G}, \tag{13}$$

and $r1$, $r2 \overset{R}{\leftarrow} Zn$, the evaluation is

$$\mathbf{Enc}(pk, m_1 + m_2) = g^{m1+m2} h^{r1+r2} = g^{m1} h^{r1} g^{m2} h^{r2} = c_1 c_2. \tag{14}$$

Multiplication. More importantly, anyone can multiply two encrypted messages once using the bilinear map. Set $g_1 = e(g, g)$, $h_1 = e(g, h)$, $c = e(c_1, c_2)$ ${h_1}^r \in G_1$. Then,

$$c = e(c_1, c_2) h_1^r = e\left(g^{m1} h^{r1} g^{m2} h^{r2}\right) h_1^r g_1^{m1m2} h_1^{m1r2+r2m1+\alpha q2r1r2+r1} = g_1^{rm1m2} h_1^{\tilde{r}} = \mathrm{Enc}(pk, m_1 m_2), \tag{15}$$

where $\tilde{r} = m_1 r_2 + r_2 m_1 + \alpha q_2 r_1 r_2 + r$ is distributed uniformly in Z_n as required. This means that in equation (1), the operation $\bullet$ over the set M is addition and multiplication, we denote

$$c_1 \oplus c_2 = \mathrm{Enc}(pk, m_1 + m_2) = c_1 c_2,\ c_1 \otimes c_2 = \mathrm{Enc}(pk, m_1 m_2) = e(c_1, c_2) h_1^r = e(g, g)^{m1m2} h_1^{\tilde{r}}, \tag{16}$$

respectively.

4. Encryption of Controller

In this section, we first introduce the encrypted controller, give the basic concept, and then present two representative existing theoretical schemes. Then, we propose a new encrypted controller scheme based on BGN encryption. Finally, we compare the three schemes and list the advantages and disadvantages of the encrypted controller using BGN encryption.

4.1. Encrypted Controller. Consider a discrete-time linear controller of the following form:

$$f: \begin{cases} x(t+1) = Ax(t) + By(t), \\ u(t) = Cx(t) + Dy(t), \end{cases} \tag{17}$$

where A, B, C, and D are parameters of the controller, $x \in R^{nc}$ is a state of it, t is a step, $y \in R^{mc}$ is an input to the controller and $u \in R^{lc}$ is an output of the controller. Equation (13) is equivalently rewritten in the following:

$$\begin{bmatrix} x(t+1) \\ u(t) \end{bmatrix} = f(\Phi, \xi(t)) = \Phi \xi(t), \tag{18}$$

where the parameter Φ and the input ξ are as follows:

$$\Phi := \begin{bmatrix} A & B \\ C & D \end{bmatrix} \in R^{\alpha \times \beta}\ \xi := \begin{bmatrix} x \\ v \end{bmatrix} \in R^{\beta}, \tag{19}$$

with $\alpha := n_c + l_c$ and $\beta := n_c + m_c$.

Definition 1. Assumes that given a linear controller f in equation (2) for a NCS, controller's input v and output u are encrypted by an encryption scheme $E = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$. If there exists a map f_E such that an equation:

$$f_E\left(\mathrm{Enc}(k_p, \Phi), \mathrm{Enc}(k_p, \xi)\right) = \mathrm{Enc}\left(k_p, f(\Phi, \xi)\right), \tag{20}$$

holds, then f_E is an encrypted controller to f. Here, $\Phi \in M^{\alpha \times \beta}$, $\xi \in M^{\beta}$, and $f(\cdot) \in M$ plaintexts obtained from $\Phi \in R^{\alpha \times \beta}$, $\xi \in R^{\beta}$, and $f(\cdot) \in R^{\alpha}$ are the α, respectively.

Considering the practice of the encrypted controller, the key problem is: how to manipulate ciphertext to complete the original control operation. Kogiso proposed the first theoretical scheme to implement the encrypted controller using RSA and ElGamal encryption. The following describes the scheme of the encrypted controller using RSA.

