applied sciences

MDPI

Article

Secure Control of Linear Controllers Using Fully Homomorphic Encryption

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

[1] College of Information Science and Engineering, Ocean University of China, Qingdao 266100, China; panjsh@sdas.org (J.P.); weizhq@sdas.org (Z.W.)
[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; liuwen@jnist.cn
[4] Faculty of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Science), Jinan 250102, China; sui_tongtong@163.com (T.S.); konglingrui125@163.com (L.K.); zy13296440360@163.com (Y.Z.)
[5] Quancheng Laboratory, Jinan 250100, China
* Correspondence: wangjzh@sdas.org

Abstract: In actual operation, there are security risks to the data of the network control system, mainly in the form of possible eavesdropping of signals in the transmission channel and parameters in the controller leading to data leakage. In this paper, we propose a scheme for encrypting linear controllers using fully homomorphic encryption, which effectively removes these security risks and substantially improves the security of networked control systems. Meanwhile, this paper uses precomputation to handle data encryption, which eliminates the encryption time and solves the drawback of fully homomorphic encryption that it is difficult to apply due to the efficiency problem. Compared to previous schemes with precomputation, for the first time, we propose two methods to mitigate the problem of the slight security degradation caused by precomputation, which makes our scheme more secure. Finally, we provide numerical simulation results to support our scheme, and the data show that the encrypted controller achieves normal control and improves safety and efficiency.

Keywords: encrypted controller; networked control system; fully homomorphic; BFV encryption

Citation: Pan, J.; Sui, T.; Liu, W.; Wang, J.; Kong, L.; Zhao, Y.; Wei, Z. Secure Control of Linear Controllers Using Fully Homomorphic Encryption. *Appl. Sci.* **2023**, *13*, 13071. https://doi.org/10.3390/app132413071

Academic Editors: Andrea Prati, Peter R.J. Trim and Yang-Im Lee

Received: 18 October 2023
Revised: 27 November 2023
Accepted: 4 December 2023
Published: 7 December 2023

1. Introduction

As control technology, network communication technology, and computer technology advance, the network control system (NCS) also advances steadily. It is a feedback control system which realizes a closed-loop control through the control network. The key benefits of a network control system include less system connection, a high dependability, a flexible structure, a simple system extension, and the possibility to implement resource sharing for information. This has led to its widespread application in key infrastructure such as water, transportation, and power. However, NCSs are not completely secure [1], if a malicious user has invaded the controller without authorization, it can lead to the leakage of important information of the control system, which can make infrastructure failures such as power plants sustain huge failures and losses [2–4]. Therefore, the security of network control systems is becoming increasingly important and has attracted the attention of researchers.

The traditional antieavesdropping method is communication encryption, as shown in Figure 1a, which is to encrypt the data sampled by the sensor to hide the data. This is equivalent to putting a lock on the data, and it is difficult for a malicious attacker without a corresponding key to open the lock and eavesdrop on the data. However, this also prevents the controller from operating on the locked data. It is necessary to decrypt the data into plaintext after transmission to the controller, and then the computation of the plaintext data is completed in the controller. Then, the computed signal is encrypted by

the controller and transmitted to the actuator to perform decryption. However, in this process, this conventional communication encryption not only requires two encryptions and decryptions, but the data in the controller are in plaintext as well, and thus does not protect this part of the data from eavesdropping. Kogiso et al. [5] proposed the concept of encrypted controller in 2015 to make up for the deficiency of communication encryption. The ideal encryption controller can directly calculate the encrypted data. As shown in Figure 1b, the data exist in ciphertext throughout the network control loop, and a malicious attacker would have no way to get at it. In this way, the encryption controller greatly improves the security of the data in the NCS compared to the NCS without the encrypted controller.

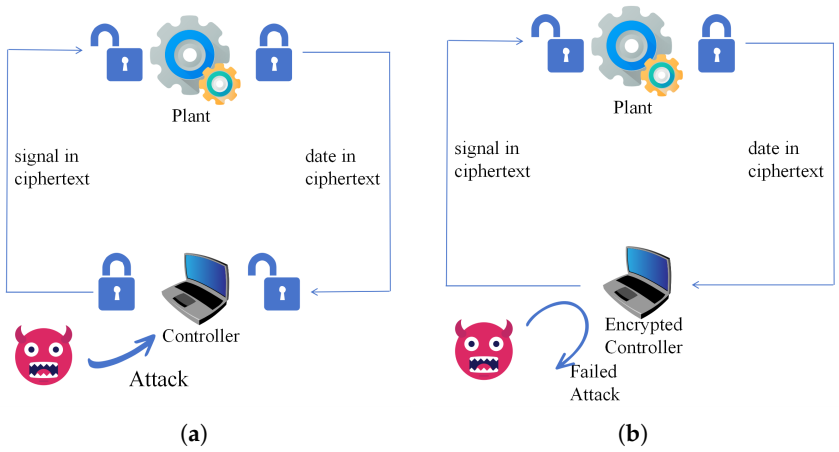


Figure 1. The NCS (**a**) without encrypted controller; (**b**) with encrypted controller.

However, the concrete encryption controller scheme proposed by Kogiso et al. [5] was implemented by RSA [6] and ElGamal [7] partially homomorphic encryption. Subsequent research studies on encrypted controllers have also mostly been conducted on partially homomorphic encryption controllers, including a series of research studies based on encrypted controllers [8–11] with Paillier [12] encryption and ElGamal encryption. Homomorphic encryption allows direct operations on ciphertext. Partially homomorphic encryption is homomorphic encryption that supports only one operation in addition or multiplication. Therefore, it may not be possible to complete the operation in the encrypted controller using partially homomorphic encryption. This leads to the fact that operations inside the encrypted controller are often guaranteed at the expense of data security. Therefore, can we implement the encrypted controller using a fully homomorphic encryption scheme? For the first time, a fully homomorphic encrypted controller was shown by Kim et al. [13]. But the scheme faced two problems. On the one hand, they suggested running multiple controllers and a catch-up mechanism to solve the problem because the controller could not function while the bootstrapping of encrypted variables was performed. On the other hand, the finite life of the encrypted variable was reduced with the operation. To solve this problem, a tree-based algorithm was introduced. But this also led to an increased complexity of the control systems.

In this paper, we propose a encrypted controller scheme that is more secure and efficient while less complex. The highlights of this paper are as follows. Firstly, we adopt a fully homomorphic encryption scheme [14,15] proposed by Brakerski, Fan, and Vercauteren to encrypt the controller in this paper, which we usually refer to as BFV encryption. Secondly, we use the method of generating tables by precomputation [16] to improve encryption efficiency. Thirdly, we propose to continuously update the table to improve security. Finally, we describe the attack scenarios [17–19] in this study and discuss the security of the scheme.

