이학석사 학위논문

Securing Networked Control System with Homomorphic Authenticated Encryptions

(동형암호인증을 이용한 제어 시스템의 안전성 증진)

2017년 2월

서울대학교 대학원

수리과학부

이지은

Abstract

Enhancing the security of the control system is necessary since it can directly cause physical problems in real world. It is important to protect the controller itself and the signal between the controller and the plant since an attacker can steal or manipulate it during the process. There is a paper proposing a solution to prevent the data from stealing. They use fully homomorphic encryption (FHE) to secure the controller. However, even though the attacker can not steal the signal, he or she still can manipulate the signal or break the system. In this paper, we propose a homomorphic authenticated encryption(HAE) scheme which can enable the plant to verify whether the input data is correctly evaluated. By using the label, discrete fourier transform(DFF) and ring learning with errors(RLWE), we can protect the data from the stealing and manipulating. Furthermore, we implement the scheme with quadruple water tank model.

Key words: Homomorphic Authenticated Encrytion, Control System, Discrete Fourier Transform, Ring-LWE
Student Number: 2015-20274

Contents

Chapter 1

Introduction

Cyber-physical systems(CPSs) means cyber process, physical process and their links. It uses sensor and actuator to monitor physical process and this system can provide new ability to the physical system. The importance of the integrate process of physical systems and networks is increasing since it is a key sector in the IoT(Internet of Things). Since CPSs is closely related to the real world, it is vulnerable to the various attacks. For example, eavesdropping is one of the well-known attacks to the CPSs. Through this attack, the attacker can get data illegally. More troubling is that an cyber-physical attacker can cause malfuction, which is sending false data and causing more serious problems.

The simplest way to protect CPSs from the attacker is to protect the communication channels between the plant and the controller. However, in this conventional way, the controller had to decrypt the input data to evaluate and had to encrypt it again to send a signal to the actuator. That is, the controller have to keep the secret key inside. This system can let the attacker get the secret key or data from the controller by injecting malicious signals. Therefore, there was a need of fully homomorphic enryption(FHE).

There is a paper using FHE to encrypt the controller. [**?**] By using FHE, the controller doesn't have to encrypt or decrypt the signal. Therefore there is no need to have the secret key in the controller. This way the attacker can only get encrypted data from the controller. However, there is an important problem left. That is, the attacker cannot steal the data but he or she still can manipulate the signal or break the system. To avoid this problem, the actuator have to verify the evaluated signal.

In this paper, we propose homomorphic authenticated encryption(HAE) to secure the networked control system for the first time. Similar to many

homomorphic authentication schemes, this scenario is based on the unsecure controller. By using HAE, it is possible to protect the signal from both stealing and changing.

First, we generate the tag of the signal. After that, it is necessary to hide the signal and its tag so that the attacker cannot convert the signal without changing its tag. Therefore, we use discrete fourier transform(DFF) to hide them and generate the message for ring learning with errors(RLWE). In this system the most difficult part was the existance of the controller state. The output of the plant refreshes everytime so generating its label and encrypting itself are not a big problem. However, the controller state is updated inside of the controller and the plant won't know the value. The reason is that if the plant knows the controller state, there is no need of the controller. So as you keep evaluting in the controller, the error of the controller state increases and there comes a time to reboot it. We use multi-tree [**?**] to increase the lifespan of encrypted data. The lifespan means the time between the rebootings the system. To use multi-tree, the actuator need some memory. As the controller evaluate ciphertexts, the actuator will do the same with plaintexts, specifically with the tags of the signals. By evaluating the tags, the controller can verify input signal.

After the propsal of our scheme, we implement this scheme with quadruple water tank model. [**?**]

Chapter 2

Model and Settings

In this chapter, we introduce the networked control system model which we are focusing on. We propose a HAE scheme on this model.

The Networked Control System. We propose our scheme on the discrete-time linear time-invariant controller. [**?**]

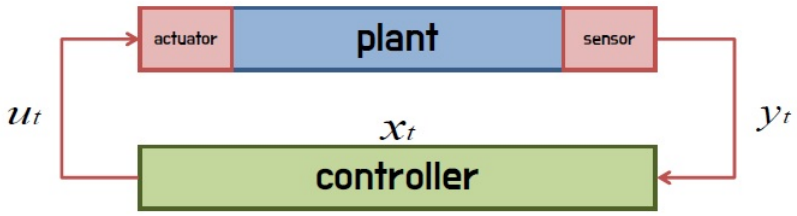

Figure 2.1: The Networked Control System

- $x_t \in \mathbb{R}^l$ is the controller state.
- $y_t \in \mathbb{R}^p$ is the controller input or the plant output.
- $u_t \in \mathbb{R}^m$ is the controller output or the pland input.
- $A \in \mathbb{R}^{l \times l}$, $B \in \mathbb{R}^{l \times p}$, $C \in \mathbb{R}^{m \times l}$, $D \in \mathbb{R}^{m \times p}$ are controller parameters.