4.2. Theoretical Scheme of Encryption Controller

4.2.1. Encrypted Controller Using RSA Encryption. RSA encryption is a partially homomorphic encryption scheme that can only support homomorphic multiplication. In order to ensure the completion of the control operation, Kogiso has modified the decryption algorithm Dec in the RSA scheme. According to equation (3), f is the multiplication operation between the parameter matrix Φ and the input vector ξ. Then, the operation f of the linear controller can be decomposed into the compound product of multiplication and addition:

$$f = f^{+} \circ f^{\times}. \tag{21}$$

Because

$$\Phi\xi = \Phi_1 \xi_1 + \Phi_2 \xi_2 + \cdots + \Phi_\beta \xi_\beta = \sum_{l=1}^{\beta} \Phi_l \xi_l, \tag{22}$$

we can define the following multiplication:

$$f^{\times}(\Phi, \xi) := \left[\Phi_1 \xi_1\ \Phi_2 \xi_2 \cdots \Phi_\beta \xi_\beta\right] =: \Psi, \tag{23}$$

and the addition is defined as

$$f^{+}(\Psi) := \sum_{l=1}^{\beta} \Psi_l, \tag{24}$$

where ξ_l denotes the l-th component of the column vector ξ, Φ_l denotes the l-th row vector of the matrix Φ, Ψ_l denotes the l-th row vector of the matrix Ψ, and β is the maximum row

number. The idea of an encrypted controller using RSA encryption is as follows:

(i) When designing the controller, the controller parameters Φ and controller initial state $x(0)$ is quantized, encrypted, and sent to the controller

(ii) The sensor collects signal y and encrypts it and then sends the ciphertext to the controller

(iii) The ciphertext homomorphic multiplication is carried out in the controller to obtain the intermediate result Ψ

(iv) The actuator block decrypts Ψ and then completes addition to obtain the final output and state and implement output $u(t)$

(v) The controller state is passed to and encrypted again in the sensor block and transmitted back to the controller to update the internal state of the controller

But the shortcomings of the scheme are exposed. The state transfer process unnecessarily increases the complexity of the control system and requires more network throughput. In addition, the controller needs to perform not only decryption but also addition operations, which also increases the capability requirements of the device.

The above-given idea is assumed in the ideal state, but in the actual scenario, the actuator cannot perform additional operations. As shown in Figure 3, after decryption at the actuator, the output of the controller needs to be sent back to the controller for addition so as to obtain the final output u and state x. Finally, u is sent to the actuator to act on the plant. This results in a significant increase in communication time compared to Figure 2. In addition, there are additional processes of plaintext transmission in the whole process, which greatly reduces the security of the whole scheme.

4.2.2. Encrypted Controller Using Paillier Encryption. Farokhi et al. implemented an encrypted discrete-time static feedback controller using Paillier encryption in [13]. Since Paillier encryption only supports homomorphic addition and homomorphic number multiplication, it can only support homomorphic multiplication of a plaintext and a ciphertext. Therefore, one of the inputs for controller parameters and sensor measurements will be plaintext. Consequently, there will be two schemes: one is the plaintext controller parameters, focusing on the output measurements protection and the other is the plaintext sensor measurements, which focuses on the protection of controller parameters. According to the idea in [13], we introduce using Paillier to encrypt linear controller (2) and focus on the protection of sensor measurement output. As shown in Figure 4, the operation flow of the scheme can be described as follows:

(i) When designing the controller, the controller parameters Φ are quantized and sent to the controller, and the controller initial state $x(0)$ is quantized, encrypted, and sent to the controller

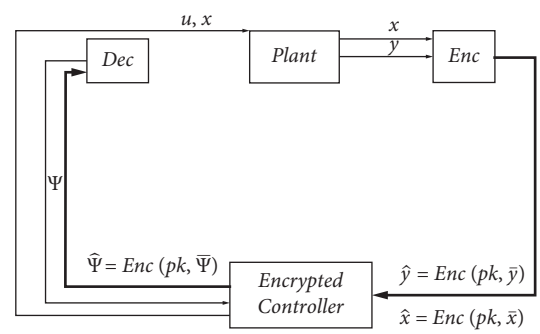

Figure 3: The schematic diagram of a networked system with RSA encryption.

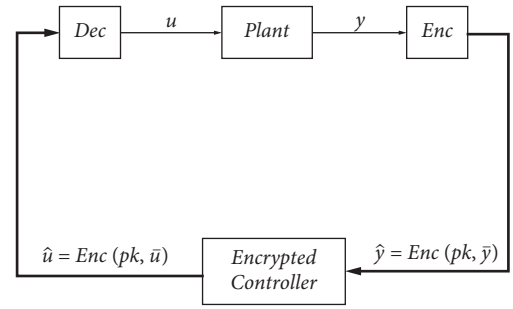

Figure 4: The schematic diagram of a networked control system with Paillier encryption.

(ii) The sensor collects signal y and encrypts it, and then sends the ciphertext to the controller

(iii) The ciphertext homomorphic operation is carried out in the controller to obtain the ciphertext output

(iv) The actuator block decrypts the output to obtain and implement $u(t)$

Compared with the former encrypted controller, operations are all completed in the encrypted controller using Paillier encryption. Therefore, it avoids the problem of the state transfer process and data insecurity in some channels. Unfortunately, one of the controller parameters and sensor measurements will be ciphertext in this scheme, which makes it impossible to protect all data simultaneously. However, both types of data are important and need to be protected at the same time in most scenarios. Therefore, in the following, we consider a scheme that can protect both types of data at the same time.

