SECURITY ANALYSIS AND DESIGN OF ENCRYPTED CONTROL SYSTEMS USING HOMOMORPHIC ENCRYPTION

Kaoru Teranishi

A thesis submitted for the Degree of Doctor of Philosophy

Department of Mechanical and Intelligent Systems Engineering

The University of Electro-Communications

December, 2023

Acknowledgment

First and foremost, I would like to express my gratitude to my supervisor Dr. Kiminao Kogiso for his guidance and support throughout this doctoral student life. He gave me a precious study opportunity and invaluable experience in his laboratory.

I would also like to thank my thesis committee members. I am grateful to Dr. Osamu Kaneko and Dr. Kenji Sawada at The University of Electro-Communications for their support in conducting my doctoral studies as advisors.

I am also grateful to Dr. Jun Ueda who hosted me when I visited the Georgia Institute of Technology. The experience as a visiting scholar at his laboratory enriched my academic life and broadened my horizons. Additionally, he and his family fully supported me when I stayed in Atlanta.

The core idea of this thesis was born through discussions with Dr. Tomonori Sadamoto at The University of Electro-Communications when he and I visited the laboratory of Dr. Aranya Chakrabortty at North Carolina State University. I am grateful to them for their insightful comments and feedback.

My academic journey started at the National Institute of Technology, Ishikawa College. I would like to thank the faculty in the Department of Electronics and Information Engineering, in particular my previous advisor Dr. Naoki Shimada. He taught me the joy of research and encouraged me to pursue my doctoral studies. If I had not met him, I probably would not be writing this thesis.

I would also like to thank Shima Nakajima and Mariko Hayashibe for supporting my life in Kimilab. Furthermore, I thank the current and past members of Kimilab for all the discussions and remembrance. They made my university life fulfilling.

The world has undergone the COVID-19 pandemic, although we are now returning to our new normal. During the pandemic disruption, it was my friends who supported me. I am grateful to my old classmates from INCT and friends who met in the metaverse.

Last but not least, I am eternally grateful to my parents Yoshinori and Mariko and my brother Yuuki for their love, support, and understanding.

SECURITY ANALYSIS AND DESIGN OF ENCRYPTED CONTROL SYSTEMS USING HOMOMORPHIC ENCRYPTION

by

Kaoru Teranishi

Department of Mechanical and Intelligent Systems Engineering

The University of Electro-Communications

December, 2023

Abstract

Security and privacy are significant concerns in realizing dependable cyber-physical systems. To ensure the security and privacy of these systems, systems and control communities have been developing encrypted control protocols that utilize various cryptographic technologies. The primary focus of encrypted control in a client-server configuration is to securely outsource the computation of controllers to untrusted third parties using homomorphic encryption. This thesis aims to establish a systematic design method for encrypted control systems using homomorphic encryption in a client-server architecture. To this end, it addresses three essential tasks to face when designing encrypted control systems: i) determination of an appropriate cryptosystem for encrypted control, ii) design of a security parameter for the cryptosystem, and iii) design of a controller to be encrypted. The thesis proposes homomorphic encryption schemes that feature mechanisms for updating key pairs and demonstrates that these cryptosystems guarantee the forward and post-compromise security of encrypted control systems. Furthermore, it presents metrics to quantify the security level of encrypted control systems and formulates the definition of their security. Under the security definition, the thesis clarifies the minimum security parameter required to achieve the desired security level of encrypted control systems. Using the minimum security parameter reduces the computational burden owing to the encryption of control protocols. The thesis also shows that an H_2 optimal controller is effective in enhancing the security level. This connects the traditional controller design and the security of encrypted control systems. The results of this thesis provide a method for dealing with security as one of the control specifications and contribute to the further development of interdisciplinary research on control theory and cryptography.

Notations

$\mathbb{N}$	set of positive integers
$\mathbb{Z}$	set of integers
$\mathbb{Z}^+$	set of non-negative integers
$\mathbb{Z}_n$	set of non-negative integers less than n
$\mathbb{R}$	set of real numbers
$\mathbb{R}^+$	set of non-negative real numbers
$\emptyset$	empty set
A^n	set of n-dimentional vectors of which elements are in set A
$A^{m\times n}$	set of m-by-n matrices of which entries are in set A
$M^\top$	transpose of matrix M
M^{-1}	inverse of matrix M
M^+	pseudo inverse of matrix M
$\|v\|_\infty$	maximum norm of vector v
$\|M\|_F$	Frobenius norm of matrix M
$\|M\|_{\max}$	max norm of matrix M
$\mathrm{tr}(M)$	trace of matrix M
$\det(M)$	determinant of matrix M
$\mathrm{vec}(M)$	vectorization of matrix M
$\mathrm{diag}(x_1,\ldots,x_n)$	n-by-n diagonal matrix of which (i,i) entry is x_i
$\|A\|$	cardinality of set A
$\|x\|$	absolute value of $x \in \mathbb{R}$
$\lfloor x \rfloor$	maximum integer less than $x \in \mathbb{R}$
$\lfloor x \rceil$	nearest integer of $x \in \mathbb{R}$
$[a]_n$	residue of $a \in \mathbb{Z}$ modulo $n \in \mathbb{N}$
$[\![a]\!]_n$	minimal residue of $a \in \mathbb{Z}$ modulo $n \in \mathbb{N}$
$\Pr[A]$	probability of event A
$\Pr[A \mid B]$	conditional probability of A given B
$\mathbb{E}[X]$	expectation of random variable X
$\mathbb{E}[X \mid Y]$	conditional expectation of X given Y
$\mathcal{N}(\mu,\Sigma)$	Gaussian distribution with mean μ and variance Σ
$\chi(\sigma)$	discrete Gaussian distribution with mean zero and variance σ

$p(\cdot)$	probability density function
$p_{\mathcal{N}}(\cdot; \mu, \Sigma)$	probability density function of $\mathcal{N}(\mu, \Sigma)$
$\log_a(\cdot)$	logarithm function base a
$\ln(\cdot)$	natural logarithm function
$\exp(\cdot)$	exponential function
λ	security parameter
$\perp$	error symbol
$\mathcal{K}$	key space
$\mathcal{M}$	plaintext space
$\mathcal{C}$	ciphertext space
$\mathcal{A}$	probabilistic polynomial-time algorithm or adversary
params	public parameters
pk	public key
sk	secret key
m	plaintext
ct	ciphertext
ut	update token
poly	(positive) polynomial
negl	negligible function
KeyGen	key generation algorithm
Setup	setup algorithm
PubKeyGen	public-key generation algorithm
SecKeyGen	secret-key generation algorithm
Enc	encryption algorithm
Dec	decryption algorithm
Eval	homomorphic evaluation algorithm
KeyUpd	key update algorithm
CtUpd	ciphertext update algorithm
ScalSetup	scaling setup algorithm
Ecd	encoder algorithm
Dcd	decoder algorithm
EC	encrypted control algorithm
Game	cryptographic game
$\boxtimes$	ciphertext multiplication
$\boxplus$	ciphertext addition

$\boxdot$	plaintext-ciphertext multiplication
$\otimes$	Kronecker product
$x \leftarrow a$	a is assigned to x
$x \leftarrow_R A$	uniform sampling of element x from set A
$x \leftarrow_R D$	random sampling of x from probability distribution D

Contents

Chapter 1

Introduction

1.1 Background

Cyber-physical systems have revolutionized traditional systems for efficiency, reliability, and sustainability by integrating digital and physical worlds that consist of sensors, communication networks, computational components, and actuators [1, 2]. These systems monitor, control, and optimize physical processes with the potential to enhance our quality of life in energy, transportation, healthcare, agriculture, manufacturing, and beyond [3, 4].

Meanwhile, integrating communication and computation processes into a physical process often faces security and privacy risks owing to insecure communication channels and untrusted third parties to whom computation is outsourced [5–11]. Once confidential information is learned by an adversary in a cyber-physical system, the adversary can design sophisticated and undetectable attacks using a target system model constructed from the disclosed information [12, 13]. Therefore, information leakage within communication and computation is a primary concern in the security and privacy of cyber-physical systems.

Encryption is a common approach for realizing secure communication via public networks to prevent information leakage in cyber-physical systems. This entails sending sensor data to computational components via communication networks while encrypting the data. The computational components then decrypt the data, make control decisions for a physical process based on deciphered messages, and encrypt the decisions again. Finally, the actuators receive and decrypt the encryption of the controls to manipulate the physical processes. Although encryption effectively protects information during transmission, the computation layer is still vulnerable to threats that attempt to learn private information. This is because traditional encryption schemes require the decryption of encrypted messages before making control decisions, which can potentially expose information during computation.

In cryptography, various advanced techniques and tools have been developed

to protect private information in communication and computation. These cryptographic tools encompass a range of concepts, such as differential privacy, homomorphic encryption, secret sharing, garbled circuits, and secure multi-party computation. Each approach addresses distinct aspects of security and privacy during the computation.

Differential privacy. Differential privacy is a statistical approach for privacy-preserving data analysis in a database [14–18]. This approach provides a method to mathematically deal with privacy and balance data utility and privacy preservation. The fundamental idea behind differential privacy is to introduce randomness or perturbation into data queries or the output of computation, making it difficult to distinguish whether particular individual data are included in the database from its statistics. Differential privacy techniques are classified into two types: global and local. Global differential privacy mechanisms randomize responses to queries, whereas local differential privacy mechanisms randomize individual data before collection [19–21]. These techniques are widely used in data mining [22–24] and machine learning [25–29] but can also be beneficially applied to filtering and controlling dynamical systems [30–37].

Homomorphic encryption. Homomorphic encryption is an encryption technique that allows the direct arithmetic evaluation of encrypted data [38–40]. This enables data to be processed in its encryption form, preserving privacy and security in outsourcing computation to untrusted third parties. Several types of homomorphic encryption schemes exist, including partially homomorphic, somewhat homomorphic, leveled fully homomorphic, and fully homomorphic. Partially homomorphic encryption supports only a single operation, such as addition or multiplication, on encrypted data [41–46]. On the other hand, somewhat homomorphic and leveled fully homomorphic encryption schemes permit both addition and multiplication, making them more tractable [47–49]. However, these encryption schemes limit the number of operations that can be performed owing to the accumulation of noise in the encrypted data for every operation, which can sometimes result in incorrect decryption. Fully homomorphic encryption overcomes this limitation by introducing bootstrapping and offers the most excellent flexibility, namely the computation of any function on encrypted data [50–59]. The potential application areas of homomorphic encryption include cloud computing [60] and machine learning [61–64].

Secret sharing. Secret sharing is a cryptographic technique for distributing a secret among a group of parties by randomly dividing it into multiple shares [65–68]. A widely used secret-sharing scheme is the (t, n) threshold scheme, which splits a secret into n shares. The original secret can be retrieved by combining at least t shares, although any set of shares less than t reveals no information about the secret. Because of this property, secret sharing is suitable for storing highly confidential and critical information, such as an encryption key, as a subset of the involved parties can recover the secret even if some shares are lost. Furthermore, secret sharing is also applied to evaluate a function with multiple inputs in a multi-party computation scenario [69–72]. One of the participants in the computation distributes the inputs of a target function to the others via a secret sharing scheme. The other participants perform computations on their respective shares and return their results. The desired output of the target function is then recovered from the results by aggregating them. In this computation process, as long as the participants who received the shares do not collude, they cannot learn any information about the original inputs.

Garbled circuit. Garbled circuits are a cryptographic protocol for computing a desired function represented by a Boolean circuit with private inputs in a two-party computation scenario while keeping those inputs entirely concealed from one another [73–77]. Rather than relying on a trusted third party, the parties involved in this protocol work together to obtain a circuit evaluation result, as follows: Let Alice be a party that builds a circuit consisting of a single gate, and Bob be another party that wants to evaluate it. Alice prepares the truth table of the circuit and generates a garbled circuit that takes random bit-string labels as input instead of 0 or 1 by garbling the truth table. To evaluate the original circuit, Alice sends the garbled circuit and the label corresponding to her input to Bob, and he obtains the other label corresponding to his input from her by oblivious transfer. Bob then computes the garbled circuit with the labels and outputs the result. Here, oblivious transfers are a cryptographic protocol that allows a sender with multiple messages to send one of them to a receiver without learning which message is sent, and the receiver cannot learn the messages that were not sent [78, 79]. Some improvements in the efficiency of garbled circuits were discussed in [80–83].

Secure multi-party computation. Secure multi-party computation is a cryptographic technique in which multiple parties jointly compute a function with their inputs while preserving their security and privacy [84–86]. To this end, computation and communication protocols are developed for every party to obtain the desired computation result without learning any other information and relying on any trusted third party. Such protocols are typically based on various cryptographic primitives, such as homomorphic encryption [87, 88], secret sharing [72, 89], garbled circuits [90, 91], and oblivious transfer [92, 93].

Such cryptographic techniques are expected to improve the confidentiality of computation in cyber-physical systems. In this sense, encrypted control is an emerging research paradigm in the interdisciplinary area of control theory and cryptography that applies cryptographic tools, in particular homomorphic encryption, to control and make decisions for dynamical systems. Kogiso and Fujita initiated this paradigm as a strategy for enhancing the security of networked control systems [94]. They reconstructed a linear time-invariant controller in the form of a matrix-vector product. They then demonstrated performing the controller computation over encrypted data without decryption, using multiplicatively homomorphic encryption. Following their work, Farokhi et al. and Kim et al. suggested utilizing additively and fully homomorphic encryption in encrypted control, respectively [95–97]. The research field on encrypted control has continued to advance, with recent studies falling into two distinct categories: encrypted control in client-server models and encrypted control in multi-agent systems. For those who want to delve deeper into encrypted control, a tutorial is available in [98].

Encrypted control in client-server models. The early studies on encrypted control have resulted in encrypting various advanced methods, such as dynamic control [99–105], polynomial control [106, 107], nonlinear control [108–110], gain-scheduled control [111], event-triggered control [108,112], discrete-event control [113], motion control [114–117], bilateral control [118, 119], data-driven control [120], learning-based control [121, 122], model predictive control [123–129], state estimation [130–132], filtering [133], quadratic optimization [134–136], and machine learning [137–139]. However, the encryption of controllers may lead to destabilization and performance degradation owing to quantization errors in the encryption. Thus, the robust and asymptotic stabilities and performance of encrypted

control systems under quantizers have been studied [96, 108, 111, 112, 140–146]. Additionally, control systems with a controller having an integer state matrix have been discussed [147–149]. Some studies have shown that encrypted control systems are vulnerable to several attacks [150–154]. To improve their integrity, additional functionalities in encrypted control systems have been investigated, including detector [155–158], key-switching mechanism [159, 160], authenticated computation [161], resilient homomorphic encryption [162–164], and keyed homomorphic encryption [165]. Furthermore, encrypted control methods have been implemented on Raspberry Pi [166, 167], FPGA [168, 169], drone [170], linear stage system [115, 171], pneumatic system [119, 157], and robot manipulator [117, 172].

Encrypted control in multi-agent systems. The implementation of distributed control protocols utilizing cryptographic tools to protect the state or weight of each agent in multi-agent systems is another focus of the encrypted control paradigm. This study area is intriguing and involves various control strategies, including consensus control [173–177], formation control [178, 179], cooperative control [180, 181], distributed state estimation [182, 183], and distributed Kalman filtering [184]. Moreover, secure distributed optimization has been studied in [185–191]. Distributed algorithms often incorporate an aggregation process to integrate local private data, with secure aggregation being investigated using homomorphic encryption in [180, 190, 192–195]. In addition, similar to encrypted control in client-server models, encrypted control with quantizers has also been discussed in consensus and formation control protocols [174, 179].

1.2 Contributions

The goal of this thesis is to establish a systematic design method for encrypted control systems using homomorphic encryption in a client-server model. The encrypted control system comprises a cryptosystem, controller, and plant, in which the plant is predetermined before the design. Hence, the following essential questions must be addressed when designing encrypted control systems:

- What type of encryption scheme is appropriate for encrypted control?
- How can we design a security parameter for the used encryption scheme?
- How can we design a controller to be encrypted?

Appropriate encryption scheme for encrypted control. The first question involves the real-time operation of encrypted control systems. A control decision must be fed back to the plant in real-time to guarantee the performance and stability of the control system. Nevertheless, the encryption of control protocols increases the computational burden, potentially spoiling the real-time computation of control systems. Hence, a security parameter as large as that in traditional information and communication systems would not be applicable to encrypted control systems. Note that a security parameter is a parameter for cryptosystems that specifies their security strength and affects their processing time.

A small security parameter generally increases the risk of adversaries compromising the secret key of a cryptosystem. Suppose the secret key used in an encrypted control system falls into the hands of an adversary. In that case, the adversary can recover all past and future messages transmitted between the plant and the controller server. One possible countermeasure to mitigate this vulnerability is to regenerate the public and secret keys of the cryptosystem. However, this countermeasure is not preferred in encrypted control systems because it requires downloading the parameters of the encrypted controller, decrypting and encrypting them using the old secret and new public keys, and then uploading them again, which induces additional communication effort.

To resolve the potential vulnerability of encrypted control systems, this thesis proposes two encryption schemes with key update mechanisms: updatable homomorphic encryption and key-updatable homomorphic encryption. Updatable homomorphic encryption is a variant of homomorphic encryption inspired by updatable encryption [196]. This encryption scheme generates a token from previous and new secret keys and updates ciphertexts encrypted by a previous public key into those corresponding to a new public key using the token instead of re-encrypting them. This simplifies the key update process because a plant only needs to send a token to the controller server. The thesis formulates updatable homomorphic encryption and constructs it from basic computational assumptions. It also presents a computational security notion and mathematically proves the security of updatable homomorphic encryption based on this notion.

Furthermore, key-updatable homomorphic encryption is formulated and constructed to improve the efficiency and security of updatable homomorphic encryption. This encryption scheme is based on multi-key homomorphic encryption [54] and can evaluate the arithmetic of ciphertexts encrypted by distinct keys. With

key-updatable homomorphic encryption, the secret key can be updated without requiring a token to update the ciphertexts in the controller server. This thesis demonstrates the security of key-updatable homomorphic encryption in the same manner as updatable homomorphic encryption.

Security parameter design. Once a cryptosystem is selected for encrypted control systems, the next step is determining an appropriate security parameter for implementing the cryptosystem. The National Institute of Standards and Technology (NIST) provides recommendations for selecting security parameters and key sizes [197]. The security parameters in most conventional studies on encrypted control systems were chosen to follow the recommendations as with information and communication systems. However, as already mentioned, selecting such a security parameter is not necessarily possible in practice because of the real-time computation requirements.

If the recommendations are not followed, a security parameter should be selected for encrypted control systems to satisfy sufficient security strength based on certain criteria. Therefore, it is necessary to develop a reliable measure for quantifying the security level of encrypted control systems. Although some recent studies have introduced various security metrics for control systems [7, 198–206], they are not suitable for encrypted control scenarios. This is because they have focused on quantifying the impact of attacks on control performance or the applicability of undetectable attacks and do not capture the confidentiality of encrypted control systems.

This thesis considers a disclosure attack to learn the system parameters of a plant or a closed-loop system, which arises from network eavesdroppers and an untrusted controller server. The adversaries collect the encrypted trajectories of a target system and subsequently estimate the system parameters using the data obtained by decrypting the encrypted trajectories. In this attack scenario, the thesis proposes two metrics to assess the security of encrypted control systems. The first metric, sample identifying complexity, is the minimum sample size required for an estimation error in the adversary's estimation to become smaller than a certain threshold. The second metric, sample deciphering time, is the computation time required to recover the original data from the encrypted trajectories.

Using the security metrics, the thesis provides a security definition tailored for encrypted control systems. It also reveals the minimum security parameter that can achieve the desired security level of an encrypted control system when the attack

target is a plant. The minimum security parameter effectively balances the trade-off between security strength and computational effort owing to encryption. Moreover, the thesis offers a suboptimal security parameter that achieves the desired security level, for which the design is more tractable than the minimum one.

Controller design. A controller affects the dynamics of a closed-loop system. Hence, when an attack target is a closed-loop system, the security level of encrypted control systems must depend on not only a security parameter but also a controller. Although designing an appropriate controller can improve the security level, conventional studies on encrypted control regard the controller as a given parameter. Therefore, designing an appropriate controller for encrypted control systems remains challenging from the perspective of security.

This thesis proposes a design method for a state-feedback controller oriented toward the security of encrypted control systems. The designed controller improves the security level of encrypted control systems by increasing the difficulty of the adversary's parameter estimation in terms of sample identifying complexity. A security parameter for a closed-loop system is then designed using the controller. Furthermore, the thesis clarifies the connection between traditional controller design and the security of encrypted control systems by showing that an H_2 optimal controller can serve as a suboptimal controller for the security level. It also demonstrates the validity of the controller and security parameter design through numerical simulations.

1.3 Organization

Chapter 2 reviews the foundation of cryptography for encrypted control. It begins by defining public-key encryption and its correctness through polynomial-time algorithms and a negligible function. The chapter also formulates a provable security notion called indistinguishability under chosen plaintext attacks via a cryptographic game and introduces well-known computational problems, the decisional Diffie-Hellman problem and the learning with errors problem, that can be used to construct provably secure cryptosystems. Homomorphic encryption is then defined as an extension of public-key encryption. The chapter provides examples of multiplicatively and additively holomorphic encryption schemes, namely the ElGamal and Regev encryption schemes, and shows their correctness, homomorphism, and se-

curity. Moreover, this chapter proposes updatable and key-updatable homomorphic encryption to improve the forward and post-compromise security of homomorphic encryption. These variants are constructed as with the ElGamal and Regev encryption schemes and are demonstrated to satisfy a notion of indistinguishability.

Chapter 3 introduces an encoder and decoder that bridges real numbers and plaintexts. These tools are utilized to handle real-valued data in homomorphic encryption. The chapter then analyzes a quantization error induced by the encoder and decoder. It shows that the decoder preserves multiplication and addition in a plaintext space as long as overflow does not occur. This feature enables the encoder and decoder to inherit homomorphism. Using the encoder and decoder, this chapter presents a unified definition of encrypted control, along with its accuracy notion. In addition, it constructs encrypted control algorithms for a linear time-invariant controller by using multiplicatively and additively homomorphic encryption. The ElGamal and Regev encryption schemes are used to realize the encrypted control algorithms. Finally, the definitions and constructions of encrypted control using homomorphic encryption are extended, even when updating public and secret keys.

Chapter 4 focuses on exploring a security definition tailored to encrypted control systems in a client-server framework. The chapter formulates the attack scenario considered in this thesis while comparing the differences between conventional secure communication and encrypted control from the perspective of a threat model and security goal. In this attack scenario, the objective of an adversary, whether an eavesdropper on a communication channel or a malicious server running an encrypted control algorithm, is to identify plant or closed-loop system parameters. To define the security of encrypted control systems under this attack scenario, this chapter considers the difficulty of system identification and the computation time required. A type of sample complexity is used to quantify the difficulty of the identification itself, whereas deciphering time is employed to measure the difficulty of breaking encryption to obtain a dataset used in the identification process. In addition, the security level of encrypted control systems is quantified through an acceptable estimation error and the desired protection period determined by a system designer.

Chapter 5 proposes a design method for a security parameter and a controller in encrypted control systems. The aim is to attain the desired security level while reducing the computational burden due to encryption. The chapter begins by considering four parameter estimation algorithms commonly employed for system iden-

tification. This indicates that these algorithms can be unified with reasonable assumptions. Under an adversary using a unified estimator, the optimal security parameter of an encrypted control system is designed when the attack target is a plant. Although this involves solving an optimization problem, directly computing the solution requires significant computational resources due to repeated iterations. Therefore, this chapter also discusses the design of a suboptimal security parameter and controller based on a lower bound of sample identifying complexity for each attack target, namely a plant and a closed-loop system, to overcome this optimization challenge. Moreover, numerical examples are provided to demonstrate the effectiveness of the proposed design method.

Chapter 6 concludes this thesis by complementing the technical and intellectual contributions and providing remarks on future research directions.

Chapter 2

Cryptographic Foundations

The focus of this chapter is on providing cryptographic foundations for encrypted control. The syntax and properties of public-key and homomorphic encryption are introduced to formulate cryptosystems and rigorously prove their security. The security of homomorphic encryption schemes is shown through cryptographic game-based reduction. Furthermore, this chapter presents several modifications for homomorphic encryption that achieve a stronger security notion than conventional homomorphic encryption schemes.

2.1 Public-key encryption

Public-key encryption is an asymmetric technique for private communication between two parties. Fig. 2.1 illustrates a standard problem setting in public-key encryption described below. Suppose Alice is a sender, and Bob is a receiver. Bob prepares encryption and decryption keys, called public and secret keys, respectively, and shares the public key with Alice before communication. Alice encrypts a message with the public key and sends the encrypted message to Bob. Bob receives and decrypts the encrypted message with the secret key. In this communication, we want to ensure that an adversary, Eve, eavesdropping on the communication cannot learn about Alice's message even though the communication channel is public, i.e., the adversary can obtain the public key.

This section provides a formal definition of public-key encryption and its security. Additionally, the section introduces two fundamental notions called a polynomial-time algorithm and a negligible function.

2.1.1 Definitions

This section begins by defining a class of efficient algorithms to formulate the hardness of computation. The running time of an algorithm basically increases depending on its input size. Thus, this section defines the efficiency of algorithms for their

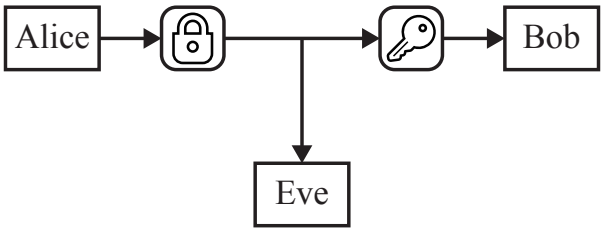


Fig. 2.1: Communication using public-key encryption under the adversary eavesdropping on the communication channel.

growth of running time. More precisely, an algorithm is considered efficient if its running time is bounded by a polynomial.

Definition 2.1 (Polynomial-time algorithm [207])**.** *An algorithm runs in polynomial time if, for every n-bit input, there exists a polynomial* poly *such that the algorithm terminates within at most* $\mathsf{poly}(n)$ *steps. A polynomial-time algorithm is an algorithm that runs in polynomial time.*

We are now ready to define the syntax of public-key encryption to specify its inputs, outputs, and operations. The syntax enables rigorous analysis of the properties and security of encryption schemes. In what follows, $y \leftarrow \mathsf{A}(x)$ denotes that the value of $\mathsf{A}(x)$ is assigned to the variable y. x and y are referred to as the input and output of A, respectively. Additionally, key, plaintext (message), and ciphertext (encrypted message) spaces are denoted by $\mathcal{K}$, $\mathcal{M}$, and $\mathcal{C}$, respectively.

Definition 2.2 (Public-key encryption)**.** *A public-key encryption scheme is a tuple of polynomial-time algorithms* KeyGen*,* Enc*, and* Dec *such that:*

- *Key generation: The key generation algorithm* $(\mathsf{params}, \mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ *consists of polynomial-time algorithms* Setup*,* $\mathsf{SecKeyGen}$*, and* $\mathsf{PubKeyGen}$*. The setup algorithm* $\mathsf{params} \leftarrow \mathsf{Setup}(1^\lambda)$ *takes as input a security parameter* $\lambda \in \mathbb{N}$ *and outputs public parameters* params*. Although* params *is input to all other algorithms, we omit it for simplicity in the following. The secret-key generation algorithm* $\mathsf{sk} \leftarrow \mathsf{SecKeyGen}()$ *outputs a secret key* sk*. The public-key generation algorithm* $\mathsf{pk} \leftarrow \mathsf{PubKeyGen}(\mathsf{sk})$ *takes as input the secret key* sk *and outputs a public key* pk*.*

- *Encryption: The encryption algorithm* $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$ *takes as input the public key* pk *and a plaintext* $\mathsf{m} \in \mathcal{M}$ *and outputs a ciphertext* $\mathsf{ct} \in \mathcal{C}$*.*

- *Decryption: The decryption algorithm* $\mathsf{m} \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ *takes as input the secret key* sk *and a ciphertext* $\mathsf{ct} \in \mathcal{C}$ *and outputs either a plaintext* $\mathsf{m} \in \mathcal{M}$ *or error symbol* $\perp$*.*

Note that the input of the key generation algorithm, 1^λ, is the unary representation of security parameter λ. The unary representation expresses a natural number using a sequence of 1. For example, $1^3 = 111$ and $1^5 = 11111$ are 3 and 5 in the decimal form, respectively. The unary representation is used to specify the input length of the key generation algorithm. In other words, the running time of the key generation algorithm is at most $\mathsf{poly}(\lambda)$ steps.

The key generation and encryption algorithms are typically probabilistic to randomize a secret key and a ciphertext. Hence, the decryption algorithm must almost always recover the original plaintext correctly. This property can be formulated as the probability of failure decryption being sufficiently small. Meanwhile, such probability depends on a security parameter because public and secret keys are generated by a key generation algorithm taking as input the security parameter. Thus, a function that is negligibly small with respect to its input is defined.

Definition 2.3 (Negligible function [207])**.** *A function* $\mathsf{negl} : \mathbb{N} \to \mathbb{R}^+$ *is negligible if, for every positive polynomial* poly*, there exists* $N \in \mathbb{N}$ *such that* $\mathsf{negl}(n) < 1/\mathsf{poly}(n)$ *holds for all* $n > N$*. A negligible function is a function that is negligible.*

A negligible function approaches zero faster than any positive polynomial as its input grows. For instance, consider the function 2^{-n} with the input n. This is an example of a negligible function that becomes rapidly small as n increases. With a negligible function, the condition of correct decryption is defined as follows.

Definition 2.4 (Correctness)**.** *A public-key encryption scheme in Definition 2.2 is correct if there exists a negligible function* negl *such that*

$$\Pr\left[\mathsf{m}' = \mathsf{m} \;\middle|\; \begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}) \\ \mathsf{m}' \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda)$$

for all $\lambda \in \mathbb{N}$ *and for all* $\mathsf{m} \in \mathcal{M}$*.*

The probability that the output of the decryption algorithm is equal to the original plaintext rapidly converges to one as the security parameter increases. Thus,

the decryption algorithm of a correct public-key encryption scheme almost always recovers the original plaintext when a security parameter is sufficiently large.

2.1.2 Security

Modern cryptography considers security only against efficient adversaries whose computational power is bounded by a polynomial. Such security is called computational security. Similarly to a polynomial-time algorithm, the adversary is referred to as a polynomial-time adversary. Additionally, such adversaries can behave in a probabilistic manner. Thus, the security notion considers guaranteeing the success probability of attacks is negligibly small. It should be noted that limiting the adversaries to efficient ones is reasonable in practice because the computation of adversaries is typically more efficient than parties who communicate with each other using a public-key encryption scheme. Recall that public-key encryption in Definition 2.2 consists of polynomial-time algorithms.

To analyze the security of encryption schemes, it is necessary to formulate a threat model and security goal with a rigorous mathematical approach. To this end, the game-based proof is employed. This section formally defines a game capturing a cryptographic protocol under the attack of an adversary by using pseudocode. The security proof demonstrates that the advantage of an adversary in winning the game is negligibly small under some computational assumptions. The game defined below formulates the most fundamental security notion, indistinguishability under chosen-plaintext attacks (IND-CPA), in public-key encryption. In what follows, $x \leftarrow_R X$ denotes uniform sampling of an element x from a set X. Similarly, we use the same symbol for random sampling of x from X if X is a probability distribution.

Definition 2.5 (IND-CPA [207])**.** *Consider a public-key encryption scheme* $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *and adversary* $\mathcal{A}$*. Define the game* $\mathsf{Game}^{\mathsf{IND-CPA}}_{\Pi,\mathcal{A}}(\lambda)$ *as follows.*

$\mathsf{Game}^{\mathsf{IND-CPA}}_{\Pi,\mathcal{A}}(\lambda)$

$(\mathsf{params}, \mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$

$(\mathsf{m}_0, \mathsf{m}_1) \leftarrow \mathcal{A}(1^\lambda, \mathsf{params}, \mathsf{pk})$

$b \leftarrow_R \{0, 1\}$

$\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_b)$

$\hat{b} \leftarrow \mathcal{A}(\mathsf{ct})$

return 1 **if** $\hat{b} = b$ **and** 0 **otherwise**

Setup. *Public parameters* params *and a key pair* $(\mathsf{pk}, \mathsf{sk})$ *are generated by running the key generation algorithm* $\mathsf{KeyGen}(1^\lambda)$.

Challenge. $\mathcal{A}$ *takes as input* 1^λ, params, *and* pk *and outputs plaintexts* $\mathsf{m}_0, \mathsf{m}_1 \in \mathcal{M}$ *of the same length. A bit* $b \in \{0,1\}$ *is chosen uniformly. A ciphertext* ct *is computed by running the encryption algorithm* $\mathsf{Enc}(\mathsf{pk}, \mathsf{m}_b)$.

Guess. $\mathcal{A}$ *takes as input* ct *and outputs a bit* $\hat{b} \in \{0,1\}$. *The game outputs* 1 *if* $\hat{b} = b$ *and* 0 *otherwise.*

We say Π *is IND-CPA secure if there exists a negligible function* negl *such that*

$$\left| \Pr\left[\mathsf{Game}^{\mathsf{IND-CPA}}_{\Pi,\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2} \right| < \mathsf{negl}(\lambda)$$

for all $\lambda \in \mathbb{N}$ *and for all probabilistic polynomial-time adversary* $\mathcal{A}$.

The left-hand side of inequality in the definition is called the advantage of adversary $\mathcal{A}$. If the advantage is negligibly small, that is, the probability of the adversary winning the game is sufficiently close to that of random guess, then such an encryption scheme is IND-CPA secure. This security definition implies that any probabilistic polynomial-time adversary cannot learn about any partial information of a plaintext from a ciphertext. It should be noted that the adversary in the game takes as input the security parameter λ, i.e., the running time of the adversary is bounded by $\mathsf{poly}(\lambda)$ steps. In other words, the security definition covers only efficient adversaries. Moreover, in the game, the adversary can access the public parameters, public key, and the encryption of m_b, referred to as the challenge ciphertext. This formulation reflects the desired property of public-key encryption described in the introduction of this chapter.

At the end of this section, we introduce two computational problems usually assumed to be hard in several constructions of encryption schemes. In what follows, the modular reduction of $x \in \mathbb{Z}$ modulo $n \in \mathbb{N}$ is denoted by $[x]_n$. Similarly, for a vector $v \in \mathbb{Z}$ and matrix $M \in \mathbb{Z}$, $[v]_n$ and $[M]_n$ represents the vector and matrix obtained by the modular reduction of each element of v and M, respectively.

Definition 2.6 (DDH problem [207])**.** *Given a cyclic group* G *of order* $q = q(\lambda)$. *Let* g *be a generator of* G, *and let* x, y, z *be random numbers uniformly sampled from* $\mathbb{Z}_q$. *The decisional Diffie-Hellman (DDH) problem is to distinguish* (g^x, g^y, g^z) *and*

(g^x, g^y, g^{xy}). *The DDH problem is hard if there exists a negligible function* negl *such that*

$$|\Pr[\mathcal{A}(G, q, g, g^x, g^y, g^z) = 1] - \Pr[\mathcal{A}(G, q, g, g^x, g^y, g^{xy}) = 1]| < \mathsf{negl}(\lambda)$$

for all $\lambda \in \mathbb{N}$ *and for all probabilistic polynomial-time algorithm* $\mathcal{A}$. *The DDH assumption is an assumption that the DDH problem is hard.*

Definition 2.7 (LWE problem [208])**.** *Let* $m = m(\lambda)$, $n = n(\lambda)$, *and* $q = q(\lambda) \geq 2$ *be positive integers, and let* $\chi = \chi(\lambda)$ *be a probability distribution over* $\mathbb{Z}$. *Given a uniformly random matrix* $A \in \mathbb{Z}_q^{m \times n}$, *uniformly random vectors* $s \in \mathbb{Z}_q^n, u \in \mathbb{Z}_q^m$, *and an error vector* $e \in \mathbb{Z}^m$ *sampled from* χ^m. *The (decisional) learning with errors (LWE) problem is to distinguish* $(A, [As + e]_q)$ *and* (A, u). *The LWE problem is hard if there exists a negligible function* negl *such that*

$$\left|\Pr[\mathcal{A}(A, [As + e]_q) = 1] - \Pr[\mathcal{A}(A, u) = 1]\right| < \mathsf{negl}(\lambda)$$

for all $\lambda \in \mathbb{N}$ *and for all probabilistic polynomial-time algorithm* $\mathcal{A}$. *The LWE assumption is an assumption that the LWE problem is hard.*

Note that the hardness of DDH and LWE problems depends on the choice of the cyclic group G and the parameters m, n, q, χ, respectively. We assume that an appropriate group and parameters are selected when the DDH and LWE assumptions are made. The readers of interest can refer to [209,210] and their references for more details.

2.2 Homomorphic encryption

Homomorphic encryption is an emerging cryptographic tool for secure outsourcing computation. Roughly speaking, homomorphic encryption allows one to compute some arithmetic directly on encrypted data without decryption. Fig. 2.2 depicts a client-server model of outsourcing computation using homomorphic encryption. The client aims to outsource the computation of a function, whose inputs are private data of the client, to the server. Meanwhile, the client wishes to keep the data secret against the server and eavesdropper in the network for privacy. The client then transmits the function and private data to the server while encrypting the data by homomorphic encryption. The server can compute and return an output

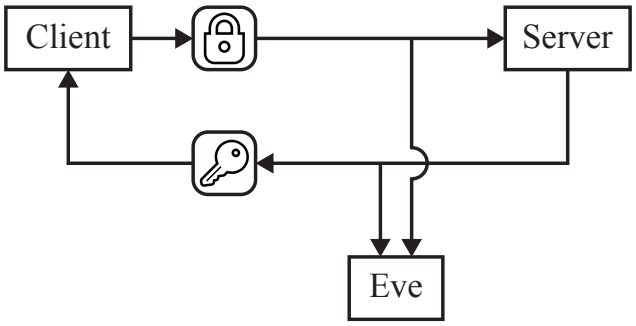


Fig. 2.2: Outsourcing computation using homomorphic encryption under the adversary eavesdropping on the communication channel.

of the function with the received ciphertexts thanks to the ability of homomorphic encryption. Hence, the client achieves its goal by decrypting the response, namely outsourcing computation while keeping its privacy.

This section serves as a mathematical formulation of homomorphic encryption and its properties. Furthermore, the section introduces two concrete constructions of homomorphic encryption based on the DDH and LWE problems.

2.2.1 Definitions

The formulation of homomorphic encryption begins with defining its syntax. The definition is an extension of public-key encryption in Definition 2.2 with an additional polynomial-time algorithm for homomorphic evaluation.

Definition 2.8 (Homomorphic encryption)**.** *A homomorphic encryption scheme is a tuple of* KeyGen*,* Enc*, and* Dec *in Definition 2.2, and a polynomial-time algorithm* Eval *such that:*

- *Homomorphic evaluation: The homomorphic evaluation algorithm* $\mathsf{ct} \leftarrow \mathsf{Eval}(f, \mathsf{ct}_1, \mathsf{ct}_2)$ *takes as input a binary operation* f *and two ciphertexts* $\mathsf{ct}_1, \mathsf{ct}_2 \in \mathcal{C}$ *and outputs a ciphertext* $\mathsf{ct} \in \mathcal{C}$*.*

Fig. 2.3 illustrates a schematic picture of the relationship among the encryption, decryption, and homomorphic evaluation algorithms. It is required that the output of homomorphic evaluation algorithm $\mathsf{Eval}(f, \mathsf{ct}_1, \mathsf{ct}_2)$ is almost always decrypted to a corresponding plaintext $f(\mathsf{m}_1, \mathsf{m}_2)$ correctly, where $\mathsf{ct}_1 \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_1)$, and $\mathsf{ct}_2 \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_2)$. Similar to the correctness in Definition 2.4, this property of homomorphic encryption is defined as follows.

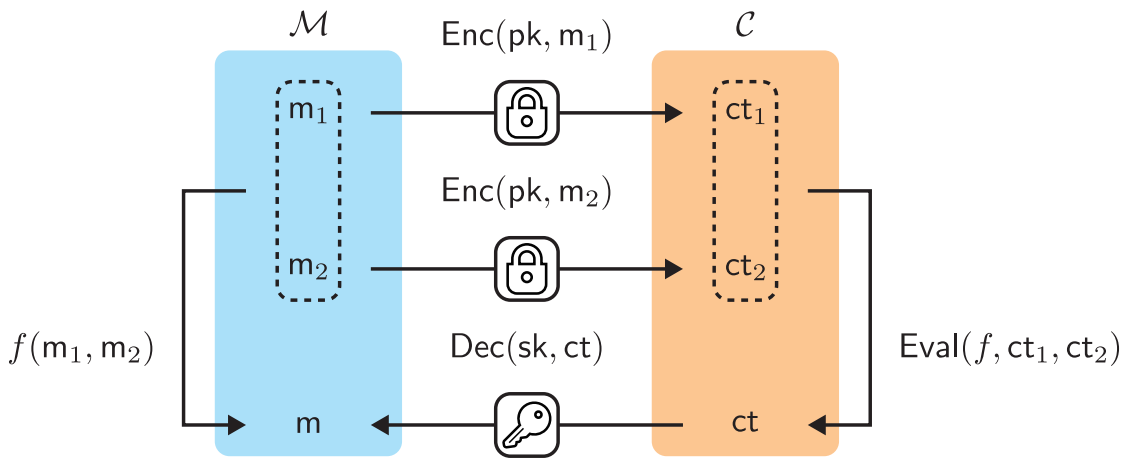


Fig. 2.3: Schematic picture of the relationship among encryption, decryption, and homomorphic evaluation algorithms in homomorphic encryption.

Definition 2.9 (Homomorphism)**.** *Let f be a binary operation. A homomorphic encryption scheme in Definition 2.8 is homomorphic for f if there exists a negligible function* negl *such that*

$$\Pr\left[\mathsf{m}' = f(\mathsf{m}_1, \mathsf{m}_2) \;\middle|\; \begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_i),\ i = 1, 2 \\ \mathsf{ct} \leftarrow \mathsf{Eval}(f, \mathsf{ct}_1, \mathsf{ct}_2) \\ \mathsf{m}' \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda)$$

for all $\lambda \in \mathbb{N}$ and for all $\mathsf{m}_1, \mathsf{m}_2 \in \mathcal{M}$.

Homomorphic encryption is classified into some types according to a binary operation in its homomorphism. If an encryption scheme is homomorphic for either multiplication $\times$ or addition $+$, it is called a multiplicatively or additively homomorphic encryption scheme, respectively. In addition, if the scheme is homomorphic for both multiplication and addition with a limited number of operations, it is called somewhat or leveled fully homomorphic encryption. If the scheme has no limitations for both operations, it is referred to as fully homomorphic encryption.

Define the binary operations over a ciphertext space, $\boxtimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C} : (\mathsf{ct}_1, \mathsf{ct}_2) \mapsto \mathsf{Eval}(\times, \mathsf{ct}_1, \mathsf{ct}_2)$ and $\boxplus : \mathcal{C} \times \mathcal{C} \to \mathcal{C} : (\mathsf{ct}_1, \mathsf{ct}_2) \mapsto \mathsf{Eval}(+, \mathsf{ct}_1, \mathsf{ct}_2)$, for the sake of simplicity. Using the binary operations, it holds that

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}_1 \boxtimes \mathsf{ct}_2) = \mathsf{m}_1 \times \mathsf{m}_2,$$

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}_1 \boxplus \mathsf{ct}_2) = \mathsf{m}_1 + \mathsf{m}_2,$$

for multiplicatively and additively homomorphic encryption schemes except with a negligible probability, where $\mathsf{ct}_1 \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_1)$ and $\mathsf{ct}_2 \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_2)$. Furthermore, the homomorphic addition allows computing multiplication between plaintext and ciphertext as

$$\mathsf{Dec}(\mathsf{sk}, \underbrace{\mathsf{ct}_2 \boxplus \cdots \boxplus \mathsf{ct}_2}_{\mathsf{m}_1 \text{ times}}) = \underbrace{\mathsf{m}_2 + \cdots + \mathsf{m}_2}_{\mathsf{m}_1 \text{ times}} = \mathsf{m}_1 \times \mathsf{m}_2.$$

Define the operation $\boxdot : \mathcal{M} \times \mathcal{C} \to \mathcal{C}$ to denote the plaintext-ciphertext multiplication, namely $\mathsf{Dec}(\mathsf{sk}, \mathsf{m}_1 \boxdot \mathsf{ct}_2) = \mathsf{m}_1 \times \mathsf{m}_2$.

2.2.2 Construction from DDH

Following the syntax and homomorphism definitions, the following sections introduce two concrete constructions of homomorphic encryption. The first construction is the ElGamal encryption [43] that relies on the hardness of the DDH problem in Definition 2.6. The encryption scheme is a multiplicatively-homomorphic encryption scheme constructed as follows.

Definition 2.10 (ElGamal encryption)**.** *The algorithms in Definition 2.8 for the ElGamal encryption are as follows.*

- *Setup: Let $q = q(\lambda)$ and $p = p(\lambda)$ be prime numbers such that $p = nq + 1$ and $n \geq 2$. Randomly compute a generator g of a cyclic group $\mathbb{G} = \{[g^i]_p \mid i \in \mathbb{Z}_q\}$ such that $[g^q]_p = 1$. Output $\mathsf{params} = (p, q, g)$. The plaintext and ciphertext spaces are $\mathcal{M} = \mathbb{G}$ and $\mathcal{C} = \mathbb{G}^2$, respectively.*
- *Secret key generation: Choose $s \leftarrow_R \mathbb{Z}_q$. Output $\mathsf{sk} = s$.*
- *Public key generation: Set $s \leftarrow \mathsf{sk}$. Output $\mathsf{pk} = [g^s]_p$.*
- *Encryption: Set $h \leftarrow \mathsf{pk}$. Choose $r \leftarrow_R \mathbb{Z}_q$. Output $\mathsf{ct} = ([g^r]_p, [\mathsf{m}h^r]_p)$.*
- *Decryption: Parse $\mathsf{ct} = (c_1, c_2)$. Set $s \leftarrow \mathsf{sk}$. Output $\mathsf{m} = \left[c_1^{-s} c_2\right]_p$.*
- *Homomorphic evaluation: Parse $\mathsf{ct}_1 = (c_{11}, c_{12})$, and $\mathsf{ct}_2 = (c_{21}, c_{22})$. Output $\mathsf{ct} = ([c_{11}c_{21}]_p, [c_{12}c_{22}]_p)$.*

The ElGamal encryption satisfies the correctness in Definition 2.4 and homomorphism in Definition 2.9 with respect to multiplication. The proofs of these properties are shown below.

Proposition 2.1. *The ElGamal encryption is correct.*

Proof. Let params, pk, and sk be as in Definition 2.10. A ciphertext of $\mathsf{m} \in \mathbb{G}$ is given as

$$\mathsf{ct} = \mathsf{Enc}(\mathsf{pk}, \mathsf{m}) = \Big([g^r]_p, [\mathsf{m}(g^s)^r]_p\Big) = \Big([g^r]_p, [\mathsf{m}g^{rs}]_p\Big),$$

where $r \in \mathbb{Z}_q$ is a random number. The decryption of ct is computed as

$$\mathsf{m}' = \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = \left[(g^r)^{-s}\mathsf{m}g^{rs}\right]_p = \left[\mathsf{m}g^{-rs}g^{rs}\right]_p = \left[\mathsf{m}g^{rs-rs}\right]_p = [\mathsf{m}]_p = \mathsf{m}.$$

This implies that $\Pr[\mathsf{m}' = \mathsf{m}] = 1$. □

Proposition 2.2. *Let p be as in Definition 2.10. The ElGamal encryption is homomorphic for multiplication modulo p.*

Proof. Let params, pk, and sk be as in Definition 2.10. Let $f : \mathcal{M} \times \mathcal{M} \to \mathcal{M} : (\mathsf{m}_1, \mathsf{m}_2) \mapsto [\mathsf{m}_1\mathsf{m}_2]_p$. Ciphertexts of $\mathsf{m}_1, \mathsf{m}_2 \in \mathbb{G}$ are given as

$$\begin{aligned}\mathsf{ct}_1 &= \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_1) = \Big([g^{r_1}]_p, [\mathsf{m}_1 g^{r_1 s}]_p\Big),\\ \mathsf{ct}_2 &= \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_2) = \Big([g^{r_2}]_p, [\mathsf{m}_2 g^{r_2 s}]_p\Big),\end{aligned}$$

where $r_1, r_2 \in \mathbb{Z}_q$ are random numbers. The output of homomorphic evaluation is obtained as

$$\begin{aligned}\mathsf{ct} = \mathsf{Eval}(f, \mathsf{ct}_1, \mathsf{ct}_2) &= \Big([g^{r_1}g^{r_2}]_p, [\mathsf{m}_1 g^{r_1 s}\mathsf{m}_2 g^{r_2 s}]_p\Big),\\ &= \Big(\left[g^{r_1+r_2}\right]_p, \left[\mathsf{m}_1\mathsf{m}_2 g^{(r_1+r_2)s}\right]_p\Big),\\ &= \Big([g^r]_p, [\mathsf{m}_1\mathsf{m}_2 g^{rs}]_p\Big),\end{aligned}$$

where $r = r_1 + r_2$. It follows from the proof of Proposition 2.1 that the decryption of ct is computed as

$$\mathsf{m}' = \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = [\mathsf{m}_1\mathsf{m}_2]_p = f(\mathsf{m}_1, \mathsf{m}_2).$$

This implies that $\Pr[\mathsf{m}' = f(\mathsf{m}_1, \mathsf{m}_2)] = 1$. □

2.2.3 Construction from LWE

The next construction of homomorphic encryption is the Regev encryption [208]. The encryption scheme is an additively homomorphic encryption scheme based on the hardness of the learning with errors (LWE) problem, and hence it is sometimes called LWE-based encryption. The construction is as follows.