Here is the structure for the rest of the article. In Section 2, we present the related work. In Section 3, we introduce the mathematical symbols and some basic knowledge. In

Section 4, we introduce our proposed encrypted controller using BFV encryption and give two methods to improve security and analyze the security of the scheme. In Section 5, a numerical example is given to demonstrate that the encrypted controller can implement regular control and to verify that the precomputation saves time. We discuss the findings, implications, and some limitations of this paper in Section 6, and in Section 7, we summarize the paper, presenting the advantages of the scheme and outlining the current shortcomings.

2. Related Work

Homomorphic encryption is a form of encryption that is capable of performing computations on encrypted data, and its research can be traced back to the idea of homomorphic encryption proposed by Rivest in [20]. The idea can be described as the ability to directly perform functions on the ciphertext without knowing the private key under an encryption scheme with homomorphic properties. For a long time thereafter, partially homomorphic encryption, where only one of the operations of addition and multiplication can be performed on the ciphertext, developed by leaps and bounds, e.g., RSA and ElGamal are multiplicative homomorphic encryption schemes, and Paillier is an additive homomorphic encryption scheme. But there has been no breakthrough in homomorphic encryption schemes that can support both additive and multiplicative operations. It was not until 2009 that Gentry [21] first proposed a fully homomorphic encryption scheme based on an ideal lattice, which allowed anyone without a private key to perform any valid computable function on the encrypted data. According to different construction ideas, fully homomorphic encryption can be roughly classified into three categories: the first category is the fully homomorphic schemes constructed based on the hard problem on an ideal lattice, which is represented by the scheme proposed by Gentry in [21] and its improvement [22]; the second category is the fully homomorphic scheme constructed based on the (R)LWE problem, which is represented by the scheme proposed by Brakershi et al. [23,24], which has improved efficiency compared to the first category, and the fully homomorphic encryption proposed in [14,15] used in this paper belongs to this category; the third category is the fully homomorphic encryption scheme that does not require any key exchange, and this category of schemes is represented by the scheme proposed by Gentry et al. in [25]. As we all know, fully homomorphic encryption is more secure but inefficient compared to partially homomorphic encryption. The efficiency of fully homomorphic encryption has been greatly improved in recent years, such as GSW encryption [25] and BGV encryption [24] in the second and third categories; both of them are more efficient fully homomorphic encryption schemes, with an encryption time reaching the *ms* level. However, this time-consuming aspect of the control system cannot be ignored. The two methods we propose improve efficiency and ensure that the security of the program is not compromised. In terms of efficiency, the time spent on encryption is completely eliminated compared to existing encryption schemes. This results in a significant increase in efficiency compared to existing fully homomorphic encryption schemes. In terms of security, the table used for encryption is constantly updated, so the scheme still maintains the high security of homomorphic encryption.

The above is the research work on homomorphic encryption, and with the rise of NCSs, there has been a focus on using homomorphic encryption as a tool to improve the security of networked control systems. Homomorphic encryption was first used for NCSs in [5], where two partially homomorphic encryption schemes, RSA and ElGamal, were used in the method. Paillier was subsequently proposed to be used for NCSs. Recent studies [26–29] have proposed many ways to further improve and optimize these schemes, such as maintaining stability and performance. Among them, ref. [26] proposed to update the key pair and ciphertext by simple update rules and modulo operations at each sampling cycle, which brought some inspiration and reference to this paper. In this paper, we apply in-cycle updating of plaintext–ciphertext pairs in a fully homomorphic encrypted controller scheme to improve the security of the scheme.

The application of fully homomorphic encryption to NCSs has been late and rare because of efficiency issues in real-world applications. Ref. [13] considers the application of fully homomorphic encryption to NCSs to alleviate the extra overhead and quantization errors caused by quantization recovery. Subsequently, ref. [16] proposed to use a non-strictly fully homomorphic encryption scheme for encryption and performed optimization. We refer to the method of [16] and propose a new fully homomorphic encryption scheme. The scheme performs well in terms of security and efficiency compared to existing schemes using fully homomorphic encryption. Specifically, the security is comparable to [13] and much higher than [16], and the efficiency is much higher than [13,16]. In addition, compared with [13], our scheme does not require multiple controllers, so the control system is simpler, which is more favorable for applications in practice. In addition to this, another popular control scheme involving optimization is model predictive control, and [30–32] consider a model predictive control scheme for related linear systems. For some of the current challenges, ref. [33] outlines them accordingly. However, since our scheme already involves a large amount of computation, we do not use this technique in this paper, but in the future, we will consider using model predictive control for our scheme.

3. Preliminaries

The paper makes use of the following notions.

In this paper, we use bold uppercase letters (e.g., $\mathbf{A}$, $\mathbf{B}$) to denote matrices and, similarly, lowercase letters (e.g., $\mathbf{a}$, $\mathbf{b}$) to denote column vectors. We use $\mathbb{R}$ to denote the set of real numbers; thus, $z \in \mathbb{R}$ is a real number. If z_1 is the closest integer to z, then we denote z_1 by $\lfloor z \rceil$, which means it is the only integer in the half open interval $(z - 1/2, z + 1/2]$.

We identify $\mathbb{Z} \bigcap (-q/2, q/2]$ as a representation of $\mathbb{Z}_q$ for an integer q and use $[z]_q$ or $r_p(z)$ to indicate the interval into which the integer z modulo q is reduced. To represent the sampling of x based on a distribution D, we use the notation $x \leftarrow D$. When D is a finite set, it means sampling from the uniform distribution over D.

Along with the explanations of the aforementioned symbols, we also provide some definitions of the fundamental terms that will be used throughout the remainder of this paper.

3.1. RLWE Problem

The RLWE problem is the underlying mathematically difficult problem of securing cryptographic methods. Before introducing the RLWE problem, it is necessary to familiarize oneself with some of the notations in the definition that follows.

Let $\Phi_M(X)$ be the Mth cyclotomic polynomial of degree $N = \phi(M)$ for a positive integer M. Let $R = \mathbb{Z}[X]/(\Phi_M(X))$ be the ring of integers in the $Q[X]/(\Phi_M(X))$ number field. For the residue ring of R modulo an integer q, we write $R_q = R/qR$. We write $R_q^\vee = R^\vee/qR^\vee$, where $R^\vee$ is the dual fractional ideal of R. For a positive integer modulus of $q \geq 2$, $s \in R_q^\vee$, $r \in (\mathbb{R}^+)^N$, and an error distribution of $\chi := \lfloor \Psi_r \rceil_{R^\vee}$.

Definition 1 ([34]). *(Ring learning with errors (RLWE) distribution) We define $A_{N,q,\chi}(s)$ as the* **RLWE** *distribution that is formed by uniformly sampling $a \leftarrow R_q$ at random, $e \leftarrow \chi$ and returning $(a, a \cdots + e) \in R_q \times R_q^\vee$.*

Definition 2 ([35]). *((Decision) RLWE) The (decision) RLWE, denoted by $\mathbf{RLWE}_{N,q,\chi}(\mathcal{D})$, is the problem of distinguishing arbitrarily many independent samples chosen according to $A_{N,q,\chi}(s)$ for a random choice of s sampled from the distribution $\mathcal{D}$ over $R^\vee$ from the same number of uniformly random and independent samples from $R_q \times R_q^\vee$.*

3.2. Fully Homomorphic Encryption

Fully homomorphic encryption plays an important role in this paper, and it is defined as follows.