4.2.3. Encrypted Controller Using BGN Encryption. Unlike the previous two partially homomorphic encryption, BGN supports both homomorphic addition and homomorphic multiplication. Therefore, the operation of the controller can be completed in the encrypted controller, and the parameters and inputs of the controller are all ciphertexts.

Note that, the decryption takes polynomial time in the size if the message space T, and one can speed up decryption by precomputing a (polynomial-size) table of powers of g^q_1 so that decryption can occur in constant time. In this

scheme, we improve the efficiency of the scheme mainly from three parts: precomputation in encryption, parallel computation during homomorphic operation, and precomputation in decryption. Because BGN encryption can only support one homomorphic multiplication operation, the ciphertext of new state $x(t+1)$ generated in the encrypted controller cannot carry out the next round of homomorphic operation. To solve this problem, we transfer the state to the actuator for decryption and then encrypt it in the sensor and return it to the encrypted controller. In addition, because the ciphertext obtained by operation and encryption is not in the same form, we can not directly use the form of plaintext-ciphertext pair when precomputing to generate the decryption query table. We describe the optimization of each part in detail:

(1) The control system consists of the following discrete-time linear plant and the following type of linear controller. From these two functions and the initial states, we can determine the range of x and y is $\{n_1, n_1+1,\ldots, m_1\}$, and the range of x and u is $\{n_2, n_2+1,\ldots, m_2\}$.

(2) Encrypting all integers in the range of y and x to generate a plaintext-ciphertext pair $(m, \widehat{m})$ stored in Table 1 and calculating $e(g, g)^{q^1 m}$ for all the integers in the value range of u and x to obtain the number pairs $(m, e(g, g)^{q^1 m})$ and store them in Table 2.

(3) When the sensor collects the y of plant and encrypts it, it only needs to refer to Table 1 to get the corresponding ciphertext.

(4) When the controller performs a homomorphic operation on the ciphertext, it computes each multiplication part in parallel to shorten the operation time.

(5) When $\widehat{u}$ and $\widehat{x}$ are output to the actuator for decryption, calculate $\widehat{u}^{q^1}$ and $\widehat{x}^{q^1}$ and then look up Table 2 to get the corresponding plaintext values.

(6) State x is passed to the sensor and encrypted by looking up Table 1 and returned to the controller.

When using the BGN scheme to encrypt the controller, suppose that parameters Φ and signals ξ are bounded by some constant $T > 0$. They are rounded before encryption to ensure that every component of Φ and ξ is represented as an element of the message space M. Here, we denote the quantized parameters as $\xi(t) \in M^{\beta}$, $\Phi(t) \in M^{\alpha\times\beta}$. As shown in Figure 5, we introduce the specific scheme:

(i) When designing the controller, the controller parameters Φ and controller initial state $x(0)$ are encrypted to obtain $\widehat{A} = \mathrm{Enc}(\mathrm{pk}, A)$, $\widehat{B} = \mathrm{Enc}(\mathrm{pk}, B)$, $\widehat{C} = \mathrm{Enc}(\mathrm{pk}, C)$, $\widehat{D} = \mathrm{Enc}(\mathrm{pk}, D)$, $\widehat{x}(0) = \mathrm{Enc}(\mathrm{pk}, x(0))$ and then sends the ciphertext to the controller.

(ii) The sensor collects signal y and encrypts it to obtain $\hat{y} = \mathrm{Enc}(\mathrm{pk}, y)$, and then transmits $\widehat{y}$ to the controller.

(iii) The ciphertext homomorphic operation is carried out in the controller to obtain the ciphertext of the controller state and output and then pass them to the actuator. According to the BGNhomomorphism operation, (2) actually run in the cipher space after encrypted as

$$\begin{aligned} \widehat{x}(t+1) &= \widehat{A} \otimes \widehat{x}(t) \oplus \widehat{B} \otimes \widehat{y}(t), \\ \widehat{u}(t) &= \widehat{C} \otimes \widehat{x}(t) \oplus \widehat{D} \otimes \widehat{x}(t). \end{aligned} \tag{25}$$

(iv) The actuator block decrypts the output to obtain the plaintext: $u(t) = \mathrm{Dec}(\mathrm{sk}, \widehat{u}(t))$, $x(t+1) = \mathrm{Dec}(\mathrm{sk}, \widehat{x}(t+1))$ and implement $u(t)$. The state $x(t+1)$ is passed to the sensor.