Definition 2.11 (Regev encryption [208])**.** *The algorithms in Definition 2.8 for the Regev encryption are as follows.*

- *Setup: Let $m = m(\lambda)$, $n = n(\lambda)$, $t = t(\lambda) \geq 2$, and $q = q(\lambda) \gg t$ be integers, and let $\chi = \chi(\sigma)$ be the discrete Gaussian distribution with mean zero and variance $\sigma = \sigma(\lambda)$. Choose $A \leftarrow_R \mathbb{Z}_q^{m \times n}$. Output $\mathsf{params} = (m, n, t, q, \chi, A)$. The plaintext and ciphertext spaces are $\mathcal{M} = \mathbb{Z}_t$ and $\mathcal{C} = \mathbb{Z}_q^{n+1}$, respectively.*
- *Secret key generation: Choose $s \leftarrow_R \mathbb{Z}_q^n$. Output $\mathsf{sk} = s$.*
- *Public key generation: Set $s \leftarrow \mathsf{sk}$. Sample $e \leftarrow_R \chi^m$. Output $\mathsf{pk} = [As + e]_q$.*
- *Encryption: Set $b \leftarrow \mathsf{pk}$. Choose $r \leftarrow_R \mathbb{Z}_2^m$. Output*

$$\mathsf{ct} = \left(\left[r^\top A \right]_q, \left[\left\lfloor \frac{q}{t} \right\rfloor \mathsf{m} + r^\top b \right]_q \right).$$

- *Decryption: Parse $\mathsf{ct} = (c_1, c_2)$. Set $s \leftarrow \mathsf{sk}$. Output*

$$\mathsf{m} = \left[\left\lfloor \frac{t}{q} [c_2 - c_1 s]_q \right\rceil \right]_t.$$

- *Homomorphic evaluation: Parse $\mathsf{ct}_1 = (c_{11}, c_{12})$ and $\mathsf{ct}_2 = (c_{21}, c_{22})$. Output $\mathsf{ct} = ([c_{11} + c_{21}]_q, [c_{12} + c_{22}]_q)$.*

The important difference between the Regev and ElGamal encryption is the use of noise sampled from the discrete Gaussian distribution in the public-key generation. As a result, the noise is injected into lower bits of the scaled plaintext $\lfloor q/t \rfloor \mathsf{m}$ in the encryption algorithm and removed by rounding after re-scaling in the decryption algorithm. Note that the rounding does not consider an ultimately large noise sampled with non-zero probability and fails when injecting the noise. However, such a noise is not sampled from the discrete Gaussian distribution except with a negligible probability. The definition below formulates the property that a probability distribution is practically bounded.

Definition 2.12 (Bounded distribution)**.** *A probability distribution $D = D(\lambda)$ over $\mathbb{Z}$ is B-bounded if there exists a negligible function* negl *such that*

$$\Pr[|x| \geq B \mid x \leftarrow_R D] < \mathsf{negl}(\lambda)$$

for all $\lambda \in \mathbb{N}$.

Under the assumption that the discrete Gaussian distribution used for public-key generation is bounded, the Regev encryption satisfies the correctness as well as the ElGamal encryption and is homomorphic for addition.

Proposition 2.3. *Let m, t, q, and χ be as in Definition 2.11. Assume that χ is $(q/(2mt) - t/m)$-bounded. The Regev encryption is correct.*

Proof. Let params, pk, and sk be as in Definition 2.11. Let $\Delta = \lfloor q/t \rfloor$. A ciphertext of $\mathsf{m} \in \mathbb{Z}_t$ is given as

$$\mathsf{ct} = \mathsf{Enc}(\mathsf{pk}, \mathsf{m}) = \Big(\big[r^\top A\big]_q, \big[\Delta \mathsf{m} + r^\top (As + e)\big]_q \Big),$$

where $r \in \mathbb{Z}_2^m$ is a random number, and e is a noise sampled from χ^m. The decryption of ct is computed as

$$\begin{aligned}
\mathsf{m}' = \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) &= \left[\left\lfloor \frac{t}{q} \big[\Delta \mathsf{m} + r^\top (As + e) - r^\top As\big]_q \right\rfloor \right]_t, \\
&= \left[\left\lfloor \frac{t}{q} \big[\Delta \mathsf{m} + r^\top e\big]_q \right\rfloor \right]_t, \\
&= \left[\left\lfloor \frac{t}{q} (\Delta \mathsf{m} + r^\top e + n_q q) \right\rfloor \right]_t, \\
&= \left[\left\lfloor \frac{t}{q} \Delta \mathsf{m} + \frac{t}{q} r^\top e + n_q t \right\rfloor \right]_t, \\
&= \left[\left\lfloor \mathsf{m} + \frac{t}{q} (r^\top e - \epsilon \mathsf{m}) + n_q t \right\rfloor \right]_t, \\
&= \left[\mathsf{m} + n_q t + \left\lfloor \frac{t}{q} (r^\top e - \epsilon \mathsf{m}) \right\rfloor \right]_t, \\
&= \mathsf{m} + \left[\left\lfloor \frac{t}{q} (r^\top e - \epsilon \mathsf{m}) \right\rfloor \right]_t,
\end{aligned}$$

where $n_q \in \mathbb{Z}$, $\epsilon = q/t - \Delta$, and $0 \leq \epsilon < 1$. $\mathsf{m}' = \mathsf{m}$ holds if $|(t/q) \cdot (r^\top e - \epsilon \mathsf{m})| < 1/2$,

and its sufficient condition is given as

$$\begin{aligned}
\left|\frac{t}{q}(r^\top e - \epsilon\mathsf{m})\right| < \frac{1}{2} &\iff |r^\top e - \epsilon\mathsf{m}| < \frac{q}{2t}, \\
&\impliedby |r^\top e| + \epsilon\mathsf{m} < \frac{q}{2t}, \\
&\impliedby \left|\sum_{i=1}^{m} r_i e_i\right| < \frac{q}{2t} - t, \\
&\impliedby \sum_{i=1}^{m} |r_i e_i| < \frac{q}{2t} - t, \\
&\impliedby \sum_{i=1}^{m} |e_i| < \frac{q}{2t} - t, \\
&\impliedby |e_i| < \frac{1}{m}\Big(\frac{q}{2t} - t\Big),\ i = 1, \ldots, m.
\end{aligned}$$

Hence, $|(t/q) \cdot (r^\top e - \epsilon\mathsf{m})| < 1/2$ holds with probability at least $1 - \mathsf{negl}(\lambda)$ because χ is $(q/(2mt) - t/m)$-bounded. This implies that $\Pr[\mathsf{m}' = \mathsf{m}] \geq 1 - \mathsf{negl}(\lambda)$. □

Proposition 2.4. *Let m, t, q, and χ be as in Definition 2.11. Assume that χ is $(q/(4mt) - t/m)$-bounded. The Regev encryption is homomorphic for addition modulo t.*

Proof. Let params, pk, and sk be as in Definition 2.11. Let $\Delta = \lfloor q/t \rfloor$, and let $f : \mathcal{M} \times \mathcal{M} \to \mathcal{M} : (\mathsf{m}_1, \mathsf{m}_2) \mapsto [\mathsf{m}_1 + \mathsf{m}_2]_t$. Ciphertext of $\mathsf{m}_1, \mathsf{m}_2 \in \mathbb{Z}_t$ are given as

$$\begin{aligned}
\mathsf{ct}_1 = \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_1) &= \Big(\big[r_1^\top A\big]_q, \big[\Delta\mathsf{m}_1 + r_1^\top(As + e)\big]_q\Big), \\
\mathsf{ct}_2 = \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_2) &= \Big(\big[r_2^\top A\big]_q, \big[\Delta\mathsf{m}_2 + r_2^\top(As + e)\big]_q\Big),
\end{aligned}$$

where $r_1, r_2 \in \mathbb{Z}_2^m$ are random numbers, and e is a noise sampled from χ^m. The output of homomorphic evaluation is obtained as

$$\begin{aligned}
\mathsf{ct} &= \mathsf{Eval}(f, \mathsf{ct}_1, \mathsf{ct}_2), \\
&= \Big(\big[r_1^\top A + r_2^\top A\big]_q, \big[\Delta\mathsf{m}_1 + r_1^\top(As + e) + \Delta\mathsf{m}_2 + r_2^\top(As + e)\big]_q\Big), \\
&= \Big(\big[(r_1 + r_2)^\top A\big]_q, \big[\Delta(\mathsf{m}_1 + \mathsf{m}_2) + (r_1 + r_2)^\top(As + e)\big]_q\Big), \\
&= \Big(\big[r^\top A\big]_q, \big[\Delta(\mathsf{m}_1 + \mathsf{m}_2) + r^\top(As + e)\big]_q\Big),
\end{aligned}$$

where $r = r_1 + r_2$. It follows from the proof of Proposition 2.3 that

$$\mathsf{m}' = \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = [\mathsf{m}_1 + \mathsf{m}_2]_t = f(\mathsf{m}_1, \mathsf{m}_2)$$

holds if $|(t/q) \cdot (r^\top e - \epsilon(\mathsf{m}_1 + \mathsf{m}_2))| < 1/2$, and its sufficient condition is given as

$$\begin{aligned}
\left|\frac{t}{q}(r^\top e - \epsilon(\mathsf{m}_1 + \mathsf{m}_2))\right| < \frac{1}{2} &\iff |r^\top e - \epsilon(\mathsf{m}_1 + \mathsf{m}_2)| < \frac{q}{2t}, \\
&\impliedby |r^\top e| + \epsilon(\mathsf{m}_1 + \mathsf{m}_2) < \frac{q}{2t}, \\
&\impliedby |r^\top e| < \frac{q}{2t} - 2t, \\
&\impliedby \left|\sum_{i=1}^{m} r_{1,i} e_i + \sum_{i=1}^{m} r_{2,i} e_i\right| < \frac{q}{2t} - 2t, \\
&\impliedby \sum_{i=1}^{m} |r_{1,i} e_i| + \sum_{i=1}^{m} |r_{2,i} e_i| < \frac{q}{2t} - 2t, \\
&\impliedby 2\sum_{i=1}^{m} |e_i| < \frac{q}{2t} - 2t, \\
&\impliedby |e_i| < \frac{1}{m}\Big(\frac{q}{4t} - t\Big),\ i = 1, \ldots, m,
\end{aligned}$$

where $\epsilon = q/t - \Delta$, and $0 \leq \epsilon < 1$. Hence, $|(t/q) \cdot (r^\top e - \epsilon(\mathsf{m}_1 + \mathsf{m}_2))| < 1/2$ holds with probability at least $1 - \mathsf{negl}(\lambda)$ because χ is $(q/(4mt) - t/m)$-bounded. This implies that $\Pr[\mathsf{m}' = f(\mathsf{m}_1, \mathsf{m}_2)] \geq 1 - \mathsf{negl}(\lambda)$. □

2.2.4 Security

This section analyzes the security of ElGamal and Regev encryption through game-based proofs. The security is usually shown by the reduction of the target security game to an idealized game via a game sequence. The idealized game represents the perfect security, i.e., the probability of an adversary winning the game is equivalent to 1/2. The reduction approach demonstrates that the difference in adversary's advantages between the target and idealized games is negligibly small by assuming that some computational problem is hard to solve. Then, if there exists an adversary who has a non-negligible probability of winning the target game, the existence contradicts the fact that the advantage in the idealized game is zero since the adversary wins the idealized one with a 1/2 probability. It implies that such an adversary does not exist under the computational assumption.

The following propositions show the IND-CPA security of ElGamal and Regev encryption. The proofs will be accomplished by reducing their IND-CPA games in Definition 2.5 to some idealized games through the DDH and LWE assumptions.

Proposition 2.5. *The ElGamal encryption is IND-CPA secure under the DDH assumption.*

Proof. We prove the statement by reduction of the following games.

$\mathsf{Game}_0(\lambda)$**:** This game is the original IND-CPA game of ElGamal encryption, shown below.

$\mathsf{Game}_0(\lambda)$
$\mathsf{params} \leftarrow \mathsf{Setup}(1^\lambda)$
$s \leftarrow_R \mathbb{Z}_q$
$(\mathsf{m}_0, \mathsf{m}_1) \leftarrow \mathcal{A}(1^\lambda, \mathsf{params}, [g^s]_p)$
$b \leftarrow_R \{0,1\}$
$\mathsf{ct} \leftarrow ([g^r]_p, [\mathsf{m}_b g^{rs}]_p),\ r \leftarrow_R \mathbb{Z}_q$
$\hat{b} \leftarrow \mathcal{A}(\mathsf{ct})$
return 1 **if** $\hat{b} = b$ **and** 0 **otherwise**

$\mathsf{Game}_1(\lambda)$**:** This game is the same as Game_0 except replacing g^{rs} in the challenge ciphertext with g^v for some random number v uniformly sampled from $\mathbb{Z}_q$.

$\mathsf{Game}_2(\lambda)$**:** This game is the same as Game_1 except replacing $\mathsf{m}_b g^v$ with g^v.

Claim 2.1. $|\Pr[\mathsf{Game}_0(\lambda) = 1] - \Pr[\mathsf{Game}_1(\lambda) = 1]|$ *is negligible under the DDH assumption.*

Proof. Consider the following algorithm.

Algorithm $\mathcal{B}(\lambda, \mathsf{params}, \alpha, \beta, \gamma)$
$(\mathsf{m}_0, \mathsf{m}_1) \leftarrow \mathcal{A}(1^\lambda, \mathsf{params}, [\alpha]_p)$
$b \leftarrow_R \{0,1\}$
$\mathsf{ct} \leftarrow ([\beta]_p, [\mathsf{m}_b \gamma]_p)$
$\hat{b} \leftarrow \mathcal{A}(\mathsf{ct})$
return 1 **if** $\hat{b} = b$ **and** 0 **otherwise**

If $(\alpha, \beta, \gamma) = (g^s, g^r, g^{rs})$ for some random numbers $r, s \leftarrow_R \mathbb{Z}_q$, the algorithm $\mathcal{B}$ simulates Game_0. This implies that

$$\Pr[\mathcal{B}(\lambda, \mathsf{params}, g^s, g^r, g^{rs}) = 1 \mid s, r \leftarrow_R \mathbb{Z}_q] = \Pr[\mathsf{Game}_0(\lambda) = 1].$$

Similarly, it follows that

$$\Pr[\mathcal{B}(\lambda, \mathsf{params}, g^s, g^r, g^v) = 1 \mid s, r, v \leftarrow_R \mathbb{Z}_q] = \Pr[\mathsf{Game}_1(\lambda) = 1].$$

Assume that

$$|\Pr[\mathsf{Game}_0(\lambda) = 1] - \Pr[\mathsf{Game}_1(\lambda) = 1]| \geq \mathsf{negl}(\lambda)$$

holds for all negligible function negl, then $\mathcal{B}$ can distinguish (g^s, g^r, g^{rs}) and (g^s, g^r, g^v) with non-negligible probability that contradicts the DDH assumption. The claim is held by contradiction. □

Claim 2.2. $\Pr[\mathsf{Game}_1(\lambda) = 1] = \Pr[\mathsf{Game}_2(\lambda) = 1]$.

Proof. If v is uniformly sampled from $\mathbb{Z}_q$, $[g^v]_p$ follows the uniform distribution on $\mathbb{G}$, and so is $[\mathsf{m}_b g^v]_p$. Hence, the modification of Game_2 does not change any probability of Game_1. □

Claim 2.3. $\Pr[\mathsf{Game}_2(\lambda) = 1] = 1/2$.

Proof. The probability $\Pr[\hat{b} = b]$ is equivalent to $1/2$ because the challenge ciphertext in Game_2 is independent of b. □

Consequently, it follows that $|\Pr[\mathsf{Game}_0(\lambda) = 1] - 1/2|$ is negligible under the DDH assumption. □

Next, the IND-CPA security of Regev encryption is shown. The logic of proof is almost similar to that of ElGamal encryption.

Proposition 2.6. *The Regev encryption is IND-CPA secure under the LWE assumption.*

Proof. We prove the statement by reduction of the following games.

$\mathsf{Game}_0(\lambda)$**:** This game is the original IND-CPA game of Regev encryption, shown below.

$\mathsf{Game}_0(\lambda)$

$\mathsf{params} \leftarrow \mathsf{Setup}(1^\lambda)$

$s \leftarrow_R \mathbb{Z}_q^n,\ e \leftarrow_R \chi^m$

$(\mathsf{m}_0, \mathsf{m}_1) \leftarrow \mathcal{A}(1^\lambda, \mathsf{params}, [As+e]_q)$

$b \leftarrow_R \{0,1\}$

$\mathsf{ct} \leftarrow \left(\left[r^\top A\right]_q, \left[\left\lfloor \frac{q}{t} \right\rfloor \mathsf{m}_b + r^\top (As+e)\right]_q\right),\ r \leftarrow_R \mathbb{Z}_2^m$

$\hat{b} \leftarrow \mathcal{A}(\mathsf{ct})$

return 1 **if** $\hat{b} = b$ **and** 0 **otherwise**

$\mathsf{Game}_1(\lambda)$**:** This game is the same as Game_0 except replacing $As+e$ in the challenge ciphertext with a random number u sampled uniformly from $\mathbb{Z}_q^m$.

$\mathsf{Game}_2(\lambda)$**:** This game is the same as Game_1 except replacing $\Delta\mathsf{m}_b + r^\top u$ with $r^\top u$.

Claim 2.4. $|\Pr[\mathsf{Game}_0(\lambda) = 1] - \Pr[\mathsf{Game}_1(\lambda) = 1]|$ *is negligible under the LWE assumption.*

Proof. Consider the following algorithm.

Algorithm $\mathcal{B}(\lambda, \mathsf{params}, \alpha, \beta, \gamma)$

$(\mathsf{m}_0, \mathsf{m}_1) \leftarrow \mathcal{A}(1^\lambda, \mathsf{params}, \alpha, [\gamma]_q)$

$b \leftarrow_R \{0,1\}$

$\mathsf{ct} \leftarrow \left(\left[\beta^\top \alpha\right]_q, \left[\Delta\mathsf{m}_b + \beta^\top \gamma\right]_q\right)$

$\hat{b} \leftarrow \mathcal{A}(\mathsf{ct})$

return 1 **if** $\hat{b} = b$ **and** 0 **otherwise**

If $(\alpha, \beta, \gamma) = (A, r, As+e)$ for some random matrix $A \leftarrow_R \mathbb{Z}_q^{m\times n}$ and vectors $s \leftarrow_R \mathbb{Z}_q^n, e \leftarrow_R \chi^m, r \leftarrow_R \mathbb{Z}_2^m$, the algorithm $\mathcal{B}$ simulates Game_0. This implies that

$$\Pr[\mathcal{B}(\lambda, \mathsf{params}, A, r, As+e) = 1 \mid A \leftarrow_R \mathbb{Z}_q^{m\times n},\ r \leftarrow_R \mathbb{Z}_2^m,\ s \leftarrow_R \mathbb{Z}_q^n,\ e \leftarrow_R \chi^m] = \Pr[\mathsf{Game}_0(\lambda) = 1].$$

Similarly, it follows that

$$\Pr[\mathcal{B}(\lambda, \mathsf{params}, A, r, u) = 1 \mid A \leftarrow_R \mathbb{Z}_q^{m\times n},\ r \leftarrow_R \mathbb{Z}_2^m,\ s \leftarrow_R \mathbb{Z}_q^n,\ u \leftarrow_R \mathbb{Z}_q^m]$$

$= \Pr[\mathsf{Game}_1(\lambda) = 1]$.

Assume that

$$|\Pr[\mathsf{Game}_0(\lambda) = 1] - \Pr[\mathsf{Game}_1(\lambda) = 1]| \geq \mathsf{negl}(\lambda)$$

holds for all negligible function negl, then $\mathcal{B}$ can distinguish $(A, r, As+e)$ and (A, r, u) with non-negligible probability that contradicts the LWE assumption. The claim is held by contradiction. □

Claim 2.5. $\Pr[\mathsf{Game}_1(\lambda) = 1] = \Pr[\mathsf{Game}_2(\lambda) = 1]$.

Proof. If r is uniformly sampled from $\mathbb{Z}_2^m$, $\left[r^\top u\right]_q$ follows the uniform distribution on $\mathbb{Z}_q$, and so is $\left[\Delta \mathsf{m}_b + r^\top u\right]_q$. Hence, the modification of Game_2 does not change any probability of Game_1. □

Claim 2.6. $\Pr[\mathsf{Game}_2(\lambda) = 1] = 1/2$.

Proof. The probability $\Pr[\hat{b} = b]$ is equivalent to $1/2$ because the challenge ciphertext in Game_2 is independent of b. □

Consequently, it follows that $|\Pr[\mathsf{Game}_0(\lambda) = 1] - 1/2|$ is negligible under the LWE assumption. □

This section concludes with some remarks on the security of homomorphic encryption. The section has viewed the IND-CPA security in Definition 2.5 models the security against an adversary eavesdropping on a communication channel. The adversary can access public parameters and a public key, which implies that the adversary can obtain a ciphertext of any plaintext. In other words, the adversary is capable of accessing an encryption oracle.

Now, consider a more capable adversary who can access a decryption oracle that receives a query of ciphertext excluding a challenge ciphertext and returns a decryption result of the received ciphertext. Such security is referred to as the indistinguishability under chosen-ciphertext attacks (IND-CCA). The IND-CCA security models a scenario in which an adversary can obtain the decryption of a modified challenge ciphertext. The adversary might learn partial information about the original plaintext from the decryption result. Thus, the IND-CCA security captures an adversary actively collapsing the secrecy of encryption compared to a passive adversary in the IND-CPA security.

To achieve the IND-CCA security, an encryption scheme should become tolerant of tampering with a ciphertext. Unfortunately, homomorphic encryption cannot satisfy the IND-CCA security because it is malleable. Malleability is the property of an encryption scheme that allows an adversary to manipulate a ciphertext without knowledge of a secret key, resulting in the change of decrypted message. For example, the ciphertext of ElGamal encryption, $\mathsf{ct} = ([g^r]_p, [\mathsf{m}h^r]_p)$, can be modified to $\mathsf{ct}' = \mathsf{ct} \boxtimes (1, k) = ([g^r]_p, [k\mathsf{m}h^r]_p)$ for some $k \in \mathcal{M}$. The decryption result of ct' is expected to be $[k\mathsf{m}]_p$, and hence the ElGamal encryption is malleable. Similarly, the ciphertext of Regev encryption $\mathsf{ct} = (\left[r^\top A\right]_q, \left\lfloor \lfloor q/t \rfloor \mathsf{m} + r^\top b \right\rceil)$ can be manipulated as $\mathsf{ct}' = \mathsf{ct} \boxplus (1, \lfloor q/t \rfloor k)$ for some $k \in \mathcal{M}$, which might be decrypted to $[\mathsf{m} + k]_t$.

Malleability and homomorphism are two sides of the same coin. Hence, an additional scheme to prevent or detect illegal manipulations for ciphertexts is required while maintaining homomorphic evaluation ability. One approach for constructing IND-CCA secure homomorphic encryption schemes is the requirement of an additional key to evaluate homomorphic computations. Such modified encryption is called keyed-homomorphic encryption [211]. The construction in [211] can satisfy the IND-CCA security against an adversary who is not capable of accessing the evaluation key.

2.3 Updatable homomorphic encryption

The previous sections introduced homomorphic encryption for secure outsourcing computation in a client-server model. Encrypted control, which will be formally defined in the next chapter, is realized based on the framework of secure outsourcing computation as the procedure below. The sensor data of a plant are encrypted by homomorphic encryption and transmitted to a server. The server then computes and returns control inputs from the encrypted data and encrypted controller parameters without decryption. It should be noted that, in the encrypted control scenario, a function and a part of input ciphertexts for the function, i.e., a control law and controller parameters, are usually stored on the server before the control.

Some practical settings of secure outsourcing computation, including encrypted control, require repeated communication between a client and server for the long term. Consider what will happen here if the secret key used for the communication is compromised by an adversary at a certain time. It is evident that the adversary can learn messages communicated between the client and server before and after

Fig. 2.4: Updatable encryption.

the time. In such a case, there is no security guaranteed by the conventional homomorphic encryption schemes. A naive countermeasure to the problem is a refresh of key pairs at every time step. However, repeated key generation is undesirable in the encrypted control scenario. This is because the client should download the ciphertexts of controller parameters, decrypt them, and re-encrypt them using a new key pair, thereby increasing computation costs and network loads.

Such a problem has been attempted to be solved by updatable encryption [196] in the context of private-key encryption for cloud storage. Note that private-key encryption is a symmetric methodology for private communication and securing data using the same key in encryption and decryption. Fig. 2.4 depicts an abstract view of updatable encryption. In updatable encryption, an update token is generated by a key owner, and a cloud server updates a ciphertext encrypted by the previous key to another ciphertext corresponding to a new key by using the update token instead of re-encryption. Boneh et al. realized an updatable encryption scheme based on a key-homomorphic pseudorandom function and formulated a security notion for the encryption [196]. In contrast, Everspaugh et al. improved the efficiency of ciphertext-dependent token generation and showed the security notion in [196] is not sufficient [212]. Additionally, they proposed a stronger security notion implicitly achieving the CCA security and ciphertext integrity (INT-CTXT) and provided an updatable encryption scheme with ciphertext-independent token generation.

When the updatable encryption was developed by Boneh et al., only the forward security (forward secrecy) was considered [47]. The forward security is a security notion that any information of data encrypted using the past keys is protected even though an adversary compromises the current key. Meanwhile, some studies have considered the post-compromise security (backward secrecy) that ensures any information of data encrypted by future keys is protected even when compromising the current key [213, 214]. Fig. 2.5 illustrates the forward security and post-compromise

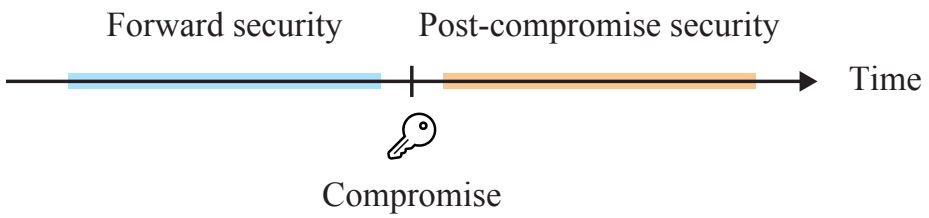


Fig. 2.5: Forward and post-compromise security.

security notions. The security notions are not conflicted and complement each other to protect past and future data from the current compromise. Lehmann and Tackmann formulated a strong security model called the indistinguishability of encryption (IND-ENC) and update (IND-UPD) for the forward and post-compromise security and revealed that the conventional schemes could not satisfy the security [214]. They then constructed an updatable encryption scheme, which is secure in their security model, under the DDH assumption. Klooß et al. modified the security notions by Everspaugh et al. and provided a general construction of a scheme achieving the modified security [215]. Boyd et al. formulated the other indistinguishability notion, IND-UE, and comprehensively analyzed the relationship between their security notion and conventional ones [216].

A homomorphic encryption scheme used in our encrypted-control scenario should satisfy both the forward and post-compromise security because an adversary may store the sequences of inputs and outputs for an encrypted controller and attempt to decipher all the data after collapsing one of them. To this end, This section introduces updatable homomorphic encryption, a public-key variant of updatable encryption satisfying homomorphism. The encryption is built based on the DDH and LWE assumptions by modifying the ElGamal and Regev encryption. This section also formulates an indistinguishability notion for updatable homomorphic encryption and provides security proofs to show that our constructions satisfy the forward and post-compromise security against a network eavesdropper.

2.3.1 Definitions

This section begins by defining the syntax of updatable homomorphic encryption, which is homomorphic encryption with key and ciphertext update algorithms.

Definition 2.13 (Updatable homomorphic encryption)**.** *An updatable homomorphic encryption scheme is a tuple of* KeyGen*,* Enc*,* Dec*, and* Eval *in Definition 2.8, and*

polynomial-time algorithms KeyUpd *and* CtUpd *such that:*

- *Key update: The key update algorithm* $(\mathsf{pk}', \mathsf{sk}', \mathsf{ut}) \leftarrow \mathsf{KeyUpd}(\mathsf{pk}, \mathsf{sk})$ *takes as input a key pair* $(\mathsf{pk}, \mathsf{sk}) \in \mathcal{K}$ *and outputs an updated key pair* $(\mathsf{pk}', \mathsf{sk}') \in \mathcal{K}$ *and update token* ut*.*

- *Ciphertext update: The ciphertext update algorithm* $\mathsf{ct}' \leftarrow \mathsf{CtUpd}(\mathsf{ct}, \mathsf{ut})$ *takes as input a ciphertext* $\mathsf{ct} \in \mathcal{C}$ *and the update token* ut *and outputs an updated ciphertext* $\mathsf{ct}' \in \mathcal{C}$*.*

The correctness condition of an updatable homomorphic encryption scheme is defined by extending the correctness in Definition 2.4 so that the updated ciphertext and the ciphertext encrypted by using the updated public key should be correctly decrypted to the original plaintext by using the updated secret key.

Definition 2.14 (Correctness)**.** *An updatable homomorphic encryption scheme in Definition 2.13 is correct if there exists a negligible function* negl *such that*

$$\Pr\left[\mathsf{m}'_k = \mathsf{m}''_k = \mathsf{m} \;\middle|\; \begin{array}{l} (\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \mathsf{ct}_0^{\mathsf{upd}} \leftarrow \mathsf{Enc}(\mathsf{pk}_0, \mathsf{m}) \\ (\mathsf{pk}_k, \mathsf{sk}_k, \mathsf{ut}_k) \leftarrow \mathsf{KeyUpd}(\mathsf{pk}_{k-1}, \mathsf{sk}_{k-1}) \\ \mathsf{ct}_k^{\mathsf{upd}} \leftarrow \mathsf{CtUpd}(\mathsf{ct}_{k-1}^{\mathsf{upd}}, \mathsf{ut}_k) \\ \mathsf{ct}_k \leftarrow \mathsf{Enc}(\mathsf{pk}_k, \mathsf{m}) \\ \mathsf{m}'_k \leftarrow \mathsf{Dec}(\mathsf{sk}_k, \mathsf{ct}_k) \\ \mathsf{m}''_k \leftarrow \mathsf{Dec}(\mathsf{sk}_k, \mathsf{ct}_k^{\mathsf{upd}}) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda)$$

for all $\lambda \in \mathbb{N}$*, for all* $\mathsf{m} \in \mathcal{M}$*, and for all* $k \in \mathbb{Z}^+$*.*

Similar to the correctness condition, this thesis requires an updatable homomorphic encryption scheme to inherit the homomorphism in Definition 2.9 among the updated ciphertext and the ciphertext encrypted by the updated public key.

Definition 2.15 (Homomorphism)**.** *Let* f *be a binary operation. An updatable homomorphic encryption scheme in Definition 2.13 is homomorphic for* f *if there*

exists a negligible function negl *such that*

$$\Pr\left[\mathsf{m}'_k = \mathsf{m}''_k = \mathsf{m}'''_k = f(\mathsf{m}_1, \mathsf{m}_2) \;\middle|\; \begin{array}{l} (\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \mathsf{ct}^{\mathsf{upd}}_{i,0} \leftarrow \mathsf{Enc}(\mathsf{pk}_0, \mathsf{m}_i),\ i = 1,2 \\ (\mathsf{pk}_k, \mathsf{sk}_k, \mathsf{ut}_k) \leftarrow \mathsf{KeyUpd}(\mathsf{pk}_{k-1}, \mathsf{sk}_{k-1}) \\ \mathsf{ct}^{\mathsf{upd}}_{i,k} \leftarrow \mathsf{CtUpd}(\mathsf{ct}^{\mathsf{upd}}_{i,k-1}, \mathsf{ut}_k),\ i = 1,2 \\ \mathsf{ct}_{i,k} \leftarrow \mathsf{Enc}(\mathsf{pk}_k, \mathsf{m}_i),\ i = 1,2 \\ \mathsf{m}'_k \leftarrow \mathsf{Dec}(\mathsf{sk}_k, \mathsf{Eval}(f, \mathsf{ct}_{1,k}, \mathsf{ct}_{2,k})) \\ \mathsf{m}''_k \leftarrow \mathsf{Dec}(\mathsf{sk}_k, \mathsf{Eval}(f, \mathsf{ct}^{\mathsf{upd}}_{1,k}, \mathsf{ct}_{2,k})) \\ \mathsf{m}'''_k \leftarrow \mathsf{Dec}(\mathsf{sk}_k, \mathsf{Eval}(f, \mathsf{ct}^{\mathsf{upd}}_{1,k}, \mathsf{ct}^{\mathsf{upd}}_{2,k})) \end{array}\right]$$
$$\geq 1 - \mathsf{negl}(\lambda)$$

for all $\lambda \in \mathbb{N}$*, for all* $\mathsf{m}_1, \mathsf{m}_2 \in \mathcal{M}$*, and for all* $k \in \mathbb{Z}^+$*.*

It should be noted that the correctness in Definition 2.14 and the homomorphism in Definition 2.15 are a generalization of those in Definition 2.4 and Definition 2.9, respectively. In what follows, the term correctness and homomorphism are used in the sense of Definition 2.14 and Definition 2.15 when a considered encryption scheme is updatable homomorphic encryption.

2.3.2 Construction from DDH

This section constructs an updatable-homomorphic encryption scheme based on the DDH assumption in Definition 2.6. To this end, key and ciphertext update algorithms are added to the ElGamal encryption. The key update algorithm generates a new secret key and computes the difference d between the new and previous keys. The previous public key is updated by multiplying the d power of a generator of a plaintext space. An update token consists of the previous public key and the difference. The new ciphertext corresponding to the updated keys is computed using the update token and re-randomized by the ciphertext update algorithm. The updatable ElGamal encryption is formally defined as follows.

Definition 2.16 (Updatable ElGamal)**.** *The algorithms in Definition 2.13 for the updatable ElGamal encryption are as follows.*

- *The key generation, encryption, decryption, and homomorphic evaluation algorithms are identical to the ElGamal encryption.*

- *Key update: Set* $h \leftarrow \mathsf{pk}$ *and* $s \leftarrow \mathsf{sk}$. *Compute* $s' \leftarrow \mathsf{SecKeyGen}()$. *Set* $d \leftarrow [s' - s]_q$ *and* $h' \leftarrow \left[hg^d\right]_p$. *Output* $(\mathsf{pk}', \mathsf{sk}', \mathsf{ut}) = (h', s', (h, d))$.
- *Ciphertext update: Parse* $\mathsf{ct} = (c_1, c_2)$ *and* $\mathsf{ut} = (h, d)$. *Choose* $r \leftarrow_R \mathbb{Z}_q$. *Output* $\mathsf{ct}' = ([c_1 g^r]_p, \left[(c_1 g^r)^d c_2 h^r\right]_p)$.

The correctness of updatable ElGamal encryption can be confirmed as follows.

Theorem 2.1. *The updatable ElGamal encryption is correct.*

Proof. Let params be as in Definition 2.10. By construction, the updated secret keys are independent of the previous ones. Suppose $\mathsf{sk}_k = s_k \in \mathbb{Z}_q$, $\mathsf{pk}_0 = [g^{s_0}]_p$, and $(\mathsf{pk}_k, \mathsf{sk}_k, \mathsf{ut}_k) \leftarrow \mathsf{KeyUpd}(\mathsf{pk}_{k-1}, \mathsf{sk}_{k-1})$ for $k \in \mathbb{Z}^+$. The sequences of public keys $\{\mathsf{pk}_k\}_{k\in\mathbb{Z}^+}$ and update token $\{\mathsf{ut}_k\}_{k\in\mathbb{Z}^+}$ are given as

$$
\begin{aligned}
\mathsf{pk}_0 &= [g^{s_0}]_p, \\
\mathsf{pk}_1 &= \left[g^{s_0} g^{[s_1 - s_0]_q}\right]_p = [g^{s_1}]_p, & \mathsf{ut}_1 &= \Big([g^{s_0}]_p, [s_1 - s_0]_q\Big), \\
\mathsf{pk}_2 &= \left[g^{s_1} g^{[s_2 - s_1]_q}\right]_p = [g^{s_2}]_p, & \mathsf{ut}_2 &= \Big([g^{s_1}]_p, [s_2 - s_1]_q\Big), \\
&\vdots & &\vdots \\
\mathsf{pk}_k &= \left[g^{s_{k-1}} g^{[s_k - s_{k-1}]_q}\right]_p = [g^{s_k}]_p, & \mathsf{ut}_k &= \Big([g^{s_{k-1}}]_p, [s_k - s_{k-1}]_q\Big), \\
&\vdots & &\vdots
\end{aligned}
$$

The encryption of $\mathsf{m} \in \mathbb{G}$ using pk_k is

$$\mathsf{ct}_k = \mathsf{Enc}(\mathsf{pk}_k, \mathsf{m}) = \Big([g^{r_k}]_p, [\mathsf{m} g^{r_k s_k}]_p\Big),$$

where $r_k \in \mathbb{Z}_q$ is a random number used in the encryptions at step k. It follows from the proof of Proposition 2.1 that the decryption of ct_k using sk_k is m. This implies that $\Pr[\mathsf{Dec}(\mathsf{sk}_k, \mathsf{ct}_k) = \mathsf{m}] = 1$ holds for all $k \in \mathbb{Z}^+$.

Suppose $\mathsf{ct}_0^{\mathsf{upd}} \leftarrow \mathsf{Enc}(\mathsf{pk}_0, \mathsf{m})$, and $\mathsf{ct}_k^{\mathsf{upd}} \leftarrow \mathsf{CtUpd}(\mathsf{ct}_{k-1}^{\mathsf{upd}}, \mathsf{ut}_k)$. The sequence of updated ciphertexts $\{\mathsf{ct}_k^{\mathsf{upd}}\}_{k\in\mathbb{Z}^+}$ is given as

$$
\begin{aligned}
\mathsf{ct}_0^{\mathsf{upd}} &= \Big([g^{v_0}]_p, [\mathsf{m} g^{v_0 s_0}]_p\Big), \\
\mathsf{ct}_1^{\mathsf{upd}} &= \Big([g^{v_0} g^{v_1}]_p, \left[(g^{v_0} g^{v_1})^{[s_1 - s_0]_q} \mathsf{m} g^{v_0 s_0} g^{s_0 v_1}\right]_p\Big), \\
&= \Big(\left[g^{v_0 + v_1}\right]_p, \left[\mathsf{m} g^{(v_0+v_1)(s_1 - s_0) + v_0 s_0 + s_0 v_1}\right]_p\Big),
\end{aligned}
$$

$$
\begin{aligned}
&= \left(\left[g^{v_0+v_1} \right]_p, \left[\mathsf{m} g^{(v_0+v_1)s_1} \right]_p \right), \\
\mathsf{ct}_2^{\mathsf{upd}} &= \left(\left[g^{v_0+v_1} g^{v_2} \right]_p, \left[(g^{v_0+v_1} g^{v_2})^{[s_2-s_1]_q} \mathsf{m} g^{(v_0+v_1)s_1} g^{s_1 v_2} \right]_p \right), \\
&= \left(\left[g^{v_0+v_1+v_2} \right]_p, \left[\mathsf{m} g^{(v_0+v_1+v_2)(s_2-s_1)+(v_0+v_1)s_1+s_1 v_2} \right]_p \right), \\
&= \left(\left[g^{v_0+v_1+v_2} \right]_p, \left[\mathsf{m} g^{(v_0+v_1+v_2)s_2} \right]_p \right), \\
&\vdots \\
\mathsf{ct}_k^{\mathsf{upd}} &= \left(\left[g^{\sum_{j=0}^{k} v_j} \right]_p, \left[\mathsf{m} g^{\left(\sum_{j=0}^{k} v_j\right) s_k} \right]_p \right) = \left(\left[g^{\bar{v}_k} \right]_p, \left[\mathsf{m} g^{\bar{v}_k s_k} \right]_p \right), \\
&\vdots
\end{aligned}
$$

where $\bar{v}_k = \sum_{j=0}^{k} v_j$, and $v_j \in \mathbb{Z}_q$ are random numbers used in the update of $\mathsf{ct}_{j-1}^{\mathsf{upd}}$ for $j > 0$. Hence, it follows from the proof of Proposition 2.1 that the decryption of $\mathsf{ct}_k^{\mathsf{upd}}$ using sk_k is m. This implies that $\Pr[\mathsf{Dec}(\mathsf{sk}_k, \mathsf{ct}_k^{\mathsf{upd}}) = \mathsf{m}] = 1$ holds for all $k \in \mathbb{Z}^+$. □

From Theorem 2.1, a ciphertext of the updatable ElGamal encryption is correctly decrypted even though a key pair is updated repeatedly. Next, the homomorphism of the encryption is confirmed. It can be easily shown from the equations for the updated secret and public keys and ciphertext at time step k.

Theorem 2.2. *Let p be as in Definition 2.10. The updatable ElGamal encryption is homomorphic for multiplication modulo p.*

Proof. Let params be as in Definition 2.10. Let $f : \mathcal{M} \times \mathcal{M} \to \mathcal{M} : (\mathsf{m}_1, \mathsf{m}_2) \mapsto [\mathsf{m}_1 \mathsf{m}_2]_p$, and let $\mathsf{sk}_k = s_k \in \mathbb{Z}_q$, $\mathsf{pk}_0 = [g^{s_0}]_p$, and $\mathsf{ct}_{i,0}^{\mathsf{upd}} \leftarrow \mathsf{Enc}(\mathsf{pk}_0, \mathsf{m}_i)$ for $k \in \mathbb{Z}^+$, $\mathsf{m}_i \in \mathbb{G}$, and $i = 1, 2$. Suppose $(\mathsf{pk}_k, \mathsf{sk}_k, \mathsf{ut}_k) \leftarrow \mathsf{KeyUpd}(\mathsf{pk}_{k-1}, \mathsf{sk}_{k-1})$ and $\mathsf{ct}_{i,k}^{\mathsf{upd}} \leftarrow \mathsf{CtUpd}(\mathsf{ct}_{i,k-1}^{\mathsf{upd}}, \mathsf{ut}_k)$. It follows from the proof of Theorem 2.1 that, for all k, the secret key, public key, encryptions of m_i, and updated ciphertexts at time step k are respectively given as

$$
\mathsf{sk}_k = s_k,\ \mathsf{pk}_k = [g^{s_k}]_p,\ \mathsf{ct}_{i,k} = \left([g^{r_{i,k}}]_p, [\mathsf{m}_i g^{r_{i,k} s_k}]_p \right),\ \mathsf{ct}_{i,k}^{\mathsf{upd}} = \left([g^{\bar{v}_{i,k}}]_p, [\mathsf{m}_i g^{\bar{v}_{i,k} s_k}]_p \right),
$$

where $\bar{v}_{i,k} = \sum_{j=0}^{k} v_{i,j}$, and $r_{i,k}, v_{i,j} \in \mathbb{Z}_q$ are random numbers. The outputs of homomorphic evaluations with $(\mathsf{ct}_{1,k}, \mathsf{ct}_{2,k})$, $(\mathsf{ct}_{1,k}^{\mathsf{upd}}, \mathsf{ct}_{2,k})$, and $(\mathsf{ct}_{1,k}^{\mathsf{upd}}, \mathsf{ct}_{2,k}^{\mathsf{upd}})$ are respectively computed as

$$
\mathsf{ct}'_k = \mathsf{Eval}(f, \mathsf{ct}_{1,k}, \mathsf{ct}_{2,k}) = \left([g^{r_{1,k}} g^{r_{2,k}}]_p, [\mathsf{m}_1 g^{r_{1,k} s_k} \mathsf{m}_2 g^{r_{2,k} s_k}]_p \right),
$$

$$
\begin{aligned}
&= \left(\left[g^{r_{1,k}+r_{2,k}} \right]_p, \left[\mathsf{m}_1 \mathsf{m}_2 g^{(r_{1,k}+r_{2,k})s_k} \right]_p \right), \\
\mathsf{ct}''_k = \mathsf{Eval}(f, \mathsf{ct}^{\mathsf{upd}}_{1,k}, \mathsf{ct}_{2,k}) &= \left(\left[g^{\bar{v}_{1,k}} g^{r_{2,k}} \right]_p, \left[\mathsf{m}_1 g^{\bar{v}_{1,k} s_k} \mathsf{m}_2 g^{r_{2,k} s_k} \right]_p \right), \\
&= \left(\left[g^{\bar{v}_{1,k}+r_{2,k}} \right]_p, \left[\mathsf{m}_1 \mathsf{m}_2 g^{(\bar{v}_{1,k}+r_{2,k})s_k} \right]_p \right), \\
\mathsf{ct}'''_k = \mathsf{Eval}(f, \mathsf{ct}^{\mathsf{upd}}_{1,k}, \mathsf{ct}^{\mathsf{upd}}_{2,k}) &= \left(\left[g^{\bar{v}_{1,k}} g^{\bar{v}_{2,k}} \right]_p, \left[\mathsf{m}_1 g^{\bar{v}_{1,k} s_k} \mathsf{m}_2 g^{\bar{v}_{2,k} s_k} \right]_p \right), \\
&= \left(\left[g^{\bar{v}_{1,k}+\bar{v}_{2,k}} \right]_p, \left[\mathsf{m}_1 \mathsf{m}_2 g^{(\bar{v}_{1,k}+\bar{v}_{2,k})s_k} \right]_p \right).
\end{aligned}
$$

It follows from the proof of Proposition 2.1 that the decryptions of ct'_k, ct''_k, and ct'''_k using sk_k are $f(\mathsf{m}_1, \mathsf{m}_2)$. Therefore, $\Pr[\mathsf{Dec}(\mathsf{sk}_k, \mathsf{ct}'_k) = \mathsf{Dec}(\mathsf{sk}_k, \mathsf{ct}''_k) = \mathsf{Dec}(\mathsf{sk}_k, \mathsf{ct}'''_k) = f(\mathsf{m}_1, \mathsf{m}_2)] = 1$ holds for all $k \in \mathbb{Z}^+$. □

Theorem 2.2 implies that the updatable ElGamal encryption inherits the multiplicative homomorphism of ElGamal encryption while updating the keys and ciphertext.

2.3.3 Construction from LWE

Similar to the updatable ElGamal encryption, we construct an LWE-based updatable-homomorphic encryption scheme by modifying the Regev encryption. The key update algorithm of updatable Regev encryption generates a new secret key and computes the difference with the previous secret key as with the updatable ElGamal encryption. A new public key is obtained by adding the difference to the previous one. An update token includes the previous public key and the difference. The ciphertext update algorithm updates the previous ciphertext to a new one, which is also re-randomized. The updatable Regev encryption is formally defined as follows.

Definition 2.17 (Updatable Regev)**.** *The algorithms in Definition 2.13 for the updatable Regev encryption are as follows.*

- *The key generation, encryption, decryption, and homomorphic evaluation algorithms are identical to the Regev encryption.*
- *Key update: Set* $b \leftarrow \mathsf{pk}$ *and* $s \leftarrow \mathsf{sk}$*. Compute* $s' \leftarrow \mathsf{SecKeyGen}()$*. Set* $d \leftarrow [s' - s]_q$ *and* $b' \leftarrow [b + Ad]_q$*. Output* $(\mathsf{pk}', \mathsf{sk}', \mathsf{ut}) = (b', s', d)$*.*
- *Ciphertext update: Parse* $\mathsf{ct} = (c_1, c_2)$*. Set* $d \leftarrow \mathsf{ut}$*. Output* $\mathsf{ct} = (c_1, [c_1 d + c_2]_q)$*.*

The following theorem shows the correctness of updatable Regev encryption. The theorem can be proven by the same logic as the updatable ElGamal encryption.

Theorem 2.3. *Let m, t, q, and χ be as in Definition 2.11. The updatable Regev encryption is correct if χ is $(q/(2mt) - t/m)$-bounded.*

Proof. Let params be as in Definition 2.11, and let $\Delta = \lfloor q/t \rfloor$. By construction, updated secret keys are independent of the previous secret keys. Suppose $\mathsf{sk}_k = s_k \in \mathbb{Z}_q^n$, $\mathsf{pk}_0 = [As_0 + e]_q$, and $(\mathsf{pk}_k, \mathsf{sk}_k, \mathsf{ut}_k) \leftarrow \mathsf{KeyUpd}(\mathsf{pk}_{k-1}, \mathsf{sk}_{k-1})$ for $k \in \mathbb{Z}^+$, where e is a noise sampled from χ^m. The sequences of public keys $\{\mathsf{pk}_k\}_{k\in\mathbb{Z}^+}$ and update token $\{\mathsf{ut}_k\}_{k\in\mathbb{Z}^+}$ are given as

$$
\begin{aligned}
\mathsf{pk}_0 &= [As_0 + e]_q, \\
\mathsf{pk}_1 &= [As_0 + e + A(s_1 - s_0)]_q = [As_1 + e]_q, && \mathsf{ut}_1 = [s_1 - s_0]_q, \\
\mathsf{pk}_2 &= [As_1 + e + A(s_2 - s_1)]_q = [As_2 + e]_q, && \mathsf{ut}_2 = [s_2 - s_1]_q, \\
&\vdots && \vdots \\
\mathsf{pk}_k &= [As_{k-1} + e + A(s_k - s_{k-1})]_q = [As_k + e]_q, && \mathsf{ut}_k = [s_k - s_{k-1}]_q, \\
&\vdots && \vdots
\end{aligned}
$$

The encryption of $\mathsf{m} \in \mathbb{Z}_t$ using pk_k is

$$
\mathsf{ct}_k = \mathsf{Enc}(\mathsf{pk}_k, \mathsf{m}) = \Big(\big[r_k^\top A\big]_q, \big[\Delta\mathsf{m} + r_k^\top (As_k + e)\big]_q \Big),
$$

where $r_k \in \mathbb{Z}_2^m$ is a random number used in the encryption at step k. It follows from the proof of Proposition 2.3 that the decryption of ct_k using sk_k becomes m with probability at least $1 - \mathsf{negl}(\lambda)$ since χ is $(q/(2mt) - t/m)$-bounded. This implies that $\Pr[\mathsf{Dec}(\mathsf{sk}_k, \mathsf{ct}_k) = \mathsf{m}] \geq 1 - \mathsf{negl}(\lambda)$ holds for all $k \in \mathbb{Z}^+$.