Definition 3. *A fully homomorphic encryption scheme* **FHE** $=$ $(\mathbf{Gem}, \mathbf{Enc}, \mathbf{Dec}, \mathbf{Eval})$ *is described as follows:*

- $\mathbf{Gen}(1^\lambda) \rightarrow (pk, sk, evk)$*: input security parameter* λ*, output* (pk, sk, evk) *where* pk *is a public key,* sk *is a secret key, and* evk *is an evaluation key.*
- $\mathbf{Enc}(m, pk) \rightarrow c$*: input message* m *and public key* pk*, output ciphertext* c*.*
- $\mathbf{Dec}(c, sk) \rightarrow m'$*: input ciphertext* c *and secret key* sk*, compute and output the plaintext* m'*.*
- $\mathbf{Evaluation}(f, c_1, c_2, \cdots, c_N, evk) \rightarrow c^*$*: input a set of ciphertexts* $(c_1, c_2, \cdots, c_N)$*, function* f*, and evaluation key* evk*. Compute and output the evaluation ciphertext* c^**.*

Fully homomorphic encryption can be performed on data in the form of ciphertexts of arbitrary complexity, which we describe more intuitively in Figure 2 below. A set of data $(m_1, m_2, \cdots, m_N)$ is encrypted with the algorithm **Enc** to obtain a set of ciphertexts $(c_1, c_2, \cdots, c_N)$, and an arbitrary computation f is performed on the ciphertext set to obtain c^*. The value of c^* after decryption should be the same as the value of the calculation done directly on the plaintext.

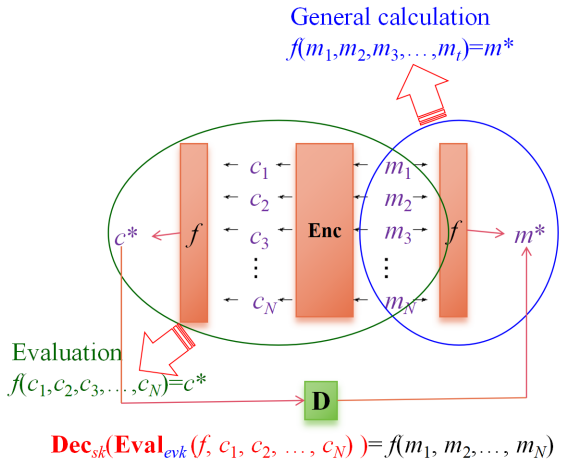


Figure 2. Graphical representation of fully homomorphic encryption

3.3. BFV Encryption

It is assumed that the security-parameter-related noise distribution χ is a discrete Gaussian distribution on the ring R, and that the uniform random noise distribution χ' is also on the ring R. The seven probabilistic polynomial time (*PPT*) algorithms (**SecretKeyGen**, **PublicKeyGen**, **EvaluateKeyGen**, **Enc**, **Dec**, **Add**, **Mult**) used in BFV encryption are as follows:

- $\mathbf{SecretKeyGen}(1^\lambda)$: input security papameter λ, sample $\mathbf{s} \leftarrow R_2$, and output secret key noted as $sk = \mathbf{s}$.
- $\mathbf{PublicKeyGen}(sk)$: input secret key, sample $\mathbf{a} \leftarrow R_q$, $\mathbf{e} \leftarrow \chi$, and output public key
$$pk = ([-(\mathbf{a} \cdot \mathbf{s} + \mathbf{e})]_q, \mathbf{a}).$$
- **EvluateKeyGen**:
  - Version 1: parameters (sk, T): for $i = 0, \cdots, l = \lfloor log_T(q) \rfloor$, sample $\mathbf{a_i}, R_q, \mathbf{e_i} \leftarrow \chi$, perform the following operation, and return
$$rlk = [([-(\mathbf{a_i} \cdot \mathbf{s} + \mathbf{e}_i) + T^i \cdot \mathbf{s}^2]_q, \mathbf{a}_i) : i \in [0 \cdots l]].$$
  - Version 2: parameters (sk, p): sample vectors $\mathbf{a} \leftarrow R_{p \cdot q}$, $\mathbf{e} \leftarrow \chi'$, and then return
$$rlk = ([-(\mathbf{a} \cdot \mathbf{s} + \mathbf{e} + p \cdot \mathbf{s}^2]_{p \cdot q}, \mathbf{a}).$$
- $\mathbf{Enc}(pk, \mathbf{m})$: to encrypt a message $m \in R_t$. We set $\mathbf{p}_0 = pk[0]$, $\mathbf{p}_1 = pk[1]$ and sample $\mathbf{u} \in R_2, \mathbf{e}_1, \mathbf{e}_2 \in \chi$, then return the final ciphertext

$$ct = ([\mathbf{p}_0 \cdot \mathbf{u} + \mathbf{e}_1 + \Delta \cdot \mathbf{m}]_q, [\mathbf{p}_1 \cdot \mathbf{u} + \mathbf{e}_2]_q).$$

- **Dec**(sk, ct): set $\mathbf{c}_0 = ct[0], \mathbf{c}_1 = ct[1]$ and compute and output the result of the decryption

$$m' = [\lfloor \frac{t \cdot [\mathbf{c}_0 + \mathbf{c}_1 \cdot \mathbf{s}]_q}{q} \rceil]_t.$$

- **Add**(ct_1, ct_2): return

$$([ct_1[0] + ct_2[0]]_q, [ct_1[1] + ct_2[0]]_q),$$

where $ct_1 = \mathbf{Enc}(pk, \mathbf{m_1}), ct_2 = \mathbf{Enc}(pk, \mathbf{m_2})$.

- **Mult**(ct_1, ct_2): compute

$$\mathbf{c}_0 = [\lfloor \frac{t \cdot (ct_1[0] \cdot ct_2[0])}{q} \rceil]_q$$

$$\mathbf{c}_1 = [\lfloor \frac{t \cdot (ct_1[0] \cdot ct_2[1] + ct_1[1] \cdot ct_2[0]}{q} \rceil]_q$$

$$\mathbf{c}_2 = [\lfloor \frac{t \cdot (ct_1[1] \cdot ct_2[1])}{q} \rceil]_q$$

  - Relinearization version 1: rewrite $\mathbf{c}_2$ equivalently to be based on T, i.e., write $\mathbf{c}_2 = \sum_{i=0}^{l} \mathbf{c}_2^{(i)} T^i$ with $\mathbf{c}_2^{(i)} \in R_T$ and set

$$\mathbf{c}_0' = [\mathbf{c}_0 + \sum_{i=0}^{l} rlk[i][0] \cdot \mathbf{c}_2^{(i)}]_q \text{ and } \mathbf{c}_1' = [\mathbf{c}_1 + \sum_{i=0}^{l} rlk[i][1] \cdot \mathbf{c}_2^{(i)}]_q.$$

    Return $(\mathbf{c}_0', \mathbf{c}_1')$.
  - Relinearization version 2: compute

$$(\mathbf{c}_{2,0}, \mathbf{c}_{2,1}) = ([\lfloor \frac{\mathbf{c}_2 \cdot rlk[0]}{p} \rceil]_q, [\lfloor \frac{\mathbf{c}_2 \cdot rlk[1]}{p} \rceil]_q),$$

    and return $([\mathbf{c}_0 + \mathbf{c}_{2,0}]_q, [\mathbf{c}_1 + \mathbf{c}_{2,1}]_q)$.