The controller will update x_{t+1} and evaluate output u_t with following linear equations.

$$x_{t+1} = A \cdot x_t + B \cdot y_t$$
$$u_t = C \cdot x_t + D \cdot y_t$$

Chapter 3

Preliminary

3.1 Labeled Programs

The labed programs is first inroduced in [**?**]. It relies on the use of FHE. Let $\mathcal{M}$ be the message space, $\mathcal{L}$ be the label space, and $\mathcal{F}$ be the function space. We call $f \in \mathcal{F}$ an admissible function and l the arity of f if any $f \in \mathcal{F}$ represent $f : \mathcal{M}^l \to \mathcal{M}$ for some $l \in \mathbb{Z}^+$ which depends on f. [**?**] A labeled program is a tuple $P = (f, \tau_1, \cdots, \tau_l)$ where $\tau_i \in \mathcal{L}$ are labels for $i = 1, \cdots, l$ for each input of f. Given labeled programs $P_1, \cdots, P_t$ and a circuit $g : \{0,1\}^t \to \{0,1\}$, we can definde the composed program denoted by $P^* = g(P_1, \cdots, P_t)$. It evaluate g with the outputs of $P_1, \cdots, P_t$. The labeled inputs of P^* are all the distinct labeled inputs of $P_1, \cdots, P_t$.

The homomorphic authenticator scheme consists of following syntax. [**?**]

- $(ek, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$. This algorithm outputs the evaluation key ek and the secret key sk where λ is a security parameter.
- $\sigma \leftarrow \mathsf{Auth}_{sk}(b, \tau)$. This algorithm gets the bit $b \in \{0,1\}$ and the label $\tau \in \{0,1\}^*$ using secretly-keyed authentication algorithm and outputs a tag σ.
- $\psi \leftarrow \mathsf{Eval}_{ek}(f, \sigma)$. This algorithm gets a vector of tags $\sigma = (\sigma_1, \cdots, \sigma_k)$ and a circuit $f : \{0,1\}^f \to \{0,1\}$ and outputs a tag ψ. If each σ_i authenticates a bit b_i as the output of a labeled program P_i, then ψ should authenticate $b^* = f(b_1, \cdots .b_k)$ as the outputs of the composed program $P^* = f(P_1, \cdots, P_k)$.
- $\{\mathsf{accept}, \mathsf{reject}\} \leftarrow \mathsf{Ver}_{sk}(e, P, \psi)$. This algorithm gets a tag ψ and check that $e \in \{0,1\}$ is the output of the program P on previoulsy authenticated labeled data.

3.2 Discrete Fourier Transform

The discrete fourier transform (DFT) of vector a is vector y, where ω_n is complex nth roots of unity. [**?**]

$$a = (a_0, \cdots, a_{n-1}) \to y = (y_0, \cdots, y_{n-1})$$
$$y_k = \sum_{i=0}^{n-1} a_i \omega_n^{ki}$$

3.3 Ring Learning With Errors

Fan and Vercauteren ported Brakerski's fully homomorphic encryption (FHE) scheme was suggested in [**?**] to the ring learing with errors (RLWE) setting. [**?**] The FV FHE scheme can be written as follows. [**?**]