(v) The sensor encrypts the state $\mathrm{x}(t+1)$ and sends it to the encryption controller.

Compared with Paillier encryption, the parameters and inputs of the encrypted controller using BGN encryption are all ciphertext, which can better protect the security of the control system. However, because of the limitation of BGN encryption only supports one multiplication, the new state $x(t+1)$ generated in the encryption controller cannot be directly carried out in the next round of homomorphic operation, and it needs to be decrypted first and then encrypted and be transmitted back to the controller. The additional state transfer process requires more channel capacity. For the encrypted controller using optimized BGN encryption, we analyze its advantages and disadvantages with RSA and Paillier encryption in the next section.

5. Attack Scenario and Security

The NCS has risks of eavesdropping attacks because the plant and controller communicate with each other via network links. The NCS with an encrypted controller which we proposed can resist eavesdropping attacks well and we briefly sketch it here. We consider an attack scenario to verify the security of the proposed scheme.

The main capabilities of A in our model are specified as follows:

(1) Adversary A can collect data within the communication channel through eavesdropping attacks

(2) Adversary A can collect data inside the controller through eavesdropping attacks

Note: the decryption machine, encryption machine, and actuator cannot be breached by attackers.

If adversary A can obtain controller parameters and signals in polynomial time, then we say that our scheme is not resistant to eavesdropping attacks; otherwise, we say that it is resistant to eavesdropping.

We analyze the security of the control system in this attack scenario. Adversary A collects the data in the controller and communication channel through eavesdropping. In our scheme, the data in the controller and the data in the communication channel are in ciphertext encrypted by the BGN encryption scheme. Adversary A needs to restore ciphertext to plaintext in order to get useful data. BGN encryption schemes satisfy semantic security under the

Table 1: Plaintext-ciphertext pairs generated by precomputation.

Index of plaintext	n_1	n_1+1	n_1+2	n_1+3	...	$m1$
Ciphertext	Enc (n_1)	Enc (n_1+1)	Enc (n_1+2)	Enc (n_1+3)	...	Enc (m_1)

Table 2: Number pairs generated by precomputation.

Index of ciphertext	$e\ (g,\ g)^{q1n2}$	$e\ (g,\ g)^{q1(n2+1)}$	$e\ (g,\ g)^{q1(n2+2)}$	$e\ (g,\ g)^{q1(n2+3)}$	...	$e\ (g,\ g)^{q1m2}$
Plaintext	n_2	n_2+1	n_2+2	n_2+3	...	$m2$

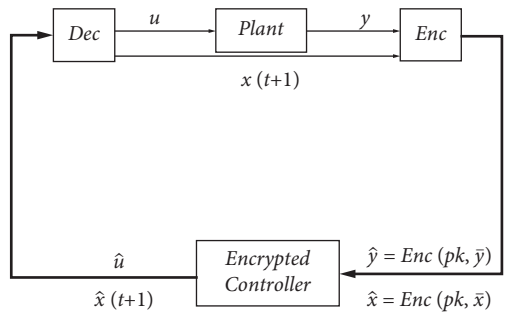

Figure 5: The schematic diagram of a networked control system with BGN encryption.