For a better understanding, the following Figure 3 represents the whole process of fully homomorphic encryption.

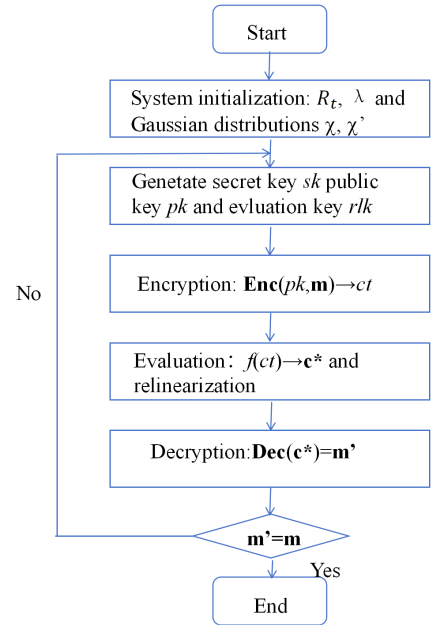


Figure 3. The process of fully homomorphic encryption.

3.4. Encrypted Controller

The discrete-time linear controller case that is under consideration in this work is summarized in the following form:

$$f : \begin{cases} \mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{y}(t) \\ \mathbf{u}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{y}(t), \end{cases} \tag{1}$$

where $\mathbf{y}(t) \in \mathbb{R}^m$ is a controller input (or a plant output), $\mathbf{u}(t) \in \mathbb{R}^l$ is a controller output (or a plant input), and t is a step. $\mathbf{A}, \mathbf{B}, \mathbf{C}$, and $\mathbf{D}$ are controller parameter values. The following is an equivalent rewriting of Equation (1):

$$\begin{bmatrix} \mathbf{x}(t+1) \\ \mathbf{u}(t) \end{bmatrix} = f(\mathbf{\Phi}, \boldsymbol{\xi}(t)) = \mathbf{\Phi}\boldsymbol{\xi}(t), \tag{2}$$

where the parameter $\mathbf{\Phi}$ and the input $\boldsymbol{\xi}$ are represented in the following form:

$$\mathbf{\Phi} := \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix} \in \mathbb{R}^{\alpha \times \beta}, \qquad \boldsymbol{\xi} := \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} \in \mathbb{R}^{\beta}.$$

with $\alpha := n + l$ and $\beta := n + m$.

Definition 4 ([27]). *For an NCS, we assume that given a linear controller f in (1) for an NCS, the controller's input y and output u are encrypted using the encryption algorithm $E = (\mathbf{Gen}, \mathbf{Enc}, \mathbf{Dec})$. If a map f_E exists such that the equation*

$$f_E(\mathbf{Enc}(k_p, \overline{\mathbf{\Phi}}), \mathbf{Enc}(k_p, \overline{\boldsymbol{\xi}})) = \mathbf{Enc}(k_p, \overline{f}(\mathbf{\Phi}, \boldsymbol{\xi})) \tag{3}$$

holds, then we call f_E the encrypted controller of f. Here, $\overline{\mathbf{\Phi}} \in \mathcal{M}^{\alpha \times \beta}$, $\overline{\boldsymbol{\xi}} \in \mathcal{M}^{\beta}$, and $\overline{f}(\cdot) \in \mathcal{M}^{\alpha}$ are the plaintexts, rounded to ensure that each component can be represented as an element of the information space.

4. Control System with Encrypted Controller

In this section, we encrypt the controller using the BFV encryption scheme to obtain Scheme 1, and we precompute to save time. Precomputation speeds up the encrypted control system's operation and reduces the amount of time required for encryption. However, the appearance of precomputation changes the underlying encryption algorithm from random encryption to deterministic encryption, which reduces the security of the BFV scheme. As a result, we suggest two schemes to strengthen scheme 1's security from two different angles.

4.1. Encrypted Controller Using BFV Encryption Scheme

First, we encrypt the controller using the BFV encryption method and follow the method in the literature [16] to adopt precomputation to save the time of encryption. We describe the process for this scheme based on Figure 4 in the following. We have drawn the flowchart as Figure 5 to help understand.

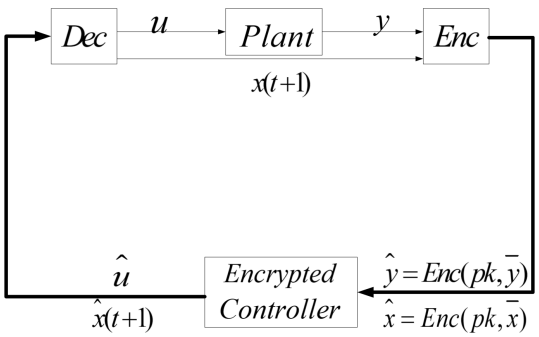


Figure 4. The schematic diagram of a networked control system with BFV encryption.

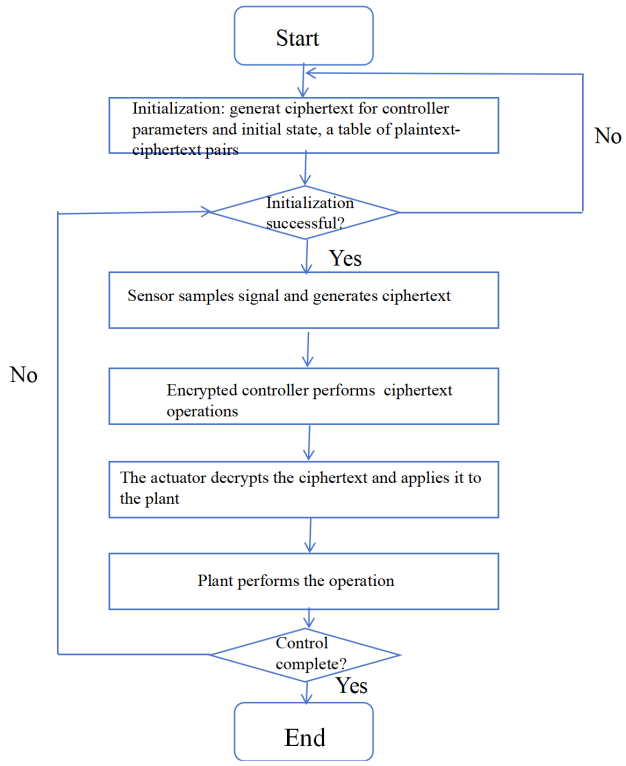


Figure 5. The flowchart of a networked control system with BFV encryption.