- $(d, q, t, \chi_{key}, \chi_{err}, w) \leftarrow \mathsf{FV.ParamsGen}(\lambda)$. The input λ is the security parameter. The output d is a fixed positive integer determines R, moduli q and t with $1 < t < q$, distributions χ_{key}, χ_{err} on R, and an integer base $w > 1$.
- $(pk, sk, ek) \leftarrow \mathsf{FV.KeyGen}(d, q, t, \chi_{key}, \chi_{err}, w)$. Choose $s \leftarrow \chi_{key}$, $a \leftarrow R_q$, $e \leftarrow \chi_{err}$, $a \leftarrow R_q^{l_{w,q}}$ and $e \leftarrow \chi_{err}^{l_{w,q}}$ uniformly randomly. Then compute $b = [-(as+e)]_q$ and $\gamma = ([\mathsf{PowersOf}_{w,q}(s^2) - (e + a \cdot s)]_q, a) \in R^{l_{w,q}}$, where $\mathsf{PowerOf}_{w,q}(a) = ([aw^i]_q)_{i=0}^{l_{w,q}-1} \in R^{l_{w,q}}$. The output $(pk, sk, ek) = ((b, a), s, \gamma)$.
- $c \leftarrow \mathsf{FV.Encrypt}((b, a), m)$. For a message $m + tR \in R/tR$, sample$u \leftarrow \chi_{key}$ and $e_1, e_2 \leftarrow \chi_{err}$. The output ciphertext is $c = ([\Delta[m]_t + bu + e_1]_q, [au + e_2]_q) \in R^2$, where $\Delta = \lfloor q/t \rfloor$.
- $m \leftarrow \mathsf{FV.Decrypt}(s, c)$. Let the input be $c = (c_0, c_1)$ and the secret key s and output is $m = [\lfloor \frac{t}{q} \cdot [c_0 + c_1 s]_q \rfloor]_t \in R$.
- $c_{add} \leftarrow \mathsf{FV.Add}(c_1, c_2)$. Let the input be two ciphertexts $c_1 = (c_{1,0}, c_{1,1})$ and $c_2 = (c_{2,0}, c_{2,1})$ and the output is $c_{add} = ([c_{1,0} + c_{2,0}]_q, [c_{1,1} + c_{2,1}]_q)$.
- $c \leftarrow \mathsf{FV.ReLin}(\tilde{c}_{mult}, ek)$. Let the input be $\tilde{c}_{mult} = (c_1, c_1, c_2)$ and $(b, a) = ek$. The output c is $([c_0 + \langle \mathsf{WordDecomp}_{w,q}(c_2), b \rangle]_q, [c_1 + \langle \mathsf{WordDecomp}_{w,q}(c_2), a \rangle]_q)$, where $\mathsf{WordDecomp}_{w,q}(a) = ([a_i]_w)_{i=0}^{l_{w,q}-1} \in R^{l_{w,q}}$.

We will skip the part about $\mathsf{FV.Mult}$ since our control system will only use homomorphic addition and constant multiplication.

Chapter 4

Homomorphic Authenticated Encryption for Securing the Control System

In this chapter, we propse our scheme of homomorphic authenticated encryption(HAE) for securing the control system. By previous work, the control system can be secured from the attacker who wants to get information from the controller by using fully homomorphic encryption(FHE). [**?**] However, the attacker can still manipulate the signal by simply adding 1 to ciphertext in the controller. Our scheme is proposed for an unsecure controller and a secure plant. Our goal is to decrypt the controller output $\bar{u}_t := Enc(u_t, \sigma(u_t))$ and verify the label $\sigma(u_t)$ to validate u_t, where the signal changes depending on time t. This is done inside of the sensor.

4.1 Encryption and Evaluation

The sensor does three steps for the encryption. We suppose it wants to send the encrypted data of y_t, where the sensor output depends on time t. First, we randomly generate the label of y_t which can be denoted as $\sigma(y_t)$. Second, we use discrete fourier transfrom(DFF) to make $m(x)$ by using y_t and $\sigma(y_t)$. Finally, we use RLWE to encrypt $m(x)$.

After that, the sensor sends the encrypted data $\bar{y}_t = Enc(y_t, \sigma(y_t))$ to the controller. The controller has its state $\bar{x}_t$ which will be self-updated inside. However, since the controller doesn't have secret key sk, the sensor have to send $\bar{x}_0$ to the controller at first. The controller evaluates $\bar{x}_{t+1}$ and

$\bar{u}_t$ by using following linear equations where A, B, C, D are fixed matrices.

$$\bar{x}_{t+1} = A \cdot \bar{x}_t + B \cdot \bar{y}_t$$
$$\bar{u}_t = C \cdot \bar{x}_t + D \cdot \bar{y}_t$$

Once the controller finish evaluation, then it update its state $\bar{x}_t$ to $\bar{x}_{t+1}$ and send $\bar{u}_t$ to the actuator. The

The encryption and the evaluation consists of following PPT algorithms.