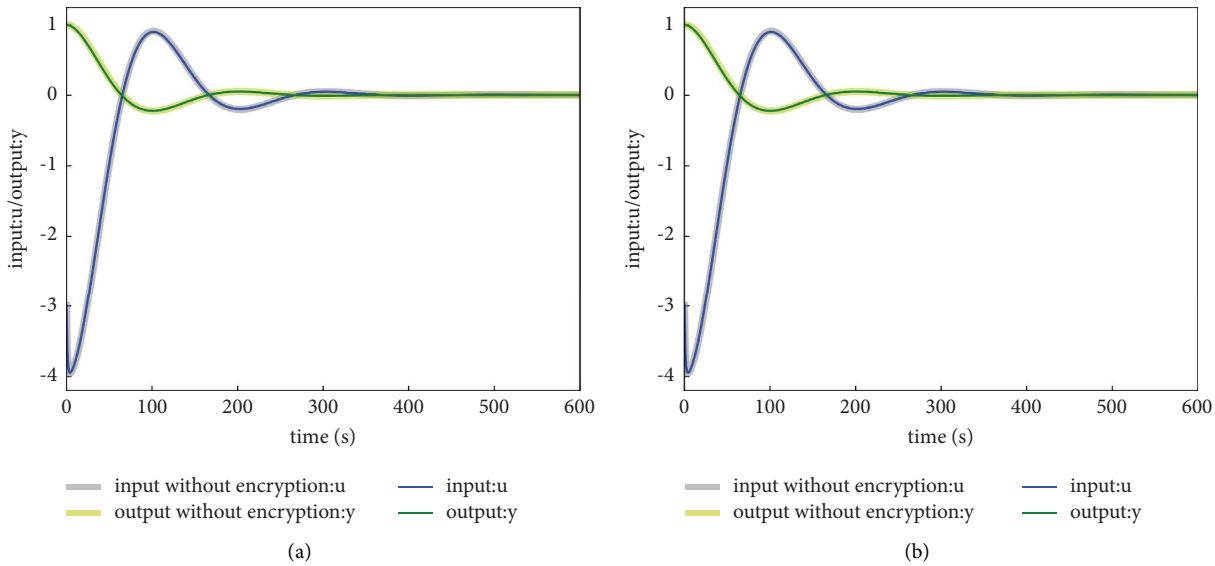

Figure 6: Continued.

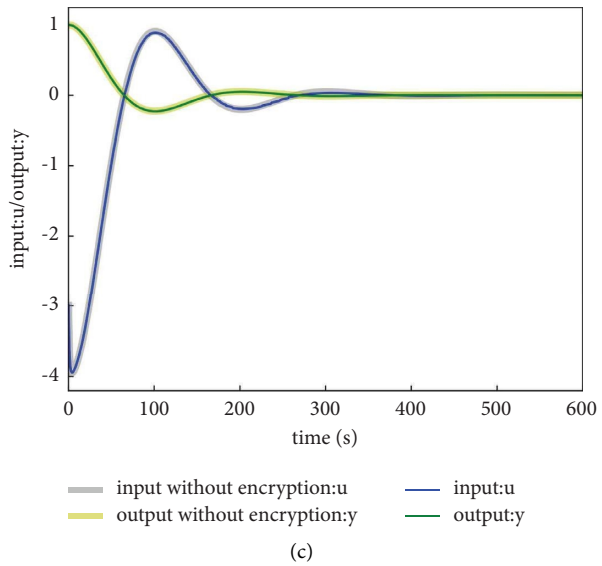

Figure 6: Comparision of output/input with and without proposed cyber-security enhancement. (a) RSA. (b) Paillier. (c) BGN.