Suppose $\mathsf{ct}_0^{\mathsf{upd}} \leftarrow \mathsf{Enc}(\mathsf{pk}_0, \mathsf{m})$, and $\mathsf{ct}_k^{\mathsf{upd}} \leftarrow \mathsf{CtUpd}(\mathsf{ct}_{k-1}^{\mathsf{upd}}, \mathsf{ut}_k)$. The sequence of updated ciphertexts $\{\mathsf{ct}_k^{\mathsf{upd}}\}_{k\in\mathbb{Z}^+}$ is given as

$$
\begin{aligned}
\mathsf{ct}_0^{\mathsf{upd}} &= \Big(\big[v_0^\top A\big]_q, \big[\Delta\mathsf{m} + v_0^\top (As_0 + e)\big]_q \Big), \\
\mathsf{ct}_1^{\mathsf{upd}} &= \Big(\big[v_0^\top A\big]_q, \big[v_0^\top A(s_1 - s_0) + \Delta\mathsf{m} + v_0^\top (As_0 + e)\big]_q \Big), \\
&= \Big(\big[v_0^\top A\big]_q, \big[\Delta\mathsf{m} + v_0^\top (As_1 + e)\big]_q \Big), \\
\mathsf{ct}_2^{\mathsf{upd}} &= \Big(\big[v_0^\top A\big]_q, \big[v_0^\top A(s_2 - s_1) + \Delta\mathsf{m} + v_0^\top (As_1 + e)\big]_q \Big),
\end{aligned}
$$

$$
\begin{aligned}
&= \Big(\big[v_0^\top A\big]_q, \big[\Delta \mathsf{m} + v_0^\top (As_2 + e)\big]_q \Big), \\
&\vdots \\
\mathsf{ct}_k^{\mathsf{upd}} &= \Big(\big[v_0^\top A\big]_q, \big[v_0^\top A(s_k - s_{k-1}) + \Delta \mathsf{m} + v_0^\top (As_{k-1} + e)\big]_q \Big), \\
&= \Big(\big[v_0^\top A\big]_q, \big[\Delta \mathsf{m} + v_0^\top (As_k + e)\big]_q \Big), \\
&\vdots
\end{aligned}
$$

where $v_0 \in \mathbb{Z}_2^m$ is a random number used in the encryption at the initial time step. Hence, it follows from the proof of Proposition 2.3 that the decryption of $\mathsf{ct}_k^{\mathsf{upd}}$ using sk_k becomes m with probability at least $1 - \mathsf{negl}(\lambda)$ since χ is $(q/(2mt) - t/m)$-bounded. This implies that $\Pr[\mathsf{Dec}(\mathsf{sk}_k, \mathsf{ct}_k^{\mathsf{upd}}) = \mathsf{m}] \geq 1 - \mathsf{negl}(\lambda)$ holds for all $k \in \mathbb{Z}^+$. □

Theorem 2.3 implies that the decryption algorithm of updatable Regev encryption can recover the original plaintext from a ciphertext under key and ciphertext updates. Next, the homomorphism of updatable Regev encryption is shown as with the updatable ElGamal encryption.

Theorem 2.4. *Let m, t, q, and χ be as in Definition 2.11. The updatable Regev encryption is homomorphic for addition modulo t if χ is $(q/(4mt) - t/m)$-bounded.*

Proof. Let params be as in Definition 2.11, and let $\Delta = \lfloor q/t \rfloor$. Let $f : \mathcal{M} \times \mathcal{M} \to \mathcal{M} : (\mathsf{m}_1, \mathsf{m}_2) \mapsto [\mathsf{m}_1 + \mathsf{m}_2]_t$, and let $\mathsf{sk}_k = s_k \in \mathbb{Z}_q^n$, $\mathsf{pk}_0 = [As_0 + e]_q$, $\mathsf{ct}_{i,0}^{\mathsf{upd}} \leftarrow \mathsf{Enc}(\mathsf{pk}_0, \mathsf{m}_i)$ for $k \in \mathbb{Z}^+$, $\mathsf{m}_i \in \mathbb{Z}_t$, and $i = 1, 2$, where e is a noise sampled from χ^m. Suppose $(\mathsf{pk}_k, \mathsf{sk}_k, \mathsf{ut}_k) \leftarrow \mathsf{KeyUpd}(\mathsf{pk}_{k-1}, \mathsf{sk}_{k-1})$ and $\mathsf{ct}_{i,k}^{\mathsf{upd}} \leftarrow \mathsf{CtUpd}(\mathsf{ct}_{i,k-1}^{\mathsf{upd}}, \mathsf{ut}_k)$. It follows from the proof of Theorem 2.3 that, for all k, the secret key, public key, encryptions of m_i, and updated ciphertexts at time step k are respectively given as

$$
\begin{aligned}
\mathsf{sk}_k = s_k, \quad \mathsf{pk}_k = [As_k + e]_q, \quad \mathsf{ct}_{i,k} &= \Big(\big[r_{i,k}^\top A\big]_q, \big[\Delta \mathsf{m}_i + r_{i,k}^\top (As_k + e)\big]_q \Big), \\
\mathsf{ct}_{i,k}^{\mathsf{upd}} &= \Big(\big[v_i^\top A\big]_q, \big[\Delta \mathsf{m}_i + v_i^\top (As_k + e)\big]_q \Big),
\end{aligned}
$$

where $r_{i,k}, v_i \in \mathbb{Z}_2^m$ are random numbers. The outputs of homomorphic evaluations with $(\mathsf{ct}_{1,k}, \mathsf{ct}_{2,k})$, $(\mathsf{ct}_{1,k}^{\mathsf{upd}}, \mathsf{ct}_{2,k})$, and $(\mathsf{ct}_{1,k}^{\mathsf{upd}}, \mathsf{ct}_{2,k}^{\mathsf{upd}})$ are respectively computed as

$$
\begin{aligned}
\mathsf{ct}'_k &= \mathsf{Eval}(f, \mathsf{ct}_{1,k}, \mathsf{ct}_{2,k}), \\
&= \Big(\big[r_{1,k}^\top A + r_{2,k}^\top A\big]_q, \big[\Delta \mathsf{m}_1 + r_{1,k}^\top (As_k + e) + \Delta \mathsf{m}_2 + r_{2,k}^\top (As_k + e)\big]_q \Big),
\end{aligned}
$$

$$
\begin{aligned}
&= \Big(\big[(r_{1,k} + r_{2,k})^\top A\big]_q, \big[\Delta(\mathsf{m}_1 + \mathsf{m}_2) + (r_{1,k} + r_{2,k})^\top (As_k + e)\big]_q \Big), \\
\mathsf{ct}''_k &= \mathsf{Eval}(f, \mathsf{ct}^{\mathsf{upd}}_{1,k}, \mathsf{ct}_{2,k}), \\
&= \Big(\big[v_1^\top A + r_{2,k}^\top A\big]_q, \big[\Delta\mathsf{m}_1 + v_1^\top (As_k + e) + \Delta\mathsf{m}_2 + r_{2,k}^\top (As_k + e)\big]_q \Big), \\
&= \Big(\big[(v_1 + r_{2,k})^\top A\big]_q, \big[\Delta(\mathsf{m}_1 + \mathsf{m}_2) + (v_1 + r_{2,k})^\top (As_k + e)\big]_q \Big), \\
\mathsf{ct}'''_k &= \mathsf{Eval}(f, \mathsf{ct}^{\mathsf{upd}}_{1,k}, \mathsf{ct}^{\mathsf{upd}}_{2,k}), \\
&= \Big(\big[v_1^\top A + v_2^\top A\big]_q, \big[\Delta\mathsf{m}_1 + v_1^\top (As_k + e) + \Delta\mathsf{m}_2 + v_2^\top (As_k + e)\big]_q \Big), \\
&= \Big(\big[(v_1 + v_2)^\top A\big]_q, \big[\Delta(\mathsf{m}_1 + \mathsf{m}_2) + (v_1 + v_2)^\top (As_k + e)\big]_q \Big).
\end{aligned}
$$

It follows from the proof of Proposition 2.4 that the decryptions of ct'_k, ct''_k, and ct'''_k using sk_k become $f(\mathsf{m}_1, \mathsf{m}_2)$ with probability at least $1 - \mathsf{negl}(\lambda)$ since χ is $(q/(4mt) - t/m)$-bounded. Therefore, $\Pr[\mathsf{Dec}(\mathsf{sk}_k, \mathsf{ct}'_k) = \mathsf{Dec}(\mathsf{sk}_k, \mathsf{ct}''_k) = \mathsf{Dec}(\mathsf{sk}_k, \mathsf{ct}'''_k) = f(\mathsf{m}_1, \mathsf{m}_2)] = 1 - \mathsf{negl}(\lambda)$ holds for all $k \in \mathbb{Z}^+$. □

It holds from Theorem 2.4 that the updatable Regev homomorphic encryption remains additively homomorphic encryption despite updating the secret and public keys of Regev encryption.

2.3.4 Security

As already mentioned, updatable homomorphic encryption used in encrypted control systems is desired to satisfy the forward and post-compromise security. This section formulates an indistinguishability notion of updatable homomorphic encryption called IND-KU-CPA. More precisely, the IND-KU-CPA security is defined such that an adversary cannot learn any partial information of the original plaintext from a challenge ciphertext under the chosen plaintext attack even though the adversary compromises secret keys before and after computing the challenge ciphertext.

Definition 2.18 (IND-KU-CPA)**.** *Consider an updatable homomorphic encryption scheme* $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval}, \mathsf{KeyUpd}, \mathsf{CtUpd})$ *and adversary* $\mathcal{A}$*. Define the game* $\mathsf{Game}^{\mathsf{IND-KU-CPA}}_{\Pi,\mathcal{A}}(\lambda)$ *as follows.*

$\mathsf{Game}^{\mathsf{IND-KU-CPA}}_{\Pi,\mathcal{A}}(\lambda)$

$k \leftarrow 0,\ \tilde{k} \leftarrow \perp,\ \mathcal{L} \leftarrow \emptyset$

$(\mathsf{params}, \mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{KeyGen}(1^\lambda)$

$(\mathsf{m}_0, \mathsf{m}_1) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KeyUpd}}, \mathcal{O}_{\mathsf{Corr}}}(1^\lambda, \mathsf{params})$

$\tilde{k} \leftarrow k$

$b \leftarrow_R \{0,1\}$

$\mathsf{ct}_{\tilde{k}} \leftarrow \mathsf{Enc}(\mathsf{pk}_{\tilde{k}}, \mathsf{m}_b)$

$\hat{b} \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KeyUpd}}, \mathcal{O}_{\mathsf{Corr}}}(\mathsf{ct}_{\tilde{k}})$

return 1 **if** $\hat{b} = b$ **and** $\tilde{k} \notin \mathcal{L}$ **and** 0 **otherwise**

Oracle $\mathcal{O}_{\mathsf{KeyUpd}}()$

$k \leftarrow k + 1$

$(\mathsf{pk}_k, \mathsf{sk}_k, \mathsf{ut}_k) \leftarrow \mathsf{KeyUpd}(\mathsf{pk}_{k-1}, \mathsf{sk}_{k-1})$

Oracle $\mathcal{O}_{\mathsf{Corr}}(\mathsf{str}, k')$

if $k' > k$ **return** $\perp$

if str $=$ secret key

  $\mathcal{L} \leftarrow \mathcal{L} \cup \{k'\}$

  return $\mathsf{sk}_{k'}$

if str $=$ public key

  return $\mathsf{pk}_{k'}$

Setup. *A time step counter k, time step of the challenge ciphertext $\tilde{k}$, and list $\mathcal{L}$ are initialized. Public parameters* params *and a key pair* $(\mathsf{pk}, \mathsf{sk})$ *are generated by running the key generation algorithm* $\mathsf{KeyGen}(1^\lambda)$.

Challenge. *$\mathcal{A}$ takes as input 1^λ and* params *and outputs plaintexts $\mathsf{m}_0, \mathsf{m}_1 \in \mathcal{M}$ of the same length while querying the oracles $\mathcal{O}_{\mathsf{KeyUpd}}$ and $\mathcal{O}_{\mathsf{Corr}}$. Set $\tilde{k} \leftarrow k$, where the time step is increased by querying $\mathcal{O}_{\mathsf{KeyUpd}}$. A bit $b \in \{0,1\}$ is chosen uniformly. A ciphertext $\mathsf{ct}_{\tilde{k}}$ is computed by running the encryption algorithm* $\mathsf{Enc}(\mathsf{pk}_{\tilde{k}}, \mathsf{m}_b)$.

Guess. $\mathcal{A}$ *takes as input* $\mathsf{ct}_{\tilde{k}}$ *and outputs a bit* $\hat{b} \in \{0,1\}$ *while querying the oracles. The game outputs* 1 *if* $\hat{b} = b$ *and* $\tilde{k} \notin \mathcal{L}$*. Otherwise, it outputs* 0*.*

We say Π *is IND-KU-CPA secure if there exists a negligible function* negl *such that*

$$\left| \Pr\left[\mathsf{Game}^{\mathsf{IND-KU-CPA}}_{\Pi,\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2} \right| < \mathsf{negl}(\lambda)$$

for all $\lambda \in \mathbb{N}$ *and for all probabilistic polynomial-time adversary* $\mathcal{A}$*.*

In the IND-KU-CPA game, the adversary can access the oracles $\mathcal{O}_{\mathsf{KeyUpd}}$ and $\mathcal{O}_{\mathsf{Corr}}$. The oracles represent the abilities of an adversary, namely updating and compromising public and secret keys. Using the oracles, the adversary can obtain as many updated public and secret keys as the adversary wants before and after computing the challenge ciphertext. The game excludes the trivial win condition that the adversary obtains the secret key corresponding to the challenge ciphertext. This condition is managed by the list $\mathcal{L}$ and examined at the last step of the game whether the time step for the challenge ciphertext is in the list.

One may think that the IND-KU-CPA security relies on the IND-CPA security. The theorem below answers this affirmatively.

Theorem 2.5. *Let* $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *be a homomorphic encryption scheme in Definition 2.8, and let* $\Pi^{\mathsf{upd}} = (\Pi.\mathsf{KeyGen}, \Pi.\mathsf{Enc}, \Pi.\mathsf{Dec}, \Pi.\mathsf{Eval}, \mathsf{KeyUpd}, \mathsf{CtUpd})$ *be an updatable homomorphic encryption scheme in Definition 2.13. Assume that* Π *is IND-CPA secure. If there exists a negligible function* negl *such that*

$$|\Pr[\mathcal{A}(\mathsf{pk}', \mathsf{sk}') = 1] - \Pr[\mathcal{A}(\mathsf{pk}, \mathsf{sk}) = 1]| < \mathsf{negl}(\lambda)$$

for all $\lambda \in \mathbb{N}$ *and for all probabilistic polynomial-time algorithm* $\mathcal{A}$*, then* Π^{upd} *is IND-KU-CPA secure, where* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \Pi.\mathsf{KeyGen}(1^\lambda)$*, and* $(\mathsf{pk}', \mathsf{sk}', \mathsf{ut}) \leftarrow \Pi^{\mathsf{upd}}.\mathsf{KeyUpd}(\mathsf{pk}, \mathsf{sk})$*.*

Proof. We prove the statement by reduction of the following games.

$\mathsf{Game}_0(\lambda)$**:** This game is the original IND-KU-CPA game in Definition 2.18.

$\mathsf{Game}_1(\lambda)$**:** This game is the same as Game_0 except for the modification of $\mathcal{O}_{\mathsf{KeyUpd}}$ before computing the challenge ciphertext to $\mathcal{O}'_{\mathsf{KeyUpd}}$ so that the key update is replaced with key generation.

Oracle $\mathcal{O}'_{\mathsf{KeyUpd}}()$
$k \leftarrow k + 1$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \Pi.\mathsf{KeyGen}(1^\lambda)$

$\mathsf{Game}_2(\lambda)$**:** This game is the same as Game_1 except that the time step of challenge ciphertext is set to zero, i.e., $\tilde{k} = 0$, in the initialization, and the adversary is prohibited from querying $\mathcal{O}'_{\mathsf{KeyUpd}}$.

$\mathsf{Game}_3(\lambda)$**:** This game is the same as Game_2 except that the adversary is given pk_0 instead of querying $\mathcal{O}_{\mathsf{Corr}}$ before computing the challenge ciphertext.

$\mathsf{Game}_4(\lambda)$**:** This game is the same as Game_3 except replacing $\mathcal{O}_{\mathsf{KeyUpd}}$ with $\mathcal{O}'_{\mathsf{KeyUpd}}$ after computing the challenge ciphertext.

$\mathsf{Game}_5(\lambda)$**:** This game is the same as Game_4 except that the adversary is prohibited from querying $\mathcal{O}'_{\mathsf{KeyUpd}}$ after computing the challenge ciphertext.

$\mathsf{Game}_6(\lambda)$**:** This game is the same as Game_5 except that the adversary is prohibited to query $\mathcal{O}_{\mathsf{Corr}}$ after computing the challenge ciphertext.

Claim 2.7. $|\Pr[\mathsf{Game}_0(\lambda) = 1] - \Pr[\mathsf{Game}_1(\lambda) = 1]|$ *is negligible.*

Proof. The claim follows from the assumption in the statement of the theorem, i.e., no probabilistic polynomial-time adversary can distinguish that the games include either $\mathcal{O}_{\mathsf{KeyUpd}}$ or $\mathcal{O}'_{\mathsf{KeyUpd}}$ except with a negligible probability. □

Claim 2.8. $|\Pr[\mathsf{Game}_1(\lambda) = 1] - \Pr[\mathsf{Game}_2(\lambda) = 1]|$ *is negligible.*

Proof. If $\tilde{k} \notin \mathcal{L}$, i.e., the secret key at $k = \tilde{k}$ is not queried, then the adversary can obtain the key pairs from $k = 0$ to $k = \tilde{k} - 1$ and the public key at $k = \tilde{k}$. The key pairs are independent of the challenge ciphertext. Thus, the modifications in Game_2 do not change any probability of Game_1. □

Claim 2.9. $\Pr[\mathsf{Game}_2(\lambda) = 1] = \Pr[\mathsf{Game}_3(\lambda) = 1]$.

Proof. Now $\mathcal{O}_{\mathsf{Corr}}$ before computing the challenge ciphertext returns only pk_0 or $\perp$ since $\tilde{k} = 0$. □

Claim 2.10. $|\Pr[\mathsf{Game}_3(\lambda) = 1] - \Pr[\mathsf{Game}_4(\lambda) = 1]|$ *is negligible.*

Proof. The claim holds from the proof of Claim 2.7. □

Claim 2.11. $|\Pr[\mathsf{Game}_4(\lambda) = 1] - \Pr[\mathsf{Game}_5(\lambda) = 1]|$ *is negligible.*

Proof. The claim holds from the proof of Claim 2.8. □

Claim 2.12. $|\Pr[\mathsf{Game}_5(\lambda) = 1] - \Pr[\mathsf{Game}_6(\lambda) = 1]|$ *is negligible.*

Proof. The claim holds from the proof of Claim 2.9. □

Claim 2.13. $|\Pr[\mathsf{Game}_6(\lambda) = 1] - 1/2|$ *is negligible if* Π *is IND-CPA secure.*

Proof. Game_6 is identical to the IND-CPA game in Definition 2.5. Hence, the claim holds by definition. □

Consequently, $|\Pr[\mathsf{Game}_0(\lambda) = 1] - 1/2]|$ is negligible. This implies that Π^{upd} is IND-KU-CPA secure. □

Theorem 2.5 implies that an updatable homomorphic encryption scheme satisfies the IND-KU-CPA security if the key update is indistinguishable and if the homomorphic encryption scheme included in the updatable one satisfies the IND-CPA security. Furthermore, the theorem helps analyze the forward and post-compromise security of our constructions because of the reduction of proving the IND-KU-CPA security itself to show the indistinguishability of key updates. Indeed, the following corollaries immediately follow from the theorem.

Corollary 2.1. *The updatable ElGamal encryption is IND-KU-CPA secure under the DDH assumption.*

Proof. Recall that the updatable ElGamal encryption consists of the ElGamal encryption and the key and ciphertext update algorithms, and the ElGamal encryption is IND-CPA secure under the DDH assumption. The statement holds from Theorem 2.5 if the key update algorithm satisfies the indistinguishability condition in the theorem. By construction, the updated secret key follows the same probability distribution as a secret key generated by the key generation algorithm. Let g and p be as in Definition 2.10, and let $s_k \in \mathbb{Z}_q$ be the updated secret key at time step k. It follows from the proof of Theorem 2.1 that the updated public key at the time step is $\mathsf{pk}_k = [g^{s_k}]_p$. This public key is identical to a public key generated by the key-generation algorithm, namely $\mathsf{pk}_k = \mathsf{PubKeyGen}(\mathsf{sk}_k)$. This completes the proof. □

Corollary 2.2. *The updatable Regev encryption is IND-KU-CPA secure under the LWE assumption.*

Proof. Similar to the proof of Corollary 2.1, we show the indistinguishability condition in Theorem 2.5 since the updatable Regev encryption is based on the Regev encryption, which is IND-CPA secure. By construction, the updated secret key follows the same probability distribution as a secret key generated by the key generation algorithm. Let A, e, and q be as in Definition 2.11, and let $s_k \in \mathbb{Z}_q^n$ be the updated secret key at time step k. It follows from the proof of Theorem 2.3 that the updated public key at the time step is $\mathsf{pk}_k = [As_k + e]_q$. Thanks to the LWE assumption, pk_k is indistinguishable from a random number $u \in \mathbb{Z}_q^m$, and so is a fresh public key. This completes the proof. □

The corollaries show the forward and post-compromise security of our constructions under leakage of past and future secret keys. It should be noted that the IND-KU-CPA security does not cover the effects of update tokens for the security. Suppose an adversary eavesdropping on the communication between a client and server obtains the current and next update tokens. In that case, the adversary can recover the previous and next secret keys from the current one as follows.

Proposition 2.7. *Consider the updatable ElGamal encryption. There exists probabilistic polynomial-time adversaries $\mathcal{A}$ and $\mathcal{B}$ such that*

$$\Pr\left[\mathcal{A}(\mathsf{sk}_k, \mathsf{ut}_k) = \mathsf{sk}_{k-1} \;\middle|\; \begin{array}{l}(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ (\mathsf{pk}_k, \mathsf{sk}_k, \mathsf{ut}_k) \leftarrow \mathsf{KeyUpd}(\mathsf{pk}_{k-1}, \mathsf{sk}_{k-1})\end{array}\right] = 1,$$

$$\Pr\left[\mathcal{B}(\mathsf{sk}_k, \mathsf{ut}_{k+1}) = \mathsf{sk}_{k+1} \;\middle|\; \begin{array}{l}(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ (\mathsf{pk}_k, \mathsf{sk}_k, \mathsf{ut}_k) \leftarrow \mathsf{KeyUpd}(\mathsf{pk}_{k-1}, \mathsf{sk}_{k-1})\end{array}\right] = 1$$

for all $k \in \mathbb{N}$. The same adversaries also exist for the updatable Regev encryption.

Proof. Let $\mathsf{sk}_{k-1} = s_{k-1}, \mathsf{sk}_k = s_k, \mathsf{sk}_{k+1} = s_{k+1}$, where $s_{k-1}, s_k, s_{k+1} \in \mathbb{Z}_q$. It follows from Proposition 2.1 that $\mathsf{ut}_k = (h_k, d_k) = ([g^{s_{k-1}}]_p, [s_k - s_{k-1}]_q)$. The adversary $\mathcal{A}$ can compute $s_{k-1} = [s_k - d_k]_q = [s_k - (s_k - s_{k-1})]_q$. Similarly, $\mathcal{B}$ can compute $s_{k+1} = [s_k + d_{k+1}]_q = [s_k + (s_{k+1} - s_k)]_q$. The proof for the updatable Regev encryption is almost the same as in the above discussion. □

One possible countermeasure against the attack in Proposition 2.7 is encrypting the update tokens by private-key encryption. Although the key agreement between

a client and server is required before communication, in practice, such a countermeasure is effective against an adversary eavesdropping on the communication. However, Proposition 2.7 implies that our constructions cannot achieve the desired security against an adversarial server because it needs the update tokens to update ciphertexts. The next section will further modify our constructions to satisfy the forward and post-compromised security against the adversarial server.

2.4 Key-updatable homomorphic encryption

The previous section has shown that updatable homomorphic encryption fulfills the forward and post-compromise security only against an eavesdropper on a network. This section considers building a homomorphic encryption scheme satisfying the security against not only the eavesdropper but also a malicious server, which performs homomorphic evaluation. Here, the malicious server is assumed to be honest but curious. An honest but curious adversary is a passive adversary who does not deviate from the default protocol but attempts to learn some information about other participants in the communication by recording and using the received messages. Note that the assumption is reasonable in the setting of secure outsourcing computation. A client is typically contracted to the server regarding a protocol for outsourcing computation. The server should avoid violating the contract, although it wishes to collect the client's private information, resulting in such a passive adversary.

The previous studies on updatable encryption developed several security notions to achieve the forward and post-compromise security against an honest-but-curious cloud. The IND-ENC [214], IND-UPD [214], IND-CTXT [215], and IND-UE [216] reviewed in the previous section capture the security when considering an adversary model having the capability to access the past and future secret keys as well as some update tokens. Although the security notions might be stronger than the IND-KU-CPA in the sense of allowing the adversary access to update tokens, the security proofs for them are significantly complicated due to the management of trivial win conditions, i.e., whether the adversary learned secret keys and update tokens corresponding to the challenge ciphertext and its updates. Furthermore, prohibiting the secret keys and update tokens directly linked to trivial win conditions implicitly ignore the possibility of the attack in Proposition 2.7 for an adversary having the current secret key even if a used encryption scheme satisfies the security notions. These facts motivates us to adopt other cryptographic mechanisms to

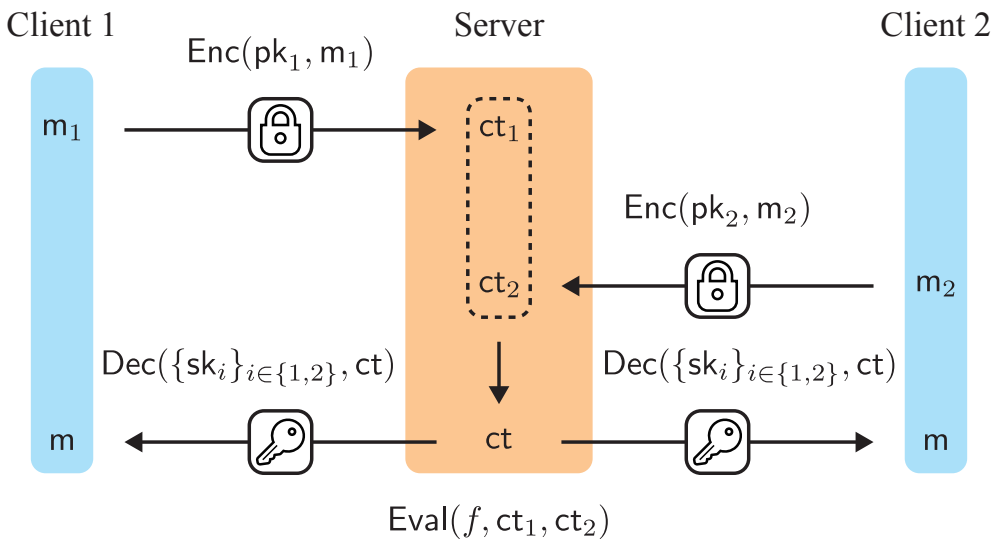


Fig. 2.6: Outsourcing computation with two clients using multi-key homomorphic encryption.

ensure the forward and post-compromise security.

The main difficulty in guaranteeing the forward and post-compromised security against an honest-but-curious server is caused by the necessity to keep update tokens secrecy to the server despite using them for updating ciphertexts in time with key updates. This section solves this problem by modifying our constructions of updatable homomorphic encryption to multi-key homomorphic encryption schemes. Multi-key homomorphic encryption was introduced by López-Alt et al. [54] as a generalization of homomorphic encryption, which enables to perform homomorphic evaluation of ciphertexts encrypted under different keys as illustrated in Fig. 2.6. The two clients in the figure encrypt their messages using each public key and transmit the ciphertexts to the server. The server computes homomorphic evaluation with the ciphertexts encrypted under the different public keys and returns the ciphertext of the computation result to the clients. The result can be recovered by joint decryption of the client.

Now, consider the clients in Fig. 2.6 as identical ones and rethink the key pair that the client 2 has is an update of client 1's. Then, Fig. 2.6 is transformed to Fig. 2.7. The client in Fig. 2.7 can obtain the evaluation result despite updating the key pair without transmitting an update token. Thus, a multi-key homomorphic encryption scheme can be regarded as an updatable homomorphic encryption scheme without update tokens if a key update rule is provided. This section extends the updatable homomorphic encryption in the previous section to a multi-key setting. The modified

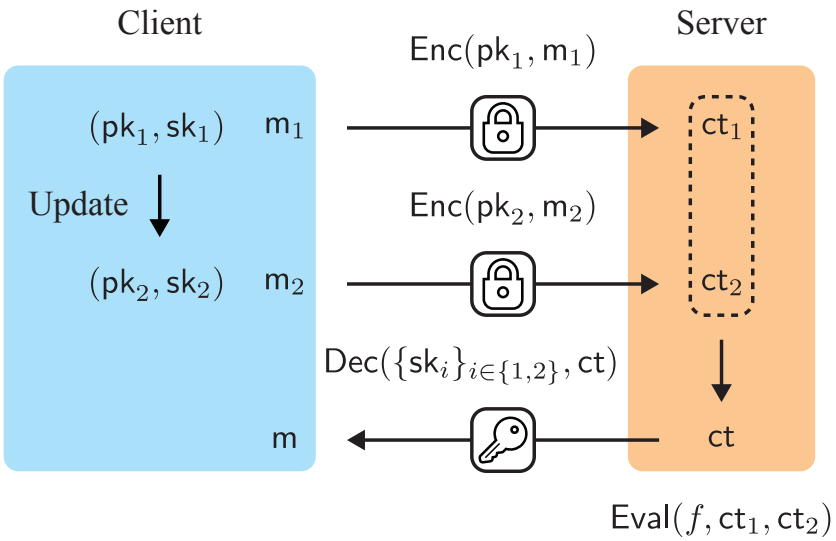


Fig. 2.7: Multi-key homomorphic encryption as homomorphic encryption with updating a key pair.

encryption equips a key update algorithm and a homomorphic evaluation algorithm with ciphertexts encrypted under different keys rather than a ciphertext update algorithm. This section builds the multi-key variants of updatable ElGamal and Regev encryption schemes in Definition 2.16 and Definition 2.17, respectively, and analyzes their forward and post-compromise security.

2.4.1 Definitions

The updatable homomorphic encryption in Definition 2.13 is modified to a variant of multi-key homomorphic encryption. The modified encryption, called key-updatable homomorphic encryption, requires multiple secret keys corresponding to a ciphertext to be decrypted. Moreover, it removes the ciphertext update algorithm and the update token included in the output of the key update algorithm. The syntax of key-updatable homomorphic encryption is formulated as follows.

Definition 2.19 (Key-updatable homomorphic encryption)**.** *A key-updatable homomorphic encryption scheme is a tuple of* KeyGen, Enc, *and* Eval *in Definition 2.8, and polynomial-time algorithms* Dec *and* KeyUpd *such that:*

- *Decryption: The decryption algorithm* $\mathsf{m} \leftarrow \mathsf{Dec}(\{\mathsf{sk}_i\}_{i\in T}, \mathsf{ct})$ *takes as input a ciphertext* $\mathsf{ct} \in \mathcal{C}$ *and a family of secret keys* sk_i *with an index set* $T \subset \mathbb{Z}^+$ *corresponding to* ct *and outputs either a plaintext* $\mathsf{m} \in \mathcal{M}$ *or error symbol* $\perp$.

- *Key update: The key update algorithm* $(\mathsf{pk}', \mathsf{sk}') \leftarrow \mathsf{KeyUpd}(\mathsf{pk}, \mathsf{sk})$ *takes as input a key pair* $(\mathsf{pk}, \mathsf{sk}) \in \mathcal{K}$ *and outputs an updated key pair* $(\mathsf{pk}', \mathsf{sk}') \in \mathcal{K}$.

The correctness and homomorphism conditions in Definition 2.14 and Definition 2.15 are also redefined along with the syntax modifications. In the key-updatable homomorphic encryption, it is required that a ciphertext encrypted by the updated public key at a certain time step is decrypted by the updated secret key at the same time step. This property is formally defined as follows.

Definition 2.20 (Correctness)**.** *A key-updatable homomorphic encryption scheme in Definition 2.19 is correct if there exists a negligible function* negl *such that*

$$\Pr\left[\mathsf{m}'_k = \mathsf{m} \;\middle|\; \begin{array}{l} (\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ (\mathsf{pk}_k, \mathsf{sk}_k) \leftarrow \mathsf{KeyUpd}(\mathsf{pk}_{k-1}, \mathsf{sk}_{k-1}) \\ \mathsf{ct}_k \leftarrow \mathsf{Enc}(\mathsf{pk}_k, \mathsf{m}) \\ \mathsf{m}'_k \leftarrow \mathsf{Dec}(\{\mathsf{sk}_i\}_{i\in\{k\}}, \mathsf{ct}_k) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda)$$

for all $\lambda \in \mathbb{N}$*, for all* $\mathsf{m} \in \mathcal{M}$*, and for all* $k \in \mathbb{Z}^+$.

Note that the index set T, in this case, contains only a particular time step corresponding to a ciphertext to be decrypted. Hence, the family of secret keys given to the decryption algorithm can be equated with a single secret key corresponding to the ciphertext.

The homomorphism condition of key-updatable homomorphic encryption is that the output of homomorphic evaluation with ciphertexts encrypted by the updated public keys not only at the same time step but also in different time steps is correctly decrypted by the secret keys corresponding to both the ciphertexts. The condition is formulated as follows.

Definition 2.21 (Homomorphism)**.** *Let* f *be a binary operation. A key-updatable homomorphic encryption scheme in Definition 2.19 is homomorphic for* f *if there exists a negligible function* negl *such that*

$$\Pr\left[\mathsf{m}'_{j,k} = f(\mathsf{m}_1, \mathsf{m}_2) \;\middle|\; \begin{array}{l} (\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ (\mathsf{pk}_k, \mathsf{sk}_k) \leftarrow \mathsf{KeyUpd}(\mathsf{pk}_{k-1}, \mathsf{sk}_{k-1}) \\ \mathsf{ct}_{1,j} \leftarrow \mathsf{Enc}(\mathsf{pk}_j, \mathsf{m}_1) \\ \mathsf{ct}_{2,k} \leftarrow \mathsf{Enc}(\mathsf{pk}_k, \mathsf{m}_2) \\ \mathsf{m}'_{j,k} \leftarrow \mathsf{Dec}(\{\mathsf{sk}_i\}_{i\in\{j,k\}}, \mathsf{Eval}(f, \mathsf{ct}_{1,j}, \mathsf{ct}_{2,k})) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda)$$

for all $\lambda \in \mathbb{N}$*, for all* $\mathsf{m}_1, \mathsf{m}_2 \in \mathcal{M}$*, for all* $k \in \mathbb{Z}^+$*, and for all* $j \leq k$*.*

Similar to the case of updatable homomorphic encryption, in what follows, an encryption scheme is said to be correct and homomorphic in the sense of Definition 2.20 and Definition 2.21 when a considered encryption scheme is key-updatable homomorphic encryption.

2.4.2 Construction from DDH

Key-updatable homomorphic encryption schemes will be constructed based on the DDH and LWE assumptions in Definition 2.6 and Definition 2.7, respectively. This section modifies the updatable ElGamal encryption in Definition 2.16 to a key-updatable variant. The encryption algorithm in the modified encryption scheme outputs a tuple consisting of three elements. The first two elements are random in a plaintext space, as with the first element of ElGamal ciphertext. The third element is a masked plaintext by the two random numbers used in the first and second elements. The modified homomorphic evaluation algorithm also outputs a tuple of the same structure. The first two elements in the output are the product of the first and second elements of each of the two input ciphertexts, and the third element is the product between the third element of the ciphertexts. Furthermore, the modified decryption algorithm is given either a single or joint secret key when attempting to recover a fresh or evaluated ciphertext. The construction is formally defined as follows.

Definition 2.22 (Key-updatable ElGamal)**.** *The algorithms in Definition 2.19 for the key-updatable ElGamal encryption are as follows.*

- *Key generation: The key generation algorithm is identical to the ElGamal encryption.*
- *Encryption: Set* $h \leftarrow \mathsf{pk}$*. Choose* $r, v \leftarrow_R \mathbb{Z}_q$*. Output* $\mathsf{ct} = ([g^r]_p, [g^v]_p, [\mathsf{m}h^{r+v}]_p)$*.*
- *Decryption: If* $|T| \geq 3$*, output* $\perp$*. If* $|T| = 1$*, set* $s_1 \leftarrow \mathsf{sk}$ *and* $s_2 \leftarrow \mathsf{sk}$*. If* $|T| = 2$*, set* $s_1 \leftarrow \mathsf{sk}_j$ *and* $s_2 \leftarrow \mathsf{sk}_k$*, where* $j, k \in T$*, and* $j < k$*. Parse* $\mathsf{ct} = (c_1, c_2, c_3)$*. Output* $\mathsf{m} = \left[c_1^{-s_1} c_2^{-s_2} c_3\right]_p$*.*
- *Homomorphic evaluation: Parse* $\mathsf{ct}_1 = (c_{11}, c_{12}, c_{13})$ *and* $\mathsf{ct}_2 = (c_{21}, c_{22}, c_{23})$*. Output* $\mathsf{ct} = ([c_{11}c_{12}]_p, [c_{21}c_{22}]_p, [c_{13}c_{23}]_p)$*.*

- *Key update: Set* $s \leftarrow \mathsf{sk}$ *and* $h \leftarrow \mathsf{pk}$. *Choose* $s' \leftarrow_R \mathbb{Z}_q$. *Set* $h' \leftarrow \left[hg^{s'-s}\right]_p$. *Output* $(\mathsf{pk}', \mathsf{sk}') = (h', s')$.

The following theorem shows the correctness of key-updatable ElGamal encryption.

Theorem 2.6. *The key-updatable ElGamal encryption is correct.*

Proof. Let params be as in Definition 2.10. Suppose $\mathsf{sk}_k = s_k \in \mathbb{Z}_q$, $\mathsf{pk}_0 = [g^{s_0}]_p$, and $(\mathsf{pk}_k, \mathsf{sk}_k) \leftarrow \mathsf{KeyUpd}(\mathsf{pk}_{k-1}, \mathsf{sk}_{k-1})$ for $k \in \mathbb{Z}^+$. It follows from the construction in Definition 2.22 and the proof of Theorem 2.1 that, for all k, the secret key, public key, and encryption of $\mathsf{m} \in \mathbb{G}$ are respectively given as

$$\mathsf{sk}_k = s_k, \quad \mathsf{pk}_k = [g^{s_k}]_p, \quad \mathsf{ct}_k = \left([g^{r_k}]_p, [g^{v_k}]_p, \left[\mathsf{m}g^{(r_k+v_k)s_k}\right]_p\right),$$

where $r_k, v_k \in \mathbb{Z}_q$ are random numbers. The decryption of ct_k using $\{\mathsf{sk}_i\}_{i\in\{k\}}$ is

$$\begin{aligned}
\mathsf{m}'_k = \mathsf{Dec}(\{\mathsf{sk}_i\}_{i\in\{k\}}, \mathsf{ct}_k) &= \left[(g^{r_k})^{-s_k}(g^{v_k})^{-s_k}\mathsf{m}g^{(r_k+v_k)s_k}\right]_p, \\
&= \left[\mathsf{m}g^{-r_k s_k - v_k s_k + (r_k+v_k)s_k}\right]_p, \\
&= [\mathsf{m}]_p, \\
&= \mathsf{m},
\end{aligned}$$

where $T = \{k\}$, and $|T| = 1$. This implies that $\Pr[\mathsf{m}'_k = \mathsf{m}] = 1$. □

The homomorphism of key-updatable ElGamal encryption is also shown below.

Theorem 2.7. *Let* p *be as in Definition 2.10. The key-updatable ElGamal encryption is homomorphic for multiplication modulo* p.

Proof. Let params be as in Definition 2.10, and let $f : \mathcal{M} \times \mathcal{M} \to \mathcal{M} : (\mathsf{m}_1, \mathsf{m}_2) \mapsto [\mathsf{m}_1\mathsf{m}_2]_p$. Suppose $\mathsf{sk}_k = s_k \in \mathbb{Z}_q$, $\mathsf{pk}_0 = [g^{s_0}]_p$, and $(\mathsf{pk}_k, \mathsf{sk}_k) \leftarrow \mathsf{KeyUpd}(\mathsf{pk}_{k-1}, \mathsf{sk}_{k-1})$ for $k \in \mathbb{Z}^+$. Similar to the proof of Theorem 2.6, for all k, the secret key, public key, and encryptions of $\mathsf{m}_i \in \mathbb{G}$ are respectively given as

$$\mathsf{sk}_k = s_k, \quad \mathsf{pk}_k = [g^{s_k}]_p, \quad \mathsf{ct}_{i,k} = \left([g^{r_{i,k}}]_p, [g^{v_{i,k}}]_p, \left[\mathsf{m}_i g^{(r_{i,k}+v_{i,k})s_k}\right]_p\right),$$

where $r_{i,k}, v_{i,k} \in \mathbb{Z}_q$ are random numbers used in the encryptions of m_i at time step k, and $i = 1, 2$. For all $j \leq k$, the output of homomorphic evaluations with $\mathsf{ct}_{1,j}$ and

$\mathsf{ct}_{2,k}$ is computed as

$$\begin{aligned}\mathsf{ct}_{j,k} &= \mathsf{Eval}(f, \mathsf{ct}_{1,j}, \mathsf{ct}_{2,k}),\\ &= \Big([g^{r_{1,j}} g^{v_{1,j}}]_p, [g^{r_{2,k}} g^{v_{2,k}}]_p, \big[\mathsf{m}_1 g^{(r_{1,j}+v_{1,j})s_j} \mathsf{m}_2 g^{(r_{2,k}+v_{2,k})s_k}\big]_p\Big),\\ &= \Big(\big[g^{r_{1,j}+v_{1,j}}\big]_p, \big[g^{r_{2,k}+v_{2,k}}\big]_p, \big[\mathsf{m}_1\mathsf{m}_2 g^{(r_{1,j}+v_{1,j})s_j+(r_{2,k}+v_{2,k})s_k}\big]_p\Big).\end{aligned}$$

If $j = k$, the output can be simplified as

$$\begin{aligned}\mathsf{ct}_{k,k} &= \Big(\big[g^{r_{1,k}+v_{1,k}}\big]_p, \big[g^{r_{2,k}+v_{2,k}}\big]_p, \big[\mathsf{m}_1\mathsf{m}_2 g^{(r_{1,k}+v_{1,k}+r_{2,k}+v_{2,k})s_k}\big]_p\Big),\\ &= \Big([g^{\bar{r}_k}]_p, [g^{\bar{v}_k}]_p, \big[\mathsf{m}_1\mathsf{m}_2 g^{(\bar{r}_k+\bar{v}_k)s_k}\big]_p\Big),\end{aligned}$$

where $\bar{r}_k = r_{1,k} + v_{1,k}$, and $\bar{v}_k = r_{2,k} + v_{2,k}$. It follows from the proof of Theorem 2.6 that the decryption of $\mathsf{ct}_{k,k}$ using $\{\mathsf{sk}_i\}_{i\in\{k\}}$ is $f(\mathsf{m}_1, \mathsf{m}_2)$. Furthermore, if $j < k$, $T = \{j, k\}$ and $|T| = 2$. Hence, the decryption of $\mathsf{ct}_{j,k}$ using $\{\mathsf{sk}_i\}_{i\in\{j,k\}}$ is computed as

$$\begin{aligned}\mathsf{Dec}(\{\mathsf{sk}_i\}_{i\in\{j,k\}}, \mathsf{ct}_{j,k}) &= \big[(g^{r_{1,j}+v_{1,j}})^{-s_j}(g^{r_{2,k}+v_{2,k}})^{-s_k}\mathsf{m}_1\mathsf{m}_2 g^{(r_{1,j}+v_{1,j})s_j+(r_{2,k}+v_{2,k})s_k}\big]_p,\\ &= \big[\mathsf{m}_1\mathsf{m}_2 g^{-(r_{1,j}+v_{1,j})s_j-(r_{2,k}+v_{2,k})s_k+(r_{1,j}+v_{1,j})s_j+(r_{2,k}+v_{2,k})s_k}\big]_p,\\ &= [\mathsf{m}_1\mathsf{m}_2]_p.\end{aligned}$$

Therefore, $\Pr[\mathsf{Dec}(\{\mathsf{sk}_i\}_{i\in\{j,k\}}, \mathsf{ct}_{j,k}) = f(\mathsf{m}_1, \mathsf{m}_2)] = 1$ holds for all $k \in \mathbb{Z}^+$ and for all $j \leq k$. □

Theorem 2.6 and Theorem 2.7 imply that the key-updatable ElGamal encryption is multiplicatively homomorphic encryption and correctly decrypts a ciphertext except with a negligible probability while updating public and secret keys.

2.4.3 Construction from LWE

This section constructs key-updatable homomorphic encryption based on the LWE assumption by modifying the updatable Regev encryption in Definition 2.17. The modification idea follows the construction of key-updatable ElGamal encryption. The construction is shown below.

Definition 2.23 (Key-updatable Regev)**.** *The algorithms in Definition 2.19 for the key-updatable Regev encryption are as follows.*

- *Key generation: The key generation algorithm is identical to the Regev encryption.*

- *Encryption: Set* $b \leftarrow \mathsf{pk}$. *Choose* $r, v \leftarrow_R \mathbb{Z}_2^m$. *Output*

$$\mathsf{ct} = \left(\left[r^\top A \right]_q, \left[v^\top A \right]_q, \left[\left\lfloor \frac{q}{t} \right\rfloor \mathsf{m} + (r+v)^\top b \right]_q \right).$$

- *Decryption: If* $|T| \geq 3$, *output* $\perp$. *If* $|T| = 1$, *set* $s_1 \leftarrow \mathsf{sk}$ *and* $s_2 \leftarrow \mathsf{sk}$. *If* $|T| = 2$, *set* $s_1 \leftarrow \mathsf{sk}_j$ *and* $s_2 \leftarrow \mathsf{sk}_k$, *where* $j, k \in T$, *and* $j < k$. *Parse* $\mathsf{ct} = (c_1, c_2, c_3)$. *Output*

$$\mathsf{m} = \left[\left\lfloor \frac{t}{q} [c_3 - c_2 s_2 - c_1 s_1]_q \right\rceil \right]_t.$$

- *Homomorphic evaluation: Parse* $\mathsf{ct}_1 = (c_{11}, c_{12}, c_{13})$ *and* $\mathsf{ct}_2 = (c_{21}, c_{22}, c_{23})$. *Output* $\mathsf{ct} = ([c_{11} + c_{12}]_q, [c_{21} + c_{22}]_q, [c_{13} + c_{23}]_q)$.

- *Key update: Set* $b \leftarrow \mathsf{pk}$. *Choose* $s' \leftarrow_R \mathbb{Z}_q^n$. *Set* $b' \leftarrow [b + A(s' - s)]_q$. *Output* $(\mathsf{pk}', \mathsf{sk}') = (b', s')$.

The theorem on the correctness of key-updatable Regev encryption is as follows.

Theorem 2.8. *Let* m, t, q, *and* χ *be as in Definition 2.11. The key-updatable Regev encryption is correct if* χ *is* $(q/(4mt) - t/(2m))$*-bounded.*

Proof. Let params be as in Definition 2.11, and let $\Delta = \lfloor q/t \rfloor$. Suppose $\mathsf{sk}_k = s_k \in \mathbb{Z}_q^n$, $\mathsf{pk}_0 = [As_0 + e]_q$, and $(\mathsf{pk}_k, \mathsf{sk}_k) \leftarrow \mathsf{KeyUpd}(\mathsf{pk}_{k-1}, \mathsf{sk}_{k-1})$ for $k \in \mathbb{Z}^+$, where e is a noise sampled from χ^m. It follows from the construction in Definition 2.23 and the proof of Theorem 2.3 that, for all k, the secret key, public key, and encryption of $\mathsf{m} \in \mathbb{Z}_t$ are respectively given as

$$\begin{aligned} &\mathsf{sk}_k = s_k, \quad \mathsf{pk}_k = [As_k + e]_q, \\ &\mathsf{ct}_k = \left(\left[r_k^\top A \right]_q, \left[v_k^\top A \right]_q, \left[\Delta \mathsf{m} + (r_k + v_k)^\top (As_k + e) \right]_q \right), \end{aligned}$$

where $r_k, v_k \in \mathbb{Z}_q$ are random numbers. The decryption of ct_k using $\{\mathsf{sk}_i\}_{i \in \{k\}}$ is computed as

$$\mathsf{m}'_k = \mathsf{Dec}(\{\mathsf{sk}_i\}_{i \in \{k\}}, \mathsf{ct}_k),$$

$$
\begin{aligned}
&= \left\lfloor \left\lfloor \frac{t}{q}\left[\Delta\mathsf{m} + (r_k + v_k)^\top (As_k + e) - r_k^\top As_k - v_k^\top As_k\right]_q \right\rfloor \right\rfloor_t, \\
&= \left\lfloor \left\lfloor \frac{t}{q}\left[\Delta\mathsf{m} + (r_k + v_k)^\top e\right]_q \right\rfloor \right\rfloor_t, \\
&= \left\lfloor \left\lfloor \frac{t}{q}(\Delta\mathsf{m} + (r_k + v_k)^\top e + n_q q) \right\rfloor \right\rfloor_t, \\
&= \left\lfloor \left\lfloor \frac{t}{q}\Delta\mathsf{m} + \frac{t}{q}(r_k + v_k)^\top e + n_q t \right\rfloor \right\rfloor_t, \\
&= \left\lfloor \left\lfloor \mathsf{m} + \frac{t}{q}\big((r_k + v_k)^\top e - \epsilon\mathsf{m}\big) + n_q t \right\rfloor \right\rfloor_t, \\
&= \left\lfloor \mathsf{m} + n_q t + \left\lfloor \frac{t}{q}\big((r_k + v_k)^\top e - \epsilon\mathsf{m}\big) \right\rfloor \right\rfloor_t, \\
&= \mathsf{m} + \left\lfloor \left\lfloor \frac{t}{q}\big(\bar{r}^\top e - \epsilon\mathsf{m}\big) \right\rfloor \right\rfloor_t,
\end{aligned}
$$

where $T = \{k\}$, $|T| = 1$, $\bar{r}_k = r_k + v_k$, $n_q \in \mathbb{Z}$, $\epsilon = q/t - \Delta$, and $0 \le \epsilon < 1$. $\mathsf{m}'_k = \mathsf{m}$ holds if $|(t/q) \cdot (\bar{r}_k^\top e - \epsilon\mathsf{m})| < 1/2$, and its sufficient condition is given as

$$
\begin{aligned}
\left|\frac{t}{q}(\bar{r}_k^\top e - \epsilon\mathsf{m})\right| < \frac{1}{2} &\iff |\bar{r}_k^\top e - \epsilon\mathsf{m}| < \frac{q}{2t}, \\
&\impliedby |\bar{r}_k^\top e| + \epsilon\mathsf{m} < \frac{q}{2t}, \\
&\impliedby \left|\sum_{i=1}^{m} r_{k,i}e_i + \sum_{i=1}^{m} v_{k,i}e_i\right| < \frac{q}{2t} - t, \\
&\impliedby \sum_{i=1}^{m} |r_{k,i}e_i| + \sum_{i=1}^{m} |v_{k,i}e_i| < \frac{q}{2t} - t, \\
&\impliedby 2\sum_{i=1}^{m} |e_i| < \frac{q}{2t} - t, \\
&\impliedby |e_i| < \frac{1}{m}\left(\frac{q}{4t} - \frac{t}{2}\right),\ i = 1, \ldots, m.
\end{aligned}
$$

Hence, $|(t/q) \cdot (\bar{r}_k^\top e - \epsilon\mathsf{m})| < 1/2$ holds with probability at least $1 - \mathsf{negl}(\lambda)$ because χ is $(q/(4mt) - t/(2m))$-bounded. This implies that $\Pr[\mathsf{m}'_k = \mathsf{m}] \ge 1 - \mathsf{negl}(\lambda)$. □

The key-updatable Regev encryption also satisfies the homomorphism, as shown below.