- $(spk, ssk, pk, sk) \leftarrow \mathsf{Kg}(1^\lambda)$. This algorithm takes security parameter λ and outputs a public signing key spk, a secret signing key ssk,a public encryption key pk and a secret decryption key sk. We may assume that pk specifies the plaintext space $\mathcal{P}$ and ciphertext space $\mathcal{C}$.
- $\sigma(y_t) \leftarrow \mathsf{Auth}_{ssk}(b, \tau(y_t))$. The algorithm takes $\tau(y_t)$ which is the label of y_t and outputs its tag $\sigma(y_t)$.
- $m(x)_{y_t} \leftarrow \mathsf{FFT}(y_t, \sigma(y_t))$. This algorithm takes y_t and its tag $\sigma(y_t)$ and outputs the message $m(x)$.
- $\bar{y}_t \leftarrow \mathsf{HEnc}(pk, m(x)_{y_t})$. This algorithm takes the public key pk and message $m(x)_{y_t}$ and outputs a ciphertext $\bar{y}_t \in \mathcal{C}$.
- $(\bar{x}\bar{y}_t)_f \leftarrow \mathsf{HEv}(ek, f, \bar{x}_t, \bar{y}_t)$. This algorithm takes the evaluation key ek, a function $f : (\{0,1\}^*)^n \to \{0,1\}^*$ and two ciphertext $\bar{x}_t, \bar{y}_t$ and outputs a ciphertext $(\bar{x}\bar{y}_t)_f$.

Label Program. Let y_t be under a label $\tau(y_t)$, then creates random value $\sigma(y_t)$ from $\mathbb{Z}_{p^n}$.

DFF. Let r_i be uniformly randomly choosen from $\mathbb{Z}_{p^n}$ where $2 \leq i \leq n-1$. We are going to use DFT of vector $(y_t := r_0, \sigma(y_t) := r_1, r_2, \cdots, r_{n-1})$ as a message of RLWE. We will consider all the components of the vector as $r_i = \sum_{j=0}^{n-1} r_{ij} \cdot \alpha^j$ where α is primitive nth root of unity. Then it is natural to think of y_t as $y_t = r_0 = \sum_{j=0}^{n-1} r_{0j} \cdot \alpha^j$ where $r_{0j} = 0$ for all $1 \leq j$ and $\sigma(y_t)$ as $\sigma(y_t) = r_1 = \sum_{j=0}^{n-1} r_{1j} \cdot \alpha^j$ where $r_{1j} = 0$ for all $k < j$. After using DFT, we can get a vector $(m_{y_t,0}, m_{y_t,1}, \cdots, m_{y_t,n-1})$ where $m_{y_t,k} = \sum_{j=0}^{n-1} r_j \cdot \alpha^{jk}$, $\sum_{j=0}^{n-1} \alpha^{jk} = 0$. We use $m_{y_t}(x) = m_{y_t,0} + m_{y_t,1}x + \cdots + m_{y_t,n-1}x^{n-1}$ as message of RLWE.

RLWE. Let

- $a(x)$ be a polynomial from the ring $Z_q[x]/\Phi(x)$ with all the coefficients from F_q.

- $e(x)$ be a random unknown polynomial from the ring $Z_q[x]/\Phi(x)$ with all the coefficients are less than an integer b which is much less than q.
- $s(x)$ be a small unknown polynomial with all the coefficients relative to the same bound b.
- $b(x)$ be the set of polynomials such that $b(x) = a(x) \cdot s(x) + e(x)$.

Note that $\Phi(x)$ is a cyclotomic polynomial.

The sensor encrypt $m_{y_t}(x)$ by using RLWE and send the ciphertext $\bar{y}_t = m_{y_t}(x) + b(x)$ to the controller.

Evaluation. The controller gets $\bar{y}_t$ and use the controller state $\bar{x}_t$ to evaluate following linear equations.

$$\begin{aligned} \bar{x}_{t+1} &= A \cdot \bar{x}_t + B \cdot \bar{y}_t \\ \bar{u}_t &= C \cdot \bar{x}_t + D \cdot \bar{y}_t \end{aligned}$$

Then, replace $\bar{x}_{t+1}$ as the controller state for the next evaluation and output ciphertext $\bar{u}_t$ to the actuator.

4.2 Decryption and Authentication

The actuator gets the ciphertext $\bar{u}_t$ and decrypt it to get message $m'(x)_{u_t}$. By using IFFT, we can get u_t and $\sigma(u_t)$.

Decryption. The decryption consists of following PPT algorithms.

- $m'(x)_{u_t} \leftarrow \mathsf{HDec}(sk, \bar{u}_t)$. This algorithm takes the secret key sk and ciphertext $\bar{u}_t$ and outputs message $m'(x)_{u_t}$.
- $(u_t, \sigma(u_t)) \leftarrow \mathsf{IFFT}(m'(x)_{u_t})$. This algorithm takes the message $m'(x)_{u_t}$ and outputs plaintext u_t and its tag $\sigma(u_t)$.