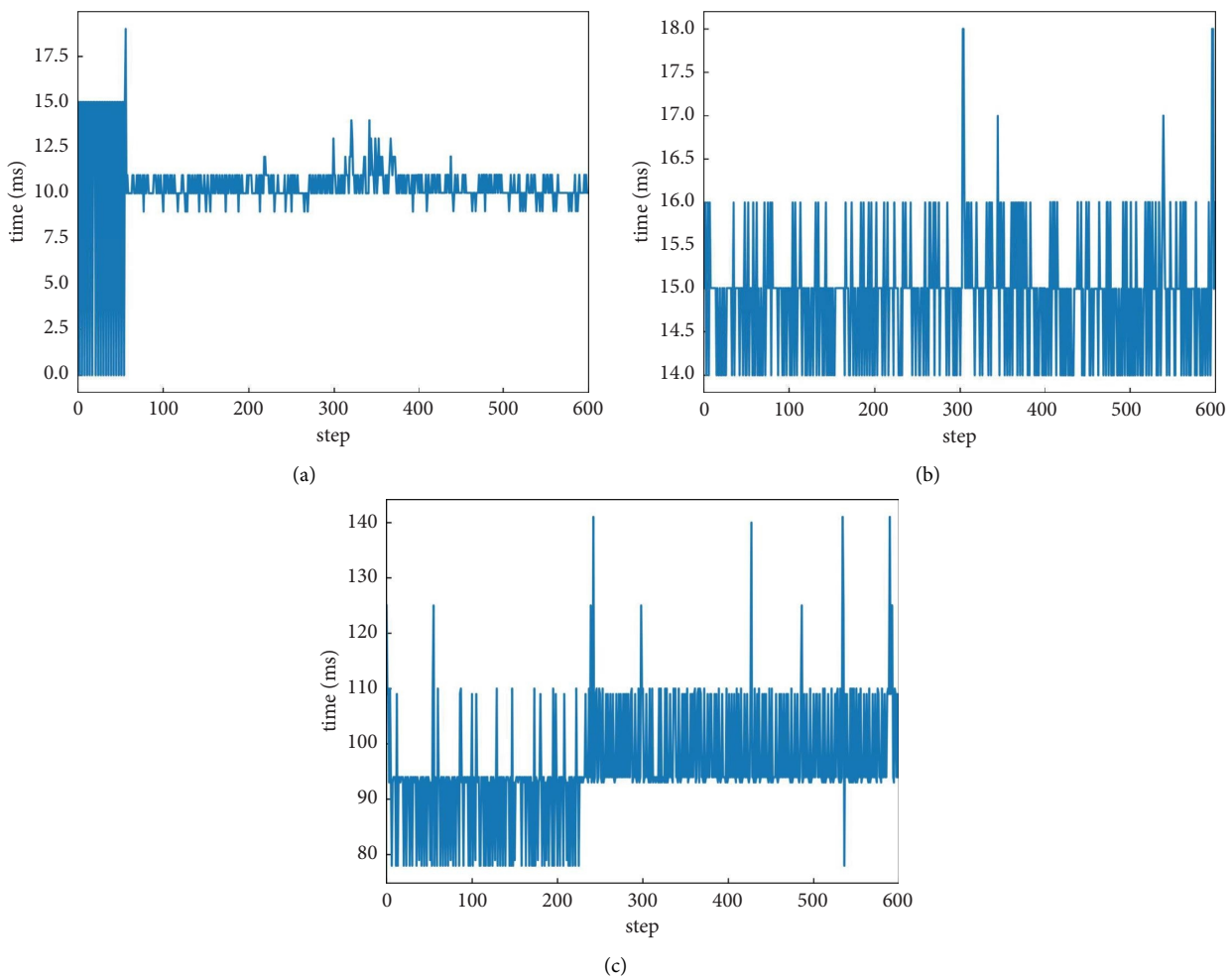

Figure 7: A time variation of each iteration calculation by encrypted controller with long sampling period to perform all tasks. (a) RSA. (b) Paillier. (c) BGN.

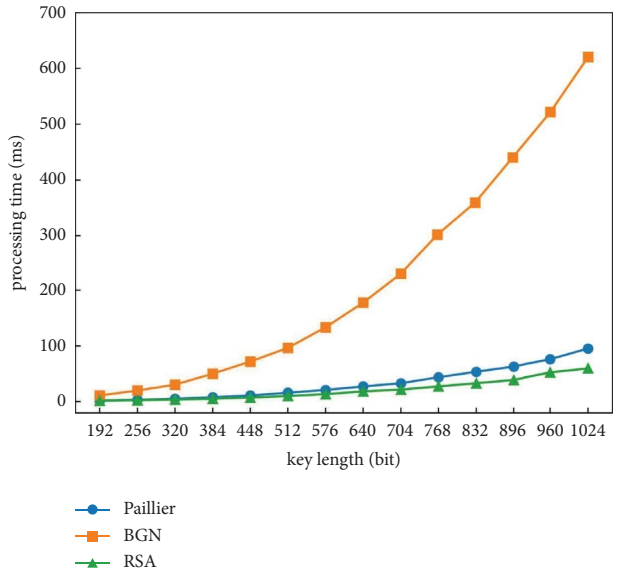

Figure 8: Comparision of the average processing time of three encrypted controllers.

Table 3: Analysis of three control systems.

Networked control system	With RSA	With Paillier encryption	With BGN encryption
Data in channel	Plaintext and ciphertext	Ciphertext	Ciphertext
Controller parameters	Ciphertext	Plaintext	Ciphertext
The average processing time (the key length is 512 bits)	10 + a ms	16 ms	96 ms
Additional state transfer	Yes	No	Yes
Precomputation	No	No	Yes

Table 4: Pros and cons of three control systems.

Networked control system	With RSA encryption	With Paillier encryption	With BGN encryption
Security (antieavesdropping)	Poor	Worse	Good
Efficiency	High	High	Low
Equipment requirements	Higher channel capacity	Normal	Higher channel capacity and controller performance

subgroup decision assumption, so cannot obtain useful plaintext data in polynomial time. Therefore, our scheme is resistant to eavesdropping attacks.

6. Numerical Example

According to the numerical example in [8], the control system consists of the following discrete-time linear plant and the following type of linear controller. The internal states of plant are $p_1(t)$, $p_2(t)$, which are satisfied.

$$\begin{aligned}\begin{bmatrix} p_1(t+1) \\ p_2(t+1) \end{bmatrix} &= \begin{bmatrix} 0.9999 & 0.0197 \\ -0.0197 & 0.97025 \end{bmatrix} \begin{bmatrix} p_1(t) \\ p_2(t) \end{bmatrix} \\ &\quad + \begin{bmatrix} 0.0000999 \\ 0.008508 \end{bmatrix} u(t), \\ y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} p_1(t) \\ p_2(t) \end{bmatrix}, \end{aligned} \tag{26}$$

where the initial states are $p_1(0) = 1$, $p_2(0) = 0$, and the internal states of linear controller are $x_1(t)$, $x_2(t)$, which is satisfied:

$$\begin{aligned}\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \end{bmatrix} &= \begin{bmatrix} 1 & 0.0063 \\ 0 & 0.3678 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0.0063 \end{bmatrix} y(t), \\ u(t) &= \begin{bmatrix} 10 & -99.9 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} - 3y(t). \end{aligned} \tag{27}$$