Scheme 1:

We describe each step in detail based on the above flowchart:

- The parameters $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$, and $\boldsymbol{D}$ as well as the controller's initial state $\boldsymbol{x}(\mathbf{0})$, are encrypted to produce its ciphertext $\hat{\boldsymbol{A}}, \hat{\boldsymbol{B}}, \hat{\boldsymbol{C}}, \hat{\boldsymbol{D}}, \hat{\boldsymbol{x}}(\mathbf{0})$ in the controller's design, and this ciphertext is transmitted to the controller.
- The sensor collects the signal $\boldsymbol{y}$ and then uses the plaintext index to look up the table generated by precomputation to obtain the corresponding ciphertext $\hat{\boldsymbol{y}} = \mathbf{Enc}(\boldsymbol{y})$, thus realizing the encryption process, which is then passed to the encrypted controller.
- The encryption controller performs homomorphic operations after obtaining the ciphertext signal $\hat{\boldsymbol{y}}$. The BFV homomorphism operation states that (1) really operates in the cipher space after being encrypted as:

$$\begin{aligned} \hat{\boldsymbol{x}}(t+1) &= \hat{\boldsymbol{A}} \times \hat{\boldsymbol{x}}(t) + \hat{\boldsymbol{B}} \times \hat{\boldsymbol{y}}(t) \\ \hat{\boldsymbol{u}}(t) &= \hat{\boldsymbol{C}} \times \hat{\boldsymbol{x}}(t) + \hat{\boldsymbol{D}} \times \hat{\boldsymbol{x}}(t) \end{aligned} \tag{4}$$

  The ciphertext $\hat{\boldsymbol{x}}(t+1)$ of the state and the ciphertext $\hat{\boldsymbol{u}}(t)$ of the output are passed to the actuator after the homomorphic operation is completed.
- The actuator block decrypts the controller's output cipher to obtain $\boldsymbol{u}(t)$ and applies it to the plant, decrypts the $\hat{\boldsymbol{x}}(t+1)$ to obtain the state $\boldsymbol{x}(t+1)$ and passes it to the sensor.
- The state $\boldsymbol{x}(t+1)$ is encrypted by the sensor and sent to the encrypted controller.

Remark 1. *(About the table generated by the precomputation)*

1. *Based on the plant, the control function, and the initial condition, we can identify the range of* $\boldsymbol{x}$ *and* $\boldsymbol{y}$.
2. *We run the control system, create plaintext–ciphertext pairs* $(m, \hat{m})$ *by encryption, and place them in a table. The final generated table contains all pairs between* $\boldsymbol{y}$ *and* $\boldsymbol{x}$.
3. *The table can determine the corresponding ciphertext when the sensor gathers the value* $\boldsymbol{y}$ *of the plant and encrypts it.*

Remark 2. *(About state $\boldsymbol{x}(t)$) To avoid the problem that the noise of the ciphertext $\boldsymbol{x}(t)$ increases significantly after homomorphic addition and homomorphic multiplication in the encrypted controller, the following measures are taken. The ciphertext of state $\boldsymbol{x}(t+1)$ after homomorphic encryption is sent to the actuator for decryption and then returned to the cryptographic controller for a new round of homomorphic computation after sensor encryption.*

With regard to Figure 5, we explain the illustration in detail by means of the following example. We want to control a conveyor belt to move a table through the control system. Then, at the beginning, we need to initialize the settings. The transmission speed has an upper limit, so we generate a plaintext–ciphertext table for the range of the speed. After successful initialization, we can start the subsequent operation. Suppose we input speed v; the axis begins to rotate and moves the table. At this point, the following workflow begins. The sensor obtains the axis rotation speed v_1, consults the table to generate the ciphertext value c_{v_1}, and sends it to the encrypted controller. The encrypted controller carries out operations in the form of ciphertext to generate new states and signals. Then, the ciphertext of signals is output to the actuator to decrypt and control the axis to accelerate or decelerate its rotation. The cycle repeats itself until the completion of the transmission of the table.

In scheme 1, the controller input and parameters are ciphertext since the encrypted controller allows homomorphic additions and homomorphic multiplications. As shown in the Figure 4, in the whole control process, from the sensor sampling data and encrypting until the actuator decrypts the ciphertext, the data in the transmission channel and encrypted controller are all ciphertext, which greatly improves the security compared with the previous partially homomorphic encrypted controller. The table is generated in advance, so that the ciphertext can be obtained only by looking up the table according to the plaintext index, saving the time of encryption. However, we know that BFV encryption is a random encryption scheme, and the precomputation causes the search table to obtain the same ciphertext from a plaintext m, that is to say, from random encryption to deterministic encryption. This process reduces the security of the scheme. Therefore, we propose two approaches below to remedy this deficiency.

4.2. Security Enhancement

In the previous section, we saw that although the precomputation apparently improves the efficiency, it also reduces the security of the scheme to a certain extent. Therefore, we propose two approaches below to solve this problem from two aspects. In the following scheme, we use the method of periodically updating the table to enhance security.

We just present a general idea here; the control system's average computation time for each iteration is provided in Section 5 below. Additionally, specific values may be substituted.

4.2.1. Periodic Update Table

The two methods we propose to improve security have no difference with scheme 1 in the general process framework, except for the specific operations in the second step related to obtaining the ciphertext in the second step. Since each ciphertext has only one fixed corresponding ciphertext in a table, we consider updating the table generated by the precalculation regularly. The most intuitive way to solve this problem is to ensure that each cycle of the control system has a new table, but in general, the control system takes much longer to compute each iteration than it takes to generate an estimated table. In this way, to complete the table update, the computing power of the precalculation process must be greatly improved, and the precalculation time must be guaranteed to be less than the control cycle. In this paper, we do not consider excessive requirements on the hardware of the control system; we hope to complete the security improvement through a "natural" method. Therefore, we do the next best thing and consider updating the table regularly.

First, we assume that the control system in scheme 1 takes approximately a ms per iteration to calculate, while the time spent on generating a precomputed encryption table is

about b ms. Therefore, a table can be updated after about b/a iterations of the control system. In order to ensure that the table can be updated, we set an update every $([b/a] + 1)$ cycle.

4.2.2. Provide Multiple Tables

From another perspective, if we randomly select one of the encrypted tables to search for plaintext–ciphertext pairs each time, the problem that there is only one ciphertext value corresponding to the plaintext m can be avoided.