Theorem 2.9. *Let m, t, q, and χ be as in Definition 2.11. The key-updatable Regev encryption is homomorphic for addition modulo t if χ is $(q/(8mt) - t/(2m))$-bounded.*

Proof. Let params be as in Definition 2.11, and let $f : \mathcal{M} \times \mathcal{M} \to \mathcal{M} : (\mathsf{m}_1, \mathsf{m}_2) \mapsto [\mathsf{m}_1 + \mathsf{m}_2]_t$. Suppose $\mathsf{sk}_k = s_k \in \mathbb{Z}_q^n$, $\mathsf{pk}_0 = [As_0 + e]_q$, and $(\mathsf{pk}_k, \mathsf{sk}_k) \leftarrow \mathsf{KeyUpd}(\mathsf{pk}_{k-1}, \mathsf{sk}_{k-1})$ for $k \in \mathbb{Z}^+$, where e is a noise sampled from χ^m. Similar to the proof of Theorem 2.8, for all k, the secret key, public key, and encryptions of $\mathsf{m}_i \in \mathbb{Z}_t$ are respectively given as

$$\begin{aligned}
\mathsf{sk}_k &= s_k, \quad \mathsf{pk}_k = [As_k + e]_q, \\
\mathsf{ct}_{i,k} &= \Big(\big[r_{i,k}^\top A\big]_q, \big[v_{i,k}^\top A\big]_q, \big[\Delta \mathsf{m}_i + (r_{i,k} + v_{i,k})^\top (As_k + e)\big]_q \Big),
\end{aligned}$$

where $r_{i,k}, v_{i,k} \in \mathbb{Z}_2^m$ are random numbers used in the encryptions of m_i at time step k, and $i = 1, 2$. For all $j \le k$, the output of homomorphic evaluations with $\mathsf{ct}_{1,j}$ and $\mathsf{ct}_{2,k}$ is computed as

$$\begin{aligned}
\mathsf{ct}_{j,k} &= \mathsf{Eval}(f, \mathsf{ct}_{1,j}, \mathsf{ct}_{2,k}), \\
&= \Big(\big[r_{1,j}^\top A + v_{1,j}^\top A\big]_q, \big[r_{2,k}^\top A + v_{2,k}^\top A\big]_q, \\
&\qquad \big[\Delta \mathsf{m}_1 + (r_{1,j} + v_{1,j})^\top (As_j + e) + \Delta \mathsf{m}_2 + (r_{2,k} + v_{2,k})^\top (As_k + e)\big]_q \Big), \\
&= \Big(\big[(r_{1,j} + v_{1,j})^\top A\big]_q, \big[(r_{2,k} + v_{2,k})^\top A\big]_q, \\
&\qquad \big[\Delta(\mathsf{m}_1 + \mathsf{m}_2) + (r_{1,j} + v_{1,j})^\top (As_j + e) + (r_{2,k} + v_{2,k})^\top (As_k + e)\big]_q \Big), \\
&= \Big(\big[\bar{r}_j^\top A\big]_q, \big[\bar{v}_k^\top A\big]_q, \big[\Delta(\mathsf{m}_1 + \mathsf{m}_2) + \bar{r}_j^\top (As_j + e) + \bar{v}_k^\top (As_k + e)\big]_q \Big),
\end{aligned}$$

where $\bar{r}_j = r_{1,j} + v_{1,j}$, and $\bar{v}_k = r_{2,k} + v_{2,k}$. The decryption of $\mathsf{ct}_{j,k}$ using $\{\mathsf{sk}_i\}_{i \in \{j,k\}}$ is computed as

$$\begin{aligned}
\mathsf{m}'_k &= \mathsf{Dec}(\{\mathsf{sk}_i\}_{i \in \{j,k\}}, \mathsf{ct}_{j,k}), \\
&= \left\lfloor \left\lfloor \frac{t}{q} \big[\Delta(\mathsf{m}_1 + \mathsf{m}_2) + \bar{r}_j^\top (As_j + e) + \bar{v}_k^\top (As_k + e) - \bar{r}_j^\top As_j - \bar{v}_k^\top As_k\big]_q \right\rceil \right\rfloor_t, \\
&= \left\lfloor \left\lfloor \frac{t}{q} \big[\Delta(\mathsf{m}_1 + \mathsf{m}_2) + (\bar{r}_j + \bar{v}_k)^\top e\big]_q \right\rceil \right\rfloor_t, \\
&= \left\lfloor \left\lfloor \frac{t}{q} (\Delta(\mathsf{m}_1 + \mathsf{m}_2) + (\bar{r}_j + \bar{v}_k)^\top e + n_q q) \right\rceil \right\rfloor_t, \\
&= \left\lfloor \left\lfloor \frac{t}{q} \Delta(\mathsf{m}_1 + \mathsf{m}_2) + \frac{t}{q} (\bar{r}_j + \bar{v}_k)^\top e + n_q t \right\rceil \right\rfloor_t, \\
&= \left\lfloor \left\lfloor \mathsf{m}_1 + \mathsf{m}_2 + \frac{t}{q} \big((\bar{r}_j + \bar{v}_k)^\top e - \epsilon(\mathsf{m}_1 + \mathsf{m}_2) \big) + n_q t \right\rceil \right\rfloor_t,
\end{aligned}$$

$$
\begin{aligned}
&= \left[\mathsf{m}_1 + \mathsf{m}_2 + n_q t + \left\lfloor \frac{t}{q}\big((\bar{r}_j + \bar{v}_k)^\top e - \epsilon(\mathsf{m}_1 + \mathsf{m}_2)\big) \right\rfloor\right]_t, \\
&= [\mathsf{m}_1 + \mathsf{m}_2]_t + \left[\left\lfloor \frac{t}{q}\big((\bar{r}_j + \bar{v}_k)^\top e - \epsilon(\mathsf{m}_1 + \mathsf{m}_2)\big) \right\rfloor\right]_t,
\end{aligned}
$$

where $n_q \in \mathbb{Z}$, $\epsilon = q/t - \Delta$, and $0 \leq \epsilon < 1$. $\mathsf{m}'_k = \mathsf{m}$ holds if $|(t/q) \cdot (\bar{r}_j + \bar{v}_k)^\top e - \epsilon(\mathsf{m}_1 + \mathsf{m}_2))| < 1/2$, and its sufficient condition is given as

$$
\begin{aligned}
\left|\frac{t}{q}(\bar{r}_j + \bar{v}_k)^\top e - \epsilon(\mathsf{m}_1 + \mathsf{m}_2))\right| < \frac{1}{2} &\Longleftrightarrow |(\bar{r}_j + \bar{v}_k)^\top e - \epsilon(\mathsf{m}_1 + \mathsf{m}_2)| < \frac{q}{2t}, \\
&\Longleftarrow |(\bar{r}_j + \bar{v}_k)^\top e| + \epsilon(\mathsf{m}_1 + \mathsf{m}_2) < \frac{q}{2t}, \\
&\Longleftarrow \left|\sum_{i=1}^{m}(r_{1,j,i} + v_{1,j,i} + r_{2,k,i} + v_{2,k,i})e_i\right| < \frac{q}{2t} - 2t, \\
&\Longleftarrow \sum_{i=1}^{m}|(r_{1,j,i} + v_{1,j,i} + r_{2,k,i} + v_{2,k,i})e_i| < \frac{q}{2t} - 2t, \\
&\Longleftarrow 4\sum_{i=1}^{m}|e_i| < \frac{q}{2t} - 2t, \\
&\Longleftarrow |e_i| < \frac{1}{m}\left(\frac{q}{8t} - \frac{t}{2}\right),\ i = 1, \ldots, m.
\end{aligned}
$$

Hence, $|(t/q) \cdot (\bar{r}_k^\top e - \epsilon\mathsf{m})| < 1/2$ holds with probability at least $1 - \mathsf{negl}(\lambda)$ because χ is $(q/(8mt) - t/(2m))$-bounded. This implies that $\Pr[\mathsf{m}'_k = \mathsf{m}] \geq 1 - \mathsf{negl}(\lambda)$. □

From Theorem 2.9, the key-updatable Regev encryption is additively homomorphic encryption. It should be noted that the modifications of the key-updatable Regev encryption from the updatable one require the bound of noise in half for correctness and homomorphism. In other words, the ciphertext modulus of key-updatable Regev encryption should be increased from the one of updatable Regev encryption by $4mtB$ to retain the noise size B in the updatable Regev encryption. This requirement leads to an increase in ciphertext size.

2.4.4 Security

This section reveals the forward and post-compromise security of key-updatable ElGamal and Regev encryption, namely the IND-KU-CPA security in Definition 2.18. Following Theorem 2.5, it shows that the constructions satisfy the IND-CPA security in Definition 2.5, and the probability distribution of the output of each key-update

algorithm is indistinguishable from that of a fresh key pair generated by each key-generation algorithm.

The section begins by showing the IND-KU-CPA security of key-updatable ElGamal encryption. The lemma below is on the IND-CPA security of the encryption. The proof is fulfilled by slightly modifying the cryptographic games in the proof of Proposition 2.5.

Lemma 2.1. *The key-updatable ElGamal encryption is IND-CPA secure under the DDH assumption.*

Proof. We prove the statement by reduction of the following games as with the ElGamal encryption.

$\mathsf{Game}_0(\lambda)$**:** This game is the original IND-CPA game of key-updatable ElGamal encryption, shown below.

$\mathsf{Game}_0(\lambda)$

$\mathsf{params} \leftarrow \mathsf{Setup}(1^\lambda)$

$s \leftarrow_R \mathbb{Z}_q$

$(\mathsf{m}_0, \mathsf{m}_1) \leftarrow \mathcal{A}(1^\lambda, \mathsf{params}, [g^s]_p)$

$b \leftarrow_R \{0,1\}$

$\mathsf{ct} \leftarrow \left([g^r]_p, [g^v]_p, \left[\mathsf{m}_b g^{(r+v)s}\right]_p\right),\ r \leftarrow_R \mathbb{Z}_q,\ v \leftarrow_R \mathbb{Z}_q$

$\hat{b} \leftarrow \mathcal{A}(\mathsf{ct})$

return 1 **if** $\hat{b} = b$ **and** 0 **otherwise**

$\mathsf{Game}_1(\lambda)$**:** This game is the same as Game_0 except replacing $g^{(r+v)s}$ in the challenge ciphertext with g^{u_1+vs} for some random number u_1 uniformly sampled from $\mathbb{Z}_q$.

$\mathsf{Game}_2(\lambda)$**:** This game is the same as Game_1 except replacing g^{u_1+vs} in the challenge ciphertext with $g^{u_1+u_2}$ for some random number u_2 uniformaly sampled from $\mathbb{Z}_q$.

$\mathsf{Game}_3(\lambda)$**:** This game is the same as Game_2 except replacing $\mathsf{m}_b g^{u_1+u_2}$ with $g^{u_1+u_2}$.

Claim 2.14. $|\Pr[\mathsf{Game}_0(\lambda) = 1] - \Pr[\mathsf{Game}_1(\lambda) = 1]|$ *is negligible under the DDH assumption.*

Proof. Consider the following algorithm.

Algorithm $\mathcal{B}(\lambda, \mathsf{params}, \alpha, \beta, \gamma, \delta)$

$(\mathsf{m}_0, \mathsf{m}_1) \leftarrow \mathcal{A}(1^\lambda, \mathsf{params}, [\alpha]_p)$

$b \leftarrow_R \{0,1\}$

$\mathsf{ct} \leftarrow ([\beta]_p, [\gamma]_p, [\mathsf{m}_b\delta]_p)$

$\hat{b} \leftarrow \mathcal{A}(\mathsf{ct})$

return 1 **if** $\hat{b} = b$ **and** 0 **otherwise**

If $(\alpha, \beta, \gamma, \delta) = (g^s, g^r, g^v, g^{rs}g^{vs})$ for some random numbers $r, v, s \leftarrow_R \mathbb{Z}_q$, the algorithm $\mathcal{B}$ simulates Game_0. This implies that

$$\Pr[\mathcal{B}(\lambda, \mathsf{params}, g^s, g^r, g^v, g^{rs}g^{vs}) = 1 \mid s, r, v \leftarrow_R \mathbb{Z}_q] = \Pr[\mathsf{Game}_0(\lambda) = 1].$$

Similarly, it follows that

$$\Pr[\mathcal{B}(\lambda, \mathsf{params}, g^s, g^r, g^v, g^{u_1}g^{vs}) = 1 \mid s, r, v, u_1 \leftarrow_R \mathbb{Z}_q] = \Pr[\mathsf{Game}_1(\lambda) = 1].$$

Assume that

$$|\Pr[\mathsf{Game}_0(\lambda) = 1] - \Pr[\mathsf{Game}_1(\lambda) = 1]| \geq \mathsf{negl}(\lambda)$$

holds for all negligible function negl, then $\mathcal{B}$ can distinguish (g^s, g^r, g^{rs}) and (g^s, g^r, g^{u_1}) with non-negligible probability that contradicts the DDH assumption. The claim is held by contradiction. □

Claim 2.15. $|\Pr[\mathsf{Game}_1(\lambda) = 1] - \Pr[\mathsf{Game}_2(\lambda) = 1]|$ *is negligible under the DDH assumption.*

Proof. We use the algorithm $\mathcal{B}$ again to prove the claim. It follows that

$$\Pr[\mathcal{B}(\lambda, \mathsf{params}, g^s, g^r, g^v, g^{u_1}g^{vs}) = 1 \mid s, r, v, u_1 \leftarrow_R \mathbb{Z}_q] = \Pr[\mathsf{Game}_1(\lambda) = 1]$$

and

$$\Pr[\mathcal{B}(\lambda, \mathsf{params}, g^s, g^r, g^v, g^{u_1}g^{u_2}) = 1 \mid s, r, v, u_1, u_2 \leftarrow_R \mathbb{Z}_q] = \Pr[\mathsf{Game}_2(\lambda) = 1].$$

Assume that

$$|\Pr[\mathsf{Game}_1(\lambda) = 1] - \Pr[\mathsf{Game}_2(\lambda) = 1]| \geq \mathsf{negl}(\lambda)$$

holds for all negligible function negl, then $\mathcal{B}$ can distinguish (g^s, g^v, g^{vs}) and (g^s, g^v, g^{u_2}) with non-negligible probability that contradicts the DDH assumption. The claim is held by contradiction. □

Claim 2.16. $\Pr[\mathsf{Game}_2(\lambda) = 1] = \Pr[\mathsf{Game}_3(\lambda) = 1]$.

Proof. If u_1 and u_2 are uniformly sampled from $\mathbb{Z}_q$, $[g^{u_1+u_2}]_p$ follows the uniform distribution on $\mathbb{G}$, and so is $[\mathsf{m}_b g^{u_1+u_2}]_p$. Hence, the modification of Game_2 does not change any probability of Game_1. □

Claim 2.17. $\Pr[\mathsf{Game}_3(\lambda) = 1] = 1/2$.

Proof. The probability $\Pr[\hat{b} = b]$ is equivalent to $1/2$ because the challenge ciphertext in Game_3 is independent of b. □

Consequently, it follows that $|\Pr[\mathsf{Game}_0(\lambda) = 1] - 1/2|$ is negligible under the DDH assumption. □

The following lemma shows the indistinguishability of key updates in the key-updatable ElGamal encryption. Its proof is almost based on that of Corollary 2.1.

Lemma 2.2. *Given the key-updatable ElGamal encryption. There exists a negligible function* negl *such that*

$$|\Pr[\mathcal{A}(\mathsf{pk}', \mathsf{sk}') = 1] - \Pr[\mathcal{A}(\mathsf{pk}, \mathsf{sk}) = 1]| < \mathsf{negl}(\lambda)$$

for all $\lambda \in \mathbb{N}$ *and for all probabilistic polynomial-time algorithm* $\mathcal{A}$*, where* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$*, and* $(\mathsf{pk}', \mathsf{sk}') \leftarrow \mathsf{KeyUpd}(\mathsf{pk}, \mathsf{sk})$.

Proof. Let params be as in Definition 2.10, and let $(\mathsf{sk}, \mathsf{pk}) = (s, [g^s]_p) \in \mathbb{Z}_q \times \mathbb{G}$ be a fresh key pair. By construction, the updated secret key $\mathsf{sk}' = s'$ follows the same probability distribution as the fresh secret key. It follows from the proof of Theorem 2.6 that the updated public key is $\mathsf{pk}' = \left[g^{s'}\right]_p$. This public key is identical to a public key generated by $\mathsf{pk}' \leftarrow \mathsf{PubKeyGen}(\mathsf{sk}')$. This completes the proof. □

The lemmas conclude that the key-updatable ElGamal encryption is IND-KU-CPA secure.

Theorem 2.10. *The key-updatable ElGamal encryption is IND-KU-CPA secure under the DDH assumption.*

Proof. The theorem follows from Theorem 2.5, Lemma 2.1, and Lemma 2.2. □

Next, the proof for the IND-KU-CPA security of key-updatable Regev encryption is shown. Unlike the key-updatable ElGamal encryption, the IND-CPA security of the construction immediately follows from Proposition 2.6 by considering the LWE assumption.

Lemma 2.3. *The key-updatable Regev encryption is IND-CPA secure under the LWE assumption.*

Proof. The theorem follows from Proposition 2.6. □

Furthermore, the indistinguishability of key update in the key-updatable Regev encryption can be shown almost the same way to prove Lemma 2.2.

Lemma 2.4. *Given the key-updatable Regev encryption. There exists a negligible function* negl *such that*

$$|\Pr[\mathcal{A}(\mathsf{pk}', \mathsf{sk}') = 1] - \Pr[\mathcal{A}(\mathsf{pk}, \mathsf{sk}) = 1]| < \mathsf{negl}(\lambda)$$

for all $\lambda \in \mathbb{N}$ *and for all probabilistic polynomial-time algorithm* $\mathcal{A}$ *under the LWE assumption, where* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$*, and* $(\mathsf{pk}', \mathsf{sk}') \leftarrow \mathsf{KeyUpd}(\mathsf{pk}, \mathsf{sk})$*.*

Proof. This proof is almost the same as one for Lemma 2.2. Let params be as in Definition 2.11, and let $(\mathsf{sk}, \mathsf{pk}) = (s, [As + e]_q) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^m$ be a fresh key pair, where e is a noise sampled from χ^m. By construction, the updated secret key $\mathsf{sk}' = s'$ follows the same probability distribution as the fresh secret key. It follows from the proof of Theorem 2.8 that the updated public key is $\mathsf{pk}' = [As' + e]_q$. With the LWE assumption, the updated public key is indistinguishable from a random number $u \in \mathbb{Z}_q^m$, and so is a fresh public key generated by $\mathsf{pk}' \leftarrow \mathsf{PubKeyGen}(\mathsf{sk}')$. This completes the proof. □

Consequently, the following theorem on the IND-KU-CPA security of key-updatable Regev encryption is obtained.

Theorem 2.11. *The key-updatable Regev encryption is IND-KU-CPA secure under the LWE assumption.*

Proof. The theorem follows from Theorem 2.5, Lemma 2.3, and Lemma 2.4. □

Theorem 2.10 and Theorem 2.11 imply that the key-updatable ElGamal and Regev encryption satisfies the forward and post-compromise security in the sense of IND-KU-CPA as with the updatable ones. However, it should be noted that the security achieved by key-updatable constructions is clearly stronger than that of updatable ones because of the removal of update tokens. In other words, the key-updatable schemes fulfill the security in the present against an adversary who is given all the past and future information communicated on a network. Thus, the key-updatable ElGamal and Regev encryption achieves the forward and post-compromise security against both an eavesdropper on the network and an honest-but-curious server.

Chapter 3

Encrypted Control

This chapter presents a general formulation of encrypted control using homomorphic encryption and its correctness notion. To this end, an encoding scheme is introduced for converting a real number to plaintext because a controller typically operates over real numbers rather than positive integers, which can be evaluated in homomorphic encryption. The encrypted control is extended to use updatable and key-updatable homomorphic encryption.

3.1 Encoder and decoder

Chapter 2 viewed homomorphic encryption and its extensions for secure outsourcing computation. So far, the data in the outsourcing computation has been considered as a positive integer. However, in general, a controller whose computation we wish to outsource operates over real numbers rather than integers. This gap between control theory and cryptography scenarios suggests the need for developing an encoding and decoding mechanism to convert real numbers to integers and vice versa.

Such transformation from a real number to an integer was a classical problem in digital and quantized controls. For this reason, there are two main types of encoding schemes in encrypted control, depending on which context they are derived from. The first type represents a real number as a signed fixed-point number, a basic digital control method. Farokhi et al. employed rounding a real number to the number representation in base two for encoding in the Paillier encryption, which is additively homomorphic encryption [95, 96]. Additionally, Darup generalized the encoding to any base [106].

The second encoding type is rounding a real number to the nearest plaintext after scaling, a basic scheme in quantized control. The main difficulty of this type of encoding is how to convert negative real numbers to plaintexts, which are positive integers while preserving homomorphism. Kogiso and Fujita solved this problem by splitting the plaintext space of ElGamal encryption into two spaces and assigning

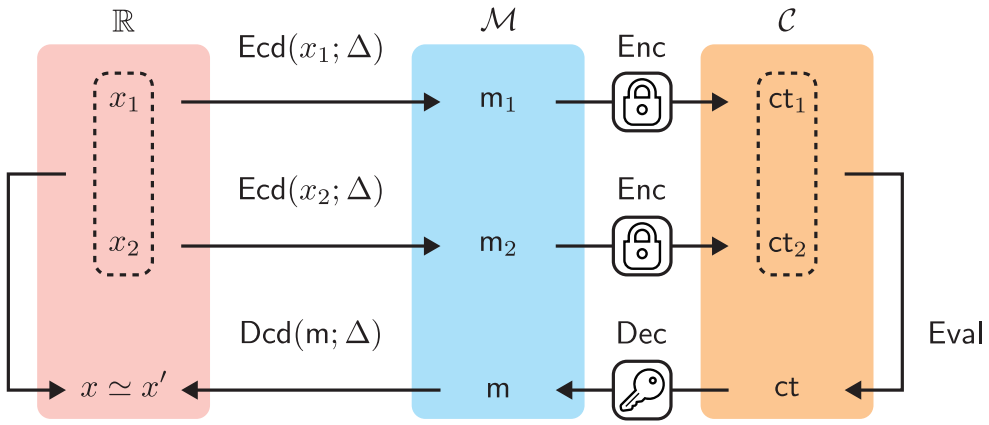


Fig. 3.1: Encoder and decoder for homomorphic encryption.

positive and negative real numbers into each space [94]. More specifically, they considered using plaintexts from 1 to $(p-1)/2$ for positive numbers and from $(p-1)/2+1$ to $p-1$ for negative numbers, where p is the modulus of plaintext space. Kishida applied this encoding to a mid-tread uniform quantizer in the Paillier encryption [108]. Moreover, Kim et al. used the rounding after scaling encoding in the LWE-based encryption [104].

This section provides a general encoding scheme of the second type shown in Fig. 3.1 and analyzes its properties. The encoder Ecd in the figure converts the real numbers x_1 and x_2 to the plaintexts m_1 and m_2 with the scaling factor Δ, respectively. A binary operation between m_1 and m_2 is computed over the ciphertext space by the homomorphic evaluation algorithm Eval, and its result m is recovered by decrypting the output of Eval. The decrypted computation result is restored to the real number x' by the decoder Dcd with the scaling factor used in the encoder. The formal definitions of encoder and decoder are as follows.

Definition 3.1 (Encoding scheme)**.** *An encoding scheme consists of polynomial-time algorithms* $\mathsf{ScalSetup}$*,* Ecd*, and* Dcd *such that:*

- *Scaling setup: The scaling setup algorithm* $\Delta \leftarrow \mathsf{ScalSetup}(\mathsf{params}, \mathcal{X})$ *takes as input public parameters* params *and a bounded set* $\mathcal{X} \subset \mathbb{R}$ *and outputs a scaling factor* $\Delta \in \mathbb{R}^+$*.*

- *Encoder: The encoder algorithm* $\mathsf{m} \leftarrow \mathsf{Ecd}(x;\Delta)$ *takes as input a real number* $x \in \mathcal{X}$ *and the scaling parameter* Δ *and outputs a plaintext* $\mathsf{m} \in \mathcal{M}$*.*

- *Decoder: The decoder algorithm* $x \leftarrow \mathsf{Dcd}(\mathsf{m};\Delta)$ *takes as input a plaintext*

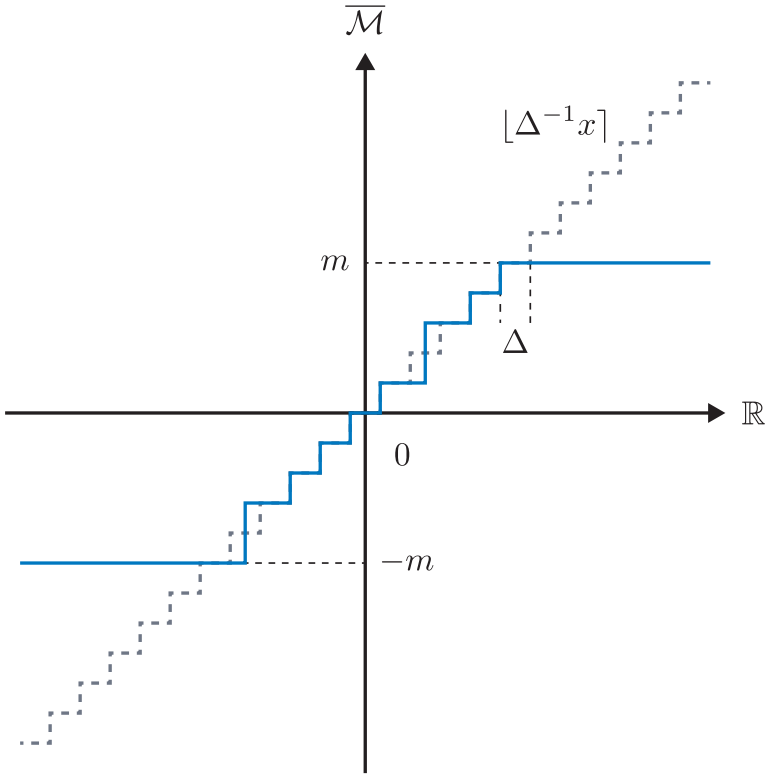


Fig. 3.2: Rounding $\mathbb{R}$ to $\overline{\mathcal{M}}$.

$\mathsf{m} \in \mathcal{M}$ *and the scaling parameter* Δ *and outputs a real number* $x \in \mathcal{X}$.

The second encoding type includes a rounding process, thereby inducing the loss of original data precision. In other words, the encoding scheme causes a quantization error according to its resolution that relies on the degree of scaling and the width of plaintext space. This quantization error is formulated using the algorithms as the difference between decoded and original values.

Definition 3.2 (Quantization error)**.** *Consider an encoding scheme in Definition 3.1. A quantization error* $e(x; \Delta)$ *for* $x \in \mathcal{X}$ *with* Δ *is defined as*

$$e(x; \Delta) \coloneqq \mathsf{Q}(x; \Delta) - x,$$

where $\mathsf{Q}(\cdot; \Delta) \coloneqq \mathsf{Dcd}(\mathsf{Ecd}(\cdot; \Delta); \Delta)$*, and* $\Delta \leftarrow \mathsf{ScalSetup}(\mathsf{params}, \mathcal{X})$.

This section considers building an encoding scheme whose encoder process is shown in Fig. 3.2 and Fig. 3.3. Suppose the modulus of plaintext space $\mathcal{M}$ is n. To begin with, the encoder rounds a real number x to the nearest integer after scaling up by multiplying Δ^{-1} as the gray dashed line in Fig. 3.2. Additionally, as illustrated

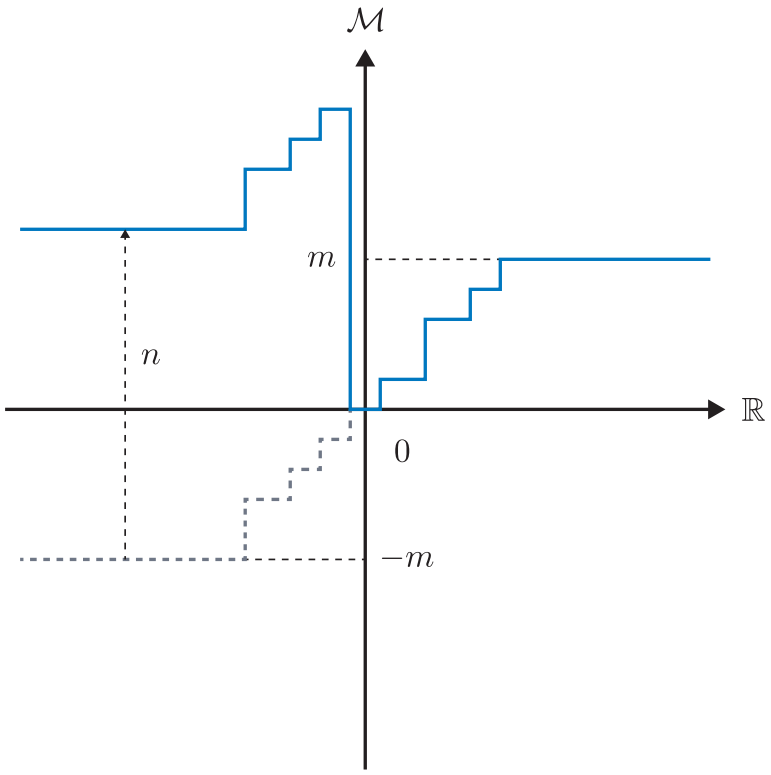


Fig. 3.3: Conversion from $\overline{\mathcal{M}}$ to $\mathcal{M}$.

by the blue solid line in the figure, the encoder chooses the nearest element of the rounded integer from $\overline{\mathcal{M}} \subset [-m, m]$, where $\overline{\mathcal{M}}$ is the set of minimal residues of elements in $\mathcal{M}$ modulo n, and $m = \lfloor (n-1)/2 \rfloor$. Then, the residue of an element in $\overline{\mathcal{M}}$ modulo n is computed as shown in Fig. 3.3. By definition, the residue of an element $\bar{\mathsf{m}}$ in $\overline{\mathcal{M}}$ modulo n is $\bar{\mathsf{m}}$ if $\bar{\mathsf{m}} \geq 0$ and $\bar{\mathsf{m}} + n$ otherwise. The corresponding decoding process is multiplying Δ by the minimal residue of an encoder output modulo n. Note that the plaintext considered here is not necessarily the set of residues modulo n. For instance, the plaintext space of ElGamal encryption in Definition 2.10 is a subset of $\mathbb{Z}_p$.

Our interest here is the magnitude of quantization errors in the encoding scheme. The following lemma reveals that it is bounded by a product of the scaling factor Δ and a half of the maximum width of the union of $\overline{\mathcal{M}}$ and $\{\pm m\}$.

Lemma 3.1. *Let $n \geq 3$ be the modulus of a plaintext space $\mathcal{M} \subset \mathbb{Z}_n$, and let*

$m = \lfloor (n-1)/2 \rfloor$. *If an encoding scheme in Definition 3.1 is given as*

$$\begin{cases} \mathsf{ScalSetup} : (\mathsf{params}, \mathcal{X}) \mapsto \Delta \geq m^{-1}B, \\ \quad \mathsf{Ecd} : x \mapsto \mathsf{m} = \left\lceil \min \arg\min_{\bar{\mathsf{m}} \in \overline{\mathcal{M}}} \left| \lfloor \Delta^{-1} x \rceil - \bar{\mathsf{m}} \right| \right\rceil_n, \\ \quad \mathsf{Dcd} : \mathsf{m} \mapsto x = \Delta [\![\mathsf{m}]\!]_n, \end{cases} \tag{3.1}$$

then it holds that

$$|e(x;\Delta)| \leq \frac{\Delta d_{\max}}{2}$$

for all $x \in \mathcal{X}$*, where* $B \coloneqq \sup\{|x| \mid x \in \mathcal{X}\}$*,* $d_{\max} \coloneqq \max_i \bar{\mathsf{m}}_i - \bar{\mathsf{m}}_{i+1}$*,* $\bar{\mathsf{m}}_i$ *is the* i*th largest element in* $\overline{\mathcal{M}} \cup \{\pm 2m\}$*,* $\overline{\mathcal{M}} \coloneqq \{[\![\mathsf{m}]\!]_n \mid \mathsf{m} \in \mathcal{M}\}$*, and* $e(x;\Delta)$ *is defined in Definition 3.2.*

Proof. Let

$$\mathsf{m}' = \min \arg\min_{\bar{\mathsf{m}} \in \overline{\mathcal{M}}} \left| \lfloor \Delta^{-1} x \rceil - \bar{\mathsf{m}} \right|.$$

The output of the encoder is given as

$$\mathsf{m} = \mathsf{Ecd}(x;\Delta) = [\mathsf{m}']_n = \begin{cases} \mathsf{m}', & x \geq 0, \\ \mathsf{m}' + n, & x < 0. \end{cases}$$

because $|\mathsf{m}'|$ is at most m. Then, the output of the decoder is

$$x' = \mathsf{Dcd}(\mathsf{m};\Delta) = \Delta [\![\mathsf{m}']\!]_n = \Delta \mathsf{m}', \quad [\![\mathsf{m}']\!]_n = \begin{cases} \mathsf{m}', & x \geq 0, \\ \mathsf{m}' - n, & x < 0. \end{cases}$$

It follows from the scaling setup algorithm that $\Delta^{-1}|x| < m$ for all $x \in \mathcal{X}$. This implies that there exist $\bar{\mathsf{m}}_i$ and $\bar{\mathsf{m}}_{i+1}$ such that $\bar{\mathsf{m}}_i > \Delta^{-1}x \geq \bar{\mathsf{m}}_{i+1}$. Consequently, it holds that

$$|e(x;\Delta)| = |x' - x| = |\Delta \mathsf{m}' - x| = \Delta |\mathsf{m}' - \Delta^{-1}x| \leq \frac{\Delta(\bar{\mathsf{m}}_i - \bar{\mathsf{m}}_{i+1})}{2} \leq \frac{\Delta d_{\max}}{2},$$

where m' is either $\bar{\mathsf{m}}_i$ or $\bar{\mathsf{m}}_{i+1}$. □

Moreover, if the plaintext space is identical to $\mathbb{Z}_n$, a quantization error is bounded by half of the scaling factor.

Corollary 3.1. *Let n and m be as in Lemma 3.1. Given the encoding scheme* (3.1)*. If $\mathcal{M} = \mathbb{Z}_n$, it holds that $|e(x;\Delta)| \leq \Delta/2$ for all $x \in \mathcal{X}$.*

Proof. Recall that $\Delta^{-1}|x| < m$ holds for all $x \in \mathcal{X}$. It follows that

$$x' = \mathsf{Dcd}(\mathsf{Ecd}(x;\Delta);\Delta) = \Delta\left\lfloor \Delta^{-1}x \right\rceil$$

because $\lfloor \Delta^{-1}x \rceil$ is always in $\overline{\mathcal{M}} = \{-m, \ldots, m\}$. Hence, it holds that

$$|e(x;\Delta)| = |x' - x| = |\Delta\left\lfloor \Delta^{-1}x \right\rceil - x| = |\Delta(\Delta^{-1}x + \epsilon) - x| = |\Delta\epsilon| \leq \frac{\Delta}{2},$$

where $\epsilon \in (-1/2, 1/2]$. □

It should be noted that the encoding scheme (3.1) is a general representation of the second type encoding, such as the schemes in [94, 104, 111, 142, 143]. Lemma 3.1 implies that quantization errors in such encoding schemes can be reduced by decreasing a scaling factor. Meanwhile, the minimum value of the scaling factor is restricted by the size of plaintext space and the range of values to be encoded. A value to be encoded in our encrypted control scenario is specified by controller parameters and outputs of a plant, namely the design of the original control system. The size of plaintext space, i.e., its modulus, generally depends on a security parameter. For instance, the plaintext space modulus of ElGamal encryption increases as its security parameter increases. A scaling factor in such a case can be reduced by increasing a security parameter. In contrast, increasing a security parameter in LWE-based encryption, including the Regev encryption, typically leads to decreasing plaintext space size because of reducing the size of ciphertext space. Hence, decreasing a scaling factor is required to degrade a security level. Note that although this compromise can be relaxed by increasing the size of a secret key, it also increases computation costs for encryption and decryption.

This thesis requires the encoding scheme (3.1) to remain homomorphism of encryption depicted in Fig. 3.1. In other words, a decrypted and decoded result x' of the output of a homomorphic evaluation algorithm should be almost the same as the computation result x of the original real numbers. Recall that x' is not necessarily equal to x due to quantization errors. To this end, the encoder and decoder need to inherit arithmetic over real numbers, similar to homomorphism in homomorphic encryption. The lemma below shows that the multiplication of decoded values is equal to a decoded value of multiplication of encoded values.

Lemma 3.2. *Given the encoding scheme* (3.1)*. Let n, m, and $\mathcal{M}$ be as in Lemma 3.1. It holds that, for every $N \in \mathbb{N}$,*

$$\prod_{i=1}^{N} \mathsf{Dcd}(\mathsf{m}_i; \Delta) = \mathsf{Dcd}\left(\prod_{i=1}^{N} \mathsf{m}_i; \Delta^N\right)$$

for all $\mathsf{m}_1, \ldots, \mathsf{m}_N \in \mathcal{M}$ such that $\left|\prod_{i=1}^{N} [\![\mathsf{m}_i]\!]_n\right| \leq m$.

Proof. The direct calculation yields the statement.

$$\begin{aligned}
\prod_{i=1}^{N} \mathsf{Dcd}(\mathsf{m}_i; \Delta) &= \prod_{i=1}^{N} \Delta [\![\mathsf{m}_i]\!]_n, \\
&= \Delta^N \prod_{i=1}^{N} [\![\mathsf{m}_i]\!]_n, \\
&= \Delta^N \left[\!\!\left[\prod_{i=1}^{N} [\![\mathsf{m}_i]\!]_n \right]\!\!\right]_n, \\
&= \Delta^N \left[\!\!\left[\prod_{i=1}^{N} \mathsf{m}_i \right]\!\!\right]_n, \\
&= \mathsf{Dcd}\left(\prod_{i=1}^{N} \mathsf{m}_i; \Delta^N\right).
\end{aligned}$$

Note that the third equality follows from $\left|\prod_{i=1}^{N} [\![\mathsf{m}_i]\!]_n\right| \leq m$. This completes the proof. □

Furthermore, the addition of decoded values becomes equivalent to a decoded value of the addition of encoded values.

Lemma 3.3. *Given the encoding scheme* (3.1)*. Let n, m, and $\mathcal{M}$ be as in Lemma 3.1. It holds that, for every $N \in \mathbb{N}$,*

$$\sum_{i=1}^{N} \mathsf{Dcd}(\mathsf{m}_i; \Delta) = \mathsf{Dcd}\left(\sum_{i=1}^{N} \mathsf{m}_i; \Delta\right)$$

for all $\mathsf{m}_1, \ldots, \mathsf{m}_N \in \mathcal{M}$ such that $\left|\sum_{i=1}^{N} [\![\mathsf{m}_i]\!]_n\right| \leq m$.

Proof. Similar to the Lemma 3.2, the direct calculation yields the statement.

$$\begin{aligned}
\sum_{i=1}^{N} \mathsf{Dcd}(\mathsf{m}_i; \Delta) &= \sum_{i=1}^{N} \Delta [\![\mathsf{m}_i]\!]_n, \\
&= \Delta \sum_{i=1}^{N} [\![\mathsf{m}_i]\!]_n, \\
&= \Delta \left[\!\!\left[\sum_{i=1}^{N} [\![\mathsf{m}_i]\!]_n \right]\!\!\right]_n, \\
&= \Delta \left[\!\!\left[\sum_{i=1}^{N} \mathsf{m}_i \right]\!\!\right]_n, \\
&= \mathsf{Dcd}\left(\sum_{i=1}^{N} \mathsf{m}_i; \Delta \right).
\end{aligned}$$

This completes the proof. □

Lemma 3.2 and Lemma 3.3 imply that the encoding scheme (3.1) preserves a kind of structures of multiplication and addition over a subset of $\mathbb{R}$. This property helps analyze quantization errors in encrypted controller computation, described in the next section.

3.2 Encrypted control using homomorphic encryption

Encrypted control is a control framework for secure outsourcing computation of controllers to an untrusted server, such as a public cloud. The control was first realized by Kogiso and Fujita using multiplicatively homomorphic encryption [94]. After their study, Farokhi et al. proposed encrypted control using additively homomorphic encryption [95, 96], and Kim et al. considered that of fully homomorphic encryption [97]. Such early studies of encrypted control treated to construct a computation methodology of a linear-time-invariant controller over encrypted data.

This section revisits the encrypted control of a linear-time-invariant controller using homomorphic encryption to give its formal definition in a cryptographic manner and to formulate the correctness notion of encrypted control as with encryption schemes in Chapter 2. Moreover, it provides two general constructions of encrypted

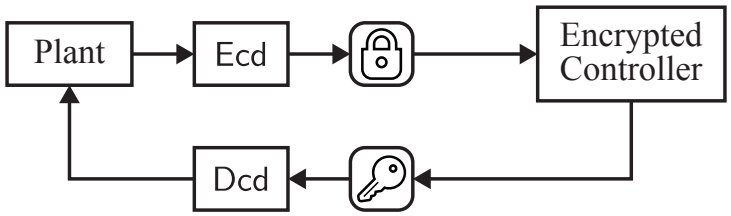


Fig. 3.4: Encrypted control system using holomorphic encryption.

control using multiplicatively and additively homomorphic encryption with the encoding scheme (3.1) and analyzes quantization errors in the constructions. Encrypted control using the ElGamal and Regev encryption is also introduced as its specific realizations. The encrypted control definition is also extended to that using updatable and key-updatable homomorphic encryption.

3.2.1 Definitions

This section provides a general definition of encrypted control using homomorphic encryption in a cryptographic manner so that the various conventional approaches for encrypted control can be treated in a unified method. Fig. 3.4 shows a schematic picture of a considered encrypted control system, which can be interpreted as a client-server model with outsourcing computation using homomorphic encryption. As a client, the plant transmits a controller input, such as sensor data, to the encrypted controller while encoding and encrypting it. As a server, the encrypted controller computes a controller output over encrypted data and returns it to the plant. The plant then recovers a control input by decrypting and decoding the controller output. The encrypted controller in such a control scenario is formally defined as follows.

Definition 3.3 (Encrypted control)**.** *Let f be a control law. An encrypted controller of f is a polynomial-time algorithm* EC *such that:*

- *Encrypted controller: The encrypted control algorithm* $\mathsf{ct} \leftarrow \mathsf{EC}(f, \varphi_1, \mathsf{ct}_2)$ *takes as input the control law, either a plaintext matrix* $\varphi_1 \in \mathcal{M}^{\alpha\times\beta}$ *or ciphertext matrix* $\varphi_1 \in \mathcal{C}^{\alpha\times\beta}$*, and a ciphertext vector* $\mathsf{ct}_2 \in \mathcal{C}^{\beta}$ *and outputs either a ciphertext vector* $\mathsf{ct} \in \mathcal{C}^{\alpha}$ *or ciphertext matrix* $\mathsf{ct} \in \mathcal{C}^{\alpha\times\beta}$*, respectively.*

In the definition, an encrypted controller is a polynomial-time algorithm that computes a ciphertext vector from the input matrix and vector, unlike encryp-

tion/decryption and encoder/decoder algorithms that perform for a scalar message. Along with this definition, in what follows, encryption/decryption and encoder/decoder algorithms are supposed to perform for each input element if they take a vector or matrix as input. It should be noted that only one of the controller inputs or parameters can be encrypted if additively homomorphic encryption is employed, as discussed in the existing studies [96]. However, both the inputs and parameters are encrypted using multiplicatively homomorphic encryption. For this reason, the input matrix in Definition 3.3 is defined as either plaintext or ciphertext.

We are now ready to define the correctness of an encrypted controller. As previously discussed, encrypting real numbers requires an encoding scheme that induces a quantization error. Hence, the correctness of the encrypted controller must be considered together with accuracy, unlike the encryption schemes in Chapter 2. To this end, here this thesis introduces δ-correctness. It means that the deviation of the decrypted and decoded result of the encryption controller output from the original controller output becomes smaller than δ except with a negligibly small probability.

Definition 3.4 (δ-correctness)**.** *Given a homomorphic encryption scheme in Definition 2.8 and an encoding scheme in Definition 3.1. Suppose $f : (\Phi, \xi) \mapsto \psi$ is a control law, where $\Phi \in \mathcal{X}^{\alpha\times\beta}$ is a controller parameter, $\xi \in \mathcal{X}^{\beta}$ is a controller input, and $\psi \in \mathcal{X}^{\alpha}$ is a controller output. An encrypted controller* EC *of f is δ-correct if there exist $\delta \in \mathbb{R}^+$ and a negligible function* negl *such that*

$$\Pr\left[\|\psi' - \psi\|_\infty \leq \delta \;\middle|\; \begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \Delta \leftarrow \mathsf{ScalSetup}(\mathsf{params}, \mathcal{X}) \\ \mathsf{ct}_\xi \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{Ecd}(\xi; \Delta)) \\ \mathsf{ct}_\psi \leftarrow \mathsf{EC}(f, \varphi_\Phi, \mathsf{ct}_\xi) \\ \psi' \leftarrow \mathsf{Dcd}(\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}_\psi); \Delta) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda)$$

for all $\lambda \in \mathbb{N}$, for all $\Phi \in \mathcal{X}^{\alpha\times\beta}$, and for all $\xi \in \mathcal{X}^{\beta}$, where $\psi = f(\Phi, \xi)$, φ_Φ is given as either $\varphi_\Phi \leftarrow \mathsf{Ecd}(\Phi, \Delta)$ or $\varphi_\Phi \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{Ecd}(\Phi; \Delta))$, and $\|\cdot\|_\infty$ is the maximum norm.

It should be stressed that the definition covers the correctness of almost all encrypted controls using any homomorphic encryption, including nonlinear control, distributed control, and so on, under the encoding scheme in Definition 3.1 because f in the definition is a general representation of control law. Meanwhile, f represents

not only a control law but also a system having input and output. This fact suggests that the definitions of encrypted control and δ-correctness are not restricted to controllers by formulating f appropriately. For example, if a Kalman filter is represented as $f : (\Phi, \xi) \mapsto \psi$, where Φ is a system parameter, ψ is an estimated state, and ξ consists of a sensor output, system input, and previous estimated-state, then its encrypted version and correctness can be defined by Definition 3.3 and Definition 3.4, respectively.

3.2.2 Constructions

Following Definition 3.3 and Definition 3.4, this section builds two general encrypted controllers of a linear-time-invariant controller using multiplicatively and additively homomorphic encryption.

The first construction using multiplicatively homomorphic encryption computes the element-wise product of a ciphertext matrix and vector, which correspond to a controller parameter and input, respectively. The decoder here is modified to perform decoding of the decrypted matrix and aggregating each row of the matrix [94] so that the overall process of the encrypted controller, decryption, and decoder is comparable to the product of the controller parameter matrix and the controller input vector. Formally, the encrypted controller is given as follows.