6.1. Numerical Results. We, respectively, use the above-given three algorithms to simulate the time response comparison of output y and control input u with decrypted plaintext y and u when the key length is 512 bits, as shown in Figure 6. Some small quantization errors can be seen in the control input responses of the three figures, but the quantization errors of the three encryption schemes are small enough to

be ignored. Figure 6(c) also shows that the encrypted controller with BGN encryption can achieve normal control, which ensures the control performance and control stability of the closed-loop system.

After the controller is encrypted with three encryption algorithms when the key length is 512 bits, the time variation of each iteration calculation of the control system is shown in Figure 7. By comparing the three figures, it can be seen that the encrypted controller using BGN encryption has a gap in efficiency compared with the previous two types of encryption, but the average processing time can reach within 100 ms, which is sufficient for the sampling period of the control system.

As the key length changes, the average processing time of the three encrypted controllers changes as shown in Figure 8. As the key length increases, the processing time of the encrypted controller using BGN encryption increases exponentially. Therefore, there is a trade-off between security and computing time. The encryption algorithm with a longer secret key is suitable for the control system.

7. Comparison of Three Schemes

We compare the three schemes in terms of data in channel and controller, average processing time and whether additional state transfer is needed, and so on, and a detailed explanation is given in the followup. Here, a represents the additional communication time required in the control system with RSA encryption. The average processing time of the three encryption algorithms is shown in the next section.

It is not difficult to find from Table 3 that in RSA encryption and Paillier encryption schemes, the controller parameters and the data transmitted in the channel during the process from the sensor to actuator cannot be guaranteed to be ciphertext, which indicates that they are not ideal in security. But the BGN encryption scheme can avoid this problem; both controller parameters and input/output data are ciphertexts. In terms of efficiency, RSA encryption is somewhat less efficient considering the extra communication time, but even so, RSA encryption and Paillier encryption are more efficient than BGN. The good thing is the optimized average processing time of BGN encryption is also suitable for the sampling period of the control system. Both RSA encryption and BGN encryption require additional state transfer and therefore require more network throughput channel capacity. From the last row of the table, we use the method of precomputation to shorten the encryption and decryption time, but the precomputation itself needs a lot of computation. In addition, according to our specific scheme, the controller adopts the method of parallel calculation of each multiplication part to shorten the computation time, but this will have certain requirements on the performance of the controller.

We give a more intuitive comparison of the pros and cons of the three schemes in Table 4.

Although BGN encryption is not perfect, it is good enough to keep data secure to tolerate these minor flaws.

8. Conclusion and Open Problem

8.1. Conclusion. In this paper, we proposed a scheme to encrypt the controller using BGN homomorphic encryption scheme. We optimized the BGN encryption scheme to fit the sampling period of the control system, 15.

And, how to use the optimized encryption scheme to implement the encrypted controller is presented. We compared the scheme with existing encrypted controllers using RSA and Paillier encryption. The most remarkable property of the proposed scheme is the encryption protection of the data in the channel and controller. At the same time, we verify that the encrypted controller can achieve normal control.

8.2. Open Problem. In addition to the antieavesdropping attacks we mentioned in Section 5, encryption controllers involving authentication (e.g., [28, 29]) are also an interesting topic that we will study in the future.

Data Availability

The data used to support the study are included in the paper.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Key R&D Plan of Shandong Province (Grant no. 2021RZB01002), China and Major Innovation Project of Science, Education and Industry of Shandong Academy of Sciences (Grant no. 800421020220220009), China.

References

[1] H. Sandberg, S. Amin, and K. Johansson, "Cyberphysical security in networked control systems: an introduction to the issue," *IEEE Control Systems Magazine*, vol. 35, no. 1, pp. 20–23, 2015.

[2] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, 2007.

[3] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, "A secure control framework for resource-limited adversaries," *Automatica*, vol. 51, pp. 135–148, 2015.

[4] L. Zhang, H. Gao, and O. Kaynak, "Network-induced constraints in networked control systems-A survey," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 403–416, 2013.

[5] M. Yampolskiy, T. R. Andel, and J. T. Mcdonald, "Intellecutal protection in additive layer manufacturing: requirements for secure outsouring," in *Proceedings of the Program Protection and Revese EngineeringWorkshop ACM*, pp. 1–9, Los Angeles CA USA, December 2014.