We assume that α tables are generated when the plaintext–ciphertext pairs are generated in the initial precomputation, and a table is randomly selected from α tables and then retrieved according to the plaintext index during each encrypted table lookup. To prevent excessive storage burden, the value of α cannot be too large. But if α is too small, randomness is not enough. At the same time, we consider incorporating the idea of Section 4.2.1 into it and completing the update of a table after b/a iterations of the control system. It may take many cycles to complete the update of α tables, but this is not important, because each encrypted table lookup is a random table selected from α. This practice only further enhances the security of the scheme on this basis.

4.3. Attack Scenario and Security

NCSs are at risk of eavesdropping attacks because plants and controllers communicate with each other over network links. Our proposed network control system with encrypted controllers is well protected against eavesdropping attacks, and we briefly describe it here. We consider the following attack scenarios.

In our model, the attacker $\mathcal{A}$ has mainly the following described capabilities.

1. Adversary $\mathcal{A}$ can collect data within the communication channel through an eavesdropping attack.
2. Adversary $\mathcal{A}$ can collect data within the controller through an eavesdropping attack.

Note: In addition to the capabilities listed above, the decryptor, encryptor, and actuator of the control system cannot be compromised by an attacker $\mathcal{A}$.

We say that the scheme is not resistant to eavesdropping attacks if the attacker $\mathcal{A}$ can obtain the controller parameters A, B, C, and D or signals y in polynomial time; otherwise, we say that it is eavesdropping-resistant.

The security of the control system in this attack scenario is analyzed. Attacker $\mathcal{A}$ collects data in the controller and communication channel by eavesdropping. In this scenario, the data in the controller and the data in the communication channel are in the form of ciphertext, which is encrypted using the BFV encryption scheme. In order to obtain useful data, the attacker $\mathcal{A}$ needs to reduce the ciphertext to plaintext. The BFV encryption scheme is based on the difficult problem of RLWE and hence cannot obtain useful plaintext data in polynomial time. Therefore, our scheme is resistant to eavesdropping attacks.

4.4. Comparison of Four Schemes

In terms of safety and efficiency, we contrast the scheme proposed in this study with a number of traditional schemes that have been previously offered. For their specific processing time, we refer to the literature [16], and the average processing time in this paper will be given in the next section.

It can be seen from Table 1 that in the previous partially homomorphic encryption schemes, the security of data transmitted in the channel during the process from sensor to actuator and data in the controller cannot be ensured at the same time. This suggests that security is not ideal. Subsequent BGN encryption schemes can well avoid this problem. The BFV encrypted controller scheme proposed in this paper and BGN encrypted controller can ensure that the data inside the controller and in the transmission channel are ciphertext. In terms of efficiency, the two partially homomorphic encryption schemes are efficient, and BGN is relatively inefficient. But this time can also be suitable for the control system's sampling cycle. However, by accelerating precomputation, the scheme using the BFV technique suggested in this study achieves an efficiency that is almost identical to partially

homomorphic encryption. On the other hand, considering the homomorphic operation inside the controller, the homomorphic multiplication of BGN can only be performed once, but partially homomorphic encryption and the BFV scheme proposed in this paper do not have this defect.

Table 1. Analysis of three control systems.

	Data Security	Homomorphic Operation	Average Processing Time [1]
With RSA encryption	Only the data inside the controller	Many times	$(10 + a)$ [2] ms
With Paillier encryption	Only the data in the channel	Many times	16 ms
With BGN encryption	All data are secure	Add multiple times and multiply once	96 ms
With BFV encryption	All data are secure	Many times	24 ms

[1] The data are obtained with a key length of 512 bits; [2] "a" represents the additional communication time required.

Through the above analysis, the BFV scheme is excellent in both security and efficiency.

5. Numerical Example

In this section, we first give a concrete numerical example of a control system, then simulate with our scheme to obtain a series of results, and subsequently analyze the obtained graphs to support our scheme.

The control system is made up of the following discrete-time linear plant and the following kind of linear controller, according to the numerical example in [5]. $\boldsymbol{p_1}(t)$ and $\boldsymbol{p_2}(t)$ are the internal states of the plant, and they satisfy:

$$\begin{aligned} \begin{bmatrix} \boldsymbol{p_1}(t+1) \\ \boldsymbol{p_2}(t+1) \end{bmatrix} &= \begin{bmatrix} 0.99998 & 0.0197 \\ -0.0197 & 0.97025 \end{bmatrix} \begin{bmatrix} \boldsymbol{p_1}(t) \\ \boldsymbol{p_2}(t) \end{bmatrix} + \begin{bmatrix} 0.0000999 \\ 0.0098508 \end{bmatrix} \boldsymbol{u}(t), \\ \boldsymbol{y}(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{p_1}(t) \\ \boldsymbol{p_2}(t) \end{bmatrix}, \end{aligned} \tag{5}$$

where the initial states are $\boldsymbol{p_1}(0) = 1$ and $\boldsymbol{p_2}(0) = 0$, and the linear controller's internal states are $\boldsymbol{x_1}(t)$ and $\boldsymbol{x_2}(t)$, satisfying:

$$\begin{aligned} \begin{bmatrix} \boldsymbol{x_1}(t+1) \\ \boldsymbol{x_2}(t+1) \end{bmatrix} &= \begin{bmatrix} 1 & 0.0063 \\ 0 & 0.3678 \end{bmatrix} \begin{bmatrix} \boldsymbol{x_1}(t) \\ \boldsymbol{x_2}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0.0063 \end{bmatrix} \boldsymbol{y}(t), \\ \boldsymbol{u}(t) &= \begin{bmatrix} 10 & -99.9 \end{bmatrix} \begin{bmatrix} \boldsymbol{x_1}(t) \\ \boldsymbol{x_2}(t) \end{bmatrix} - 3\boldsymbol{y}(t), \end{aligned} \tag{6}$$

where the initial states are $\boldsymbol{x_1}(0) = 0$ and $\boldsymbol{x_2}(0) = 0$.

Numerical Results

The BFV encryption was implemented through Microsoft's Simple Encrypted Arithmetic Library (SEAL). The following diagram was obtained by calling SEAL to simulate the encryption of a specific number of cases.

Figure 6 shows the simulation results corresponding to the time of input y and output u. The control input response shows some minor quantization errors, but such quantization errors are so small that they can be ignored. As can be seen from Figure 6, the closed-loop system's control performance and stability can be realized with the help of the BFV encryption controller.

Figure 7 depicts the time change for computing the iterations of the controlled system following the BFV encryption of the controller. Figure 7a represents the calculation time of each iteration of the control system without BFV encryption using precomputation, with an average time of 32.40 ms; Figure 7b represents the time after precomputation, with an average time of 23.99 ms. As can be seen from the comparison of the two pictures, it is estimated that about 35% of the time will be saved, which is still considerable.

The suggested encryption control system's histogram of ciphertext is displayed in Figure 8. It can be assumed that the ciphertext in the suggested cryptosystem follows a discrete uniform distribution because the histogram distribution is nearly flat.