Theorem 3.1. *Given a multiplicatively homomorphic encryption scheme. Consider the encoding scheme* (3.1) *and redefine the decoder as* $\mathsf{Dcd}_{\mathsf{MHE}}(\cdot;\Delta) \coloneqq \mathsf{Sum}(\mathsf{Dcd}(\cdot;\Delta^2))$ *with*

$$\mathsf{Sum} : \mathbb{R}^{\alpha\times\beta} \to \mathbb{R}^{\alpha} : M \mapsto \begin{bmatrix} \sum_{j=1}^{\beta} M_{1j} \\ \vdots \\ \sum_{j=1}^{\beta} M_{\alpha j} \end{bmatrix}.$$

Let m, B, and $d_{\max}$ be as in Lemma 3.1. Suppose $f : \mathcal{X}^{\alpha\times\beta} \times \mathcal{X}^{\beta} \to \mathcal{X}^{\alpha} : (\Phi, \xi) \mapsto \psi = \Phi\xi$ is a linear-time-invariant control law. If $\Delta^{-2}\|\mathsf{Q}(\Phi;\Delta)\|_{\max}\|\mathsf{Q}(\xi;\Delta)\|_{\infty} \le m$, the algorithm

$$\mathsf{EC} : (f, \mathsf{ct}_{\Phi}, \mathsf{ct}_{\xi}) \mapsto \begin{bmatrix} \mathsf{ct}_{\Phi_{11}} \boxtimes \mathsf{ct}_{\xi_1} & \cdots & \mathsf{ct}_{\Phi_{1\beta}} \boxtimes \mathsf{ct}_{\xi_\beta} \\ \vdots & \ddots & \vdots \\ \mathsf{ct}_{\Phi_{\alpha 1}} \boxtimes \mathsf{ct}_{\xi_1} & \cdots & \mathsf{ct}_{\Phi_{\alpha\beta}} \boxtimes \mathsf{ct}_{\xi_\beta} \end{bmatrix} \tag{3.2}$$

is a $\beta(\Delta d_{\max} B + (\Delta d_{\max}/2)^2)$-correct encrypted controller of f, where $\mathsf{ct}_{\Phi} \leftarrow \mathsf{Enc}(\mathsf{pk},$

$\mathsf{Ecd}(\Phi;\Delta))$, $\mathsf{ct}_\xi \leftarrow \mathsf{Enc}(\mathsf{pk},\mathsf{Ecd}(\xi;\Delta))$, $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$, $\Delta \leftarrow \mathsf{ScalSetup}(\mathsf{params}, \mathcal{X})$, Q *is defined in Definition 3.2, and* $\|\cdot\|_{\max}$ *is the max norm.*

Proof. Let $\mathsf{ct}_\psi \leftarrow \mathsf{EC}(f,\mathsf{ct}_\Phi,\mathsf{ct}_\xi)$. It follows from the multiplicative homomorphism of the encryption scheme that the decryption of ct_ψ is computed as

$$\begin{aligned}
\mathsf{m}_\psi &= \mathsf{Dec}(\mathsf{sk},\mathsf{ct}_\psi),\\
&= \begin{bmatrix} \mathsf{Dec}(\mathsf{sk},\mathsf{ct}_{\Phi_{11}} \boxtimes \mathsf{ct}_{\xi_1}) & \cdots & \mathsf{Dec}(\mathsf{sk},\mathsf{ct}_{\Phi_{1\beta}} \boxtimes \mathsf{ct}_{\xi_\beta}) \\ \vdots & \ddots & \vdots \\ \mathsf{Dec}(\mathsf{sk},\mathsf{ct}_{\Phi_{\alpha 1}} \boxtimes \mathsf{ct}_{\xi_1}) & \cdots & \mathsf{Dec}(\mathsf{sk},\mathsf{ct}_{\Phi_{\alpha\beta}} \boxtimes \mathsf{ct}_{\xi_\beta}) \end{bmatrix},\\
&= \begin{bmatrix} \mathsf{Ecd}(\Phi_{11};\Delta)\mathsf{Ecd}(\xi_1;\Delta) & \cdots & \mathsf{Ecd}(\Phi_{1\beta};\Delta)\mathsf{Ecd}(\xi_\beta;\Delta) \\ \vdots & \ddots & \vdots \\ \mathsf{Ecd}(\Phi_{\alpha 1};\Delta)\mathsf{Ecd}(\xi_1;\Delta) & \cdots & \mathsf{Ecd}(\Phi_{\alpha\beta};\Delta)\mathsf{Ecd}(\xi_\beta;\Delta) \end{bmatrix},
\end{aligned}$$

with probability at least $1-\mathsf{negl}(\lambda)$. The condition $\|\mathsf{Q}(\Phi;\Delta)/\Delta\|_{\max}\|\mathsf{Q}(\xi;\Delta)/\Delta\|_\infty \le m$ yields

$$\left|[\![\mathsf{Ecd}(\Phi_{ij};\Delta)]\!]_n [\![\mathsf{Ecd}(\xi_j;\Delta)]\!]_n\right| \le m$$

for all $1 \le i \le \alpha$ and for all $1 \le j \le \beta$ because

$$\begin{aligned}
&\Delta^{-2}\|\mathsf{Q}(\Phi;\Delta)\|_{\max}\|\mathsf{Q}(\xi;\Delta)\|_\infty\\
&= \Delta^{-2}\max_{i,j}|\mathsf{Q}(\Phi_{ij};\Delta)|\max_j|\mathsf{Q}(\xi_j;\Delta)|,\\
&= \max_{i,j}|\mathsf{Dcd}(\mathsf{Ecd}(\Phi_{ij};\Delta);\Delta)/\Delta|\max_j|\mathsf{Dcd}(\mathsf{Ecd}(\xi_j;\Delta);\Delta)/\Delta|,\\
&= \max_{i,j}\left|[\![\mathsf{Ecd}(\Phi_{ij};\Delta);\Delta)]\!]_n\right|\max_j\left|[\![\mathsf{Ecd}(\xi_j;\Delta);\Delta)]\!]_n\right|,\\
&\ge \max_{i,j}\left|[\![\mathsf{Ecd}(\Phi_{ij};\Delta);\Delta)]\!]_n [\![\mathsf{Ecd}(\xi_j;\Delta);\Delta)]\!]_n\right|.
\end{aligned}$$

Hence, Lemma 3.2 implies that the decoded result of the (i,j)-entry of m_ψ is given as

$$\begin{aligned}
\mathsf{Dcd}\big(\mathsf{m}_{\psi_{ij}};\Delta^2\big) &= \mathsf{Dcd}\big(\mathsf{Ecd}(\Phi_{ij};\Delta)\mathsf{Ecd}(\xi_j;\Delta);\Delta^2\big),\\
&= \mathsf{Dcd}(\mathsf{Ecd}(\Phi_{ij};\Delta);\Delta)\mathsf{Dcd}(\mathsf{Ecd}(\xi_j;\Delta);\Delta),\\
&= \mathsf{Q}(\Phi_{ij};\Delta)\mathsf{Q}(\xi_j;\Delta).
\end{aligned}$$

Let $\psi' \leftarrow \mathsf{Dcd}_{\mathsf{MHE}}(\mathsf{m}_\psi; \Delta)$. It holds that

$$\begin{aligned}
\psi' &= \mathsf{Dcd}_{\mathsf{MHE}}(\mathsf{m}_\psi; \Delta), \\
&= \mathsf{Sum}\left(\begin{bmatrix} \mathsf{Q}(\Phi_{11};\Delta)\mathsf{Q}(\xi_1;\Delta) & \cdots & \mathsf{Q}(\Phi_{1\beta};\Delta)\mathsf{Q}(\xi_\beta;\Delta) \\ \vdots & \ddots & \vdots \\ \mathsf{Q}(\Phi_{\alpha 1};\Delta)\mathsf{Q}(\xi_1;\Delta) & \cdots & \mathsf{Q}(\Phi_{\alpha\beta};\Delta)\mathsf{Q}(\xi_\beta;\Delta) \end{bmatrix}\right), \\
&= \begin{bmatrix} \sum_{j=1}^{\beta} \mathsf{Q}(\Phi_{1j};\Delta)\mathsf{Q}(\xi_j;\Delta) \\ \vdots \\ \sum_{j=1}^{\beta} \mathsf{Q}(\Phi_{\alpha j};\Delta)\mathsf{Q}(\xi_j;\Delta) \end{bmatrix}.
\end{aligned}$$

Moreover, it holds that

$$\begin{aligned}
\|\psi' - f(\Phi,\xi)\|_\infty &= \max_i |\psi'_i - \psi_i|, \\
&= \max_i \left| \sum_{j=1}^{\beta} \mathsf{Q}(\Phi_{ij};\Delta)\mathsf{Q}(\xi_j;\Delta) - \sum_{j=1}^{\beta} \Phi_{ij}\xi_j \right|, \\
&= \max_i \left| \sum_{j=1}^{\beta} \mathsf{Q}(\Phi_{ij};\Delta)\mathsf{Q}(\xi_j;\Delta) - \Phi_{ij}\xi_j \right|, \\
&= \max_i \left| \sum_{j=1}^{\beta} (\Phi_{ij} + e(\Phi_{ij};\Delta))(\xi_j + e(\xi_j;\Delta)) - \Phi_{ij}\xi_j \right|, \\
&= \max_i \left| \sum_{j=1}^{\beta} \Phi_{ij}e(\xi_j;\Delta) + e(\Phi_{ij};\Delta)\xi_j + e(\Phi_{ij};\Delta)e(\xi_j;\Delta) \right|.
\end{aligned}$$

Lemma 3.1 yields that $|e(\Phi_{ij};\Delta)|$ and $|e(\xi_j;\Delta)|$ are bounded from above by $\Delta d_{\max}/2$. Additionally, $|\Phi_{ij}| < B$ and $|\xi_j| < B$ hold for all i and j. Consequently, $\|\psi' - f(\Phi,\xi)\|_\infty$ is bounded as follows.

$$\begin{aligned}
\|\psi' - f(\Phi,\xi)\|_\infty &\leq \left| \sum_{j=1}^{\beta} \frac{\Delta d_{\max}}{2} B + \frac{\Delta d_{\max}}{2} B + \left(\frac{\Delta d_{\max}}{2}\right)^2 \right|, \\
&= \left| \beta \left(\Delta d_{\max} B + \left(\frac{\Delta d_{\max}}{2}\right)^2 \right) \right|,
\end{aligned}$$

$$= \beta\left(\Delta d_{\max} B + \left(\frac{\Delta d_{\max}}{2}\right)^2\right).$$

This completes the proof. □

The theorem shows that the encrypted controller (3.2) of a linear-time-invariant controller using any multiplicatively homomorphic encryption can satisfy the $\beta(\Delta d_{\max} B + (\Delta d_{\max}/2)^2)$-correctness under the modified encoding scheme of (3.1). It implies that a control system with the encrypted controller can achieve almost the same control performance as the original control system when choosing a scaling factor Δ such that $\beta(\Delta d_{\max} B + (\Delta d_{\max}/2)^2) \ll 1$. Here, the condition is equivalent to $d_{\max}^2 \Delta^2 + 4B d_{\max} \Delta \ll 4\beta^{-1}$. If $d_{\max}^2 \Delta^2 + 4B d_{\max} \Delta - 4\beta^{-1} = 0$, the scaling factor is given as $\Delta = 2d_{\max}^{-1}(\sqrt{B^2 + \beta^{-1}} - B)$. Therefore, a choice satisfying

$$\Delta \ll \frac{2}{d_{\max}}\left(\sqrt{B^2 + \beta^{-1}} - B\right)$$

is recommended. Note that, from Theorem 3.1, a scaling factor is bounded from below by

$$\Delta \geq \sqrt{\frac{\|\mathsf{Q}(\Phi;\Delta)\|_{\max}\|\mathsf{Q}(\xi;\Delta)\|_\infty}{m}}.$$

A security parameter or control law needs to be redesigned when neither condition is met.

The corollary below, which is on a realization of encrypted control using the ElGamal encryption, immediately follows from the homomorphism of ElGamal encryption and Theorem 3.1.

Corollary 3.2. *Given the ElGamal encryption scheme in Definition 2.10. Consider the encoding scheme in Theorem 3.1. Let B and $d_{\max}$ be as in Lemma 3.1, and let Φ, ξ, Δ, and β be as in Theorem 3.1. Let p be as in Definition 2.10. If $\Delta^{-2}\|\mathsf{Q}(\Phi;\Delta)\|_{\max}\|\mathsf{Q}(\xi;\Delta)\|_\infty \leq (p-1)/2$, (3.2) is $\beta(\Delta d_{\max} B + (\Delta d_{\max}/2)^2)$-correct, where Q is defined in Definition 3.2.*

Proof. The corollarry follows from Proposition 2.2 and Theorem 3.1. □

The next encrypted controller construction is based on the use of additively homomorphic encryption. As opposed to encrypted control using multiplicatively homomorphic encryption, controller parameters in the construction with additively

homomorphic encryption cannot be encrypted because computing a product of ciphertexts is impossible. Additively homomorphic encryption allows the computation of plaintext-ciphertext multiplication by repeating the addition as many times as the plaintext instead of the limitation. Hence, the encrypted controller using additively holomorphic encryption computes the product of a plaintext controller-parameter matrix and a ciphertext input vector as follows.

Theorem 3.2. *Given an additively homomorphic encryption scheme. Consider the encoding scheme* (3.1) *and redefine the decoder as* $\mathsf{Dcd}_{\mathsf{AHE}}(\cdot;\Delta) \coloneqq \mathsf{Dcd}(\cdot;\Delta^2)$. *Let* m, B, *and* $d_{\max}$ *be as in Lemma 3.1, and let* f, Φ, ξ, *and* β *be as in Theorem 3.1. If* $\Delta^{-2}\beta\|\mathsf{Q}(\Phi;\Delta)\|_{\max}\|\mathsf{Q}(\xi;\Delta)\|_\infty \leq m$, *the algorithm*

$$\mathsf{EC} : (f, \mathsf{m}_\Phi, \mathsf{ct}_\xi) \mapsto \begin{bmatrix} (\mathsf{m}_{\Phi_{11}} \boxdot \mathsf{ct}_{\xi_1}) \boxplus \cdots \boxplus (\mathsf{m}_{\Phi_{1\beta}} \boxdot \mathsf{ct}_{\xi_\beta}) \\ \vdots \\ (\mathsf{m}_{\Phi_{\alpha 1}} \boxdot \mathsf{ct}_{\xi_1}) \boxplus \cdots \boxplus (\mathsf{m}_{\Phi_{\alpha\beta}} \boxdot \mathsf{ct}_{\xi_\beta}) \end{bmatrix} \tag{3.3}$$

is a $\beta(\Delta d_{\max}B+(\Delta d_{\max}/2)^2)$*-correct encrypted controller of* f, *where* $\mathsf{m}_\Phi \leftarrow \mathsf{Ecd}(\Phi;\Delta)$, $\mathsf{ct}_\xi \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{Ecd}(\xi;\Delta))$, $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$, $\Delta \leftarrow \mathsf{ScalSetup}(\mathsf{params}, \mathcal{X})$, *and* Q *is defined in Definition 3.2.*

Proof. The proof flow is almost the same as that of Theorem 3.1. Let $\mathsf{ct}_\psi \leftarrow \mathsf{EC}(f, \mathsf{ct}_\Phi, \mathsf{ct}_\xi)$. It follows from the additive homomorphism of the encryption scheme that the decryption of ct_ψ is computed as

$$\begin{aligned}
\mathsf{m}_\psi &= \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}_\psi), \\
&= \begin{bmatrix} \mathsf{Dec}(\mathsf{sk}, (\mathsf{m}_{\Phi_{11}} \boxdot \mathsf{ct}_{\xi_1}) \boxplus \cdots \boxplus (\mathsf{m}_{\Phi_{1\beta}} \boxdot \mathsf{ct}_{\xi_\beta})) \\ \vdots \\ \mathsf{Dec}(\mathsf{sk}, (\mathsf{m}_{\Phi_{\alpha 1}} \boxdot \mathsf{ct}_{\xi_1}) \boxplus \cdots \boxplus (\mathsf{m}_{\Phi_{\alpha\beta}} \boxdot \mathsf{ct}_{\xi_\beta})) \end{bmatrix}, \\
&= \begin{bmatrix} \mathsf{Ecd}(\Phi_{11};\Delta)\mathsf{Ecd}(\xi_1;\Delta) + \cdots + \mathsf{Ecd}(\Phi_{1\beta};\Delta)\mathsf{Ecd}(\xi_\beta;\Delta) \\ \vdots \\ \mathsf{Ecd}(\Phi_{\alpha 1};\Delta)\mathsf{Ecd}(\xi_1;\Delta) + \cdots + \mathsf{Ecd}(\Phi_{\alpha\beta};\Delta)\mathsf{Ecd}(\xi_\beta;\Delta) \end{bmatrix}, \\
&= \begin{bmatrix} \sum_{j=1}^{\beta} \mathsf{Ecd}(\Phi_{1j};\Delta)\mathsf{Ecd}(\xi_j;\Delta) \\ \vdots \\ \sum_{j=1}^{\beta} \mathsf{Ecd}(\Phi_{\alpha j};\Delta)\mathsf{Ecd}(\xi_j;\Delta) \end{bmatrix},
\end{aligned}$$

with probability at least $1-\mathsf{negl}(\lambda)$. The condition $\Delta^{-2}\beta\|\mathsf{Q}(\Phi;\Delta)\|_{\max}\|\mathsf{Q}(\xi;\Delta)\|_\infty \leq m$ yields

$$\left|[\![\mathsf{Ecd}(\Phi_{i1};\Delta)]\!]_n[\![\mathsf{Ecd}(\xi_1;\Delta)]\!]_n + \cdots + [\![\mathsf{Ecd}(\Phi_{i\beta};\Delta)]\!]_n[\![\mathsf{Ecd}(\xi_\beta;\Delta)]\!]_n\right| \leq m$$

for all $1 \leq i \leq \alpha$ because

$$\begin{aligned}
&\Delta^{-2}\beta\|\mathsf{Q}(\Phi;\Delta)\|_{\max}\|\mathsf{Q}(\xi;\Delta)\|_\infty \\
&= \Delta^{-2}\beta \max_{i,j}|\mathsf{Q}(\Phi_{ij};\Delta)| \max_j|\mathsf{Q}(\xi_j;\Delta)|, \\
&= \beta \max_{i,j}|\mathsf{Dcd}(\mathsf{Ecd}(\Phi_{ij};\Delta);\Delta)/\Delta| \max_j|\mathsf{Dcd}(\mathsf{Ecd}(\xi_j;\Delta);\Delta)/\Delta|, \\
&= \beta \max_{i,j}\left|[\![\mathsf{Ecd}(\Phi_{ij};\Delta);\Delta)]\!]_n\right| \max_j\left|[\![\mathsf{Ecd}(\xi_j;\Delta);\Delta)]\!]_n\right|, \\
&\geq \beta \max_{i,j}\left|[\![\mathsf{Ecd}(\Phi_{ij};\Delta);\Delta)]\!]_n[\![\mathsf{Ecd}(\xi_j;\Delta);\Delta)]\!]_n\right|, \\
&\geq \max_i\left|\sum_{j=1}^{\beta}[\![\mathsf{Ecd}(\Phi_{ij};\Delta);\Delta)]\!]_n[\![\mathsf{Ecd}(\xi_j;\Delta);\Delta)]\!]_n\right|.
\end{aligned}$$

Hence, Lemma 3.2 and Lemma 3.3 imply that the decoded result of the ith element of m_ψ is given as

$$\begin{aligned}
\mathsf{Dcd}\big(\mathsf{m}_{\psi_i};\Delta^2\big) &= \mathsf{Dcd}\left(\sum_{j=1}^{\beta}\mathsf{Ecd}(\Phi_{ij};\Delta)\mathsf{Ecd}(\xi_j;\Delta);\Delta^2\right), \\
&= \sum_{j=1}^{\beta}\mathsf{Dcd}(\mathsf{Ecd}(\Phi_{ij};\Delta);\Delta)\mathsf{Dcd}(\mathsf{Ecd}(\xi_j;\Delta);\Delta), \\
&= \sum_{j=1}^{\beta}\mathsf{Q}(\Phi_{ij};\Delta)\mathsf{Q}(\xi_j;\Delta).
\end{aligned}$$

Let $\psi' \leftarrow \mathsf{Dcd}_{\mathsf{AHE}}(\mathsf{m}_\psi;\Delta)$. It holds that

$$\psi' = \mathsf{Dcd}_{\mathsf{AHE}}(\mathsf{m}_\psi;\Delta) = \begin{bmatrix} \sum_{j=1}^{\beta}\mathsf{Q}(\Phi_{1j};\Delta)\mathsf{Q}(\xi_j;\Delta) \\ \vdots \\ \sum_{j=1}^{\beta}\mathsf{Q}(\Phi_{\alpha j};\Delta)\mathsf{Q}(\xi_j;\Delta) \end{bmatrix}.$$

Therefore, the theorem follows from the proof of Theorem 3.1. □

One of the additively homomorphic encryption schemes for realizing the encrypted control (3.3) is the Regev encryption. The following corollary shows that the encrypted control with the encryption scheme achieves correctness if a noise used in the secret key generation is bounded by a specific size.

Corollary 3.3. *Given the Regev encryption scheme in Definition 2.11. Consider the encoding scheme in Theorem 3.2. Let B be as in Lemma 3.1, and let Φ, ξ, Δ, and β be as in Theorem 3.1. Let t and χ be as in Definition 2.11. If $\Delta^{-2}\beta \|\mathsf{Q}(\Phi;\Delta)\|_{\max}\|\mathsf{Q}(\xi;\Delta)\|_\infty \le \lfloor (t-1)/2 \rfloor$ and if χ is $(q/(2m\beta t^2) - (t/m))$-bounded, then (3.3) is $\beta(\Delta B + (\Delta/2)^2)$-correct, where Q is defined in Definition 3.2.*

Proof. Let params, pk, sk be as in Definition 2.11, and let $\Delta' = \lfloor q/t \rfloor$. By definition, it holds that

$$\begin{aligned}
\mathsf{ct}_{\psi_i} &= (\mathsf{m}_{\Phi_{i1}} \boxdot \mathsf{ct}_{\xi_1}) \boxplus \cdots \boxplus (\mathsf{m}_{\Phi_{i\beta}} \boxdot \mathsf{ct}_{\xi_\beta}) \\
&= \left(\left[\mathsf{m}_{\Phi_{i1}} r_1^\top A\right]_q, \left[\Delta' \mathsf{m}_{\Phi_{i1}} \mathsf{m}_{\xi_1} + \mathsf{m}_{\Phi_{i1}} r_1^\top (As+e)\right]_q \right) \\
&\quad \boxplus \cdots \boxplus \left(\left[\mathsf{m}_{\Phi_{i\beta}} r_\beta^\top A\right]_q, \left[\Delta' \mathsf{m}_{\Phi_{i\beta}} \mathsf{m}_{\xi_\beta} + \mathsf{m}_{\Phi_{i\beta}} r_\beta^\top (As+e)\right]_q \right), \\
&= \left(\left[\left(\sum_{j=1}^{\beta} \mathsf{m}_{\Phi_{ij}} r_j \right)^\top A \right]_q, \left[\Delta' \left(\sum_{j=1}^{\beta} \mathsf{m}_{\Phi_{ij}} \mathsf{m}_{\xi_j} \right) + \left(\sum_{j=1}^{\beta} \mathsf{m}_{\Phi_{ij}} r_j \right)^\top (As+e) \right]_q \right),
\end{aligned}$$

where $\mathsf{m}_{\Phi_{ij}} \leftarrow \mathsf{Ecd}(\Phi_{ij};\Delta)$, $\mathsf{m}_{\xi_j} \leftarrow \mathsf{Ecd}(\xi_j;\Delta)$, $\mathsf{ct}_{\xi_j} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_{\xi_j})$, and $r_j \in \mathbb{Z}_2^m$ is a random vector used in the encryption of m_{ξ_j}. It follows from the proof of Proposition 2.3 that

$$\mathsf{m}_{\psi_i} = \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}_{\psi_i}) = \left[\sum_{j=1}^{\beta} \mathsf{m}_{\Phi_{ij}} \mathsf{m}_{\xi_j} \right]_t$$

if

$$\left| \frac{t}{q} \left(\sum_{j=1}^{\beta} \mathsf{m}_{\Phi_{ij}} r_j^\top e - \epsilon \sum_{j=1}^{\beta} \mathsf{m}_{\Phi_{ij}} \mathsf{m}_{\xi_j} \right) \right| < \frac{1}{2}.$$

From the proof of Proposition 2.4, a sufficient condition of the above one is given as

$$\left| \frac{t}{q} \left(\sum_{j=1}^{\beta} \mathsf{m}_{\Phi_{ij}} r_j^\top e - \epsilon \sum_{j=1}^{\beta} \mathsf{m}_{\Phi_{ij}} \mathsf{m}_{\xi_j} \right) \right| < \frac{1}{2} \iff \left| \sum_{j=1}^{\beta} \mathsf{m}_{\Phi_{ij}} r_j^\top e - \epsilon \sum_{j=1}^{\beta} \mathsf{m}_{\Phi_{ij}} \mathsf{m}_{\xi_j} \right| < \frac{q}{2t},$$

$$
\begin{aligned}
&\Longleftarrow \sum_{j=1}^{\beta} \mathsf{m}_{\Phi_{ij}} \left| r_j^\top e \right| + \epsilon \sum_{j=1}^{\beta} \mathsf{m}_{\Phi_{ij}} \mathsf{m}_{\xi_j} < \frac{q}{2t}, \\
&\Longleftarrow \sum_{j=1}^{\beta} \left| r_j^\top e \right| + \epsilon \sum_{j=1}^{\beta} \mathsf{m}_{\xi_j} < \frac{q}{2t^2}, \\
&\Longleftarrow \sum_{j=1}^{\beta} \left| r_j^\top e \right| < \frac{q}{2t^2} - \beta t, \\
&\Longleftarrow \beta \sum_{i=1}^{m} |e_i| < \frac{q}{2t^2} - \beta t, \\
&\Longleftarrow |e_i| < \frac{1}{m} \left(\frac{q}{2\beta t^2} - t \right), \ i = 1, \ldots, m,
\end{aligned}
$$

where $\epsilon = q/t - \Delta'$, and $0 \leq \epsilon < 1$. Therefore, the corollary follows from Theorem 3.2 because χ is $(q/(2m\beta t^2) - (t/m))$-bounded. □

To conclude this section, some remarks are described on the pros and cons of the encrypted controls using multiplicatively and additively homomorphic encryption. The advantage of multiplicatively homomorphic encryption is that both a controller parameter and input can be encrypted. However, all controller computation cannot be outsourced to a controller server in the encrypted control using multiplicatively homomorphic encryption. The aggregation of each row in the matrix-vector product needs to be performed on a client, namely a plant. In contrast to this, all controller computation in the encrypted control using additively homomorphic encryption is conducted on a controller server, although a controller parameter cannot be encrypted. Additionally, the computation costs of using additively homomorphic encryption are often higher than that of using multiplicatively one due to many homomorphic evaluations for plaintext-ciphertext products. For these reasons, a system designer is required to carefully choose which homomorphic encryption scheme to use based on objectives and costs.

3.3 Encrypted control using updatable homomorphic encryption

This section extends encrypted control in the previous section to enhance its security by using updatable homomorphic encryption. The use of updatable homomor-

phic encryption is expected for encrypted control to achieve the forward and post-compromise security. It should be noted that the forward and post-compromised security in encrypted control can be easily satisfied when using additively homomorphic encryption because a controller parameter stored on a controller server is of plaintext. Thus, this section focuses on encrypted control using updatable multiplicatively homomorphic encryption.

Fig. 3.5 illustrates a schematic picture of an encrypted control system using updatable multiplicatively homomorphic encryption. Suppose a ciphertext of the controller parameter is stored on a controller server before the control. The plant transmits the controller input ciphertext ct_ξ and the update token ut generated by the key update algorithm to the controller server while updating public and secret keys. The controller server updates the stored ciphertext using ut and computes the controller output ciphertext ct_ψ with ct_ξ and the updated ciphertext. A control input of the plant is obtained by decrypting ct_ψ with the updated secret key. Note that the update token must be transmitted via a secure communication channel to achieve the forward and post-compromise security against an adversary who eavesdrops on the communication.

Along with the modification to use updatable homomorphic encryption, the correctness notion in Definition 3.4 needs to be suitable for the encrypted control scenario in Fig. 3.5. This thesis requires the encrypted control using updatable homomorphic encryption to ensure that the decrypted and decoded result of encrypted controller output by using a secret key at a certain time step does not almost always deviate so far from the original controller output, where the controller output is computed from a controller input ciphertext encrypted with a public key at the time step and a controller parameter ciphertext updated by an update token corresponding to the public and secret keys. The requirement is defined as δ-correctness with ciphertext-update as follows.

Definition 3.5 (δ-correctness with ciphertext-update)**.** *Given an updatable homomorphic encryption scheme in Definition 2.13 and encoding scheme in Definition 3.1. Let f be as in Definition 3.4. An encrypted controller* EC *of f is δ-correct*

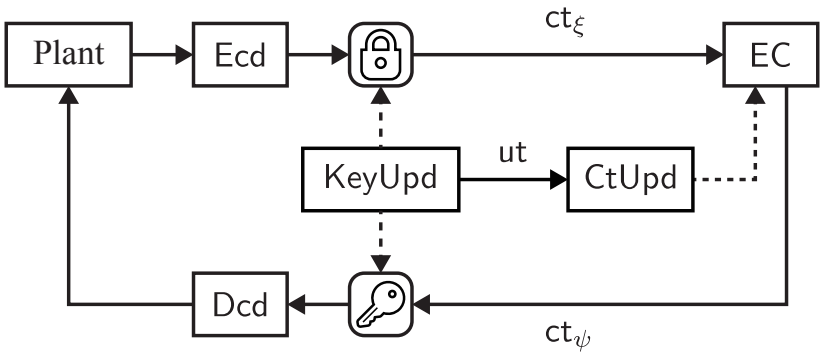


Fig. 3.5: Encrypted control system using updatable holomorphic encryption.

with ciphertext-update if there exist $\delta \in \mathbb{R}^+$ *and a negligible function* negl *such that*

$$\Pr\left[\|\psi' - f(\Phi,\xi)\|_\infty \leq \delta \;\middle|\; \begin{array}{l} (\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \Delta \leftarrow \mathsf{ScalSetup}(\mathsf{params}, \mathcal{X}) \\ (\mathsf{pk}_k, \mathsf{sk}_k, \mathsf{ut}_k) \leftarrow \mathsf{KeyUpd}(\mathsf{pk}_{k-1}, \mathsf{sk}_{k-1}) \\ \mathsf{ct}_{\xi,k} \leftarrow \mathsf{Enc}(\mathsf{pk}_k, \mathsf{Ecd}(\xi;\Delta)) \\ \mathsf{ct}_{\psi,k} \leftarrow \mathsf{EC}(f, \varphi_{\Phi,k}, \mathsf{ct}_{\xi,k}) \\ \psi' \leftarrow \mathsf{Dcd}(\mathsf{Dec}(\mathsf{sk}_k, \mathsf{ct}_{\psi,k});\Delta) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda)$$

for all $\lambda \in \mathbb{N}$*, for all* $k \in \mathbb{Z}^+$*, for all* $\Phi \in \mathcal{X}^{\alpha\times\beta}$*, and for all* $\xi \in \mathcal{X}^\beta$*, where* $\varphi_{\Phi,k}$ *is given as either* $\varphi_{\Phi,k} \leftarrow \mathsf{Ecd}(\Phi,\Delta)$ *or* $\varphi_{\Phi,k} \leftarrow \mathsf{CtUpd}(\varphi_{\Phi,k-1}, \mathsf{ut}_k)$ *with* $\varphi_{\Phi,0} \leftarrow \mathsf{Enc}(\mathsf{pk}_0, \mathsf{Ecd}(\Phi;\Delta))$.

The modified correctness notion is like a combination of the δ-correctness in Definition 3.4 and the homomorphism of updatable homomorphic encryption in Definition 2.15. It can be easily derived from the homomorphism of updatable homomorphic encryption that if an encrypted controller using homomorphic encryption satisfies the δ-correctness, the controller also fulfills the δ-correctness with ciphertext-update.

Theorem 3.3. *Given an updatable multiplicatively homomorphic encryption scheme. Consider the encoding scheme in Theorem 3.1. If* (3.2) *is* δ*-correct,* (3.2) *is also* δ*-correct with ciphertext-update.*

Proof. Let pk_k, sk_k, ut_k, Δ, Φ, ξ, and β be as in Definition 3.5, and let Q be as in Definition 3.2. It follows from the multiplicative homomorphism of the encryption scheme and the proof of Theorem 3.1 that the decoded and decrypted result of

encrypted controller output is given as

$$\psi'_k = \mathsf{Dcd}_{\mathsf{MHE}}(\mathsf{Dec}(\mathsf{sk}_k, \mathsf{ct}_{\psi,k}); \Delta) = \begin{bmatrix} \sum_{j=1}^{\beta} \mathsf{Q}(\Phi_{1j}; \Delta)\mathsf{Q}(\xi_j; \Delta) \\ \vdots \\ \sum_{j=1}^{\beta} \mathsf{Q}(\Phi_{\alpha j}; \Delta)\mathsf{Q}(\xi_j; \Delta) \end{bmatrix}$$

for all $k \in \mathbb{Z}^+$ with probability at least $1 - \mathsf{negl}(\lambda)$, where $\mathsf{ct}_{\psi,k} \leftarrow \mathsf{EC}(f, \mathsf{ct}_{\Phi,k}, \mathsf{ct}_{\xi,k})$, $\mathsf{ct}_{\Phi,k} \leftarrow \mathsf{CtUpd}(\mathsf{ct}_{\Phi,k-1}, \mathsf{ut}_k)$, $\mathsf{ct}_{\Phi,0} \leftarrow \mathsf{Enc}(\mathsf{pk}_0, \mathsf{Ecd}(\Phi; \Delta))$, $\mathsf{ct}_{\xi,k} \leftarrow \mathsf{Enc}(\mathsf{pk}_k, \mathsf{Ecd}(\xi; \Delta))$, and EC and f are defined in Theorem 3.1. There exists $\delta \in \mathbb{R}^+$ such that $\|\psi'_k - f(\Phi, \xi)\|_\infty \leq \delta$ since EC is δ-correct, and thus EC is δ-correct with ciphertext-update. □

The theorem helps design an encrypted control system with updatable homomorphic encryption because it implies that the encrypted controller and updatable homomorphic encryption can be constructed separately. Consequently, an encrypted controller with the updatable ElGamal encryption achieves the same control performance as one with the ElGamal encryption.

Corollary 3.4. *Given the updatable Elgamal encryption scheme in Definition 2.16. Consider the encoding scheme in Theorem 3.1. Let B and $d_{\max}$ be as in Lemma 3.1, and let Φ, ξ, Δ, and β be as in Theorem 3.1. Let p be as in Definition 2.10. If $\Delta^{-2}\|\mathsf{Q}(\Phi; \Delta)\|_{\max}\|\mathsf{Q}(\xi; \Delta)\|_\infty \leq (p-1)/2$, (3.2) is $\beta(\Delta d_{\max} B + (\Delta d_{\max}/2)^2)$-correct with ciphertext-update, where Q is defined in Definition 3.2.*

Proof. The corollary follows from Theorem 2.2, Corollary 3.2, and Theorem 3.3. □

Recall that an update token in updatable homomorphic encryption must be privately transmitted to a controller server against a network eavesdropper. Additionally, the encryption scheme satisfies the forward and post-compromised security against only the eavesdropper. The encrypted control using updatable homomorphic encryption inherits the limitation. In other words, the control does not fulfill the forward and post-compromised security against an honest but curious controller server. To solve this vulnerability, the next section considers encrypted control using key-updatable homomorphic encryption.

3.4 Encrypted control using key-updatable homomorphic encryption

Like with encrypted control using updatable homomorphic encryption, constructing the control using a key-updatable homomorphic encryption scheme is straightforward when the scheme is homomorphic for addition. Hence, this section considers only using key-updatable multiplicatively homomorphic encryption.

Fig. 3.6 depicts a schematic picture of an encrypted control system using key-updatable multiplicatively homomorphic encryption. The plant transmits the controller input ciphertext ct_ξ encrypted with an updated public key at a certain time to a controller server. The controller server returns the controller output ciphertext ct_ψ computed from ct_ξ and the controller parameter ciphertext, which is stored on the server in advance of the control. A control input of the plant is recovered by the decryption of ct_ψ using a joint secret key, including an updated secret key. The correctness notion of the encrypted control called δ-correctness with key-update is formally defined as follows.

Definition 3.6 (δ-correctness with key-update)**.** *Given a key-updatable homomorphic encryption scheme in Definition 2.19 and encoding scheme in Definition 3.1. Let f be as in Definition 3.4. An encrypted controller* EC *of f is δ-correct with key-update if there exist $\delta \in \mathbb{R}^+$ and a negligible function* negl *such that*

$$\Pr\left[\|\psi'_k - f(\Phi,\xi)\|_\infty \le \delta \;\middle|\; \begin{array}{l} (\mathsf{pk}_0,\mathsf{sk}_0) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \Delta \leftarrow \mathsf{ScalSetup}(\mathsf{params},\mathcal{X}) \\ (\mathsf{pk}_k,\mathsf{sk}_k) \leftarrow \mathsf{KeyUpd}(\mathsf{pk}_{k-1},\mathsf{sk}_{k-1}) \\ \mathsf{ct}_{\xi,k} \leftarrow \mathsf{Enc}(\mathsf{pk}_k,\mathsf{Ecd}(\xi;\Delta)) \\ \mathsf{ct}_{\psi,k} \leftarrow \mathsf{EC}(f,\varphi_\Phi,\mathsf{ct}_{\xi,k}) \\ \psi'_k \leftarrow \mathsf{Dcd}(\mathsf{Dec}(\{\mathsf{sk}_i\}_{i\in\{0,k\}},\mathsf{ct}_{\psi,k});\Delta) \end{array}\right] \ge 1-\mathsf{negl}(\lambda)$$

for all $\lambda \in \mathbb{N}$, for all $k \in \mathbb{Z}^+$, for all $\Phi \in \mathcal{X}^{\alpha\times\beta}$, and for all $\xi \in \mathcal{X}^\beta$, where φ_Φ is given as either $\varphi_\Phi \leftarrow \mathsf{Ecd}(\Phi,\Delta)$ or $\varphi_\Phi \leftarrow \mathsf{Enc}(\mathsf{pk}_0,\mathsf{Ecd}(\Phi;\Delta))$.

Without loss of generality, suppose that the initial time step corresponding to the encryption of the controller parameter in the definition is zero. Unlike the encrypted control using updatable homomorphic encryption, the controller parameter ciphertext does not need to be updated. Instead, the decryption algorithm requires

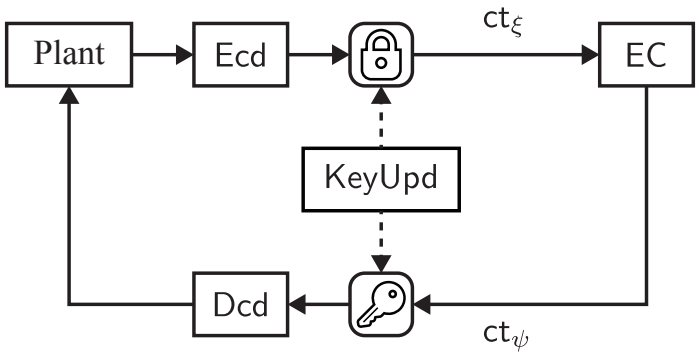


Fig. 3.6: Encrypted control system using key-updatable holomorphic encryption.

the joint secret key consisting of the initial and updated secret keys. It can be easily shown that the encrypted controller (3.2) being δ-correct is δ-correct with key-update.

Theorem 3.4. *Given a key-updatable multiplicatively homomorphic encryption scheme. Consider the encoding scheme in Theorem 3.1. If* (3.2) *is δ-correct, it is also δ-correct with key-update.*

Proof. The proof is the same as that of Theorem 3.3 except that the updatable multiplicatively homomorphic encryption is replaced with the key-updatable one. □

The theorem is a key-update variant of Theorem 3.3. Thus, it holds that the control performance of an encrypted controller with the key-updatable ElGamal encryption is the same as that with the ElGamal encryption.

Corollary 3.5. *Given the key-updatable Elgamal encryption scheme in Definition 2.22. Consider the encoding scheme in Theorem 3.1. Let B and $d_{\max}$ be as in Lemma 3.1, and let Φ, ξ, Δ, and β be as in Theorem 3.1. Let p be as in Definition 2.10. If $\Delta^{-2}\|\mathsf{Q}(\Phi;\Delta)\|_{\max}\|\mathsf{Q}(\xi;\Delta)\|_\infty \le (p-1)/2$,* (3.2) *is $\beta(\Delta d_{\max}B + (\Delta d_{\max}/2)^2)$-correct with key-update, where Q is defined in Definition 3.2.*

Proof. The corollary follows from Theorem 2.7, Corollary 3.2, and Theorem 3.4. □

The corollary implies that the encrypted control is realized using the key-updatable ElGamal encryption. Note that it is also realized using the key-updatable Regev encryption, although a controller parameter cannot be encrypted. Updating key pairs is efficient for the forward and post-compromise security of encrypted control. This security enhancement is meaningful in consideration of the control-theoretic security of encrypted control systems discussed in the next chapter.

Chapter 4

Security of Encrypted Control Systems

Chapter 2 modeled an adversary as a probabilistic polynomial-time algorithm and formulated indistinguishability of encryption schemes via a cryptographic game to prove their computational security. More specifically, an encryption scheme is IND-CPA secure if the probability of every probabilistic polynomial-time adversary winning the IND-CPA game is negligible under some computational assumptions.

Conventional studies on encrypted control implicitly assumed that an encrypted control system is secure if a used homomorphic encryption scheme is IND-CPA secure. However, the validity of this assumption is ambiguous and not verified sufficiently. Indeed, a threat model and security goal to be considered in encrypted control systems are quite different from ones in cryptosystems. This chapter tackles this problem by reconsidering an attack scenario to be supposed in encrypted control systems and builds a security definition tailored for encrypted control systems from scratch.

4.1 Attack scenario

Any formal security definition consists of a threat model, which formulates the capability of an adversary, and a security goal. As previously stated, the threat model and security goal in cryptosystems are far from the ones in encrypted control systems. Their differences are highlighted below upon consideration of defining the security of encrypted control systems.

4.1.1 Threat model

Fig. 4.1 shows an encrypted control system interpreted like a problem setting in public-key encryption, namely a two-party communication under an eavesdropper.

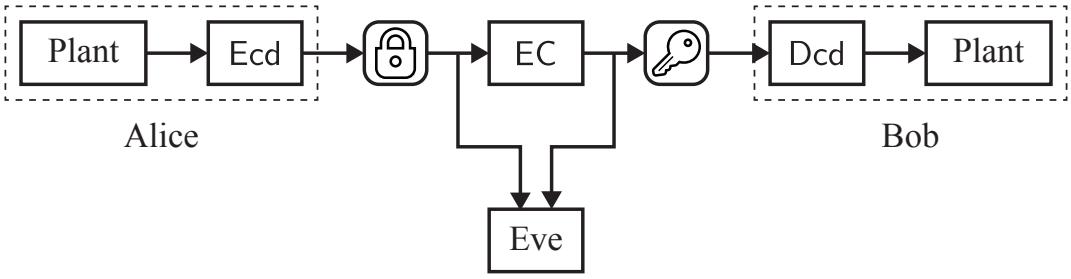


Fig. 4.1: Interpretation of encrypted control system as a two-party communication under the adversary eavesdropping on the communication channel.

The plant and encoder in the figure correspond to Alice, a sender, and the plant and decoder are regarded as Bob, a receiver. Alice transmits an encoded message to the encrypted controller while encrypting it, and Bob receives an evaluation result from the encrypted controller. Eve, an adversary, attempts to learn some information from the encrypted messages transmitted from Alice to the encrypted controller and from the controller to Bob.

The main difference of the communication in Fig. 4.1 from a basic problem setting of public-key encryption in Fig. 2.1 is that a sender and receiver are the same entity because of a feedback loop structure in a control system. Eve eavesdrops on communications of the uplink from Alice to the encrypted controller and the downlink from the controller to Bob. Thanks to the multiple communication channels, Eve can obtain more information for learning private information than the public-key encryption setting. In other words, an adversary to be considered in encrypted control systems is more capable than one in public-key encryption.

4.1.2 Security goal

The indistinguishability notion defined in Definition 2.5 captures the computational impossibility of every adversary learning any partial information about the original message from encrypted messages and public information. This formulation makes sense when private information we wish to protect is a message transmitted via an insecure communication channel or a secret key. However, such computational indistinguishability of encrypted messages is not necessarily suitable for encrypted control systems.

Fig. 4.2 illustrates the difference between security goals to be considered in cryptographic and control-theoretic problem settings. The left time-series data in the figure is an encrypted sequence transmitted in an uplink from a plant to an en-

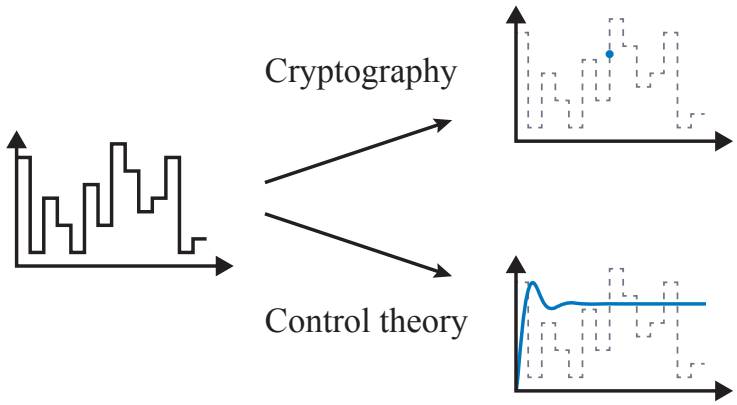


Fig. 4.2: Difference of security goals in cryptography and control theory.

crypted controller or a downlink from the controller to the plant. As the upper right in the figure, the goal of an adversary in communication is typically considered as recovering a single piece of data included in the sequence or obtaining any partial information about the data. In contrast, as the lower right in the figure, an adversary in control systems might aim to recover the original time-series data, i.e., multiple data, of a target control system. In other words, private information in control systems is the behavior of the control system rather than a single data communicated between a plant and controller. This is because eavesdropping in control systems is usually performed as the initial step to obtain a model of a target control system required for executing more sophisticated attacks.

Suppose a typical homomorphic encryption scheme is used to construct an encrypted control system. In that case, the difficulty of deciphering multiple data is identical to a single data because an adversary can decrypt any encrypted message once the adversary succeeds in breaking the encryption scheme. However, recovering multiple data is more difficult than a single data when using updatable or key-updatable homomorphic encryption because of updating key pairs.

Moreover, the aim of an adversary is not necessarily to obtain partial information about private data to be communicated. To obtain a target control system model, the adversary might estimate its parameters using the deciphered time-series data. This parameter estimation process is called system identification in control theory. If the goal of an adversary is system identification, the adversary does not need to recover time-series data completely but only a sufficient amount of data to identify a practical model.

4.1.3 Adversary protocol

The previous sections have seen that a threat model and security goal in encrypted control systems differ from typical communication via an encrypted channel. The gap between encrypted control and encrypted communication suggests the need to explore a novel security definition of control systems. This section begins by formulating an attack scenario in encrypted control systems to establish a formal security definition.

Let $\mathcal{N}(\mu, \Sigma)$ be a Gaussian distribution with mean μ and variance Σ. Consider a plant given by a discrete-time linear-time-invariant system

$$x_{t+1} = Ax_t + Bu_t + w_t, \tag{4.1}$$

where $t \in \mathbb{Z}^+$ is a time, $x \in \mathbb{R}^n$ is a state, $u \in \mathbb{R}^m$ is an input, and $w \in \mathbb{R}^n$ is a noise. Suppose x_0 and w_t are independent and identically distributed over the Gaussian distributions $\mathcal{N}(0, \Sigma_x)$ and $\mathcal{N}(0, \Sigma_w)$ with variances $\Sigma_x \in \mathbb{R}^{n\times n}$ and $\Sigma_w \in \mathbb{R}^{n\times n}$, respectively. $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{m\times n}$ are system parameters assumed to be controllable. Thanks to controllability of (A, B), there exists a state-feedback controller

$$u_t = f(F, x_t) \coloneqq Fx_t \tag{4.2}$$

that stabilizes (4.1), where $F \in \mathbb{R}^{m\times n}$ is a feedback gain to be designed.

A closed-loop system combined with (4.1) and an encrypted controller of (4.2) using multiplicatively homomorphic encryption is given as

$$x_{t+1} = Ax_t + B\mathsf{Dcd}_{\mathsf{MHE}}(\mathsf{Dec}(\mathsf{sk}, \mathsf{EC}(f, \mathsf{ct}_F, \mathsf{ct}_{x,t})); \Delta) + w_t,$$

where EC and $\mathsf{Dcd}_{\mathsf{MHE}}$ are definined in Theorem 3.1, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$, $\Delta \leftarrow \mathsf{ScalSetup}(\mathsf{params}, \mathcal{X})$, $\mathsf{m}_F \leftarrow \mathsf{Ecd}(F; \Delta)$, $\mathsf{m}_{x,t} \leftarrow \mathsf{Ecd}(x_t; \Delta)$, $\mathsf{ct}_F \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_F)$, and $\mathsf{ct}_{x,t} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_{x,t})$. Similarly, a closed-loop system with additively homomorphic encryption is given as

$$x_{t+1} = Ax_t + B\mathsf{Dcd}_{\mathsf{AHE}}(\mathsf{Dec}(\mathsf{sk}, \mathsf{EC}(f, \mathsf{m}_F, \mathsf{ct}_{x,t})); \Delta) + w_t,$$

where EC and $\mathsf{Dcd}_{\mathsf{AHE}}$ are definined in Theorem 3.2. An encrypted state-feedback control system with updatable or key-updatable homomorphic encryption can also be realized similarly. Assume that quantization errors induced by the encoder and

decoder are negligibly small. That is, for every $F \in \mathcal{X}^{m\times n}$ and $x \in \mathcal{X}^n$, a scaling parameter Δ is assumed to be chosen such that EC is δ-correct for some $\delta \ll 1$. Under this assumption, the close-loop systems can be approximated to

$$x_{t+1} = \bar{A}x_t + w_t, \tag{4.3}$$

where $\bar{A} \coloneqq A + BF$.

A system that can be a target of adversaries is a plant, controller, or closed-loop system. Potential adversaries for encrypted control systems in this setting are twofold.

Eavesdropper on a communication channel. The eavesdropper in encrypted control systems is a potential threat naively inherited from a secure communication problem using an encryption scheme. The adversary intercepts communication channels between a plant and controller server that performs an encrypted control algorithm.

Honest-but-curious controller server This type of adversary is considered when a controller server to which controller computation is outsourced is untrusted. For example, a public cloud owned by a third party can be modeled as an honest but curious server. The controller server should not perform a destructive attack directly because of a contract between a user who demands to control a plant and an owner of the controller server. Instead, the controller server might wish to collect and learn private information about a control system to utilize the information for commercial use, such as reverse engineering and advertising.

The adversaries can be treated as a unified model shown in Fig. 4.3. Eve, the adversary in the figure, collects encrypted data transmitted between the plant and the encrypted controller. The adversary then deciphers the collected data and estimates the parameters of the plant, controller, or closed-loop system. Eve represents an eavesdropper itself and a subroutine of the controller server when considering an adversary eavesdropping on the communication channels and an honest-but-curious server, respectively. The adversary considered here can access the inputs/outputs data of an encrypted controller and a public key (or multiple public keys if key pairs are updated), and the adversary's goal is to estimate the parameters of the plant or closed-loop system.

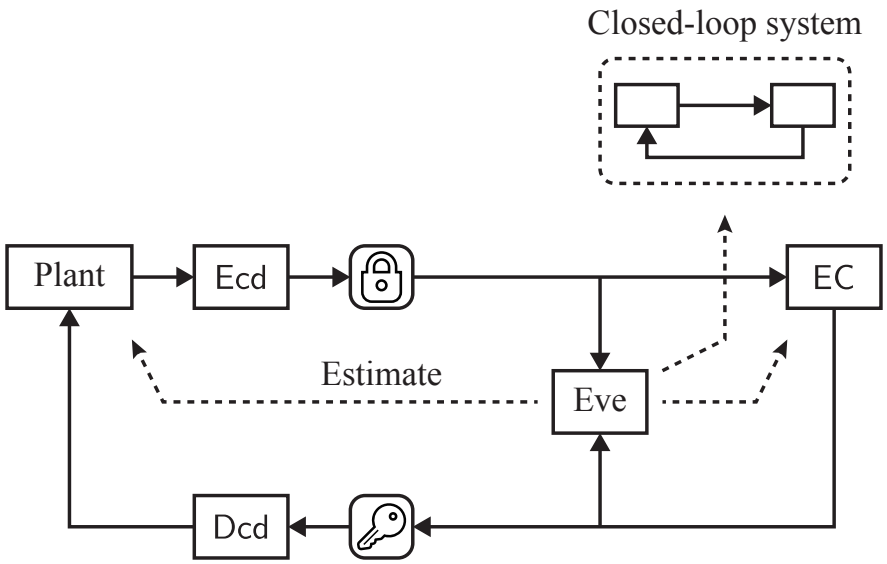


Fig. 4.3: Encrypted control under the adversary attempting to estimate parameters of the plant, controller, or closed-loop system.

It should be noted that the adversaries do not have to perform an identification process when attempting to estimate controller parameters. This is because ciphertexts of controller parameters are stored on a controller server in advance before control and not changed during the control. The adversaries are enough to decipher the ciphertexts directly as long as they can access them. The security in such a case might be equivalent to the IND-CPA security. On the other hand, they need to perform system identification to estimate plant or closed-loop system parameters because they depend on physical processes.