Authentication. Now, the actuator verifies the $\sigma(u_t)$ to guarantee u_t with following equation and save $\sigma(u_t)$ for the next calculation.
If $t = 0$,

$$\sigma(u_t) = C \cdot \sigma(x_t) + D \cdot \sigma(y_t)$$

If $t > 0$,

$$\sigma(u_t) = CAC^{-1} \cdot (\sigma(u_t) - D \cdot \sigma(y_{t-1})) + CB \cdot \sigma(y_{t-1}) + D \cdot \sigma(y_t)$$

After all this process, the sensor gives new data and it continues as above until the time t is less than the level L of FHE. Over time, a reboot of the control system is required. It doesn't have multiplication of FHE, but if the coefficients of the constant matrices are real numbers, it need to be roboot because the matrix A is multiplied i times to $\bar{x}_t$ to compute $\bar{x}_{t+i}$. We use multiplication tree structure suggested in [**?**] to delay the reboot up to 2^{L-1}.

4.3 Multiplication Tree

The multiplication tree was proposed in [**?**]. This structure is suggested to increase the lifespan of encrypted data up to 2^{L-1} while the matrices are multiplied to the data at L times. The definition of 'catch-up' vector $P_i(z_0, \cdots, z_{i-1})$ is as follows.

$$P_i(z_0, \cdots, z_{i-1}) := \sum_{j=0}^{i-1} A^{i-j-1} z_j$$

In [**?**], they propose $\mathsf{MTREE}_\mathsf{h}$ which computes the encryptions of

$$P_i(z_0), \cdots, P_{2^h}(z_0, \cdots, z_{2^h-1})$$

iteratively with ciphertexts $\bar{z}_0, \cdots, \bar{z}_{i-1}$. By using this algorithm, we can 'catch-up' directly from $\bar{x}_t$ to $\bar{x}_{t+i}$ by storing at most h ciphertexts.

It is easy to check $\bar{x}_i = P_i(B \cdot \bar{y}_0, \cdots, B \cdot \bar{y}_{i-1})$. By using $\mathsf{MTREE}_\mathsf{h}$, the error of $\bar{x}_i$ is same as $\bar{x}_{2^k}$ where $2^k \leq i < 2^{k+1}$. Therefore, the lifespan of the encrypted data increases up to 2^{L-1}.

4.4 Evaluation and Authentication using Multiplication Tree

As we keep running the system, the controller will evaluate $\bar{x}_t$ by using $P_{2^k}(B \cdot \bar{y}_0, \cdots, B \cdot \bar{y}_{2^k-1})$ while the actuator evaluates $\sigma(x_t)$ by using $P_{2^k}(B \cdot \sigma(y_0), \cdots, B \cdot \sigma(y_{2^k-1})$, where $0 \leq 2^k \leq t$.

Evaluation. The controller computes $\bar{x}_t$ by following steps. First, find $(a_0, \cdots, a_k)$ such that $t = \sum_{i=0}^{k} a_i \cdot 2^i$ where $a_i \in \{0, 1\}$. Then computes

as follows.

$$\begin{aligned}\bar{x}_t &= a_0 \cdot A^{\sum_{i=1}^{k} a_i \cdot 2^i} \cdot P_{2^0}(B \cdot \bar{y}_0) \\ &+ \sum_{i=1}^{k-1} a_0 \cdot A^{\sum_{j=i+1}^{k} a_j \cdot 2^j} \cdot P_{2^i}(B \cdot \bar{y}_0, \cdots, B \cdot \bar{y}_{2^i-1}) \\ &+ a_k \cdot P_{2^k}(B \cdot \bar{y}_0, \cdots, B \cdot \bar{y}_{2^k-1})\end{aligned}$$

Authentication. The acuator first decrypts the ciphertext $\bar{u}_t$ to get message $m'(x)_{u_t}$. By using IFFT, we can get u_t and $\sigma(u_t)$. To verify $\sigma(u_t)$, it does the same computation with $\sigma(y_t)$ as you can see below. First, find $(b_0, \cdots, b_k)$ such that $t = \sum_{i=0}^{k} b_i \cdot 2^i$ where $b_i \in \{0, 1\}$. Then computes as follows.