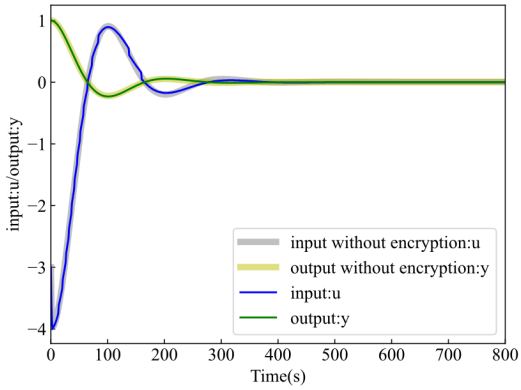
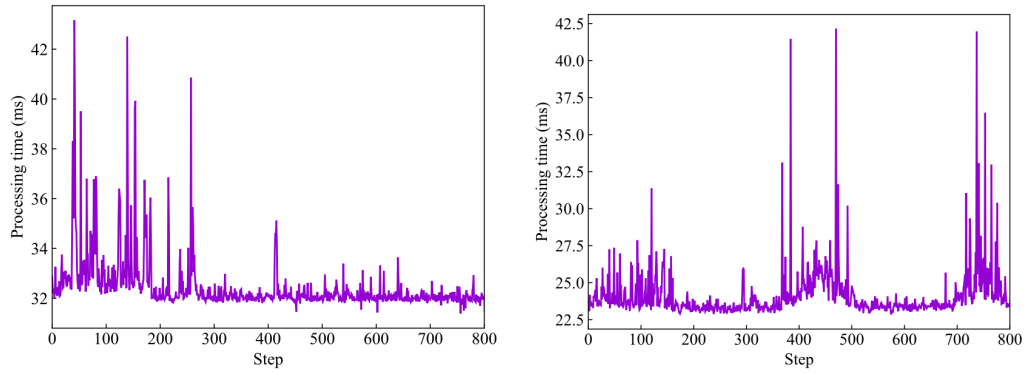


Figure 6. Comparison of output/input with and without the proposed cybersecurity enhancement: BFV encryption.



(**a**) (**b**)

Figure 7. Time variation at each iteration calculation by the encrypted controller, (**a**)without precomputation; (**b**)with precomputation.

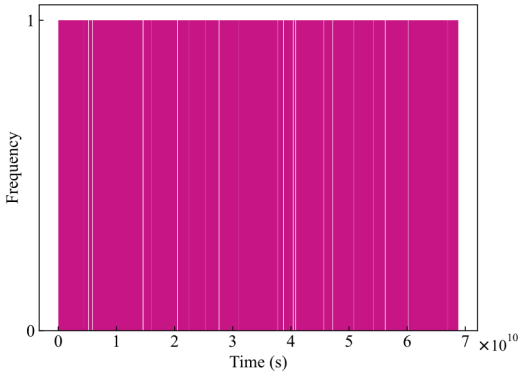


Figure 8. Histogram of the first element of the controller gain in ciphertext.

6. Discussion

In this paper, we proposed a scheme for encrypting the controller using fully homomorphic encryption. We verified that this encrypted controller could achieve a normal control and that the efficiency and security of the encrypted controller were improved using numerical examples in Section 5.

The study in this paper has the following implications. First, in terms of security, our scheme ensures that both the data in the controller and in the channel are not eavesdropped on, so the data security in the whole network control system is improved. This compensates for the lack of security in previous homomorphic encrypted controller schemes [5,13,16]. Second, in terms of efficiency, we use precomputation to alleviate the latency problem caused by fully homomorphic encryption, which reduces the iteration time of the control system and also improves the response time of the networked control system with encrypted controllers. Finally, our scheme has a simple control system and does not require more encrypted controllers compared to the scheme proposed by Kim et al. [13].

This paper also suffers from the following limitations. On the one hand, our scheme places some demands on the computational power of the device, as it requires constant computation to generate tables for encrypted access. On the other hand, homomorphic encryption, especially fully homomorphic encryption, is still difficult to apply in realistic scenarios. Only simulation results are considered in this paper to validate the scheme, which may be problematic in further practical applications specifically. Related issues will be further investigated in future work.

7. Conclusions and Future Work

7.1. Conclusions

In this paper, we proposed a scheme which effectively improved the security of an NCS by encrypting the controller using fully homomorphic encryption. Specifically, all data in the system could be well secured from eavesdropping and recording. We further reduced the time spent on encryption in the scheme by precomputation and improved the efficiency of the encryption controller. In addition, for the security of the network control system, we further proposed two methods to improve the security. An efficient and secure NCS is of great practical significance.

7.2. Future Work

The scheme proposed in this paper can provide some technical guarantee for the data security of NCSs, but this scheme needs to be improved continuously, and in the following aspects, further research needs to be conducted. Firstly, the scheme proposed in this paper is still only in the simulation stage, and further research is needed for future consideration of applications in a real environment. Secondly, in practice, the shorter the iteration time of the control system, the better; therefore, further improvement in efficiency or the design of more efficient schemes should be considered in the future.

Author Contributions: Conceptualization, J.P.; methodology, J.P.; software, T.S.; validation, L.K. and Y.Z.; formal analysis, W.L.; investigation, T.S. and J.W.; resources, J.P.; data curation, T.S.; writing—original draft preparation, J.P.; writing—review and editing, W.L. and Z.W.; supervision, J.W.; funding acquisition, Z.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Key Research and Development Program of China (No. 2023YFF0905300), the Key R&D Plan of Shandong Province (No. 2022CXGC020106), Major Innovation Project of Science, Education and Industry of Shandong Academy of Sciences (No. 2022JBZ01-01), Innovation Capacity Improvement Project for Small and Medium-sized Technology-based Enterprises of Shandong Province (No. 2023TSGC0641).