This thesis focuses on the scenario when the adversary in Fig. 4.3 identifies either (A, B) of the plant (4.1) or $\bar{A}$ of the closed-loop system (4.3) by recovering and utilizing time-series data from an encrypted data sequence. The adversary protocol is defined below to formulate the attack scenario.

Definition 4.1 (Adversary protocol)**.** *Define* $z_t \in \mathbb{R}^d$ *and* $\Theta \in \mathbb{R}^{n\times d}$ *as*

$$(z_t, \Theta) := \begin{cases} \left(\begin{bmatrix} x_t \\ u_t \end{bmatrix}, \begin{bmatrix} A & B \end{bmatrix} \right), & \textit{Attack target is } (4.1), \\ \left(x_t, \bar{A} \right), & \textit{Attack target is } (4.3), \end{cases}$$

where d *is an appropriate dimension. Let* t_s *and* t_f *be positive integers such that* $t_s < t_f < \infty$*. The adversary follows the protocol below.*

1. *If an attack target is* (4.1)*, the adversary injects malicious inputs* u_t *for* $t \in [t_s, t_f]$*, where* u_t *are independent and identically distributed over the Gaussian distribution* $\mathcal{N}(0, \Sigma_u)$ *with variance* $\Sigma_u \in \mathbb{R}^{m \times m}$*.*

2. *The adversary eavesdrops and stores the ciphertext dataset*

$$\mathcal{D}_{\mathsf{Enc}} \coloneqq \{\mathsf{ct}_{z,t} \mid t \in [t_s, t_f]\},$$

   where $\mathsf{ct}_{z,t} \leftarrow \mathsf{Enc}(\mathsf{pk}_t, \mathsf{Ecd}(z_t; \Delta))$*,* $\Delta \leftarrow \mathsf{ScalSetup}(\mathcal{X})$*,* $(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{KeyGen}(1^\lambda)$*, and* $(\mathsf{pk}_t, \mathsf{sk}_t) \leftarrow \mathsf{KeyUpd}(\mathsf{pk}_{t-1}, \mathsf{sk}_{t-1})$*.*

3. *The adversary deciphers* $\mathcal{D}_{\mathsf{Enc}}$ *to obtain the original dataset*

$$\mathcal{D} \coloneqq \{z_t \mid t \in [t_s, t_f]\}.$$

4. *The adversary estimates* Θ *using* $\mathcal{D}$*.*

An estimation error of the attack in Definition 4.1 is defined as follows.

Definition 4.2 (Estimation error)**.** *An estimation error of an estimator* $\hat{\Theta}$ *for* Θ *given* $\mathcal{D}$ *is defined as*

$$\varepsilon\Big(\Theta, \hat{\Theta}(\mathcal{D})\Big) \coloneqq \frac{1}{nd}\Big\|\Theta - \hat{\Theta}(\mathcal{D})\Big\|_F^2,$$

where Θ *and* $\mathcal{D}$ *are defined in Definition 4.1, and* $\|\cdot\|_F$ *is the Frobenius norm.*

Note that the estimation error is a mean square error by definition of the Frobenius norm.

4.2 Qualitative vs. quantitative security

There are two approaches to define the security of encrypted control systems under the adversary in Definition 4.1. One approach is to define security in a qualitative manner, such as the IND-CPA security. Such qualitative security is defined against a class of adversaries, e.g., polynomial-time adversaries. Additionally, an attack in the security is modeled by using the capabilities of the adversaries instead of considering specific attack methods, such as a brute force attack and side-channel attack. The qualitative security definition is effective in guaranteeing the security

of broad encrypted control systems even against unknown attacks as well as various attacks covered in an attack model. However, such a security definition cannot state how secure an encrypted control system is. In other words, we cannot compare how "good" one of the two given systems is against the other under a qualitative security definition.

Another approach to defining the security of encrypted control systems is to quantify the security level of the systems. In this approach, the security is formulated for an adversary having a specific computational power and performing a particular attack algorithm. This type of security definition is valid for evaluating the security strength of encrypted control systems and enables a comparison of multiple systems in terms of security. Hence, such a quantitative security definition can quantify how good any system is in specific attack scenarios.

One of the goals of control theory is to design a controller so that a control system satisfies the required control specifications. It is necessary to quantify the security level to regard security as one of the control specifications and treat it as a control-theoretic problem of designing an encrypted control system that satisfies the desired security strength. For this reason, this thesis adopts the quantitative approach to define the security of encrypted control systems. The security definition below is now considered to quantify the security level of encrypted control systems under the adversary in Definition 4.1.

Definition 4.3 (Informal)**.** *An encrypted control system is secure if an adversary cannot estimate plant (or closed-loop system) parameters of high precision during a certain given period by using an estimation algorithm.*

The crucial quantities in the definition are the accuracy of the adversary's estimation and the given period. According to the security definition, an encrypted control system might be secure if the accuracy is sufficiently low or if the computation time required for the estimation is longer than the given period, even though its accuracy is high. In other words, the combination of a threshold of estimation error to be accepted by a system designer and a certain period, such as the lifespan of a control system, can be used as a security level of an encrypted control system. The remaining task for formalizing the security definition is to quantify the estimation accuracy and computation time required for the estimation. In what follows, two notions, sample identifying complexity and sample deciphering time, are introduced to address the task.

4.3 Sample identifying complexity

The accuracy of the adversary's estimation can be quantified by the magnitude of the estimation error on a given sample size, namely the error rate. Sample complexity was introduced in computational learning theory to treat an error rate of learning algorithms. A sample complexity evaluates the minimum amount of data required for learning a function of a given precision. More precisely, for $\epsilon > 0$ and $\delta > 0$, it represents the minimum sample size so that

$$\Pr[\varepsilon < \epsilon] \geq 1 - \delta,$$

where ε is an estimation error. Hence, an estimation error with a sample size larger than the sample complexity is almost always smaller than ϵ when choosing a sufficiently small δ.

An estimation error of the adversary's estimation can be quantified by using a sample complexity with a sufficiently small δ. Given a sample complexity, a larger ϵ means stronger security from the viewpoint of our attack scenario. Unfortunately, a sample complexity is often conservative when δ is sufficiently small because of considering extremely rare cases of estimation. This means that the traditional sample complexity is not suitable for our objective in practice.

To overcome the conservativeness, this section introduces a modified sample complexity corresponding to the expectation of estimation error, called *sample identifying complexity*.

Definition 4.4 (Sample identifying complexity)**.** *Let Θ and $\mathcal{D}$ be as in Definition 4.1. A sample identifying complexity of an estimator $\hat{\Theta}$ for Θ given $\mathcal{D}$ with respect to γ_c is the minimum sample size $N^* = |\mathcal{D}|$ such that*

$$\mathbb{E}\Big[\varepsilon\Big(\Theta, \hat{\Theta}(\mathcal{D})\Big)\Big] < \gamma_c,$$

where ε is the estimation error in Definition 4.2.

A sample identifying complexity of $\hat{\Theta}$ is a function of γ_c and Θ that represents the minimum sample size such that the expectation of estimation error becomes smaller than γ_c, where $\gamma_c > 0$ is a constant corresponding to ϵ in the traditional sample complexity. A sample complexity quantifies some type of average error rate of the adversary's estimation, unlike the traditional sample complexity. Thus, a sample

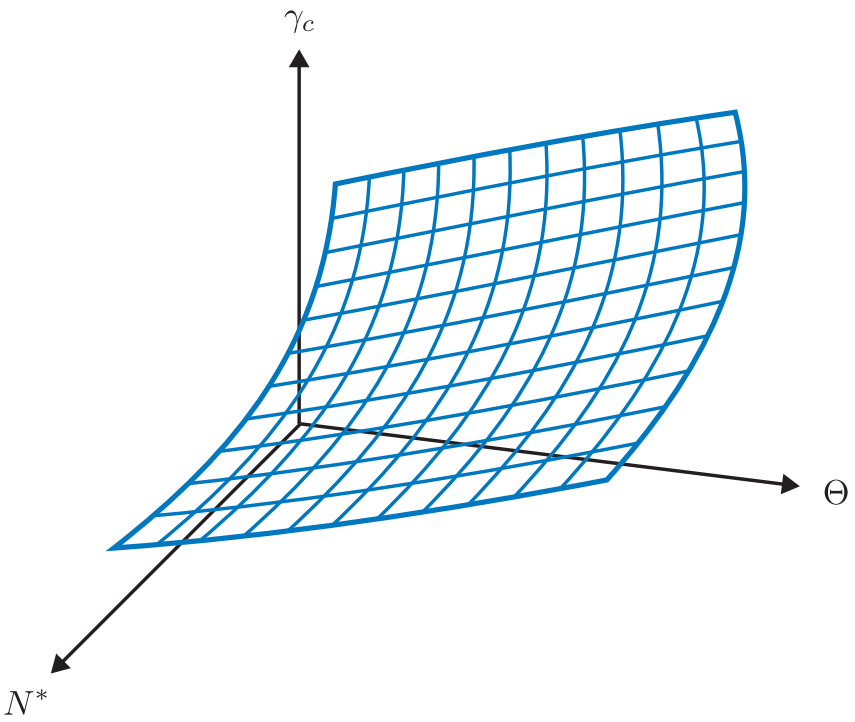


Fig. 4.4: Relationship among a sample identifying complexity, a threshold, and a system parameter.

complexity is expected to be less conservative than the traditional one because it does not consider extremely rare cases of estimation.

Fig. 4.4 illustrates a relationship among a sample identifying complexity N^*, threshold γ_c, and system parameter Θ under an estimator $\hat{\Theta}$. A sample identifying complexity increases as a threshold decreases for a certain system parameter because more samples are required for a smaller estimation error. In the figure, larger Θ implies the one that is harder to estimate, and so a threshold can be increased as a system parameter increases for a fixed sample identifying complexity. Similarly, a sample identifying complexity can be larger by increasing a system parameter for a fixed threshold.

4.4 Sample deciphering time

This section attempts to quantify the computation time required for the adversary's estimation. The computation time mainly consists of two parts: the computation times of an estimation algorithm and an algorithm that deciphers ciphertexts. In general, the computation time of an estimation algorithm is from a few seconds to

a few days at most. By contrast, the computation time for breaking a cryptosystem is usually years to decades. Therefore, in what follows, this thesis regards the computation time of the adversary's estimation as that of an algorithm breaking a cryptosystem.

Chapter 2 viewed the security of an encryption scheme in an asymptotic manner. The probability that an adversary succeeds in breaking a secure encryption scheme is negligibly small when choosing a sufficiently large security parameter. The security notions in the chapter make sense only when a security parameter is sufficiently large. Thus, the notions give us no implication about a computation time with a specific security parameter. This fact implies that another approach is needed to quantify a computation time explicitly.

To this end, this section introduces a notion called bit security. Cryptosystems relying on different computational problems, such as the DDH and LWE problems, provide different security strengths. Bit security, defined below, is usually used for evaluating the security level of cryptosystems regardless of the type of computational problem.

Definition 4.5 (Bit security)**.** *An encryption scheme is λ-bit secure if at least 2^λ operations are required for breaking the scheme on average.*

It should be noted that a security parameter in the context of bit security represents the number of bits, namely the security level itself. Using bit security, a computation time required for breaking a λ-bit secure encryption scheme by using a computer of Υ floating point number operations per second (FLOPS) can be estimated as $2^\lambda/\Upsilon$ s, where one operation in an algorithm breaking a cryptosystem is assumed to be performed by one floating-point operation. In this light, this section defines *sample deciphering time* to quantify the computation time of the adversary's estimation using N samples.

Definition 4.6 (Sample deciphering time)**.** *A sample deciphering time is a computation time τ required for breaking N ciphertexts encrypted by using N_c key pairs of a λ-bit secure encryption scheme, i.e.,*

$$\tau(N_c, \lambda; \Upsilon) \coloneqq \frac{2^\lambda N_c}{\Upsilon}, \tag{4.4}$$

where $N, N_c \in \mathbb{N}$, $N_c \leq N$, and an adversary is supposed to use a computer of Υ FLOPS.

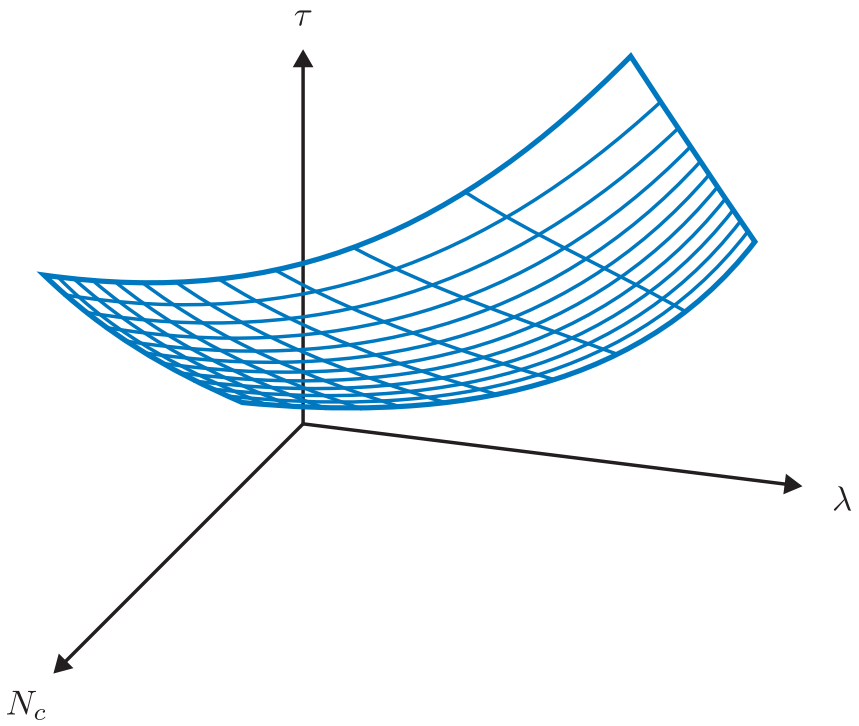


Fig. 4.5: Relationship among a sample deciphering time, a security parameter, and the number of executions of a key update algorithm.

A sample deciphering time represents the computation time for deciphering N samples of ciphertexts with N_c key pairs. Recall that such computation time does not depend on the number of key pairs when using a typical homomorphic encryption scheme because an identical key pair is used for encrypting all messages. Hence, the computation time can be estimated similarly for breaking a cryptosystem as already described, namely $\tau(1, \lambda; \Upsilon)$. By contrast, key pairs are updated when using an updatable or key-updatable homomorphic encryption scheme. A computation time in such a case should be proportional to the number of key pairs used in encryption. A relationship among a sample deciphering time τ, the number of key pairs N_c, and a security parameter λ under a fixed computer performance Υ is illustrated in Fig. 4.5.

4.5 Security definition

We are now ready to define the security of encrypted control systems. Our security definition is formally described below by using a sample identifying complexity and a sample deciphering time.

Definition 4.7 (Security of encrypted control system)**.** *Let Θ and $\mathcal{D}$ be as in Definition 4.1, and let $\hat{\Theta}$ be an estimator of Θ given $\mathcal{D}$. Suppose every sample in $\mathcal{D}$ is encrypted by using one of the N_c key pairs, where $N_c \in \mathbb{N}$, and $N_c \leq |\mathcal{D}|$. For $1 \leq i \leq N_c$, define $\mathcal{D}_i$ as the set of all samples encrypted by using the ith key pair, namely $|\mathcal{D}| = \sum_{i=1}^{N_c} |\mathcal{D}_i|$. An encrypted control system is (γ_c, τ_c)-secure if it holds that*

$$\sum_{i \in \mathcal{I}} |\mathcal{D}_i| < N^*\left(\gamma_c, \Theta; \hat{\Theta}\right) \quad \vee \quad \tau(|\mathcal{I}|, \lambda; \Upsilon) > \tau_c$$

for all $\mathcal{I} \subset \{1, \ldots, N_c\}$ except the empty set, where γ_c and N^* are defined in Definition 4.4, and τ_c, τ, λ, and Υ are defined in Definition 4.6. Otherwise, an encrypted control system is insecure on (γ_c, τ_c).

The first part of the condition implies that the number of ciphertext samples encrypted by using $|\mathcal{I}|$ key pairs is less than a sample identifying complexity. By Definition 4.4, a sample identifying complexity is the minimum sample size such that the expectation of estimation error is smaller than a given threshold γ_c. Hence, the condition is equivalent to

$$\mathbb{E}\left[\varepsilon\left(\Theta, \hat{\Theta}\left(\bigcup_{i \in \mathcal{I}} \mathcal{D}_i\right)\right)\right] \geq \gamma_c.$$

It means that the expectation of estimation error given $\bigcup_{i \in \mathcal{I}} \mathcal{D}_i$ is larger than or equal to γ_c. Furthermore, the second part of the condition implies that the computation time required for deciphering $|\bigcup_{i \in \mathcal{I}} \mathcal{D}_i|$ ciphertext samples is longer than a given period τ_c. Recall that this thesis has assumed that the computation time of the adversary's estimation is equal to that of deciphering ciphertexts to be used for the estimation. Consequently, the conditions imply that, for every dataset available to an adversary, the expectation of estimation error for the adversary's estimation is larger than γ_c or the computation time taken to perform the estimation is longer than τ_c.

By considering contraposition of the condition in Definition 4.7, it is equivalent to that there does not exist $\mathcal{I} \subset \{1, \ldots, N_c\}$ such that

$$\mathbb{E}\left[\varepsilon\left(\Theta, \hat{\Theta}\left(\bigcup_{i \in \mathcal{I}} \mathcal{D}_i\right)\right)\right] < \gamma_c \quad \wedge \quad \tau(|\mathcal{I}|, \lambda; \Upsilon) \leq \tau_c.$$

The equivalent condition implies that an adversary cannot estimate Θ using $\hat{\Theta}$ so

that $\mathbb{E}[\varepsilon]$ is less than γ_c within τ_c. In this context, the constants γ_c and τ_c represent an estimation error to be accepted by a system designer and a period that the designer wants to protect an encrypted control system, respectively. In this light, γ_c and τ_c are referred to as *accepted estimation error* and *defense period*.

Consider two circumstances for $N_c = 1$ and $N_c = |\mathcal{D}|$ to illustrate the security of encrypted control systems. When $N_c = 1$, a key pair is not updated throughout the adversary's estimation, corresponding to traditional homomorphic encryption. In this case, the possible subset $\mathcal{I}$ is only $\{1\}$, and thus it holds that $\sum_{i\in\mathcal{I}} |\mathcal{D}_i| = |\mathcal{D}_1| = |\mathcal{D}|$ and $|\mathcal{I}| = 1$. Hence, for every $\gamma_c > 0$, the condition to make an encrypted control system (γ_c, τ_c)-secure is $\tau(1, \lambda; \Upsilon) \geq \tau_c$ as long as an adversary collects $|\mathcal{D}|$ samples such that $|\mathcal{D}| \geq N^*(\gamma_c, \Theta; \hat{\Theta})$. Note that such a condition is the same as a security requirement in a typical cryptographic setup.

When $N_c = |\mathcal{D}|$, the number of used key pairs is equal to the sample size, which means that a key update algorithm is performed after every encryption. In such a case, $|\mathcal{D}_i| = 1$ holds for all $1 \leq i \leq N_c$, implying $\sum_{i\in\mathcal{I}} |\mathcal{D}_i| = |\mathcal{I}|$. Therefore, the condition of an encrypted control system to be (γ_c, τ_c)-secure is that

$$|\mathcal{I}| < N^*\Big(\gamma_c, \Theta; \hat{\Theta}\Big) \quad \vee \quad \tau(|\mathcal{I}|, \lambda; \Upsilon) > \tau_c.$$

holds for all $1 \leq |\mathcal{I}| \leq |\mathcal{D}|$. Without loss of generality, a sample size $|\mathcal{D}|$ is now regarded as the number of key pairs $|\mathcal{I}|$ because an adversary can choose the sample size freely. The condition is then equivalent to that there does not exist a dataset $\mathcal{D}$ such that

$$\mathbb{E}\Big[\varepsilon\Big(\Theta, \hat{\Theta}(\mathcal{D})\Big)\Big] < \gamma_c \quad \wedge \quad \tau(|\mathcal{D}|, \lambda; \Upsilon) \leq \tau_c.$$

Fig. 4.6 depicts sample identifying complexities N_1^* and N_2^* of an estimator $\hat{\Theta}$ for two system parameters Θ_1 and Θ_2 with respect to a certain acceptable estimation error γ_c. The two curves show the expectation of estimation error $\mathbb{E}[\varepsilon]$ against a sample size $|\mathcal{D}|$ in each system parameter. It should be noted that, for a fixed system parameter, $\mathbb{E}[\varepsilon]$ is a function of $|\mathcal{D}|$ because of performing the expectation on a dataset $\mathcal{D}$. By definition, the sample identifying complexities are determined as the minimum sample size such that the expectation of estimation error becomes smaller than the acceptable estimation error. Furthermore, Fig. 4.7 shows a sample deciphering time τ for a security parameter λ and computer performance Υ when $N_c = |\mathcal{D}|$. With the defense period τ_c, it can be shown that $\tau(N_1^*, \lambda; \Upsilon) < \tau_c < \tau(N_2^*, \lambda; \Upsilon)$. From these figures, an encrypted control system with Θ_1 is insecure on (γ_c, τ_c) because

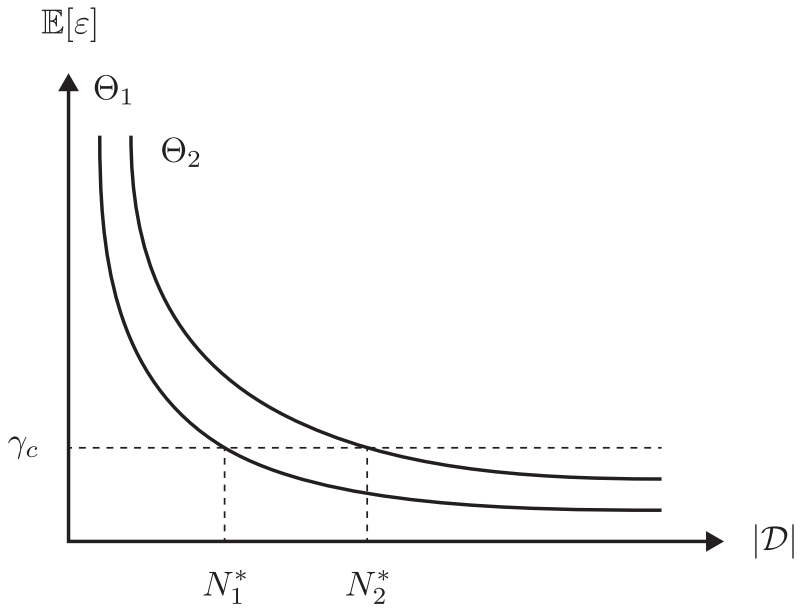


Fig. 4.6: Sample identifying complexities of an estimator for the two system parameters with respect to the acceptable estimation error.

there exists a dataset $\mathcal{D}$ such that $\mathbb{E}[\varepsilon(\Theta_1, \hat{\Theta}; \mathcal{D})] < \gamma_c$ and $\tau(|\mathcal{D}|, \lambda; \Upsilon) \leq \tau_c$, which is illustrated as the blue area in Fig. 4.7. Meanwhile, an encrypted control system with Θ_2 is (γ_c, τ_c)-secure because such $\mathcal{D}$ does not exist.

The blue area in Fig. 4.7 shrinks as N_1^* gets closer to N_2^* and vanishes when a sample deciphering time with N_1^* becomes larger than τ_c. This observation suggests that the security of encrypted control systems can be determined only by whether a sample deciphering time on a sample identifying complexity is larger than a defense period. Such a criterion works when the number of key pairs is equal to a sample size, as shown in the theorem below.

Theorem 4.1. *Consider the notations in Definition 4.7. Suppose $N_c = |\mathcal{D}|$. An encrypted control system is (γ_c, τ_c)-secure if and only if $\tau(N^*(\gamma_c, \Theta; \hat{\Theta}), \lambda; \Upsilon) > \tau_c$.*

Proof. If part: As already described, the security condition in Definition 4.7 is equivalent to that there does not exist $\mathcal{D}$ such that $\mathbb{E}[\varepsilon(\Theta, \hat{\Theta}(\mathcal{D}))] < \gamma_c$ and $\tau(|\mathcal{D}|, \lambda; \Upsilon) \leq \tau_c$. For every positive integer $N < N^*(\gamma_c, \Theta; \hat{\Theta})$, it holds that $\mathbb{E}[\varepsilon(\Theta, \hat{\Theta}(\mathcal{D}))] \geq \gamma_c$ for all dataset $\mathcal{D}$ of sample size N by the definition of a sample identifying complexity. Additionally, for every positive integer $N \geq N^*(\gamma_c, \Theta; \hat{\Theta})$, $\tau(N, \lambda; \Upsilon) > \tau_c$ holds from the assumption. Therefore, it holds that $\mathbb{E}[\varepsilon(\Theta, \hat{\Theta}(\mathcal{D}))] \geq \gamma_c$ or $\tau(|\mathcal{D}|, \lambda; \Upsilon) > \tau_c$ for all $\mathcal{D}$.

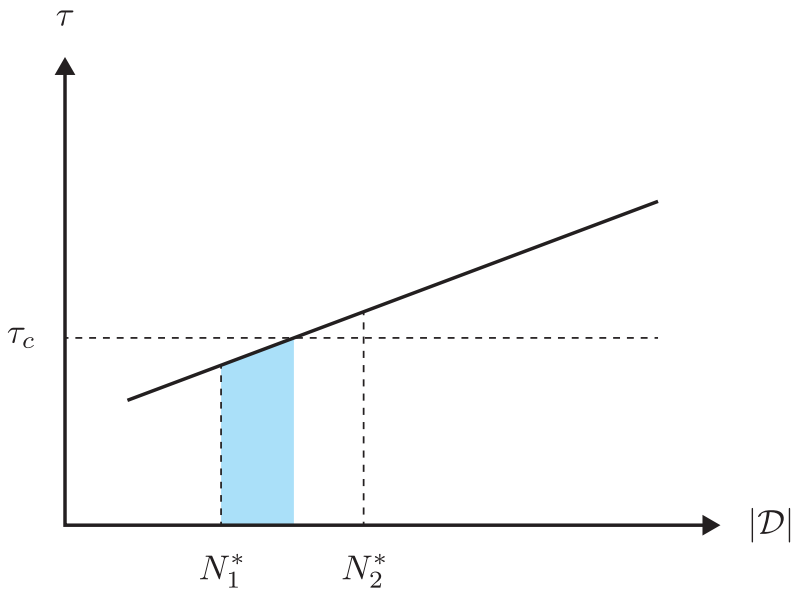


Fig. 4.7: Sample deciphering time for a certain security parameter and the defense period.

Only if part: We prove the contraposition, i.e., an encrypted control system is insecure on (γ_c, τ_c) if $\tau(N^*(\gamma_c, \Theta; \hat{\Theta}), \lambda; \Upsilon) \leq \tau_c$. It is obvious that a dataset of sample size N^* saisfies $\mathbb{E}[\varepsilon(\Theta, \hat{\Theta}(\mathcal{D}))] < \gamma_c$ and $\tau(N^*, \lambda; \Upsilon) \leq \tau_c$. □

This chapter has explored the security of encrypted control systems. It has been shown that a threat model and security goal considered in encrypted control differ from traditional private communication using a cryptosystem. A sample identifying complexity and a sample deciphering time have been introduced to define the quantitative security of encrypted control systems under an adversary who attempts to estimate a controlled or closed-loop system parameter. The security level of encrypted control systems has been quantified by an acceptable estimation error and a defense period by using a sample identifying complexity and sample deciphering time.

The security definition provides a way to determine whether an encrypted control system satisfies a desired security level. It should be noted that the security level does not make sense only with either an acceptable estimation error or a defense period. We cannot compare the security levels of (γ_c, τ_c)-secure and (γ_c', τ_c')-secure encrypted control systems when $\gamma_c > \gamma_c'$ and $\tau_c < \tau_c'$. To compare the systems, either an acceptable estimation error or defense period must be aligned with the same value.

Chapter 5

Design of Encrypted Control Systems

Chapter 4 has so far considered determining whether an encrypted control system is secure or insecure under a given security level based on a sample identifying complexity and sample deciphering time. On the other hand, how do we design encrypted control systems to achieve a given security level? It is obvious that any desired security level can be satisfied by choosing a significantly large security parameter. However, such a security parameter often increases the computational burden due to encryption, thereby making real-time computation difficult. Real-time computation is essential for guaranteeing the stability and performance of control systems. Meanwhile, a small security parameter may only satisfy an inadequate security level, although a control system can perform in real-time. This dilemma motivates us to design the minimum security parameter required for achieving a desired security level.

This chapter aims to solve the following problem for designing an encrypted control system that satisfies a desired security level while operating in real-time.

Problem 5.1. *Consider an encrypted control system that consists of* (4.1) *and an encrypted controller of* (4.2) *with an updatable or key-updatable homomorphic encryption scheme. Given an acceptable estimation error* γ_c *and defense period* τ_c*. Design the minimum security parameter* λ^* *such that the encrypted control system is* (γ_c, τ_c)*-secure under the adversary in Definition 4.1.*

With typical homomorphic encryption, a security parameter making an encrypted control system secure can be determined from a sample deciphering time (4.4) and Definition 4.7 as

$$\lambda > \log_2(\tau_c \Upsilon),$$

where Υ is a computer performance used by an adversary. In contrast, if updatable or key-updatable homomorphic encryption is used, such a security parameter

depends on the number of key pairs for encrypting a dataset. From (4.4), a security parameter is reduced by increasing the number of key pairs for a certain defense period. Hence, in what follows, this thesis assumes that the number of key pairs is chosen to be maximized.

Assumption 5.1. *A key update algorithm is performed every time step, namely* $N_c = |\mathcal{D}|$.

Under this assumption, Theorem 4.1 showed that an encrypted control system is (γ_c, τ_c)-secure if and only if $\tau(N^*(\gamma_c, \Theta; \hat{\Theta}), \lambda; \Upsilon) > \tau_c$ holds. A sample identifying complexity N^* was defined in Definition 4.4 using the expectation of estimation error. Additionally, the estimation error was defined in Definition 4.2 as a type of mean square error between a system parameter and its estimate. Thus, an estimator used by an adversary needs to be fixed to compute a sample identifying complexity, which is essential to design the minimum security parameter. To this end, the next section will discuss representative estimators for a system parameter in our attack scenario.

5.1 Parameter estimation algorithms

This section introduces four representative parameter estimation algorithms as candidates for estimators used by an adversary. Furthermore, the section shows that the parameter estimation algorithms can be unified under reasonable conditions in our attack scenario.

In what follows, three assumptions are made to simplify the derivation of the parameter estimation algorithms.

Assumption 5.2. *For the attack in Definition 4.1, the following assumptions are made.*

- $\bar{A}$ *is Schur.*
- A *is Schur if an attack target is* (4.1).
- x_{t_s} *follows the Gaussian distribution* $\mathcal{N}(0, \Sigma_w)$.

The first assumption is about the stability of closed-loop system (4.3). This thesis assumes that (4.3) is stable because a system designer should design a control

system to be stable with and without attack. The second assumption is that (4.1) is stable if it is an attack target. This is a usual assumption in open-loop system identification. The third assumption is about the probability distribution of the initial data. An adversary generally needs to specify a probability distribution of the initial data for parameter estimation. However, its computation is impossible because the adversary cannot obtain all of the past data and determine the initial time step. In such a case, the assumption is reasonable because the control system is stabilized as the first assumption and driven by only the noise w of which probability distribution is $\mathcal{N}(0, \Sigma_w)$.

5.1.1 Ordinary least squares estimation

This section begins by introducing the ordinary least squares (OLS) estimator. The OLS estimator is the most popular estimator in practice because of its simplicity and efficiency.

It follows from (4.1) and (4.3) that

$$X_f = \Theta Z_p + W_p, \quad Z_p \coloneqq \begin{cases} \begin{bmatrix} X_p \\ U_p \end{bmatrix}, & \Theta = \begin{bmatrix} A & B \end{bmatrix}, \\ X_p, & \Theta = \bar{A}, \end{cases} \tag{5.1}$$

where

$$\begin{aligned} X_f &\coloneqq \begin{bmatrix} x_{t_s+1} & \cdots & x_{t_f} \end{bmatrix} \in \mathbb{R}^{n\times(|\mathcal{D}|-1)}, \; X_p \coloneqq \begin{bmatrix} x_{t_s} & \cdots & x_{t_f-1} \end{bmatrix} \in \mathbb{R}^{n\times(|\mathcal{D}|-1)}, \\ U_p &\coloneqq \begin{bmatrix} u_{t_s} & \cdots & u_{t_f-1} \end{bmatrix} \in \mathbb{R}^{m\times(|\mathcal{D}|-1)}, \; W_p \coloneqq \begin{bmatrix} w_{t_s} & \cdots & w_{t_f-1} \end{bmatrix} \in \mathbb{R}^{n\times(|\mathcal{D}|-1)}, \end{aligned} \tag{5.2}$$

and $\mathcal{D}$, t_s, and t_f are defined in Definition 4.1. Let $\hat{\Theta}$ be an estimator for Θ given $\mathcal{D}$. Using the estimator, a prediction of X_f is obtained as

$$\hat{X}_f = \hat{\Theta}(\mathcal{D}) Z_p.$$

We now regard a good estimator as an estimator that gives an accurate prediction. In this light, our objective is to find an estimator minimizing a residual sum of squares $\|X_f - \hat{X}_f\|_F^2$, and the OLS estimator is defined as its solution.

Definition 5.1 (OLS estimator)**.** *Let Θ and $\mathcal{D}$ be as in Definition 4.1. The ordinary*

least squares (OLS) estimator $\hat{\Theta}_{\mathrm{OLS}}$ *for* Θ *given* $\mathcal{D}$ *is defined as*

$$\hat{\Theta}_{\mathrm{OLS}}(\mathcal{D}) \coloneqq \arg\min_{\Theta} \|X_f - \Theta Z_p\|_F^2,$$

where Z_p *and* X_f *are defined in* (5.1) *and* (5.2)*, respectively.*

In our attack scenario, the OLS estimator is given as a solution to an equation consisting of data matrices X_f and Z_p.

Theorem 5.1. *The OLS estimator in Definition 5.1 is given as a solution* Θ *to the equation,*

$$\Theta Z_p Z_p^\top = X_f Z_p^\top,$$

where Z_p *and* X_f *are defined in* (5.1) *and* (5.2)*, respectively.*

Proof. The cost $\|X_f - \Theta Z_p\|_F^2$ is minimized by a parameter Θ that satisfies

$$\frac{\partial}{\partial \Theta}\|X_f - \Theta Z_p\|_F^2 = 2(X_f - \Theta Z_p)Z_p^\top = O.$$

This yields the equation in the theorem. □

Furthermore, the OLS estimator is explicitly obtained if Z_p is full row rank.

Corollary 5.1. *Let* $\mathcal{D}$ *be as in Definition 4.1. If* Z_p *in* (5.1) *is full row rank, the OLS estimator in Definition 5.1 is given as*

$$\hat{\Theta}_{\mathrm{OLS}}(\mathcal{D}) = X_f Z_p^+, \tag{5.3}$$

where Z_p^+ *is the pseudo inverse matrix of* Z_p*.*

Proof. It follows from Theorem 5.1 that

$$\Theta Z_p Z_p^\top = X_f Z_p^\top \iff \Theta = X_f Z_p^\top \left(Z_p Z_p^\top\right)^{-1} = X_f Z_p^+,$$

where $Z_p Z_p^\top$ is nonsingular if Z_p is full row rank. □

It should be noted that the assumption is met almost surely as a sample size $|\mathcal{D}|$ approaches infinity. That is, Z_p is almost always full row rank in practice.

5.1.2 Maximum likelihood estimation

This section introduces the maximum likelihood (ML) estimator. Suppose a dataset $\mathcal{D}$ follows a conditional probability distribution given a system parameter Θ. Let $p(\mathcal{D} \mid \Theta)$ be a probability density function of the conditional probability distribution. Now consider the probability density function as a likelihood of Θ after observing $\mathcal{D}$ and define a likelihood function $l(\Theta \mid \mathcal{D}) \coloneqq p(\mathcal{D} \mid \Theta)$. The ML estimator is then defined as an estimator that maximizes $l(\Theta \mid \mathcal{D})$ given $\mathcal{D}$. In other words, the ML estimator gives the most likely system parameter under a given dataset.

Definition 5.2 (ML estimator)**.** *The maximum likelihood (ML) estimator* $\hat{\Theta}_{\mathrm{ML}}$ *for* Θ *given* $\mathcal{D}$ *is defined as*

$$\hat{\Theta}_{\mathrm{ML}}(\mathcal{D}) \coloneqq \arg\max_{\Theta} l(\Theta \mid \mathcal{D}),$$

where Θ *and* $\mathcal{D}$ *are defined in Definition 4.1, and* $l(\Theta \mid \mathcal{D}) = p(\mathcal{D} \mid \Theta)$ *is a likelihood function.*

Similar to the OLS estimator, the ML estimator in our attack scenario is given as a solution to an equation with data.

Theorem 5.2. *The ML estimator in Definition 5.2 is given as a solution* Θ *to the equation*

$$\left(\left(Z_p Z_p^\top\right) \otimes \Sigma_w^{-1}\right) \mathrm{vec}(\Theta) = \left(Z_p \otimes \Sigma_w^{-1}\right) \mathrm{vec}(X_f),$$

where Z_p *and* X_f *are respectively defined in* (5.1) *and* (5.2), Σ_w *is definined in* (4.1), $\otimes$ *is the Kronecker product, and* vec *is the vectorization of a matrix.*

Proof. Let Θ and $\mathcal{D}$ be as in Definition 4.1. If $\Theta = [A\ B]$, the likelihood function is given from (4.1) as

$$\begin{aligned}
&l\left(\begin{bmatrix} A & B \end{bmatrix} \mid \left\{\begin{bmatrix} x_t \\ u_t \end{bmatrix}\right\}_{t=t_s}^{t_f}\right), \\
&= p\left(\left\{\begin{bmatrix} x_t \\ u_t \end{bmatrix}\right\}_{t=t_s}^{t_f} \mid \begin{bmatrix} A & B \end{bmatrix}\right), \\
&= \prod_{t=t_s}^{t_f} p\left(\begin{bmatrix} x_t \\ u_t \end{bmatrix} \mid \begin{bmatrix} A & B \end{bmatrix}\right),
\end{aligned}$$

$$
\begin{aligned}
&= \prod_{t=t_s}^{t_f} p\Big(x_t \mid \begin{bmatrix} A & B \end{bmatrix}\Big) p(u_t), \\
&= p(x_{t_s}) p(u_{t_s}) \prod_{t=t_s+1}^{t_f} p_{\mathcal{N}}\left(x_t \mid \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} x_{t-1} \\ u_{t-1} \end{bmatrix}, \Sigma_w\right) p(u_t), \\
&= p(x_{t_s}) p(u_{t_s}) \prod_{t=t_s+1}^{t_f} p_{\mathcal{N}}\left(x_t; \left(\begin{bmatrix} x_{t-1} \\ u_{t-1} \end{bmatrix} \otimes I\right)^\top \mathrm{vec}\Big(\begin{bmatrix} A & B \end{bmatrix}\Big), \Sigma_w\right) p(u_t),
\end{aligned}
$$

where the probability density function of a Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ is denoted by $p_{\mathcal{N}}(\cdot; \mu, \Sigma)$. Moreover, the logarithm of the loss function is given as

$$
\begin{aligned}
&\ln\left(l\left(\begin{bmatrix} A & B \end{bmatrix} \mid \left\{\begin{bmatrix} x_t \\ u_t \end{bmatrix}\right\}_{t=t_s}^{t_f}\right)\right) \\
&= \sum_{t=t_s+1}^{t=t_f} \ln\left(p_{\mathcal{N}}\left(x_t; \left(\begin{bmatrix} x_{t-1} \\ u_{t-1} \end{bmatrix} \otimes I\right)^\top \mathrm{vec}\Big(\begin{bmatrix} A & B \end{bmatrix}\Big), \Sigma_w\right)\right) + \text{const.}, \\
&= -\frac{1}{2} \sum_{t=t_s+1}^{t=t_f} \left[\left(x_t - \left(\begin{bmatrix} x_{t-1} \\ u_{t-1} \end{bmatrix} \otimes I\right)^\top \mathrm{vec}\Big(\begin{bmatrix} A & B \end{bmatrix}\Big)\right)^\top \Sigma_w^{-1} \right. \\
&\quad \left. \left(x_t - \left(\begin{bmatrix} x_{t-1} \\ u_{t-1} \end{bmatrix} \otimes I\right)^\top \mathrm{vec}\Big(\begin{bmatrix} A & B \end{bmatrix}\Big)\right)\right] + \text{const.}, \\
&= -\frac{1}{2} \mathrm{vec}\Big(\begin{bmatrix} A & B \end{bmatrix}\Big)^\top \left[\sum_{t=t_s+1}^{t=t_f} \left(\begin{bmatrix} x_{t-1} \\ u_{t-1} \end{bmatrix} \otimes I\right) \Sigma_w^{-1} \left(\begin{bmatrix} x_{t-1} \\ u_{t-1} \end{bmatrix} \otimes I\right)^\top\right] \mathrm{vec}\Big(\begin{bmatrix} A & B \end{bmatrix}\Big) \\
&\quad + \left[\sum_{t=t_s+1}^{t=t_f} x_t^\top \Sigma_w^{-1} \left(\begin{bmatrix} x_{t-1} \\ u_{t-1} \end{bmatrix} \otimes I\right)^\top\right] \mathrm{vec}\Big(\begin{bmatrix} A & B \end{bmatrix}\Big) + \text{const.},
\end{aligned}
$$

where the probability density functions $p(x_{t_s})$ and $p(u_t)$ are regardless of the system parameter due to Definition 4.1 and Assumption 5.1. Similarly, if $\Theta = \bar{A}$, the likelihood function and its logarithm are given from (4.3) as

$$
\begin{aligned}
l\Big(\bar{A} \mid \{x_t\}_{t=t_s}^{t_f}\Big) &= \prod_{t=t_s}^{t_f} p\big(x_t \mid \bar{A}\big) = p(x_{t_s}) \prod_{t=t_s+1}^{t_f} p_{\mathcal{N}}\big(x_t; (x_{t-1} \otimes I)^\top \mathrm{vec}\big(\bar{A}\big), \Sigma_w\big), \\
\ln\Big(l\Big(\bar{A} \mid \{x_t\}_{t=t_s}^{t_f}\Big)\Big) &= -\frac{1}{2} \mathrm{vec}\big(\bar{A}\big)^\top \left[\sum_{t=t_s+1}^{t_f} (x_{t-1} \otimes I) \Sigma_w^{-1} (x_{t-1} \otimes I)^\top\right] \mathrm{vec}\big(\bar{A}\big)
\end{aligned}
$$

$$+\left[\sum_{t=t_s+1}^{t_f} x_t \Sigma_w^{-1}(x_{t-1}\otimes I)^\top\right]\operatorname{vec}\big(\bar{A}\big)+\text{const.}$$

Hence, it holds that

$$\begin{aligned}\ln(l(\Theta\mid\mathcal{D})) = -\frac{1}{2}\operatorname{vec}(\Theta)^\top\left[\sum_{t=t_s+1}^{t_f}(z_{t-1}\otimes I)\Sigma_w^{-1}(z_{t-1}\otimes I)^\top\right]\operatorname{vec}(\Theta)\\ +\left[\sum_{t=t_s+1}^{t_f} x_t\Sigma_w^{-1}(z_{t-1}\otimes I)^\top\right]\operatorname{vec}(\Theta)+\text{const.}\end{aligned}$$

It follows that

$$\begin{aligned}\sum_{t=t_s+1}^{t=t_f}(z_{t-1}\otimes I)\Sigma_w^{-1}(z_{t-1}\otimes I)^\top &= \begin{bmatrix} z_{t_s}\otimes I & \cdots & z_{t_f-1}\otimes I\end{bmatrix}\big(I\otimes\Sigma_w^{-1}\big)\begin{bmatrix} z_{t_s}^\top\otimes I\\ \vdots\\ z_{t_f-1}^\top\otimes I\end{bmatrix},\\ &= \left(\begin{bmatrix} z_{t_s} & \cdots & z_{t_f-1}\end{bmatrix}\otimes I\right)\big(I\otimes\Sigma_w^{-1}\big)\left(\begin{bmatrix} z_{t_s}^\top\\ \vdots\\ z_{t_f-1}^\top\end{bmatrix}\otimes I\right),\\ &= (Z_p\otimes I)\big(I\otimes\Sigma_w^{-1}\big)\big(Z_p^\top\otimes I\big),\\ &= \big(Z_pZ_p^\top\big)\otimes\Sigma_w^{-1},\\ \sum_{t=t_s+1}^{t=t_f} x_t^\top\Sigma_w^{-1}(z_{t-1}\otimes I)^\top &= \begin{bmatrix} x_{t_s+1}^\top & \cdots & x_{t_f}^\top\end{bmatrix}\big(I\otimes\Sigma_w^{-1}\big)(Z_p^\top\otimes I),\\ &= \operatorname{vec}(X_f)^\top\big(Z_p^\top\otimes\Sigma_w^{-1}\big).\end{aligned}$$

The likelihood function is maximized when its logarithm is maximized, and a maximizer Θ of the logarithm satisfies

$$\begin{aligned}&\frac{\partial}{\partial\operatorname{vec}(\Theta)}\ln(l(\Theta\mid\mathcal{D}))\\ &= \frac{\partial}{\partial\operatorname{vec}(\Theta)}\left(-\frac{1}{2}\operatorname{vec}(\Theta)^\top\big((Z_pZ_p^\top)\otimes\Sigma_w^{-1}\big)\operatorname{vec}(\Theta)+\operatorname{vec}(X_f)^\top(Z_p^\top\otimes\Sigma_w^{-1})\operatorname{vec}(\Theta)\right),\\ &= -\big((Z_pZ_p^\top)\otimes\Sigma_w^{-1}\big)\operatorname{vec}(\Theta)+\big(\operatorname{vec}(X_f)^\top(Z_p^\top\otimes\Sigma_w^{-1})\big)^\top,\\ &= -\big(\big(Z_pZ_p^\top\big)\otimes\Sigma_w^{-1}\big)\operatorname{vec}(\Theta)+\big(Z_p\otimes\Sigma_w^{-1}\big)\operatorname{vec}(X_f)=0.\end{aligned}$$

This yields the equation in the theorem. □

Interestingly, the ML estimator coincides with the OLS estimator if Z_p is full row rank.

Corollary 5.2. *Let $\mathcal{D}$ be as in Definition 4.1. If Z_p in (5.1) is full row rank, it holds that*

$$\hat{\Theta}_{\mathrm{ML}}(\mathcal{D}) = \hat{\Theta}_{\mathrm{OLS}}(\mathcal{D}),$$

where $\hat{\Theta}_{\mathrm{OLS}}$ and $\hat{\Theta}_{\mathrm{ML}}$ are definined in Definition 5.1 and Definition 5.2, respectively.

Proof. Corollary 5.1 yields $\hat{\Theta}_{\mathrm{OLS}} = X_f Z_p^+$. It follows from Theorem 5.2 that

$$\begin{aligned}
&\left(\left(Z_p Z_p^\top\right) \otimes \Sigma_w^{-1}\right) \mathrm{vec}(\Theta) = \left(Z_p \otimes \Sigma_w^{-1}\right) \mathrm{vec}(X_f), \\
\iff\ &\mathrm{vec}(\Theta) = \left(\left(Z_p Z_p^\top\right) \otimes \Sigma_w^{-1}\right)^{-1} \left(Z_p \otimes \Sigma_w^{-1}\right) \mathrm{vec}(X_f), \\
&= \left(\left(Z_p Z_p^\top\right)^{-1} \otimes \Sigma_w\right) \left(Z_p \otimes \Sigma_w^{-1}\right) \mathrm{vec}(X_f), \\
&= \left(\left(Z_p Z_p^\top\right)^{-1} Z_p \otimes I\right) \mathrm{vec}(X_f), \\
&= \mathrm{vec}\left(X_f Z_p^\top \left(Z_p Z_p^\top\right)^{-1}\right), \\
&= \mathrm{vec}\left(X_f Z_p^+\right),
\end{aligned}$$

where if Z_p is full row rank, $Z_p Z_p^\top$ is nonsingular, and so is $(Z_p Z_p^\top) \otimes \Sigma_w^{-1}$. □

Recall that the assumption on the rank of Z_p is almost always satisfied. Hence, the corollary implies that the OLS estimator gives the most likely estimate of a system parameter in our attack scenario.

5.1.3 Maximum a posteriori estimation

The maximum a posterior (MAP) estimator is closely related to the ML estimator. Suppose a system parameter Θ is a random variable following a probability distribution with a probability density function $p(\Theta)$. The ML estimator maximizes a likelihood function $l(\Theta \mid \mathcal{D}) = p(\mathcal{D} \mid \Theta)$ given a dataset $\mathcal{D}$. In contrast, the MAP estimator is defined as an estimator that maximizes a posterior probability density function $p(\Theta \mid \mathcal{D})$ after observing $\mathcal{D}$.

Definition 5.3 (MAP estimator)**.** *The maximum a posteriori (MAP) estimator $\hat{\Theta}_{\mathrm{MAP}}$ for Θ given $\mathcal{D}$ is defined as*

$$\hat{\Theta}_{\mathrm{MAP}}(\mathcal{D}) \coloneqq \arg\max_{\Theta} p(\Theta \mid \mathcal{D}),$$

where Θ and $\mathcal{D}$ are defined in Definition 4.1.

By definition, it is obvious that the MAP estimator is equivalent to a maximizer of the product between a likelihood function and a prior probability density function.

Proposition 5.1. *The MAP estimator in Definition 5.3 satisfies*

$$\hat{\Theta}_{\mathrm{MAP}}(\mathcal{D}) = \arg\max_{\Theta} p(\mathcal{D} \mid \Theta)p(\Theta),$$

where Θ and $\mathcal{D}$ are defined in Definition 4.1.

Proof. It follows from Bayes' theorem that

$$\hat{\Theta}_{\mathrm{MAP}}(\mathcal{D}) = \arg\max_{\Theta} p(\Theta \mid \mathcal{D}) = \arg\max_{\Theta} \frac{p(\mathcal{D} \mid \Theta)p(\Theta)}{p(\mathcal{D})}.$$

The proposition holds because $p(\mathcal{D} \mid \Theta)p(\Theta)/p(\mathcal{D})$ can be maximized regardless of $p(\mathcal{D})$. □

The proposition shows that an adversary is required to specify a prior probability density function of a system parameter for obtaining the MAP estimator. Meanwhile, the adversary cannot know the prior, so it should be assumed. One of the reasonable prior distributions in our attack scenario is a uniform distribution because it represents that an adversary does not have information about a system parameter. In such a case, the MAP estimator is equivalent to the ML estimator.