$$\begin{aligned}\sigma(x_t) &= b_0 \cdot A^{\sum_{i=1}^{k} a_i \cdot 2^i} \cdot P_{2^0}(B \cdot \sigma(y_0)) \\ &+ \sum_{i=1}^{k-1} a_0 \cdot A^{\sum_{j=i+1}^{k} a_j \cdot 2^j} \cdot P_{2^i}(B \cdot \sigma(y_0), \cdots, B \cdot \sigma(y_{2^i-1})) \\ &+ a_k \cdot P_{2^k}(B \cdot \sigma(y_0), \cdots, B \cdot \sigma(y_{2^k-1}))\end{aligned}$$

The actuator knows $\sigma(y_t)$ since the sensor will share it directly. After the computation, the actuator knows both $\sigma(y_t)$ and $\sigma(x_t)$ so it can verify $\sigma(u_t)$ by following simple calculation.

$$\sigma(u_t) = C \cdot \sigma(x_t) + D \cdot \sigma(y_t))$$

Chapter 5

Implementing HAE for Securing the Water Tank Model

We Implement HAE for securing a quadruple water tank system introduced in [**?**].

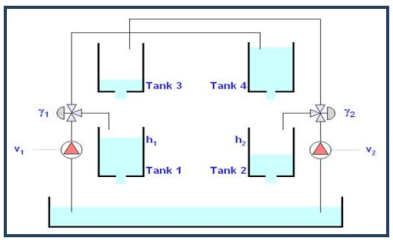

Figure 5.1: The Quadruple Water Tank System

This model is the same one used to implement FHE in [**?**]

Test environment. This scheme was implemented in C++ using NTL library and HElib library for the BGV cyptosystem. We compiled using g++ on MacOS 10.12.1. We ran on a computer with 2.6 GHz Intel Core i5 processor and 8GB RAM. Our purpose of implementation is to show that our scheme works properly.

Parameter selection. For the BGV scheme via HElib, we set the security parameter $\lambda = 80$, the multiplicative depth $L = 9$, and the plaintext space

$\mathbb{Z}_M$ where $M = 2^{28}$. Moreover, by simple calculation the controller takes the fixed matrices as follows. [**?**]

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, C = \begin{pmatrix} 0.05 & 0 \\ 0 & 0.034 \end{pmatrix}, D = \begin{pmatrix} 3.025 & 0 \\ 0 & 2.717 \end{pmatrix}$$

Performances.

Operations	CPU Time (msec)
Encryption	24.425
Decryption	9.1485
Addition	0.203
Multiplication by constant	3.867
Circuit	9460.73

Securing the controller with our scheme successfully operates as you can see above.

Chapter 6

Summary

Securing CPSs is considered critical but quite difficult since there are various types of the networked control systems. The main purpose of this paper is to take the first step toward the secure CPS.

In this work, we suggest new HAE scheme to secure the simple version of the networked control system. Our idea is to use the tag, DFT and RLWE to have both secure and authenticated system. Our implementation can show this scheme is appliable to the physical system.

More potential studies can be done for the complementation of HAE to reduce rebooting the system. Also, this scheme can be more secure by changing the way to create the tag. Furthermore, applying a homomorphic encryption scheme allowing floating-point operations between ciphertext [**?**] can allow the signals to be real numbers.

국문초록

제어시스템은 다양한 물리적 문제들과 직접적인 관련이 있기 때문에 보안이 매우 중요하다. 보안을 위해서는 해커가 정보를 훔치거나 신호를 변형하지 못하도록 컨트롤러와 플랜트 사이의 신호, 그리고 컨트롤러 자체를 암호화하는 것이 필수적이다. 이와 관련된 연구로 해커가 정보를 습득하지 못하게 완전동형암호를 이용해 컨트롤러를 암호화하는 논문이 2016년에 발표되었다. 하지만, 해커가 정보를 습득하지 못하더라도 암호화된 상태에서 정보를 변형하거나 심한 경우 시스템을 망가뜨릴 수는 있다. 이 논문에서 우리는 플랜트가 컨트롤러로부터 받은 값이 바르게 계산된 값인지 확인할 수 있는 동형암호인증 스킴을 제안하였다. 라벨, 이산 푸리에 변환, 그리고 환에서의 완전동형암호를 이용하여 해커가 정보를 습득하는 것 뿐 아니라 변형하는 것도 어렵게 만들었다. 또한, 이 스킴을 직접 물탱크 모델에 구현해봄으로써 실제로 사용가능하다는 것을 확인하였다.

주요어휘: 동형암호인증, 제어 시스템, 이산 푸리에 변환, 동형암호
학번: 2015-20274