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yohanandhan, R.V.; Elavarasan, R.M.; Manoharan, P.; Mihet-Popa, L. Cyber-Physical Power System (CPPS): A Review on Modeling, Simulation, and Analysis with Cyber Security Applications. *IEEE Access* **2020**, *8*, 151019–151064. [CrossRef]
2. Yampolskiy, M.; Andel, T.R.; Mcdonald, J.T.; Glisson, W.B.; Yasinsac, A. Intellecutal protection in additive layer manufacturing: Requirements for secure outsouring. In Proceedings of the 4th Program Protection and Reverse Engineering Workshop, New Orleans, LA, USA, 9 December 2014.
3. Wall, D.S.; Yar, M. Intellecutal property crime and the Internet: Cyber-piracy and 'stealing' information intangibles. In *Handbook of Internet Crime*; Wall, D.S., Yar, M., Eds.; Willan: London, UK, 2010; pp. 255–272.
4. Mclaughlin, S. On dynamic malware payloads aimed at programmable logic controllers. In Proceedings of the 6th USENIX Conference on Hot Topics in Security, San Francisco, CA, USA, 9 August 2011; p. 10.
5. Kogiso, K.; Fujita, T. Cyber-Security Enhancement of Networked Control Systems Using Homomorphic Encryption. In Proceedings of the IEEE Conference on Decision and Control, Osaka, Japan, 15–18 December 2015.
6. Rivest, R.L.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystem. *Commun. ACM* **1978**, *21*, 120–126. [CrossRef]
7. ElGamal, T. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inf. Theory* **1985**, *31*, 469–472. [CrossRef]
8. Farokhi, F.; Shames, I.; Batterham, N. Secure and private control using semi-homomorphic encryption. *Control Eng. Pract.* **2017**, *67*, 13–20. [CrossRef]
9. Lin, Y.; Farokhi, F.; Shames, I.; Nesic, D. Secure control of nonlinear systems using semi-homomorphic encryption. In Proceedings of the IEEE Conference on Decision and Control, Miami, FL, USA, 17–19 December, 2018.
10. Murguia, C.; Farokhi, F.; Shames, I. Secure and private implementation of dynamic controllers using semi-homomorphic encryption. *IEEE Trans. Autom. Control* **2020**, *65*, 3950–3957. [CrossRef]
11. Kosieradzki, S.; Zhao, X.; Kawase, H.; Qiu, Y.; Kogiso, K.; Ueda, J. Secure teleoperation control using somewhat homomorphic encryption. *IFAC-PapersOnLine* **2020**, *55*, 593–600. [CrossRef]
12. Paillier, P. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology—Eurocrypt'99, Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, 2–6 May 1999*; Stern, J., Ed.; Springer: Berlin/Heidelberg, Germany, 1999; pp. 223–238.
13. Kim, J.; Lee, C.; Shim, H.; Cheon, J.H.; Kim, A.; Kim, M.; Song, Y. Encrypting controller using fully homomorphic encryption for security of cyber-physical systems. *IFAC-PapersOnLine* **2020**, *49*, 175–180. [CrossRef]
14. Brakerski, Z. Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP. 2012. Available online: https://eprint.iacr.org/2012/078 (accessed on 13 June 2023).
15. Fan, J.; Vercauteren, F. Somewhat Practical Fully Homomorphic Encryption. Cryptology Eprint Archive. 2012. Available online: https://eprint.iacr.org/2012/144 (accessed on 9 June 2023).
16. Pan, J.; Sui, T.; Liu, W.; Wang, J.; Kong, L.; Zhao, Y. Secure Control Using Homomorphic Encryption and Efficiency Analysis. *Secur. Commun. Netw.* **2023**, *2023*, 6473497 . [CrossRef]
17. Wang, D.; Li, W.; Wang, P. Measuring Two-Factor Authentication Schemes for Real-Time Data Access in Industrial Wireless Sensor Networks. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4081–4092. [CrossRef]
18. Wang, Q.; Wang, D. Understanding Failures in Security Proofs of Multi-Factor Authentication for Mobile Devices. *IEEE Trans. Inf. Forensics Secur.* **2022**, *18*, 597–612. [CrossRef]
19. Yu, Y.; Xu, G.; Wang, X. Provably Secure NTRU Instances over Prime Cyclotomic Rings. In *Public-Key Cryptography—PKC 2017, Proceedings of the 20th IACR International Conference on Practice and Theory in Public-Key Cryptography, Amsterdam, The Netherlands, 28–31 March 2017*; Lecture Notes in Computer Science; Fehr, S., Ed.; Springer: Berlin/Heidelberg, Germany, 2017; pp. 409–434.
20. Rivest, R.; Adleman, L.; Deryouzos, M. On data banks and privacy homomorphisms. In *Foundations of Secure Computation*; Academic Press, Inc.: Orlando, FL, USA, 1978; pp. 169–180.
21. Gentry, C. Fully homomorphic encryption using ideal lattices. In Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing, Bethesda, MD, USA, 31 May 2009; pp. 169–178.
22. Garg, S.; Gentry, C.; Halevi, S.; Raykova, M.; Sahai, A.; Waters, B. Candidate indistinguishability obfuscation and functional encryption for all circuits. In Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, Berkeley, CA, USA, 26–29 October 2013; pp. 40–19.
23. Brakerski, Z.; Vaikuntanathan, V. Efficient fully homomorphic encryption from (standard) LWE. In Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, Palm Springs, CA, USA, 22–25 October 2011.
24. Brakerski, Z.; Gentry, C.; Vaikuntanathan, V. (Leveled) fully homomorphic encryption without bootstrapping. In Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, Cambridge, MA, USA, 8–10 January 2012; pp. 309–325.
25. Gentry, C.; Sahai, A.; Waters, B. Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based. Available online: https://eprint.iacr.org/2013/340 (accessed on 9 June 2013).
26. Teranishi, K.; Kogiso, K.; Shimada, N. Stability-guaranteed dynamic ElGamal cryptosystem for encrypted control systems. *IET Control Theory Appl.* **2020**, *14*, 2242–2252. [CrossRef]
27. Kogiso, K. Upper-Bound Analysis of Performance Degradation in Encrypted Control System. In Proceedings of the 2018 Annual American Control Conference, Milwaukee, WI, USA, 27–29 June 2018.

28. Tran, J.; Farokhi, F.; Cantoni, M.; Shames, I. Implementing homomorphic encryption based secure feedback control. *Control Eng. Pract.* **2020**, *97*, 104350.1–104350.12. [CrossRef]
29. Shoukry, Y.; Gatsis, K.; Alanwar, A.; Pappas, G.J.; Seshia, S.A.; Srivastava, M.; Tabuada, P. Privacy-aware quadratic optimization using partially homomorphic encryption. In Proceedings of the 2016 IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, USA, 12–14 December 2016.
30. Darup, M.S.; Redder, A.; Quevedo, D.E. Encrypted cloud-based MPC for linear systems with input constraints. *IFAC-PapersOnLine* **2018**, *51*, 535–542. [CrossRef]
31. Alexandru, A.B.; Morari, M.; Pappas, G.J. Cloud-Based MPC with Encrypted Data. In Proceedings of the 2018 IEEE Conference on Decision and Control (CDC), Miami, FL, USA, 17–19 December 2018.
32. Darup, M.S. Encrypted MPC based on ADMM real-time iterations. *IFAC-PapersOnLine* **2020**, *53*, 3508–3514. [CrossRef]
33. Darup, M.S.; Alexandru, A.B.; Quevedo D.E.; Pappas, G.J. Encrypted Control for Networked Systems: An Illustrative Introduction and Current Challenges. *IEEE Control Syst. Mag.* **2021**, *41*, 58–78. [CrossRef]
34. Song, C.; Huang, R. Secure Convolution Neural Network Inference Based on Homomorphic Encryption. *Appl. Sci.* **2023**, *13*, 6117. [CrossRef]
35. Lyubashevsky, V.; Peikert, C.; Regev, O. On Ideal Lattices and Learning with Errors over Rings. *Commun. ACM* **2013**, *60*, 43.1–43.35. [CrossRef]