Corollary 5.3. *Let Θ and $\mathcal{D}$ be as in Definition 4.1. If the vectorization of Θ in Definition 4.1 follows a uniform distribution, it holds that*

$$\hat{\Theta}_{\mathrm{MAP}}(\mathcal{D}) = \hat{\Theta}_{\mathrm{ML}}(\mathcal{D}),$$

where $\hat{\Theta}_{\mathrm{ML}}$ and $\hat{\Theta}_{\mathrm{MAP}}$ are definined in Definition 5.2 and Definition 5.3, respectively.

Proof. From the assumption, $p(\Theta)$ in Proposition 5.1 is constant, namely $p(\Theta) \propto 1$. Hence, it holds that

$$\hat{\Theta}_{\mathrm{MAP}}(\mathcal{D}) = \arg\max_{\Theta} p(\mathcal{D} \mid \Theta) = \arg\max_{\Theta} l(\Theta \mid \mathcal{D}) = \hat{\Theta}_{\mathrm{ML}}(\mathcal{D}),$$

where $l(\Theta \mid \mathcal{D}) = p(\mathcal{D} \mid \Theta)$. □

5.1.4 Bayesian estimation

The last introduced representative estimator is the Bayes estimator. Similar to the MAP estimator, suppose a system parameter Θ is a random variable following a certain probability distribution. The Bayes estimator is then defined as an estimator that minimizes the Bayes risk, a type of average loss in selecting an estimator.

Definition 5.4 (Bayes risk)**.** *Let Θ and $\mathcal{D}$ be as in Definition 4.1, and let $\hat{\Theta}$ be an estimator for Θ given $\mathcal{D}$. The Bayes risk of $\hat{\Theta}$ is defined as*

$$\mathbb{E}\Big[\ell\Big(\Theta, \hat{\Theta}(\mathcal{D})\Big)\Big] = \int \ell\Big(\Theta, \hat{\Theta}(\mathcal{D})\Big) p(\mathcal{D} \mid \Theta) d\mathcal{D} \int p(\Theta) d\Theta,$$

where $\ell(\Theta, \hat{\Theta}(\mathcal{D})) \geq 0$ is a loss function.

Definition 5.5 (Bayes estimator)**.** *The Bayes estimator $\hat{\Theta}_{\mathrm{Bayes}}$ for Θ given $\mathcal{D}$ is defined as a minimizer of the Bayes risk in Definition 5.4, namely*

$$\hat{\Theta}_{\mathrm{Bayes}}(\mathcal{D}) \coloneqq \arg\min_{\hat{\Theta}} \mathbb{E}\Big[\ell\Big(\Theta, \hat{\Theta}(\mathcal{D})\Big)\Big],$$

where Θ and $\mathcal{D}$ are defined in Definition 4.1.

A loss function in Definition 5.4 represents a cost in selecting an estimator. The cost should be constructed to become smaller when a better estimator is chosen. In this light, an estimation error defined in Definition 4.2 is one of the reasonable choices of a loss function in our attack scenario. In such a case, the Bayes estimator is given as the conditional expectation of a system parameter given a dataset.

Lemma 5.1. *If a loss function in Definition 5.4 is ε in Definition 4.2, i.e.,*

$$\ell\Big(\Theta, \hat{\Theta}(\mathcal{D})\Big) = \varepsilon\Big(\Theta, \hat{\Theta}(\mathcal{D})\Big) = \frac{1}{nd}\Big\|\Theta - \hat{\Theta}(\mathcal{D})\Big\|_F^2, \tag{5.4}$$

then the Bayes estimator in Definition 5.5 satisfies

$$\hat{\Theta}_{\mathrm{Bayes}}(\mathcal{D}) = \mathbb{E}[\Theta \mid \mathcal{D}],$$

where Θ and $\mathcal{D}$ are defined in Definition 4.1.

Proof. It follows from Bayes' theorem that

$$\hat{\Theta}_{\mathrm{Bayes}}(\mathcal{D}) = \arg\min_{\hat{\Theta}} \mathbb{E}\Big[\ell\Big(\Theta, \hat{\Theta}(\mathcal{D})\Big)\Big],$$

$$
\begin{aligned}
&= \arg\min_{\hat{\Theta}} \int \ell\Big(\Theta, \hat{\Theta}(\mathcal{D})\Big) p(\mathcal{D} \mid \Theta) d\mathcal{D} \int p(\Theta) d\Theta, \\
&= \arg\min_{\hat{\Theta}} \int \ell\Big(\Theta, \hat{\Theta}(\mathcal{D})\Big) p(\Theta \mid \mathcal{D}) d\Theta \int p(\mathcal{D}) d\mathcal{D}, \\
&= \arg\min_{\hat{\Theta}} \int \ell\Big(\Theta, \hat{\Theta}(\mathcal{D})\Big) p(\Theta \mid \mathcal{D}) d\Theta, \\
&= \arg\min_{\hat{\Theta}} \mathbb{E}\Big[\ell\Big(\Theta, \hat{\Theta}(\mathcal{D})\Big) \mid \mathcal{D}\Big], \\
&= \arg\min_{\hat{\Theta}} \mathbb{E}\left[\frac{1}{nd}\left\|\Theta - \hat{\Theta}(\mathcal{D})\right\|_F^2 \mid \mathcal{D}\right], \\
&= \arg\min_{\hat{\Theta}} \mathbb{E}\left[\left\|\Theta - \hat{\Theta}(\mathcal{D})\right\|_F^2 \mid \mathcal{D}\right].
\end{aligned}
$$

The conditional expectation is calculated as

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\Theta - \hat{\Theta}(\mathcal{D})\right\|_F^2 \mid \mathcal{D}\right] \\
&= \mathbb{E}\left[\left\|\Theta + \mathbb{E}[\Theta \mid \mathcal{D}] - \mathbb{E}[\Theta \mid \mathcal{D}] - \hat{\Theta}(\mathcal{D})\right\|_F^2 \mid \mathcal{D}\right], \\
&= \mathbb{E}\Big[\mathrm{tr}\Big(\Big(\Theta + \mathbb{E}[\Theta \mid \mathcal{D}] - \mathbb{E}[\Theta \mid \mathcal{D}] - \hat{\Theta}(\mathcal{D})\Big) \\
&\quad \Big(\Theta + \mathbb{E}[\Theta \mid \mathcal{D}] - \mathbb{E}[\Theta \mid \mathcal{D}] - \hat{\Theta}(\mathcal{D})\Big)^\top\Big) \mid \mathcal{D}\Big], \\
&= \mathbb{E}\big[\mathrm{tr}\big((\Theta - \mathbb{E}[\Theta \mid \mathcal{D}])(\Theta - \mathbb{E}[\Theta \mid \mathcal{D}])^\top\big) \mid \mathcal{D}\big] \\
&\quad + 2\,\mathbb{E}\left[\mathrm{tr}\Big((\Theta - \mathbb{E}[\Theta \mid \mathcal{D}])\Big(\mathbb{E}[\Theta \mid \mathcal{D}] - \hat{\Theta}(\mathcal{D})\Big)^\top\Big) \mid \mathcal{D}\right] \\
&\quad + \mathbb{E}\left[\mathrm{tr}\Big(\Big(\mathbb{E}[\Theta \mid \mathcal{D}] - \hat{\Theta}(\mathcal{D})\Big)\Big(\mathbb{E}[\Theta \mid \mathcal{D}] - \hat{\Theta}(\mathcal{D})\Big)^\top\Big) \mid \mathcal{D}\right], \\
&= \mathbb{E}[\|\Theta - \mathbb{E}[\Theta \mid \mathcal{D}]\|_F^2 \mid \mathcal{D}] + \mathbb{E}\left[\left\|\mathbb{E}[\Theta \mid \mathcal{D}] - \hat{\Theta}(\mathcal{D})\right\|_F^2 \mid \mathcal{D}\right] \\
&\quad + 2\,\mathbb{E}\left[\mathrm{tr}\Big((\Theta - \mathbb{E}[\Theta \mid \mathcal{D}])\Big(\mathbb{E}[\Theta \mid \mathcal{D}] - \hat{\Theta}(\mathcal{D})\Big)^\top\Big) \mid \mathcal{D}\right], \\
&= \mathbb{E}[\|\Theta - \mathbb{E}[\Theta \mid \mathcal{D}]\|_F^2 \mid \mathcal{D}] + \left\|\mathbb{E}[\Theta \mid \mathcal{D}] - \hat{\Theta}(\mathcal{D})\right\|_F^2.
\end{aligned}
$$

Consequently, the Bayes risk is bounded as

$$
\begin{aligned}
\mathbb{E}\left[\left\|\Theta - \hat{\Theta}(\mathcal{D})\right\|_F^2 \mid \mathcal{D}\right] &= \mathbb{E}[\|\Theta - \mathbb{E}[\Theta \mid \mathcal{D}]\|_F^2 \mid \mathcal{D}] + \left\|\mathbb{E}[\Theta \mid \mathcal{D}] - \hat{\Theta}(\mathcal{D})\right\|_F^2, \\
&\geq \mathbb{E}[\|\Theta - \mathbb{E}[\Theta \mid \mathcal{D}]\|_F^2 \mid \mathcal{D}],
\end{aligned}
$$

where the equality holds if $\hat{\Theta}(\mathcal{D}) = \mathbb{E}[\Theta \mid \mathcal{D}]$. This completes the proof. □

By definition, Lemma 5.1 implies that the Bayes estimator minimizes the expectation of estimation error in Definition 4.2. Recall that the estimation error is a mean square error between a system parameter and an estimate. Hence, the Bayes estimator is often called the minimum mean square error estimator.

From Lemma 5.1 and Bayes' theorem, the Bayes estimator is obtained as

$$\hat{\Theta}_{\mathrm{Bayes}} = \mathbb{E}[\Theta \mid \mathcal{D}] = \int \Theta p(\Theta \mid \mathcal{D}) d\Theta = \int \Theta \frac{p(\mathcal{D} \mid \Theta) p(\Theta)}{p(\mathcal{D})} d\Theta.$$

A probability density function $p(\mathcal{D})$ is, in general, difficult to calculate analytically and required to be estimated by a probabilistic algorithm, such as Markov chain Monte Carlo methods. However, if a prior probability distribution of a system parameter is a Gaussian distribution, $p(\mathcal{D})$ can be derived analytically, and hence the Bayes estimator is obtained explicitly.

Theorem 5.3. *Suppose a loss function is* (5.4)*. If the vectorization of* Θ *follows the Gaussian distribution* $\mathcal{N}(\mu_\Theta, \Sigma_\Theta)$ *with mean* $\mu_\Theta \in \mathbb{R}^{nd}$ *and variance* $\Sigma_\Theta \in \mathbb{R}^{nd \times nd}$*, it holds that*

$$\mathrm{vec}\Big(\hat{\Theta}_{\mathrm{Bayes}}(\mathcal{D})\Big) = \hat{\mu}_\Theta, \tag{5.5}$$

where

$$\begin{aligned}
\hat{\mu}_\Theta &= \hat{\Sigma}_\Theta\big(\Sigma_\Theta^{-1}\mu_\Theta + \big(Z_p \otimes \Sigma_w^{-1}\big)\,\mathrm{vec}(X_f)\big),\\
\hat{\Sigma}_\Theta &= \big(\Sigma_\Theta^{-1} + \big(Z_p Z_p^\top\big) \otimes \Sigma_w^{-1}\big)^{-1},
\end{aligned}$$

Z_p and X_f are respectively defined in (5.1) *and* (5.2)*, and Σ_w is defined in* (4.1)*.*

Proof. Lemma 5.1 and Bayes' theorem yield that

$$\begin{aligned}
\mathrm{vec}\Big(\hat{\Theta}_{\mathrm{Bayes}}(\mathcal{D})\Big) &= \mathrm{vec}(\mathbb{E}[\Theta \mid \mathcal{D}]),\\
&= \int \mathrm{vec}(\Theta) p(\mathrm{vec}(\Theta) \mid \mathcal{D}) d\,\mathrm{vec}(\Theta),\\
&= \int \mathrm{vec}(\Theta) \frac{p(\mathcal{D} \mid \mathrm{vec}(\Theta)) p(\mathrm{vec}(\Theta))}{p(\mathcal{D})} d\,\mathrm{vec}(\Theta).
\end{aligned}$$

Let $p_{\mathcal{N}}(\cdot; \mu, \Sigma)$ be the probability density function of a Gaussian distribution with mean μ and variance Σ. The probability density function $p(\mathrm{vec}(\Theta))$ is given from

the assumption as

$$\begin{aligned} p(\operatorname{vec}(\Theta)) &= p_{\mathcal{N}}(\operatorname{vec}(\Theta); \mu_\Theta, \Sigma_\Theta), \\ &= c_\Theta \exp\left(-\frac{1}{2}(\operatorname{vec}(\Theta) - \mu_\Theta)^\top \Sigma_\Theta^{-1}(\operatorname{vec}(\Theta) - \mu_\Theta)\right), \\ &= p_\Theta c_\Theta \exp\left(-\frac{1}{2}\big(\operatorname{vec}(\Theta)^\top \Sigma_\Theta^{-1} \operatorname{vec}(\Theta) - 2\mu_\Theta^\top \Sigma_\Theta^{-1} \operatorname{vec}(\Theta)\big)\right), \end{aligned}$$

where $c_\Theta = ((2\pi)^{nd} \det(\Sigma_\Theta))^{-1/2}$, and $p_\Theta = \exp(-(1/2)\mu_\Theta^\top \Sigma_\Theta^{-1} \mu_\Theta)$. Additionally, the conditional probability density function $p(\mathcal{D} \mid \operatorname{vec}(\Theta))$ is given as

$$\begin{aligned} &p(\mathcal{D} \mid \operatorname{vec}(\Theta)) \\ &= \prod_{t=t_s}^{t_f} p(z_t \mid \operatorname{vec}(\Theta)), \\ &= p_z \prod_{t=t_s+1}^{t_f} p_{\mathcal{N}}\big(x_t \mid (z_t \otimes I)^\top \operatorname{vec}(\Theta), \Sigma_w\big), \\ &= p_z c_{\mathcal{D}} \exp\left(-\frac{1}{2} \sum_{t=t_s+1}^{t_f} \big(x_t - (z_t \otimes I)^\top \operatorname{vec}(\Theta)\big)^\top \Sigma_w^{-1} \big(x_t - (z_t \otimes I)^\top \operatorname{vec}(\Theta)\big)\right), \end{aligned}$$

where

$$p_z = \begin{cases} p(x_{t_s}) \displaystyle\prod_{t=t_s}^{t_f} p(u_t), & \Theta = \begin{bmatrix} A & B \end{bmatrix}, \\ p(x_{t_s}), & \Theta = \bar{A}, \end{cases}$$

$c_{\mathcal{D}} = ((2\pi)^n \det(\Sigma_w))^{-|\mathcal{D}|/2}$, and z_t is definined in Definition 4.1. Similarly to the proof of Theorem 5.2, it holds that

$$\begin{aligned} &-\frac{1}{2} \sum_{t=t_s+1}^{t_f} \big(x_t - (z_t \otimes I)^\top \operatorname{vec}(\Theta)\big)^\top \Sigma_w^{-1} \big(x_t - (z_t \otimes I)^\top \operatorname{vec}(\Theta)\big) \\ &= -\frac{1}{2}\left(\operatorname{vec}(\Theta)^\top \left(\sum_{t=t_s+1}^{t_f} (z_t \otimes I)\Sigma_w^{-1}(z_t \otimes I)^\top\right) \operatorname{vec}(\Theta)\right. \\ &\quad \left. -2\left(\sum_{t=t_s+1}^{t=t_f} x_t^\top \Sigma_w^{-1}(z_{t-1} \otimes I)^\top\right) \operatorname{vec}(\Theta) + \sum_{t=t_s+1}^{t_f} x_t^\top \Sigma_w^{-1} x_t\right), \\ &= -\frac{1}{2}\big(\operatorname{vec}(\Theta)^\top \big(\big(Z_p Z_p^\top\big) \otimes \Sigma_w^{-1}\big) \operatorname{vec}(\Theta) - 2 \operatorname{vec}(X_f)^\top \big(Z_p^\top \otimes \Sigma_w^{-1}\big) \operatorname{vec}(\Theta) \end{aligned}$$

$$+ \operatorname{vec}(X_f)^\top (I \otimes \Sigma_w^{-1}) \operatorname{vec}(X_f)\big),$$

where

$$\begin{aligned}\sum_{t=t_s+1}^{t_f} x_t^\top \Sigma_w^{-1} x_t &= \begin{bmatrix} x_{t_s+1}^\top & \cdots & x_{t_f}^\top \end{bmatrix} (I \otimes \Sigma_w^{-1}) \begin{bmatrix} x_{t_s+1} \\ \vdots \\ x_{t_f} \end{bmatrix}, \\ &= \operatorname{vec}(X_f)^\top (I \otimes \Sigma_w^{-1}) \operatorname{vec}(X_f).\end{aligned}$$

Hence, $p(\mathcal{D} \mid \operatorname{vec}(\Theta))p(\operatorname{vec}(\Theta))$ is obtained as

$$\begin{aligned}&p(\mathcal{D} \mid \operatorname{vec}(\Theta))p(\operatorname{vec}(\Theta)) \\ &= p_z p_\Theta p_\mathcal{D} c_\Theta c_\mathcal{D} \exp\left(-\frac{1}{2}\left(\operatorname{vec}(\Theta)^\top \hat{\Sigma}_\Theta^{-1} \operatorname{vec}(\Theta) - 2\hat{\mu}_\Theta^\top \hat{\Sigma}_\Theta^{-1} \operatorname{vec}(\Theta)\right)\right),\end{aligned}$$

where $p_\mathcal{D} = \exp(-(1/2) \operatorname{vec}(X_f)^\top (I \otimes \Sigma_w^{-1}) \operatorname{vec}(X_f))$. The probability density function $p(\mathcal{D})$ can be computed by integrating $p(\mathcal{D} \mid \operatorname{vec}(\Theta))p(\operatorname{vec}(\Theta))$ as

$$\begin{aligned}p(\mathcal{D}) &= \int p(\mathcal{D} \mid \operatorname{vec}(\Theta))p(\operatorname{vec}(\Theta)) d\operatorname{vec}(\Theta), \\ &= \int p_z p_\Theta p_\mathcal{D} c_\Theta c_\mathcal{D} \exp\left(-\frac{1}{2}\left(\operatorname{vec}(\Theta)^\top \hat{\Sigma}_\Theta^{-1} \operatorname{vec}(\Theta) - 2\hat{\mu}_\Theta^\top \hat{\Sigma}_\Theta^{-1} \operatorname{vec}(\Theta)\right)\right) d\operatorname{vec}(\Theta), \\ &= p_z p_\Theta p_\mathcal{D} c_\Theta c_\mathcal{D} \int \exp\left(-\frac{1}{2} \operatorname{vec}(\Theta)^\top \hat{\Sigma}_\Theta^{-1} \operatorname{vec}(\Theta) + \hat{\mu}_\Theta^\top \hat{\Sigma}_\Theta^{-1} \operatorname{vec}(\Theta)\right) d\operatorname{vec}(\Theta), \\ &= p_z p_\Theta p_\mathcal{D} c_\Theta c_\mathcal{D} \sqrt{\frac{(2\pi)^{nd}}{\det\left(\hat{\Sigma}_\Theta^{-1}\right)}} \exp\left(\frac{1}{2}\left(\hat{\mu}_\Theta^\top \hat{\Sigma}_\Theta^{-1}\right)\left(\hat{\Sigma}_\Theta^{-1}\right)^{-1}\left(\hat{\mu}_\Theta^\top \hat{\Sigma}_\Theta^{-1}\right)^\top\right), \\ &= p_z p_\Theta p_\mathcal{D} c_\Theta c_\mathcal{D} \sqrt{(2\pi)^{nd} \det\left(\hat{\Sigma}_\Theta\right)} \exp\left(\frac{1}{2} \hat{\mu}_\Theta^\top \hat{\Sigma}_\Theta^{-1} \hat{\mu}_\Theta\right),\end{aligned}$$

where the fourth equality follows from the Gaussian integral. Consequently, the vectorization of the Bayes estimator is

$$\begin{aligned}&\operatorname{vec}\left(\hat{\Theta}_{\text{Bayes}}(\mathcal{D})\right) \\ &= \int \operatorname{vec}(\Theta) \frac{p(\mathcal{D} \mid \operatorname{vec}(\Theta))p(\operatorname{vec}(\Theta))}{p(\mathcal{D})} d\operatorname{vec}(\Theta),\end{aligned}$$

$$
\begin{aligned}
&= \int \mathrm{vec}(\Theta) \frac{\exp\left(-\frac{1}{2}\left(\mathrm{vec}(\Theta)^\top \hat{\Sigma}_\Theta^{-1} \mathrm{vec}(\Theta) - 2\hat{\mu}_\Theta^\top \hat{\Sigma}_\Theta^{-1} \mathrm{vec}(\Theta)\right)\right)}{\sqrt{(2\pi)^{nd} \det\left(\hat{\Sigma}_\Theta\right)} \exp\left(\frac{1}{2}\hat{\mu}_\Theta^\top \hat{\Sigma}_\Theta^{-1} \hat{\mu}_\Theta\right)} d\,\mathrm{vec}(\Theta), \\
&= \int \mathrm{vec}(\Theta) \frac{1}{\sqrt{(2\pi)^{nd} \det\left(\hat{\Sigma}_\Theta\right)}} \\
&\quad \exp\left(-\frac{1}{2}\left(\mathrm{vec}(\Theta)^\top \hat{\Sigma}_\Theta^{-1} \mathrm{vec}(\Theta) - 2\hat{\mu}_\Theta^\top \hat{\Sigma}_\Theta^{-1} \mathrm{vec}(\Theta) - \hat{\mu}_\Theta^\top \hat{\Sigma}_\Theta^{-1} \hat{\mu}_\Theta\right)\right) d\,\mathrm{vec}(\Theta), \\
&= \int \mathrm{vec}(\Theta) \frac{1}{\sqrt{(2\pi)^{nd} \det\left(\hat{\Sigma}_\Theta\right)}} \\
&\quad \exp\left(-\frac{1}{2}(\mathrm{vec}(\Theta) - \hat{\mu}_\Theta)^\top \hat{\Sigma}_\Theta^{-1} (\mathrm{vec}(\Theta) - \hat{\mu}_\Theta)\right) d\,\mathrm{vec}(\Theta), \\
&= \int \mathrm{vec}(\Theta) p_{\mathcal{N}}\left(\mathrm{vec}(\Theta); \hat{\mu}_\Theta, \hat{\Sigma}_\Theta\right) d\,\mathrm{vec}(\Theta), \\
&= \hat{\mu}_\Theta.
\end{aligned}
$$

This completes the proof. □

Moreover, if a posterior probability distribution is a Gaussian distribution, the Bayes estimator becomes equivalent to the MAP estimator.

Corollary 5.4. *Let Θ and $\mathcal{D}$ be as in Definition 4.1. Suppose a loss function is* (5.4)*. If the vectorization of Θ given $\mathcal{D}$ follows a Gaussian distribution, it holds that*

$$\hat{\Theta}_{\mathrm{Bayes}}(\mathcal{D}) = \hat{\Theta}_{\mathrm{MAP}}(\mathcal{D}),$$

where $\hat{\Theta}_{\mathrm{MAP}}$ and $\hat{\Theta}_{\mathrm{Bayes}}$ are definined in Definition 5.3 and Definition 5.5, respectively.

Proof. By definition, the vectorization of the MAP estimator is

$$\mathrm{vec}\left(\hat{\Theta}_{\mathrm{MAP}}\right) = \arg\max_{\Theta} p(\mathrm{vec}(\Theta) \mid \mathcal{D}).$$

The maximizer of $p(\mathrm{vec}(\Theta) \mid \mathcal{D})$ is given as its expectation, namely

$$\mathrm{vec}\left(\hat{\Theta}_{\mathrm{MAP}}\right) = \mathbb{E}[\mathrm{vec}(\Theta) \mid \mathcal{D}] = \mathrm{vec}(\mathbb{E}[\Theta \mid \mathcal{D}]) = \mathrm{vec}\left(\hat{\Theta}_{\mathrm{Bayes}}\right).$$

where the last equality follows from Lemma 5.1. □

Note that, in our attack scenario, if a prior probability distribution is a Gaussian distribution, a posterior probability distribution is also a Gaussian distribution. Hence, Corollary 5.4 means that the Bayes and MAP estimators are obtained as (5.5) if a prior probability distribution is a Gaussian distribution.

A prior probability distribution represents knowledge of an adversary about a system parameter in advance before estimation. Now consider an adversary who has no knowledge about a system parameter. As already described in Section 5.1.3, in this case, a prior probability distribution is a uniform distribution. Fortunately, a posterior probability distribution becomes a Gaussian distribution even when employing a uniform prior distribution. Therefore, if a dataset is sufficiently large, the Bayes estimator coincides with the MAP, ML, and OLS estimators.

Theorem 5.4. *Let Θ and $\mathcal{D}$ be as in Definition 4.1, and let X_f be as in* (5.2)*. Suppose a loss function is* (5.4)*. If the vectorization of Θ follows a uniform distribution, and if Z_p in* (5.1) *is full row rank, then it holds that*

$$\hat{\Theta}_{\mathrm{Bayes}}(\mathcal{D}) = \hat{\Theta}_{\mathrm{MAP}}(\mathcal{D}) = \hat{\Theta}_{\mathrm{ML}}(\mathcal{D}) = \hat{\Theta}_{\mathrm{OLS}}(\mathcal{D}) = X_f Z_p^+,$$

where $\hat{\Theta}_{\mathrm{OLS}}$, $\hat{\Theta}_{\mathrm{ML}}$, $\hat{\Theta}_{\mathrm{MAP}}$, and $\hat{\Theta}_{\mathrm{Bayes}}$ are defnined in Definition 5.1, Definition 5.2, Definition 5.3, and Definition 5.5, respectively.

Proof. By similar computation in the proof of Theorem 5.3, it follows that

$$p(\mathcal{D} \mid \mathrm{vec}(\Theta)) = p_z p_{\mathcal{D}} c_{\mathcal{D}} \exp\left(-\frac{1}{2}\left(\mathrm{vec}(\Theta)^\top \hat{\Sigma}_\Theta^{-1} \mathrm{vec}(\Theta) - 2\hat{\mu}_\Theta^\top \hat{\Sigma}_\Theta^{-1} \mathrm{vec}(\Theta)\right)\right),$$

where

$$\begin{aligned}
\hat{\mu}_\Theta &= \hat{\Sigma}_\Theta\left(Z_p \otimes \Sigma_w^{-1}\right) \mathrm{vec}(X_f), \\
\hat{\Sigma}_\Theta &= \left(Z_p Z_p^\top\right)^{-1} \otimes \Sigma_w, \\
p_z &= \begin{cases} p(x_{t_s}) \displaystyle\prod_{t=t_s}^{t_f} p(u_t), & \Theta = \begin{bmatrix} A & B \end{bmatrix}, \\ p(x_{t_s}), & \Theta = \bar{A}, \end{cases} \\
p_{\mathcal{D}} &= \exp(-(1/2)\,\mathrm{vec}(X_f)^\top (I \otimes \Sigma_w^{-1})\,\mathrm{vec}(X_f)), \\
c_{\mathcal{D}} &= ((2\pi)^n \det(\Sigma_w))^{-|\mathcal{D}|/2},
\end{aligned}$$

and Σ_w is defined in (4.1). The probability density function of $\mathcal{D}$ can be computed

as

$$
\begin{aligned}
&p(\mathcal{D}) \\
&= \int p(\mathcal{D} \mid \mathrm{vec}(\Theta)) p(\mathrm{vec}(\Theta)) d\,\mathrm{vec}(\Theta), \\
&= p(\mathrm{vec}(\Theta)) \int p(\mathcal{D} \mid \mathrm{vec}(\Theta)) d\,\mathrm{vec}(\Theta), \\
&= p_z p_{\mathcal{D}} c_{\mathcal{D}} p(\mathrm{vec}(\Theta)) \int \exp\left(-\frac{1}{2}\left(\mathrm{vec}(\Theta)^\top \hat{\Sigma}_\Theta^{-1}\,\mathrm{vec}(\Theta) - 2\hat{\mu}_\Theta^\top \hat{\Sigma}_\Theta^{-1}\,\mathrm{vec}(\Theta)\right)\right) d\,\mathrm{vec}(\Theta), \\
&= p_z p_{\mathcal{D}} c_{\mathcal{D}} p(\mathrm{vec}(\Theta)) \sqrt{(2\pi)^{nd} \det\left(\hat{\Sigma}_\Theta\right)} \exp\left(\frac{1}{2}\hat{\mu}_\Theta^\top \hat{\Sigma}_\Theta^{-1} \hat{\mu}_\Theta\right).
\end{aligned}
$$

The posterior probability density function is given as

$$
\begin{aligned}
p(\mathrm{vec}(\Theta) \mid \mathcal{D}) &= \frac{p(\mathcal{D} \mid \mathrm{vec}(\Theta)) p(\mathrm{vec}(\Theta))}{p(\mathcal{D})}, \\
&= \frac{1}{\sqrt{(2\pi)^{nd} \det\left(\hat{\Sigma}_\Theta\right)}} \exp\left(-\frac{1}{2}(\mathrm{vec}(\Theta) - \hat{\mu}_\Theta)^\top \hat{\Sigma}_\Theta^{-1} (\mathrm{vec}(\Theta) - \hat{\mu}_\Theta)\right), \\
&= p_{\mathcal{N}}\left(\mathrm{vec}(\Theta); \hat{\mu}, \hat{\Sigma}\right),
\end{aligned}
$$

where $p_{\mathcal{N}}(\cdot; \hat{\mu}, \hat{\Sigma})$ is the probability density function of the Gaussian distribution with mean $\hat{\mu}$ and variance $\hat{\Sigma}$. Therefore, the theorem follows from Corollary 5.1, Corollary 5.2, Corollary 5.3, and Corollary 5.4. □

The theorem implies that, in our attack scenario, the representative estimators can be unified by the OLS estimator (5.3) under some reasonable assumptions. In summary,

- if the sample size of a dataset $\mathcal{D}$ in Definition 4.1 is sufficiently large such that Z_p in (5.1) is full row rank,
- and if the prior probability distribution of a system parameter Θ in Definition 4.1 is a uniform distribution,

then the OLS estimator satisfies that

- it gives the most accurate prediction of X_f in (5.1) in the sense of mean square error,

- it gives the most likely estimate of Θ,
- and it minimizes the estimation error in Definition 4.2.

For these reasons, in what follows, this thesis supposes that an adversary employs the OLS estimator to identify a system parameter in the attack of Definition 4.1.

5.2 Optimal design

Following the discussions in Section 5.1, this section derives a solution to Problem 5.1 for each case where an attack target is a plant (4.1) or closed-loop system (4.3) under an adversary who uses the OLS estimator.

5.2.1 Open-loop case

Recall that, under Assumption 5.1, an encrypted control system is (γ_c, τ_c)-secure if and only if $\tau(N^*(\gamma_c, \Theta; \hat{\Theta}), \lambda; \Upsilon) > \tau_c$ holds. Here, an acceptable estimation error γ_c and a defense period τ_c are design parameters. A system parameter $\Theta = [A\ B]$ is determined by a plant (4.1) when it is an attack target. An estimator $\hat{\Theta}$ is the OLS estimator (5.3), and a computer performance Υ is specified from the capability of an anticipated adversary. Consequently, if an attack target is (4.1), our objective is to design the minimum security parameter λ^* such that $\tau(N^*(\gamma_c, \Theta; \hat{\Theta}), \lambda^*; \Upsilon) > \tau_c$ with the given conditions.

A sample identifying complexity must be computed to design the minimum security parameter and is obtained from the expectation of an estimation error and an acceptable estimation error. The following lemma shows the expectation of estimation error in Definition 4.2 under the OLS estimator.

Lemma 5.2. *Consider the OLS estimator* (5.3). *Let* Θ *and* $\mathcal{D}$ *be as in Definition 4.1, and let* Z_p *be as in* (5.1). *The expectation of estimation error in Definition 4.2 satisfies*

$$\mathbb{E}\Big[\varepsilon\Big(\Theta, \hat{\Theta}_{\mathrm{OLS}}(\mathcal{D})\Big)\Big] = \frac{1}{nd}\operatorname{tr}(\Sigma_w)\operatorname{tr}\Big(\mathbb{E}\Big[\big(Z_p Z_p^\top\big)^{-1}\Big]\Big),$$

where Σ_w *is defined in* (4.1).

Proof. It follows from Definition 4.2, (5.1), and (5.3) that

$$\begin{aligned}
\mathbb{E}\Big[\varepsilon\Big(\Theta, \hat{\Theta}_{\mathrm{OLS}}(\mathcal{D})\Big)\Big] &= \frac{1}{nd}\,\mathbb{E}\Big[\Big\|\Theta - \hat{\Theta}_{\mathrm{OLS}}(\mathcal{D})\Big\|_F^2\Big], \\
&= \frac{1}{nd}\,\mathbb{E}\Big[\big\|\Theta - X_f Z_p^+\big\|_F^2\Big], \\
&= \frac{1}{nd}\,\mathbb{E}\Big[\big\|\Theta - (\Theta Z_p + W_p) Z_p^+\big\|_F^2\Big], \\
&= \frac{1}{nd}\,\mathbb{E}\Big[\big\|W_p Z_p^+\big\|_F^2\Big], \\
&= \frac{1}{nd}\,\mathbb{E}\Big[\mathrm{tr}\Big(W_p Z_p^+ \big(W_p Z_p^+\big)^\top\Big)\Big], \\
&= \frac{1}{nd}\,\mathbb{E}\Big[\mathrm{tr}\Big(W_p^\top W_p Z_p^+ \big(Z_p^+\big)^\top\Big)\Big], \\
&= \frac{1}{nd}\,\mathbb{E}\Big[\mathrm{tr}\Big(Z_p^+ \big(Z_p^+\big)^\top W_p^\top W_p\Big)\Big].
\end{aligned}$$

Let $\bar{Z} = Z_p^+ (Z_p^+)^\top$. The expectation of trace is computed as

$$\begin{aligned}
&\mathbb{E}\big[\mathrm{tr}\big(\bar{Z} W_p^\top W_p\big)\big], \\
&= \mathbb{E}\left[\mathrm{tr}\left(\begin{bmatrix} \bar{Z}_{11} & \cdots & \bar{Z}_{1,t_f-t_s} \\ \vdots & \ddots & \vdots \\ \bar{Z}_{t_f-t_s,1} & \cdots & \bar{Z}_{t_f-t_s,t_f-t_s} \end{bmatrix} \begin{bmatrix} w_{t_s}^\top w_{t_s} & \cdots & w_{t_s}^\top w_{t_f-1} \\ \vdots & \ddots & \vdots \\ w_{t_f-1}^\top w_{t_s} & \cdots & w_{t_f-1}^\top w_{t_f-1} \end{bmatrix}\right)\right], \\
&= \mathbb{E}\Big[\Big(\bar{Z}_{11} w_{t_s}^\top w_{t_s} + \cdots + \bar{Z}_{1,t_f-t_s} w_{t_f-1}^\top w_{t_s}\Big) + \cdots \\
&\quad + \Big(\bar{Z}_{t_f-t_s,1} w_{t_s}^\top w_{t_f-1} + \cdots + \bar{Z}_{t_f-t_s,t_f-t_s} w_{t_f-1}^\top w_{t_f-1}\Big)\Big], \\
&= \mathbb{E}\left[\left(\sum_{k=1}^{t_f-t_s} \bar{Z}_{1k} w_{t_s-1+k}^\top w_{t_s}\right) + \cdots + \left(\sum_{k=1}^{t_f-t_s} \bar{Z}_{t_f-t_s,k} w_{t_s-1+k}^\top w_{t_f-1}\right)\right], \\
&= \mathbb{E}\left[\sum_{j=1}^{t_f-t_s} \sum_{k=1}^{t_f-t_s} \bar{Z}_{jk} w_{t_s-1+k}^\top w_{t_s-1+j}\right], \\
&= \mathbb{E}\left[\sum_{k=1}^{t_f-t_s} \bar{Z}_{kk} w_{t_s-1+k}^\top w_{t_s-1+k}\right] + \mathbb{E}\left[\sum_{j\neq k} \sum_{k=1}^{t_f-t_s} \bar{Z}_{jk} w_{t_s-1+k}^\top w_{t_s-1+j}\right], \\
&= \mathbb{E}\left[\sum_{k=1}^{t_f-t_s} \bar{Z}_{kk} w_{t_s-1+k}^\top w_{t_s-1+k}\right], \\
&= \mathbb{E}\Big[\mathrm{tr}\Big(\bar{Z}\,\mathrm{diag}\Big(w_{t_s}^\top w_{t_s}, \ldots, w_{t_f-1}^\top w_{t_f-1}\Big)\Big)\Big], \\
&= \mathbb{E}\Big[\mathrm{tr}\Big(\bar{Z}\,\mathrm{diag}\Big(\mathrm{tr}\big(w_{t_s} w_{t_s}^\top\big), \ldots, \mathrm{tr}\Big(w_{t_f-1} w_{t_f-1}^\top\Big)\Big)\Big)\Big],
\end{aligned}$$

$$
\begin{aligned}
&= \mathrm{tr}\Big(\mathbb{E}\big[\bar{Z}\big]\,\mathbb{E}\Big[\mathrm{diag}\Big(\mathrm{tr}\big(w_{t_s}w_{t_s}^\top\big),\dots,\mathrm{tr}\Big(w_{t_f-1}w_{t_f-1}^\top\Big)\Big)\Big]\Big),\\
&= \mathrm{tr}\big(\mathbb{E}\big[\bar{Z}\big]\,\mathrm{diag}(\mathrm{tr}(\Sigma_w),\dots,\mathrm{tr}(\Sigma_w))\big),\\
&= \mathrm{tr}\big(\mathbb{E}\big[\bar{Z}\big]\,\mathrm{tr}(\Sigma_w)I\big),\\
&= \mathrm{tr}(\Sigma_w)\,\mathrm{tr}\Big(\mathbb{E}\Big[Z_p^+\big(Z_p^+\big)^\top\Big]\Big),\\
&= \mathrm{tr}(\Sigma_w)\,\mathrm{tr}\Big(\mathbb{E}\Big[Z_p^\top\big(Z_pZ_p^\top\big)^{-1}\big(Z_pZ_p^\top\big)^{-1}Z_p\Big]\Big),\\
&= \mathrm{tr}(\Sigma_w)\,\mathrm{tr}\Big(\mathbb{E}\Big[\big(Z_pZ_p^\top\big)^{-1}\Big]\Big).
\end{aligned}
$$

Therefore, we obtain the equation in the theorem. □

Moreover, using Lemma 5.2, the theorem below shows a sample identifying complexity of the OLS estimator with respect to an acceptable estimation error.

Theorem 5.5. *Let Θ and $\mathcal{D}$ be as in Definition 4.1, and let Z_p be as in* (5.1)*. Suppose an estimation error is defined as Definition 4.2. The sample identifying complexity of the OLS estimator* (5.3) *with respect to an acceptable estimation error γ_c is given as*

$$
N^*\Big(\gamma_c,\Theta;\hat{\Theta}_{\mathrm{OLS}}\Big)=\arg\min_{|\mathcal{D}|}|\mathcal{D}|\quad s.t.\quad \mathrm{tr}\Big(\mathbb{E}\Big[\big(Z_pZ_p^\top\big)^{-1}\Big]\Big)<nd\,\mathrm{tr}(\Sigma_w)^{-1}\gamma_c,
$$

where Σ_w is defined in (4.1)*.*

Proof. The theorem immediately follows from Definition 4.4 and Lemma 5.2. □

It should be noted that the sample identifying complexity in the theorem cannot be expressed as a closed form due to the expectation of the inverse Gramian. In other words, it is required that the sample identifying complexity is numerically computed by using Monte Carlo methods. The minimum security parameter is given as follows with the sample identifying complexity in Theorem 5.5.

Theorem 5.6. *Let Θ and $\mathcal{D}$ be as in Definition 4.1. Suppose an estimation error is defined as Definition 4.2. The security parameter*

$$
\lambda^*\Big(\gamma_c,\tau_c;\Theta,\hat{\Theta}_{\mathrm{OLS}},\Upsilon\Big)=\left\lfloor\log_2\left(\frac{\Upsilon\tau_c}{N^*\Big(\gamma_c,\Theta;\hat{\Theta}_{\mathrm{OLS}}\Big)}\right)\right\rfloor+1
$$

is the minimum security parameter, such that an encrypted control system with (4.1) *and* (4.2) *is (γ_c,τ_c)-secure under the adversary in Definition 4.1 using the*

OLS estimator (5.3)*, where* Υ *is defined in Definition 4.6, and* N^* *is defined in Theorem 5.5.*

Proof. From Theorem 4.1, the encrypted control system is (γ_c, τ_c)-secure under the adversary in Definition 4.1 using the OLS estimator (5.3) if and only if

$$\tau\Big(N^*\Big(\gamma_c, \Theta; \hat{\Theta}_{\mathrm{OLS}}\Big), \Theta; \Upsilon\Big) > \tau_c,$$

where τ is a sample deciphering time in Definition 4.6. The condition can be transformed as

$$\begin{aligned}
\tau\Big(N^*\Big(\gamma_c, \Theta; \hat{\Theta}_{\mathrm{OLS}}\Big), \Theta; \Upsilon\Big) > \tau_c &\iff \frac{2^\lambda N^*\Big(\gamma_c, \Theta; \hat{\Theta}_{\mathrm{OLS}}\Big)}{\Upsilon} > \tau_c, \\
&\iff 2^\lambda > \frac{\tau_c \Upsilon}{N^*\Big(\gamma_c, \Theta; \hat{\Theta}_{\mathrm{OLS}}\Big)}, \\
&\iff \lambda > \log_2 \left(\frac{\tau_c \Upsilon}{N^*\Big(\gamma_c, \Theta; \hat{\Theta}_{\mathrm{OLS}}\Big)} \right).
\end{aligned}$$

Since a security parameter is a positive integer, the minimum one is given as $\lambda^*(\gamma_c, \tau_c; \Theta, \hat{\Theta}_{\mathrm{OLS}}, \Upsilon)$. □

The security parameter λ^* in the theorem is the solution to Problem 5.1 when an attack target is a plant (4.1).

5.2.2 Closed-loop case

Consider Problem 5.1 when an attack target is a closed-loop system (4.3). Unlike the optimal design in the open-loop case, a feedback gain of a controller (4.2) is a design parameter for the minimum security parameter because a feedback gain can tune a system parameter. The minimum security parameter is computed in the same manner of Theorem 5.6 once a feedback gain maximizing a security parameter is obtained.

It can be seen from Theorem 5.6 that increasing a sample identifying complexity reduces the minimum security parameter. By Definition 4.4, a sample identifying complexity increases as the expectation of estimation error increases. Hence, an optimal controller for designing the minimum security parameter should be designed to maximize the expectation of estimation error. In this light, Lemma 5.2 implies

that a feedback gain of such a controller is given as a solution to the optimization problem

$$\max_{F} \operatorname{tr}\Big(\mathbb{E}\Big[\big(X_p X_p^\top\big)^{-1}\Big]\Big).$$

Note that X_p depends implicitly on a feedback gain F.

Unfortunately, it is difficult to solve the above problem analytically. This thesis compromises the design of an optimal controller and explores a suboptimal controller that increases the trace of the expectation of the inverse Gramian in the later section.

5.3 Suboptimal design

The previous sections have solved Problem 5.1. The minimum security parameter to achieve the desired security level of an encrypted control system is designed as in Theorem 5.6 if an attack target is a plant. Although a sample identifying complexity for the minimum security parameter can be computed using Monte Carlo methods, the computation often takes a long time owing to repeated operations. Furthermore, designing a feedback gain that maximizes a sample identifying complexity is challenging due to the complicated optimization.

This section derives suboptimal solutions to Problem 5.1 to mitigate a computation time for the minimum security parameter design and avoid the difficulty of controller design. The section introduces a lower bound of the expectation of estimation error in each case where an attack target is a plant or a closed-loop system and designs a security parameter and controller based on the lower bound.

5.3.1 Open-loop case

A lower bound of the expectation of estimation error in Definition 4.2 under the OLS estimator (5.3) is given as follows if an attack target is a plant (4.1).

Lemma 5.3. *Consider the OLS estimator* (5.3)*. If* $\Theta = [A\ B]$*, the expectation of estimation error in Definition 4.2 is bounded from below by*

$$\begin{aligned}&\mathbb{E}\Big[\varepsilon\Big(\begin{bmatrix}A & B\end{bmatrix}, \hat{\Theta}_{\mathrm{OLS}}(\mathcal{D})\Big)\Big]\\ &\geq \gamma\Big(|\mathcal{D}|, \begin{bmatrix}A & B\end{bmatrix}\Big) \coloneqq \frac{m+n}{n}\frac{\operatorname{tr}(\Sigma_w)}{\operatorname{tr}(\Psi_x) + (|\mathcal{D}|-1)[\operatorname{tr}(\Sigma_u)+\operatorname{tr}(\Psi_u)+\operatorname{tr}(\Psi_w)]},\end{aligned} \tag{5.6}$$

where

$$\Psi_x \coloneqq \sum_{k=0}^{\infty} A^k \Sigma_x \big(A^k\big)^\top, \quad \Psi_u \coloneqq \sum_{k=0}^{\infty} A^k B \Sigma_u B^\top \big(A^k\big)^\top, \quad \Psi_w \coloneqq \sum_{k=0}^{\infty} A^k \Sigma_w \big(A^k\big)^\top,$$

Σ_x *and* Σ_w *are defined in* (4.1)*, and* Σ_u *is defined in Definition 4.1.*

Proof. If $\Theta = [A\ B]$, Z_p and d are given from (5.1) as $Z_p = [X_p^\top\ U_p^\top]^\top$ and $d = m+n$. It holds from Lemma 5.2 that

$$\mathbb{E}\Big[\varepsilon\Big(\begin{bmatrix} A & B \end{bmatrix}, \hat{\Theta}_{\mathrm{OLS}}(\mathcal{D})\Big)\Big] = \frac{1}{n(m+n)} \operatorname{tr}(\Sigma_w) \operatorname{tr}\left(\mathbb{E}\left[\left(\begin{bmatrix} X_p \\ U_p \end{bmatrix}\begin{bmatrix} X_p^\top & U_p^\top \end{bmatrix}\right)^{-1}\right]\right).$$

Let $\lambda_i(M)$ be the ith eiganvalue of a matrix M. Jensen's inequality yields that

$$\begin{aligned}
\operatorname{tr}\left(\left(\begin{bmatrix} X_p \\ U_p \end{bmatrix}\begin{bmatrix} X_p^\top & U_p^\top \end{bmatrix}\right)^{-1}\right) &= \sum_{i=1}^{m+n} \lambda_i\left(\left(\begin{bmatrix} X_p \\ U_p \end{bmatrix}\begin{bmatrix} X_p^\top & U_p^\top \end{bmatrix}\right)^{-1}\right), \\
&= \sum_{i=1}^{m+n} \lambda_i\left(\begin{bmatrix} X_p \\ U_p \end{bmatrix}\begin{bmatrix} X_p^\top & U_p^\top \end{bmatrix}\right)^{-1}, \\
&= (m+n)\sum_{i=1}^{m+n} \frac{1}{m+n}\lambda_i\left(\begin{bmatrix} X_p \\ U_p \end{bmatrix}\begin{bmatrix} X_p^\top & U_p^\top \end{bmatrix}\right)^{-1}, \\
&\geq (m+n)\left(\sum_{i=1}^{m+n} \frac{1}{m+n}\lambda_i\left(\begin{bmatrix} X_p \\ U_p \end{bmatrix}\begin{bmatrix} X_p^\top & U_p^\top \end{bmatrix}\right)\right)^{-1}, \\
&= (m+n)^2\left(\sum_{i=1}^{m+n} \lambda_i\left(\begin{bmatrix} X_p \\ U_p \end{bmatrix}\begin{bmatrix} X_p^\top & U_p^\top \end{bmatrix}\right)\right)^{-1}, \\
&= (m+n)^2 \operatorname{tr}\left(\begin{bmatrix} X_p \\ U_p \end{bmatrix}\begin{bmatrix} X_p^\top & U_p^\top \end{bmatrix}\right)^{-1}.
\end{aligned}$$

Hence, we obtain

$$\begin{aligned}
\operatorname{tr}\left(\mathbb{E}\left[\left(\begin{bmatrix} X_p \\ U_p \end{bmatrix}\begin{bmatrix} X_p^\top & U_p^\top \end{bmatrix}\right)^{-1}\right]\right) &= \mathbb{E}\left[\operatorname{tr}\left(\left(\begin{bmatrix} X_p \\ U_p \end{bmatrix}\begin{bmatrix} X_p^\top & U_p^\top \end{bmatrix}\right)^{-1}\right)\right], \\
&\geq (m+n)^2\,\mathbb{E}\left[\operatorname{tr}\left(\begin{bmatrix} X_p \\ U_p \end{bmatrix}\begin{bmatrix} X_p^\top & U_p^\top \end{bmatrix}\right)^{-1}\right],
\end{aligned}$$

$$
\begin{aligned}
&\geq (m+n)^2\,\mathbb{E}\left[\mathrm{tr}\left(\begin{bmatrix}X_p\\U_p\end{bmatrix}\begin{bmatrix}X_p^\top & U_p^\top\end{bmatrix}\right)\right]^{-1},\\
&= (m+n)^2\,\mathbb{E}\left[\mathrm{tr}\left(\begin{bmatrix}X_pX_p^\top & X_pU_p^\top\\U_pX_p^\top & U_pU_p^\top\end{bmatrix}\right)\right]^{-1},\\
&= (m+n)^2\big(\mathbb{E}\big[\mathrm{tr}\big(X_pX_p^\top\big)\big]+\mathbb{E}\big[\mathrm{tr}\big(U_pU_p^\top\big)\big]\big)^{-1},
\end{aligned}
$$

where the third inequality also follows from Jensen's inequality. Moreover, the expectations of traces are computed from (4.1) as

$$
\begin{aligned}
&\mathbb{E}\big[\mathrm{tr}\big(X_pX_p^\top\big)\big]\\
&= \mathbb{E}\left[\mathrm{tr}\left(\sum_{t=t_s}^{t_f-1}x_tx_t^\top\right)\right],\\
&= \mathbb{E}\left[\sum_{t=t_s}^{t_f-1}\mathrm{tr}\left(\left(A^tx_0+\sum_{k=0}^{t-1}A^kBu_{t-1-k}+\sum_{k=0}^{t-1}A^kBw_{t-1-k}\right)\right.\right.\\
&\quad\left.\left.\left(A^tx_0+\sum_{k=0}^{t-1}A^kBu_{t-1-k}+\sum_{k=0}^{t-1}A^kBw_{t-1-k}\right)^\top\right)\right],\\
&= \mathbb{E}\left[\sum_{t=t_s}^{t_f-1}\mathrm{tr}\left(A^tx_0x_0^\top\big(A^t\big)^\top+\sum_{k=0}^{t-1}A^kBu_{t-1-k}u_{t-1-k}^\top B^\top\big(A^k\big)^\top\right.\right.\\
&\quad\left.\left.+\sum_{k=0}^{t-1}A^kBw_{t-1-k}w_{t-1-k}^\top B^\top\big(A^k\big)^\top\right)\right],\\
&= \sum_{t=t_s}^{t_f-1}\mathrm{tr}\left(A^t\,\mathbb{E}\big[x_0x_0^\top\big]\big(A^t\big)^\top+\sum_{k=0}^{t-1}A^kB\,\mathbb{E}\big[u_{t-1-k}u_{t-1-k}^\top\big]B^\top\big(A^k\big)^\top\right.\\
&\quad\left.+\sum_{k=0}^{t-1}A^kB\,\mathbb{E}\big[w_{t-1-k}w_{t-1-k}^\top\big]B^\top\big(A^k\big)^\top\right),\\
&= \sum_{t=t_s}^{t_f-1}\mathrm{tr}\left(A^t\Sigma_x\big(A^t\big)^\top+\sum_{k=0}^{t-1}A^kB\Sigma_uB^\top\big(A^k\big)^\top+\sum_{k=0}^{t-1}A^kB\Sigma_wB^\top\big(A^k\big)^\top\right),\\
&\leq \sum_{t=t_s}^{t_f-1}\mathrm{tr}\left(A^t\Sigma_x\big(A^t\big)^\top+\sum_{k=0}^{\infty}A^kB\Sigma_uB^\top\big(A^k\big)^\top+\sum_{k=0}^{\infty}A^kB\Sigma_wB^\top\big(A^k\big)^\top\right),\\
&= \mathrm{tr}\left(\sum_{t=t_s}^{t_f-1}A^t\Sigma_x\big(A^t\big)^\top+\sum_{t=t_s}^{t_f-1}\Psi_u+\sum_{t=t_s}^{t_f-1}\Psi_w\right),
\end{aligned}
$$

$$
\begin{aligned}
&\leq \operatorname{tr}\left(\sum_{t=0}^{\infty} A^t \Sigma_x \big(A^t\big)^\top + (|\mathcal{D}|-1)\Psi_u + (|\mathcal{D}|-1)\Psi_w\right), \\
&= \operatorname{tr}(\Psi_x) + (|\mathcal{D}|-1)\operatorname{tr}(\Psi_u) + (|\mathcal{D}|-1)\operatorname{tr}(\Psi_w),
\end{aligned}
$$

and

$$
\mathbb{E}\big[\operatorname{tr}\big(U_p U_p^\top\big)\big] = \mathbb{E}\left[\operatorname{tr}\left(\sum_{t=t_s}^{t_f-1} u_t u_t^\top\right)\right] = \sum_{t=t_s}^{t_f-1} \operatorname{tr}\big(\mathbb{E}\big[u_t u_t^\top\big]\big) = (|\mathcal{D}|-1)\operatorname{tr}(\Sigma_u).
$$

Therefore, $\mathbb{E}[\operatorname{tr}(X_p X_p^\top)] + \mathbb{E}[\operatorname{tr}(U_p U_p^\top)]$ is bounded from above by

$$
\begin{aligned}
&\mathbb{E}\big[\operatorname{tr}\big(X_p X_p^\top\big)\big] + \mathbb{E}\big[\operatorname{tr}\big(U_p U_p^\top\big)\big] \\
&\leq \operatorname{tr}(\Psi_x) + (|\mathcal{D}|-1)\operatorname{tr}(\Psi_u) + (|\mathcal{D}|-1)\operatorname{tr}(\Psi_w) + (|\mathcal{D}|-1)\operatorname{tr}(\Sigma_u), \\
&= \operatorname{tr}(\Psi_x) + (|\mathcal{D}|-1)[\operatorname{tr}(\Sigma_u) + \operatorname{tr}(\Psi_u) + \operatorname{tr}(\Psi_w)].
\end{aligned}
$$

Consequently, we obtain

$$
\begin{aligned}
&\mathbb{E}\Big[\varepsilon\Big(\begin{bmatrix} A & B \end{bmatrix}, \hat{\Theta}_{\mathrm{OLS}}(\mathcal{D})\Big)\Big] \\
&\geq \frac{1}{n(m+n)} \operatorname{tr}(\Sigma_w) \cdot (m+n)^2 \cdot \frac{1}{\operatorname{tr}(\Psi_x) + (|\mathcal{D}|-1)[\operatorname{tr}(\Sigma_u) + \operatorname{tr}(\Psi_u) + \operatorname{tr}(\Psi_w)]}, \\
&= \frac{m+n}{n} \frac{\operatorname{tr}(\Sigma_w)}{\operatorname{tr}(\Psi_x) + (|\mathcal{D}|-1)[\operatorname{tr}(\Sigma_u) + \operatorname{tr}(\Psi_u) + \operatorname{tr}(\Psi_w)]}.
\end{aligned}
$$

This completes the proof. □

The lemma shows that the convergence rate of the lower bound is $1/|\mathcal{D}|$ and suggests that the expectation of estimation error relates to the variances Σ_u and Σ_w and the weighted controllability Gramians Ψ_x, Ψ_u, and Ψ_w. The lower bound increases and decreases as the traces of noise and input variances increase, respectively. The traces of the variances are equivalent to the signal powers of noise and input, and the parameter estimation accuracy generally depends on a ratio of the powers. Moreover, the lower bound increases as the traces of the Gramians decrease. Eigenvalues of a controllability Gramian represent the ease of state excitation in response to external inputs. The more excited the system is, the easier parameter estimation becomes. Therefore, the lower bound captures the properties of parameter estimation and seems an effective measure to evaluate the expectation of estimation error, namely the difficulty of parameter estimation.