[6] D. S. Wall and M. Yar, "Intellecutal property crime and the Internet: cyber-piracy and "stealing" intangibles," *Handbook of Internet Crime*, pp. 255–272, Routledge, Oxfordshire, England, UK, 2010.

[7] S. Mclaughlin, "On dynamic malware payloads aimed at programmable logic controllers," in *Proceedings of the Usenix Conference on Hot Topics in Security. USENIX Association*, p. 10, San Francisco CA, August 2011.

[8] K. Kosigo and T. Fujita, "Cyber-security enhancement of networked control systems using homo- morphic encryption," in *Proceedings of the 54th IEEE Conference on Decision and Control*, pp. 6836–6843, Osaka, Japan, December 2015.

[9] R. L. Rivest, L. Adleman, and M. Deryouzos, "On data banks and privacy homomorphisms," *Foundations of Secure Computation*, vol. 19, pp. 169–180, 1978.

[10] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[11] T. E. Gamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *Proceedings of CRYPTO*, vol. 196, pp. 10–18, 1984.

[12] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," *Ad- vances in Cryptology-Eurocrypt'99*, vol. 1592, pp. 223–238, 1999.

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

[14] Y. Lin, F. Farokhi, and I. Shames, "Secure control of nonlinear systems using semi- homomorphic encryption," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 5002–5007, Miami, FL, USA, December 2018.

[15] C. Murguia, F. Farokhi, and I. Shames, "Secure and private implementation of dynamic controllers Using semi-homomorphic encryption," *IEEE Transactions on Automatic Control*, vol. 65, no. 9, pp. 3950–3957, 2020.

[16] J. Tran, F. Farokhi, and M. Cantoni, "Implementing homomorphic encryption based secure feedback control," *Control Engineering Practice*, vol. 97, pp. 1043501–10435012, 2020.

[17] J. Kim, C. Lee, H. Shim et al., "Encrypting controller using fully homomorphic encryption for security of cyber-physical Systems∗∗The work of J. Kim, C. Lee, and H. Shim was supported by ICT R & D program of MSIP/IITP grant number 14-824-09-013, resilient cyber-physical systems research. The work of J. H. Cheon, A. Kim, M. Kim, and Y. Song was supported by IT R & D program of MSIP/KEIT [No. 0450-21060006] and samsung electronics Co., ltd. (No. 0421-20150074) security of cyber-physical systems," *IFAC-PapersOnLine*, vol. 49, no. 22, pp. 175–180, 2016.

[18] D. Boneh, E. J. Goh, and K. Nissim, *Evaluating 2-DNF Formulas on Ciphertexts*, Springer, Berlin, Germany, 2005.

[19] D. Wang, W. Li, and P. Wang, "Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks access in industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4081–4092, 2018.

[20] Q. Wang and D. Wang, "Understanding failures in security proofs of multi-factor authentication for mobile devicesfor mobile devices," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 597–612, 2023.

[21] Y. Yu, G. Xu, and X. Wang, "Provably secure NTRU instances over prime cyclotomic rings," *PublicKey Cryptography*, vol. 23, pp. 409–434, 2017.

[22] M. Schulze Darup, A. Redder, and D. E. Quevedo, "Encrypted cooperative control based on struc- tured feedback," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 37–42, 2019.

[23] A. B. Alexandru, M. Schulze Darup, and G. J. Pappas, "Encrypted cooperative control revisited," in *Proceedings of the 2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 7196–7202, Nice, France, December 2019.

[24] Y. Lu and M. Zhu, "Privacy preserving distributed optimization using homomorphic encryption," *Automatica*, vol. 96, pp. 314–325, 2018.

[25] M. S. Darup, A. Redder, and D. E. Quevedo, "Encrypted cloud-based MPC for linear systems with input constraints input constraints," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 535–542, 2018.

[26] A. B. Alexandru, M. Morari, and G. J. Pappas, "Cloud-based MPC with encrypted data," in *Proceedings of the IEEE Conference on Decision and Control (CDC)*, pp. 5014–5019, Miami, FL, USA, December 2018.

[27] M. S. Darup, "Encrypted MPC based on ADMM real-time iterations," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 3508–3514, 2020.

[28] S. Qiu, D. Wang, and G. Xu, "Practical and provably secure three-factor authentication protocol based on extended chaotic-maps for mobile lightweight devices," *IEEE Transactionson 2022 on Dependable and Secure Computing*, vol. 19, pp. 1–14, 2022.

[29] D. Wang and P. Wang, "Two-factor authentication with security beyond conventional bound," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, pp. 708–722, 2018.