A lower bound of a sample identifying complexity is obtained using the lower bound of the expectation of estimation error.

Theorem 5.7. *Suppose $\Theta = [A\ B]$, and an estimation error is defined as Definition 4.2. The sample size*

$$N^{**}\left(\gamma_c, \begin{bmatrix} A & B \end{bmatrix}; \hat{\Theta}_{\mathrm{OLS}}\right) = \max\left\{2, \left\lfloor \frac{\frac{m+n}{n}\gamma_c^{-1}\operatorname{tr}(\Sigma_w) - \operatorname{tr}(\Psi_x)}{\operatorname{tr}(\Sigma_u) + \operatorname{tr}(\Psi_u) + \operatorname{tr}(\Psi_w)} \right\rfloor + 2\right\} \tag{5.7}$$

is a lower bound of the sample identifying complexity of the OLS estimator (5.3) *with respect to γ_c, where Σ_w is defined in* (4.1)*, Σ_u is defined in Definition 4.1, and Ψ_x, Ψ_u, and Ψ_w are defined in Lemma 5.3.*

Proof. From Lemma 5.3, a lower bound of the sample identifying complexity with respect to γ_c is given as the minimum sample size such that $\gamma(|\mathcal{D}|, [A\ B]) < \gamma_c$. It follows from (5.6) that

$$\begin{aligned}
&\gamma\left(|\mathcal{D}|, \begin{bmatrix} A & B \end{bmatrix}\right) < \gamma_c, \\
&\iff \frac{m+n}{n}\frac{\operatorname{tr}(\Sigma_w)}{\operatorname{tr}(\Psi_x) + (|\mathcal{D}|-1)[\operatorname{tr}(\Sigma_u) + \operatorname{tr}(\Psi_u) + \operatorname{tr}(\Psi_w)]} < \gamma_c, \\
&\iff |\mathcal{D}| > \frac{\frac{m+n}{n}\gamma_c^{-1}\operatorname{tr}(\Sigma_w) - \operatorname{tr}(\Psi_x)}{\operatorname{tr}(\Sigma_u) + \operatorname{tr}(\Psi_u) + \operatorname{tr}(\Psi_w)} + 1.
\end{aligned}$$

Since a sample size $|\mathcal{D}|$ is a positive integer larger than or equal to two, the minimum one is given as (5.6). □

A suboptimal security parameter can be computed based on the lower bound of a sample identifying complexity alike with Theorem 5.6.

Theorem 5.8. *Suppose $\Theta = [A\ B]$, and an estimation error is defined as Definition 4.2. The security parameter*

$$\lambda^{**} = \left\lfloor \log_2\left(\frac{\Upsilon_{\tau_c}}{N^{**}\left(\gamma_c, \begin{bmatrix} A & B \end{bmatrix}; \hat{\Theta}_{\mathrm{OLS}}\right)}\right)\right\rfloor + 1 \tag{5.8}$$

is the minimum security parameter with respect to (5.7)*, such that an encrypted control system with* (4.1) *and* (4.2) *is (γ_c, τ_c)-secure under the adversary in Defini-*

tion 4.1 using the OLS estimator (5.3)*, where* Υ *is defined in Definition 4.6, and* N^{**} *is defined in* (5.7)*.*

Proof. The theorem follows the same manner as the proof of Theorem 5.6. □

Although the suboptimal security parameter has the same form as the optimal security parameter in Theorem 5.6, the suboptimal one is an explicit form of system and design parameters. Note that the weighted controllability Gramians can be obtained by solving corresponding discrete Lyapunov equations. For instance, Ψ_x is unique and satisfies $A\Psi_x A^\top - \Psi_x + \Sigma_x = O$. Consequently, the computation costs of the suboptimal security parameter can be less than the optimal one.

5.3.2 Closed-loop case

We have seen in Section 5.2.2 that the minimum security parameter cannot be designed analytically due to the difficulty of finding an appropriate controller when an attack target is a closed-loop system (4.3). This section derives a suboptimal security parameter based on the following lower bound of the expectation of estimation error as with the open-loop case in Section 5.3.1.

Lemma 5.4. *Consider the OLS estimator* (5.3)*. Let* $\mathcal{D}$ *be as in Definition 4.1. If* $\Theta = \bar{A}$*, the expectation of estimation error in Definition 4.2 is bounded from below by*

$$\mathbb{E}\Big[\varepsilon\Big(\bar{A}, \hat{\Theta}_{\mathrm{OLS}}(\mathcal{D})\Big)\Big] \geq \gamma\big(|\mathcal{D}|, \bar{A}\big) \coloneqq \frac{\mathrm{tr}(\Sigma_w)}{\mathrm{tr}\big(\bar{\Psi}_x\big) + (|\mathcal{D}| - 1)\,\mathrm{tr}\big(\bar{\Psi}_w\big)}, \tag{5.9}$$

where

$$\bar{\Psi}_x = \bar{\Psi}_x(F) \coloneqq \sum_{k=0}^{\infty} \bar{A}^k \Sigma_x \big(\bar{A}^k\big)^\top, \quad \bar{\Psi}_w = \bar{\Psi}_w(F) \coloneqq \sum_{k=0}^{\infty} \bar{A}^k \Sigma_w \big(\bar{A}^k\big)^\top,$$

and Σ_x *and* Σ_w *are defined in* (4.1)*.*

Proof. If $\Theta = \bar{A}$, Z_p and d are given from (5.1) as $Z_p = X_p$ and $d = n$. It holds from Lemma 5.2 that

$$\mathbb{E}\Big[\varepsilon\Big(\bar{A}, \hat{\Theta}_{\mathrm{OLS}}(\mathcal{D})\Big)\Big] = \frac{1}{n^2}\,\mathrm{tr}(\Sigma_w)\,\mathrm{tr}\Big(\mathbb{E}\Big[\big(X_p X_p^\top\big)^{-1}\Big]\Big).$$

Similar to the proof of Lemma 5.3, it follows that

$$\mathrm{tr}\Big(\mathbb{E}\Big[\big(X_p X_p^\top\big)^{-1}\Big]\Big) \geq n^2\,\mathbb{E}\big[\mathrm{tr}\big(X_p X_p^\top\big)\big]^{-1}$$

Furthermore, the expectation of trace is computed from (4.3) as

$$
\begin{aligned}
&\mathbb{E}\big[\mathrm{tr}\big(X_p X_p^\top\big)\big] \\
&= \mathbb{E}\left[\mathrm{tr}\left(\sum_{t=t_s}^{t_f-1} x_t x_t^\top\right)\right], \\
&= \mathbb{E}\left[\sum_{t=t_s}^{t_f-1} \mathrm{tr}\left(\left(\bar{A}^t x_0 + \sum_{k=0}^{t-1} \bar{A}^k w_{t-1-k}\right)\left(\bar{A}^t x_0 + \sum_{k=0}^{t-1} \bar{A}^k w_{t-1-k}\right)^\top\right)\right], \\
&= \mathbb{E}\left[\sum_{t=t_s}^{t_f-1} \mathrm{tr}\left(\bar{A}^t x_0 x_0^\top \big(\bar{A}^t\big)^\top + \sum_{k=0}^{t-1} \bar{A}^k w_{t-1-k} w_{t-1-k}^\top \big(\bar{A}^k\big)^\top\right)\right], \\
&= \sum_{t=t_s}^{t_f-1} \mathrm{tr}\left(\bar{A}^t\, \mathbb{E}\big[x_0 x_0^\top\big] \big(\bar{A}^t\big)^\top + \sum_{k=0}^{t-1} \bar{A}^k\, \mathbb{E}\big[w_{t-1-k} w_{t-1-k}^\top\big] \big(\bar{A}^k\big)^\top\right), \\
&= \sum_{t=t_s}^{t_f-1} \mathrm{tr}\left(\bar{A}^t \Sigma_x \big(\bar{A}^t\big)^\top + \sum_{k=0}^{t-1} \bar{A}^k \Sigma_w \big(\bar{A}^k\big)^\top\right), \\
&\leq \sum_{t=t_s}^{t_f-1} \mathrm{tr}\left(\bar{A}^t \Sigma_x \big(\bar{A}^t\big)^\top + \sum_{k=0}^{\infty} \bar{A}^k \Sigma_w \big(\bar{A}^k\big)^\top\right), \\
&= \mathrm{tr}\left(\sum_{t=t_s}^{t_f-1} \bar{A}^t \Sigma_x \big(\bar{A}^t\big)^\top + \sum_{t=t_s}^{t_f-1} \bar{\Psi}_w\right), \\
&\leq \mathrm{tr}\left(\sum_{t=0}^{\infty} \bar{A}^t \Sigma_x \big(\bar{A}^t\big)^\top + (|\mathcal{D}|-1)\bar{\Psi}_w\right), \\
&= \mathrm{tr}\big(\bar{\Psi}_x\big) + (|\mathcal{D}|-1)\,\mathrm{tr}\big(\bar{\Psi}_w\big).
\end{aligned}
$$

Therefore, we obtain

$$
\begin{aligned}
\mathbb{E}\Big[\varepsilon\Big(\bar{A}, \hat{\Theta}_{\mathrm{OLS}}(\mathcal{D})\Big)\Big] &\geq \frac{1}{n^2}\,\mathrm{tr}(\Sigma_w) \cdot n^2 \cdot \frac{1}{\mathrm{tr}\big(\bar{\Psi}_x\big) + (|\mathcal{D}|-1)\,\mathrm{tr}\big(\bar{\Psi}_w\big)}, \\
&= \frac{\mathrm{tr}(\Sigma_w)}{\mathrm{tr}\big(\bar{\Psi}_x\big) + (|\mathcal{D}|-1)\,\mathrm{tr}\big(\bar{\Psi}_w\big)}.
\end{aligned}
$$

This completes the proof. □

The lower bound in the lemma can be further simplified when a sample size is sufficiently large.

Corollary 5.5. *Let $\mathcal{D}$ be as in Definition 4.1. γ in Theorem 5.4 holds*

$$\gamma\big(|\mathcal{D}|, \bar{A}\big) = \hat{\gamma}\big(|\mathcal{D}|, \bar{A}\big)(1 + o(1)), \quad \hat{\gamma}\big(|\mathcal{D}|, \bar{A}\big) \coloneqq \frac{\operatorname{tr}(\Sigma_w)}{(|\mathcal{D}| - 1)\operatorname{tr}\big(\bar{\Psi}_w\big)} \tag{5.10}$$

as $|\mathcal{D}| \to \infty$, where $o(\cdot)$ is the little-o notation, Σ_w is defined in (4.1), and $\bar{\Psi}_w$ is defined in Lemma 5.4.

Proof. It follows from Lemma 5.4 that

$$\begin{aligned}
\lim_{|\mathcal{D}|\to\infty} \frac{\gamma\big(|\mathcal{D}|, \bar{A}\big)}{\hat{\gamma}\big(|\mathcal{D}|, \bar{A}\big)} &= \lim_{|\mathcal{D}|\to\infty} \frac{(|\mathcal{D}| - 1)\operatorname{tr}\big(\bar{\Psi}_w\big)}{\operatorname{tr}\big(\bar{\Psi}_x\big) + (|\mathcal{D}| - 1)\operatorname{tr}\big(\bar{\Psi}_w\big)}, \\
&= \lim_{|\mathcal{D}|\to\infty} \frac{\operatorname{tr}\big(\bar{\Psi}_w\big)}{(|\mathcal{D}| - 1)^{-1}\operatorname{tr}\big(\bar{\Psi}_x\big) + \operatorname{tr}\big(\bar{\Psi}_w\big)}, \\
&= 1.
\end{aligned}$$

Therefore, we obtain

$$\frac{\gamma\big(|\mathcal{D}|, \bar{A}\big)}{\hat{\gamma}\big(|\mathcal{D}|, \bar{A}\big)} - 1 = o(1) \iff \gamma\big(|\mathcal{D}|, \bar{A}\big) = \hat{\gamma}\big(|\mathcal{D}|, \bar{A}\big)(1 + o(1))$$

by definition of little-o notation. □

Lemma 5.4 suggests that the expectation of estimation error relies on the variance Σ_w and the weighted controllability Gramians $\bar{\Psi}_x$ and $\bar{\Psi}_w$ even when an attack target is a closed-loop system. Corollary 5.5 implies that the lower bound (5.9) approaches (5.10) as a sample size increases. Thus, the expectation of estimation error is considered to mainly depend on Σ_w and $\bar{\Psi}_w$ in practice. From these results, a suboptimal controller should be designed to minimize the trace of $\bar{\Psi}_w$ for reducing a security parameter required for a desired security level. The theorem below shows that such a controller is an optimal H_2 controller for a fictitious system whose system and input matrices correspond to $\bar{A}$ and Σ_w, respectively.

Theorem 5.9. *Let $\mathcal{D}$ be as in Definition 4.1. Suppose $(\eta^{**}, P^{**}, Q^{**}) \in \mathbb{R} \times \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m}$ is a solution to the problem*

$$\min_{\eta,P,Q} \eta \quad s.t. \quad \eta > \operatorname{tr}(P),\ P > 0,\ \begin{bmatrix} P & AP + BQ & \Sigma_w^{1/2} \\ (AP + BQ)^\top & P & O \\ \Sigma_w^{1/2} & O & I \end{bmatrix} > 0,$$

where $\Sigma_w = \Sigma_w^{1/2}\Sigma_w^{1/2}$*, and* Σ_w *is defined in* (4.1)*. The controller* (4.2) *with the feedback gain*

$$F^{**} = Q^{**}(P^{**})^{-1} \tag{5.11}$$

stabilizes (4.3) *and maximizes* $\hat{\gamma}$ *in* (5.10)*, thereby maximizing* γ *in* (5.9) *as* $|\mathcal{D}| \to \infty$*.*

Proof. From Corollary 5.5, γ in (5.9) is maximized when maximizing $\hat{\gamma}$ in (5.10) as $|\mathcal{D}| \to \infty$. The parameter of $\hat{\gamma}$ depending on a feedback gain F is the Gramian $\bar{\Psi}_w = \bar{\Psi}_w(F)$ only. Hence, a controller minimizing $\mathrm{tr}(\bar{\Psi}_w)$ maximizes $\hat{\gamma}$.

Now, consider the fictitious system

$$G : \begin{cases} \tilde{x}_{t+1} = \bar{A}\tilde{x}_t + \Sigma_w^{1/2}\tilde{u}_t, \\ \tilde{y}_t = \tilde{x}_t, \end{cases}$$

where $\tilde{x} \in \mathbb{R}^n$ is a fictirious state, $\tilde{u} \in \mathbb{R}^m$ is a fictirious input, and $\tilde{y} \in \mathbb{R}^n$ is a fictirious output. Note that $\Sigma_w^{1/2}$ always exists because Σ_w is positive definite. From Parseval's theorem, it follows that

$$\|G\|_{H_2} = \sqrt{\sum_{t=-\infty}^{\infty} \mathrm{tr}(h_t h_t^\top)} = \sqrt{\mathrm{tr}\big(\bar{\Psi}_w\big)}, \quad h_t = \begin{cases} \bar{A}^t\Sigma_w^{1/2}, & t \geq 0, \\ 0, & t < 0, \end{cases}$$

where $\|\cdot\|_{H_2}$ is the H_2 norm, and h_t is the impulse response of G. Therefore, a controller minimizing $\|G\|_{H_2}^2$ maximizes $\hat{\gamma}$.

Let $\eta \in \mathbb{R}^+$. Suppose G is stable, i.e., $\bar{A}$ is Schur. It is well known that $\|G\|_{H_2}^2 < \eta$ holds if and only if there exists a positive definite matrix P such that $\mathrm{tr}(P) < \eta$ and $\bar{A}P\bar{A}^\top - P + \Sigma_w < 0$ [217, 218]. It follows that

$$\begin{aligned}
\bar{A}P\bar{A}^\top - P + \Sigma_w < 0 &\iff P - \Sigma_w - \bar{A}P\bar{A}^\top > 0, \\
&\iff (P - \Sigma_w) - \big(\bar{A}P\big)P^{-1}\big(\bar{A}P\big)^\top > 0, \\
&\iff \begin{bmatrix} P - \Sigma_w & \bar{A}P \\ \big(\bar{A}P\big)^\top & P \end{bmatrix} > 0, \\
&\iff \begin{bmatrix} P & \bar{A}P \\ \big(\bar{A}P\big)^\top & P \end{bmatrix} - \begin{bmatrix} \Sigma_w & O \\ O & O \end{bmatrix} > 0, \\
&\iff \begin{bmatrix} P & \bar{A}P \\ \big(\bar{A}P\big)^\top & P \end{bmatrix} - \begin{bmatrix} \Sigma_w^{1/2} \\ O \end{bmatrix} I^{-1} \begin{bmatrix} \Sigma_w^{1/2} & O \end{bmatrix} > 0,
\end{aligned}$$

$$
\begin{aligned}
&\iff \begin{bmatrix} P & \bar{A}P & \Sigma_w^{1/2} \\ (\bar{A}P)^\top & P & O \\ \Sigma_w^{1/2} & O & I \end{bmatrix} > 0, \\
&\iff \begin{bmatrix} P & (A+BF)P & \Sigma_w^{1/2} \\ ((A+BF)P)^\top & P & O \\ \Sigma_w^{1/2} & O & I \end{bmatrix} > 0, \\
&\iff \begin{bmatrix} P & AP+BQ & \Sigma_w^{1/2} \\ (AP+BQ)^\top & P & O \\ \Sigma_w^{1/2} & O & I \end{bmatrix} > 0,
\end{aligned}
$$

where $Q = FP$, and the third and sixth transformations follow from the property of Schur complement, namely

$$
\begin{bmatrix} X & Y \\ Y^\top & Z \end{bmatrix} > 0 \iff Z > 0 \ \wedge \ X - YZ^{-1}Y^\top > 0
$$

for block matrices X, Y, and Z. Consequently, $\|G\|_{H_2}^2$ is minimized by solving the problem in the theorem, and (5.11) holds by definition of Q. □

Suppose $\tilde{u}_t$ of the fictitious system G in the above proof follows the Gaussian distribution $\mathcal{N}(0, I)$. It follows that

$$
\begin{aligned}
&\lim_{t\to\infty} \mathbb{E}\big[\mathrm{tr}\big(\tilde{x}_t\tilde{x}_t^\top\big)\big], \\
&= \lim_{t\to\infty} \mathbb{E}\left[\mathrm{tr}\left(\bar{A}^t\tilde{x}_0\tilde{x}_0^\top\big(\bar{A}^t\big)^\top + \sum_{k=0}^{t-1} \bar{A}^{t-1-k}\Sigma_w^{1/2}\tilde{u}_k\tilde{u}_k^\top\big(\Sigma_w^{1/2}\big)^\top\big(\bar{A}^{t-1-k}\big)^\top\right)\right], \\
&= \lim_{t\to\infty} \mathrm{tr}\left(\bar{A}^t\tilde{x}_0\tilde{x}_0^\top\big(\bar{A}^t\big)^\top + \sum_{k=0}^{t-1} \bar{A}^{t-1-k}\Sigma_w^{1/2}\,\mathbb{E}\big[\tilde{u}_k\tilde{u}_k^\top\big]\big(\Sigma_w^{1/2}\big)^\top\big(\bar{A}^{t-1-k}\big)^\top\right), \\
&= \lim_{t\to\infty} \mathrm{tr}\left(\bar{A}^t\tilde{x}_0\tilde{x}_0^\top\big(\bar{A}^t\big)^\top + \sum_{k=0}^{t-1} \bar{A}^{t-1-k}\Sigma_w\big(\bar{A}^{t-1-k}\big)^\top\right), \\
&= \lim_{t\to\infty} \mathrm{tr}\left(\bar{A}^t\tilde{x}_0\tilde{x}_0^\top\big(\bar{A}^t\big)^\top + \sum_{k=0}^{t-1} \bar{A}^{k}\Sigma_w\big(\bar{A}^{k}\big)^\top\right), \\
&= \mathrm{tr}\left(\sum_{k=0}^{\infty} \bar{A}^{k}\Sigma_w\big(\bar{A}^{k}\big)^\top\right) = \mathrm{tr}\big(\bar{\Psi}_w\big) = \|G\|_{H_2}^2,
\end{aligned}
$$

where $\tilde{x}_t = \bar{A}^t\tilde{x}_0 + \sum_{k=0}^{t-1} \bar{A}^{t-1-k}\Sigma_w^{1/2}\tilde{u}_t$. Let $\tilde{v}_t = \Sigma_w^{1/2}\tilde{u}_t$ be an affine transformation

of $\tilde{u}_t$. The fictitious system G is then equivalent to the closed-loop system (4.3) because $\tilde{v}_t$ follows the Gaussian distribution $\mathcal{N}(0, \Sigma_w)$. Therefore, the suboptimal controller (5.11) can be interpreted as attenuating a steady-state variance of (4.3) to the noise w. In other words, the controller makes parameter estimation difficult by reducing the sensitivity of (4.3) to the external input.

With the suboptimal controller, a lower bound of a sample identifying complexity is obtained as follows.

Theorem 5.10. *Suppose $\Theta = \bar{A}$, and an estimation error is defined as Definition 4.2. The sample size*

$$N^{**}\left(\gamma_c, \bar{A}; \hat{\Theta}_{\mathrm{OLS}}\right) = \max\left\{2, \left\lfloor \frac{\gamma_c^{-1}\,\mathrm{tr}(\Sigma_w) - \mathrm{tr}\left(\bar{\Psi}_x^{**}\right)}{\mathrm{tr}\left(\bar{\Psi}_w^{**}\right)} \right\rfloor + 2\right\} \tag{5.12}$$

is a lower bound of the sample identifying complexity of the OLS estimator (5.3) *with respect to γ_c, where Σ_w is defined in* (4.1)*, $\bar{\Psi}_x^{**} = \bar{\Psi}_x(F^{**})$ and $\bar{\Psi}_w^{**} = \bar{\Psi}_w(F^{**})$ are defined in Lemma 5.4, and F^{**} is defined in* (5.11)*.*

Proof. The theorem follows the same manner as the proof of Theorem 5.7. □

Moreover, a suboptimal security parameter is given as follows.

Theorem 5.11. *Suppose $\Theta = \bar{A}$, and an estimation error is defined as Definition 4.2. The security parameter*

$$\lambda^{**} = \left\lfloor \log_2 \left(\frac{\Upsilon \tau_c}{N^{**}\left(\gamma_c, \bar{A}; \hat{\Theta}_{\mathrm{OLS}}\right)} \right) \right\rfloor + 1 \tag{5.13}$$

is the minimum security parameter with respect to (5.12)*, such that an encrypted control system with* (4.1) *and* (4.2) *is (γ_c, τ_c)-secure under the adversary in Definition 4.1 using the OLS estimator* (5.3)*, where Υ is defined in Definition 4.6, and N^{**} is defined in* (5.12)*.*

Proof. The theorem follows the same manner as the proof of Theorem 5.8. □

It should be noted that parameters in Definition 2.10 and Definition 2.11 must be chosen appropriately to implement the (updatable and key-updatable) ElGamal and Regev encryption cryptosystems that satisfy λ-bit security. In the case of the ElGamal-based schemes, a prime number q is chosen as an ℓ-bit prime number,

where ℓ is called a key length. The minimum key length to achieve λ-bit security can be computed as

$$\ell^* = \arg\min_{\ell \in \mathbb{N}} \Omega(\ell) \quad \text{s.t.} \quad \Omega(\ell) \geq 2^{\lambda^*},$$

where $\Omega(\ell)$ is the time complexity of the known fastest algorithm for breaking the encryption scheme. Furthermore, the parameters m, n, t, q, and σ in the Regev-based schemes can be selected by using the lattice-estimator [210]. Therefore, the cryptosystems can be effectively implemented utilizing the optimal and suboptimal security parameters.

5.4 Numerical example

This section investigates our design of encrypted control systems through numerical examples. Consider the quadruple-tank process in [219],

$$\begin{aligned}
\frac{dh_1}{dt} &= -\frac{a_1}{A_1}\sqrt{2gh_1} + \frac{a_3}{A_1}\sqrt{2gh_3} + \frac{\gamma_1 k_1}{A_1}v_1, \\
\frac{dh_2}{dt} &= -\frac{a_2}{A_2}\sqrt{2gh_2} + \frac{a_4}{A_2}\sqrt{2gh_4} + \frac{\gamma_2 k_2}{A_2}v_2, \\
\frac{dh_3}{dt} &= -\frac{a_3}{A_3}\sqrt{2gh_3} + \frac{(1-\gamma_2)k_2}{A_3}v_2, \\
\frac{dh_4}{dt} &= -\frac{a_4}{A_4}\sqrt{2gh_4} + \frac{(1-\gamma_1)k_1}{A_4}v_1,
\end{aligned}$$

where A_i is a cross-section of the ith tank, a_i is a cross-section of the ith tank's outlet hole, h_i is a water level of the ith tank, v_1 and v_2 are voltages applied to the first and second pumps, k_1 and k_2 are input gains, γ_1 and γ_2 are model parameters, and g is the gravitational acceleration. The linearized system of the process at an operating point $(h_1^0, h_2^0, h_3^0, h_4^0, v_1^0, v_2^0)$ is given by

$$\frac{dx}{dt} = \begin{bmatrix} -\frac{1}{T_1} & 0 & \frac{A_3}{A_1 T_3} & 0 \\ 0 & -\frac{1}{T_2} & 0 & \frac{A_4}{A_2 T_4} \\ 0 & 0 & -\frac{1}{T_3} & 0 \\ 0 & 0 & 0 & -\frac{1}{T_4} \end{bmatrix} x + \begin{bmatrix} \frac{\gamma_1 k_1}{A_1} & 0 \\ 0 & \frac{\gamma_2 k_2}{A_2} \\ 0 & \frac{(1-\gamma_2)k_2}{A_3} \\ \frac{(1-\gamma_1)k_1}{A_4} & 0 \end{bmatrix} u,$$

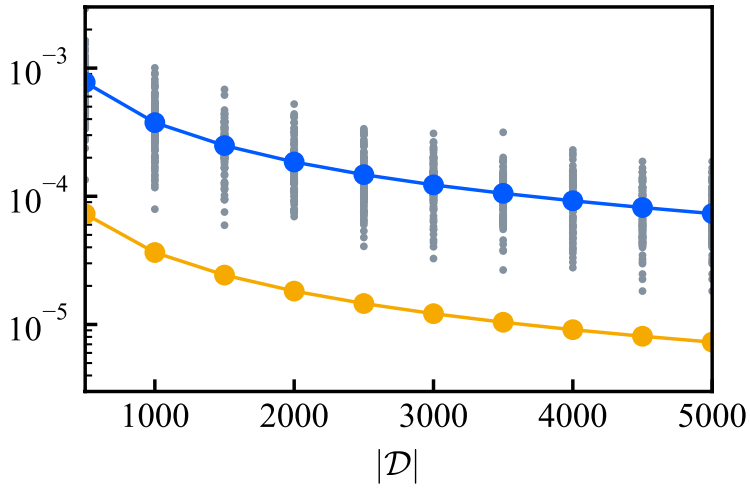


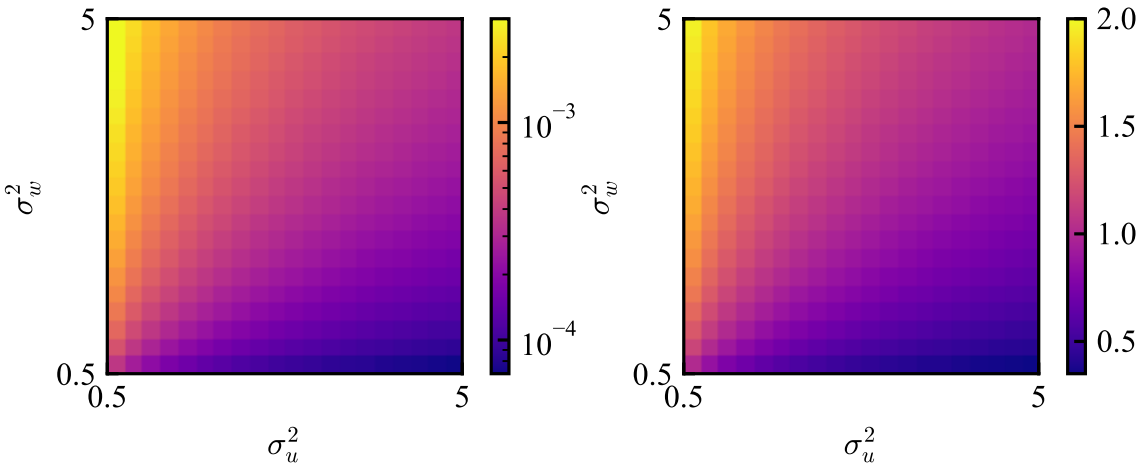
Fig. 5.1: The expectation of estimation error and its lower bound when an attack target is a plant.

where $x = [x_1\ x_2\ x_3\ x_4]^\top$, $u = [u_1\ u_2]^\top$, $x_i = h_i - h_i^0$, $u_i = v_i - v_i^0$, and $T_i = (A_i/a_i)\sqrt{2h_i^0/g}$. Following [219], set $A_1 = A_3 = 28$ cm^2, $A_2 = A_4 = 32$ cm^2, $a_1 = a_3 = 0.071$ cm^2, $a_2 = a_4 = 0.057$ cm^2, $k_1 = 3.33$ cm^3/Vs, $k_2 = 3.35$ cm^3/Vs, $\gamma_1 = 0.70$, $\gamma_2 = 0.60$, and $g = 981$ cm/s^2. Suppose the initial state follows the Gaussian distribution with mean zero and variance Σ_x. Then, by discretizing with the sampling period of 1 s and adding a noise w independent and identically distributed over the Gaussian distribution with mean zero and variance Σ_w, the linearized system at the operating point $(h_1^0, h_2^0, h_3^0, h_4^0, v_1^0, v_2^0) = (12.4, 12.7, 1.8, 1.4, 3.00, 3.00)$ is obtained as (4.1), where

$$A = \begin{bmatrix} 0.9842 & 0 & 0.0407 & 0 \\ 0 & 0.9890 & 0 & 0.0326 \\ 0 & 0 & 0.9590 & 0 \\ 0 & 0 & 0 & 0.9672 \end{bmatrix}, \quad B = \begin{bmatrix} 0.0826 & 0.0010 \\ 0.0005 & 0.0625 \\ 0 & 0.0469 \\ 0.0307 & 0 \end{bmatrix}. \tag{5.14}$$

Additionally, for the sake of simplicity, set $\Sigma_x = I$, $\Sigma_w = \sigma_w^2 I$, and $\Sigma_u = \sigma_u^2 I$ throughout this section.

Fig. 5.1 depicts the expectation of estimation error and its lower bound when an attack target is a plant (4.1) with the system parameters (5.14). The gray dots in the figure are the estimation errors defined in Definition 4.2, the blue line is the expectation of estimation error in Lemma 5.2, and the orange line is the lower bound (5.6), where the attack of Definition 4.1 was performed 100 times for each



(a) The expectation of estimation error with various variances.

(b) The gap between the expectation of estimation error and its lower bound.

Fig. 5.2: The expectation of estimation error and its gap to the lower bound in changing input and noise variances.

sample size using the OLS estimator (5.3) with $t_s = 0$, and $\sigma_w^2 = \sigma_u^2 = 1$. The figure demonstrates that both the expectation of estimation error and lower bound decrease as a sample size $|\mathcal{D}|$ increases at a similar rate.

Next, the change in the expectation of estimation error is confirmed when the noise and input variances are varied. Fig. 5.2(a) illustrates the expectation of estimation error in the 400 combinations of σ_w^2 and σ_u^2 for $|\mathcal{D}| = 1000$, where each variance is spaced in 20 equal parts from 0.5 to 5. It can be seen that the expectation of estimation error decreases as σ_w^2 and σ_u^2 respectively decreases and increases. This result suggests that an adversary would select the input variance as large as possible compared to the noise variance. Moreover, Fig. 5.2(b) depicts the gap between the expectation of estimation error and its lower bound defined by

$$\log_{10}\Big(\mathbb{E}\Big[\varepsilon\Big(\begin{bmatrix}A & B\end{bmatrix}, \hat{\Theta}_{\mathrm{OLS}}(\mathcal{D})\Big)\Big]\Big) - \log_{10}\Big(\gamma\Big(|\mathcal{D}|, \begin{bmatrix}A & B\end{bmatrix}\Big)\Big),$$

where ε, $\hat{\Theta}_{\mathrm{OLS}}$, and γ are defined in Definition 4.2, (5.3), and (5.6), respectively. The figure shows that the lower bound becomes tighter if the input variance is sufficiently larger than the noise variance. Therefore, (5.6) would be a reasonable choice for a lower bound of the expectation of estimation error in practice.

Using (5.6), the lower bound (5.7) of the sample identifying complexity in Theorem 5.5 and the suboptimal security parameter in (5.8) are computed as shown in Table 5.1, where Υ in (5.8) is set to 4.42×10^{17} FLOPS that is the performance of

Table 5.1: Suboptimal security parameters when an attack target is a plant.

γ_c	N^{**}	τ_c (years)				
		1	3	5	10	50
10^{-8}	3641747	62	64	65	66	68
10^{-7}	364175	66	67	68	69	71
10^{-6}	36418	69	70	71	72	75
10^{-5}	3642	72	74	75	76	78
10^{-4}	365	76	77	78	79	81
–	1	84	86	86	87	90

Supercomputer Fugaku[1]. Note that the security parameters in the case of $N^{**} = 1$ are optimal values when a key pair is not updated. For instance, with the acceptable estimation error $\gamma_c = 10^{-6}$ and the defense period $\tau_c = 10$ years, it can be shown that updatable (or key-updatable) homomorphic encryption reduces a security parameter by 15 bit compared with typical homomorphic encryption. This result offers the effectiveness of updatable and key-updatable homomorphic encryption in improving the security level of encrypted control systems or reducing computation costs due to encryption while keeping a security level.

We move on to examining the design of encrypted control systems when an attack target is a closed-loop system (4.3). The suboptimal feedback gain (5.11) of a controller (4.2) for (4.1) with (5.14) is computed by using CVXPY [220, 221] as

$$F^{**} = \begin{bmatrix} -10.6621 & -0.4077 & 0.3075 & -3.3205 \\ -0.1287 & -11.5672 & -5.5034 & 0.1435 \end{bmatrix},$$

where $\sigma_w^2 = 1$, and

$$P^{**} = \begin{bmatrix} 3.3727 & -2.3306 & 4.1041 & -6.2997 \\ -2.3306 & 5.4510 & -8.5637 & 4.8929 \\ 4.1041 & -8.5638 & 17.6025 & -8.1816 \\ -6.2997 & 4.8929 & -8.1816 & 18.5025 \end{bmatrix},$$

$$Q^{**} = \begin{bmatrix} -12.8298 & 3.7464 & -7.6879 & 1.2201 \\ 3.0332 & -14.9199 & 0.4824 & -8.1044 \end{bmatrix}.$$

[1] https://www.top500.org/system/179807/

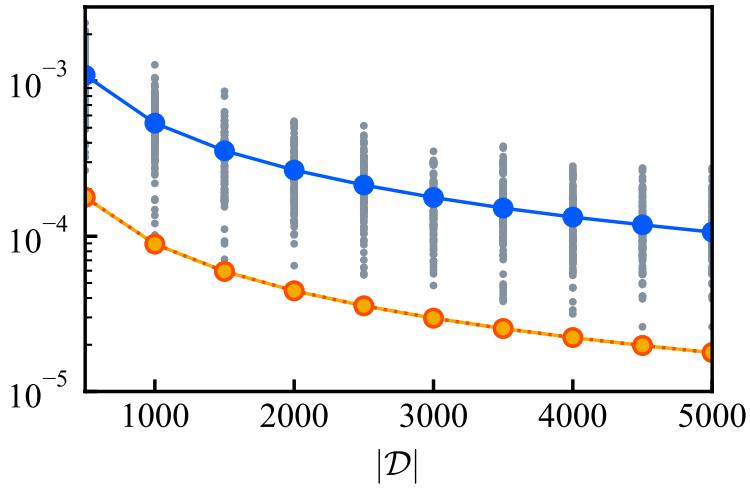


Fig. 5.3: The expectation of estimation error and its lower bounds when an attack target is a closed-loop system.

The system parameter of (4.3) is then given as

$$\bar{A} = \begin{bmatrix} 0.1034 & -0.0452 & 0.0606 & -0.2741 \\ -0.0134 & 0.2658 & -0.3438 & 0.0399 \\ -0.0060 & -0.5425 & 0.7009 & 0.0067 \\ -0.3273 & -0.0125 & 0.0094 & 0.8653 \end{bmatrix}. \tag{5.15}$$

Fig. 5.3 depicts the expectation of estimation error and its lower bounds under the OLS estimator for (4.3) with (5.15), where $t_s = 0$. Except for the attack target, the gray dots and the blue line are the same as Fig. 5.1. The orange solid line is the lower bound (5.9), and the red dotted line is the approximated lower bound (5.10). Similar to the open-loop case, the expectation of estimation error and the lower bounds decrease as $|\mathcal{D}|$ increases. Furthermore, the figure shows that the lower bound and its approximation are almost identical. Thus, the approximation can be used for designing the suboptimal feedback gain instead of the lower bound.

With the suboptimal controller, Table 5.2 shows the lower bound (5.12) of the sample identifying complexity in Theorem 5.5 and the suboptimal security parameter in (5.13), which are computed under the same conditions as Table 5.1. Recall that the security parameters in the case of $N^{**} = 1$ are optimal values when using typical homomorphic encryption. The result reveals the effectiveness of updatable and key-updatable homomorphic encryption in reducing a security parameter while keeping a certain security level even when an attack target is a closed-loop system.

Table 5.2: Suboptimal security parameters with the suboptimal controller when an attack target is a closed-loop system.

γ_c	N^{**}	τ_c (years)				
		1	3	5	10	50
10^{-8}	8903047	61	63	63	64	67
10^{-7}	890305	64	66	67	68	70
10^{-6}	89031	68	69	70	71	73
10^{-5}	8904	71	72	73	74	77
10^{-4}	891	74	76	77	78	80
–	1	84	86	86	87	90

Table 5.3: Suboptimal security parameters with a controller designed by pole placement when an attack target is a closed-loop system.

γ_c	N^{**}	τ_c (years)				
		1	3	5	10	50
10^{-8}	516	75	77	77	78	81
10^{-7}	52	78	80	81	82	84
10^{-6}	6	81	83	84	85	87
10^{-5}	2	83	85	85	86	89
10^{-4}	2	83	85	85	86	89
–	1	84	86	86	87	90

Furthermore, Table 5.3 presents the lower bound of the sample identifying complexity and the suboptimal security parameter when using a feedback gain that places the eigenvalues of $\bar{A}$ in (4.3) to ± 0.01 and $\pm 0.01i$, where i is the imaginary unit. Comparing Table 5.2 with Table 5.3, we can see that the suboptimal controller significantly increases the lower bound of the sample identifying complexity and contributes to reducing a security parameter.

Chapter 6

Conclusion

Encrypted control is a state-of-the-art technology used to enhance the confidentiality of control systems. This thesis focused on the security and design of encrypted control systems that utilize homomorphic encryption to establish a fundamental theory for encrypted control. Our investigation delved into appropriate security measures for encrypted control systems and determined their security level based on the complexity and computation time of the system identification. The security level of encrypted control systems can be quantified by an acceptable estimation error and a defense period induced by a sample identifying complexity and a sample deciphering time. The thesis stressed the significance of a key update mechanism in ensuring forward and post-compromise security. It examined the correlations among the security strength, security parameter, and system parameter in encrypted control systems. Based on these relationships, it also formulated a design methodology for a security parameter and controller to achieve the desired security level of an encrypted control system.

The design method allows for a systematic approach to creating encrypted control systems. With this method, a system designer can determine the appropriate security parameter and controller for a specific plant and cryptosystem without trying different options through trial and error. This results in a lower computational burden caused by encryption and can reduce implementation time and costs. Furthermore, the security level achieved using this design method can be used as a benchmark to standardize the security strength in encrypted control systems.

When designing control systems, security has now become one of the control specifications, such as stability and robustness, and is no longer ambiguous. Consequently, control systems can be analyzed and synthesized with a cryptographic security lens, building a theory for encrypted control systems. In other words, a theoretical foundation for interdisciplinary research on control theory and cryptography can be constructed based on our security definition and measures. Some remarks are delineated to guide future research endeavors in this domain.

Output feedback and nonlinearity. In most cases, the states of dynamical systems cannot be directly obtained and are partially observed through sensors. In addition, controlling a dynamical system sometimes involves dealing with nonlinear behavior. When a controller determines a control input based on sensor observations, it is called an output-feedback controller. A feedback system that includes nonlinear components is known as a nonlinear control system.

An adversary cannot access the states of a plant when using an output-feedback controller and cannot apply the estimation algorithms in Section 5.1 directly to nonlinear control systems. Therefore, an adversary may use more advanced algorithms, such as subspace identification methods or machine learning, to estimate system parameters. As a result, extending the design method discussed in this thesis to encrypted output feedback and nonlinear control systems requires deriving lower bounds for the expectation of an estimation error and sample identifying complexity in such advanced estimation algorithms.

Multi-agent systems security. This thesis discussed the security and design of encrypted control systems in a client-server model. However, a multi-agent system is another primary class for encrypted control. Defining a threat model and security goal is essential for formulating security tailored for multi-agent systems, similar to the discussion in Chapter 4. Potential adversaries for a multi-agent system are comparable to those for a client-server model, including a network eavesdropper outside the system, a malicious agent within the system, and a malicious server that aggregates distributed agents. Meanwhile, caution is warranted when considering a security goal, that is, private information to be protected.

Private information differs for each system under consideration. Thus, it is necessary first to clarify which information should be protected. Moreover, some information in multi-agent systems cannot be protected by encryption. For example, a network topology represented by an adjacency matrix or a graph Laplacian is an example of distinctive private information in a multi-agent system. However, it can be disclosed by tracking a packet flow without deciphering encrypted data. Therefore, it is crucial to determine whether private information of interest can be protected using cryptography when establishing a security goal in an encrypted control framework for multi-agent systems.

Multi-objective synthesis of encrypted controllers. Chapter 4 and Chapter 5 discussed the analysis and design of encrypted control systems, respectively. The analysis determined the security level of encrypted control systems from a given security parameter and system parameter. Conversely, the design addressed deciding on a security parameter and controller to achieve the desired security level. In the design, Theorem 5.9 revealed that a suboptimal controller is given by solving a semidefinite programming problem in which the constraints are linear matrix inequalities. Linear matrix inequalities can also represent various control objectives and constraints on disturbance rejection, tracking performance, H_∞ performance, overshoot, rise time, settling time, pole region, and so on [217]. Hence, combining a controller design to reduce the security parameter with other control objectives and constraints would be feasible. An interesting research direction here is the multi-objective synthesis of encrypted controllers to satisfy the required control specifications, including security.

Bibliography

[1] E. A. Lee, "Cyber physical systems: Design challenges," in *IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing*, 2008, pp. 363–369.

[2] Y. Liu, Y. Peng, B. Wang, S. Yao, and Z. Liu, "Review on cyber-physical systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 1, pp. 27–40, 2017.

[3] R. R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: The next computing revolution," in *Design Automation Conference*, 2010, pp. 731–736.

[4] J. Shi, J. Wan, H. Yan, and H. Suo, "A survey of cyber-physical systems," in *International Conference on Wireless Communications and Signal Processing*, 2011, pp. 1–6.

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

[6] A. Teixeira, K. C. Sou, H. Sandberg, and K. H. Johansson, "Secure control systems: A quantitative risk management approach," *IEEE Control Systems Magazine*, vol. 35, no. 1, pp. 24–45, 2015.

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

[8] I. Zografopoulos, J. Ospina, X. Liu, and C. Konstantinou, "Cyber-physical energy systems security: Threat modeling, risk assessment, resources, metrics, and case studies," *IEEE Access*, vol. 9, pp. 29 775–29 818, 2021.

[9] W. Duo, M. Zhou, and A. Abusorrah, "A survey of cyber attacks on cyber physical systems: Recent advances and challenges," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 5, pp. 784–800, 2022.

[10] H. Kayan, M. Nunes, O. Rana, P. Burnap, and C. Perera, "Cybersecurity of industrial cyber-physical systems: A review," *ACM Computing Surveys*, vol. 54, no. 11s, 2022.

[11] S. Kim, K.-J. Park, and C. Lu, "A survey on network security for cyber-physical systems: From threats to resilient design," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 3, pp. 1534–1573, 2022.

[12] 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.

[13] M. S. Chong, H. Sandberg, and A. M. Teixeira, "A tutorial introduction to security and privacy for cyber-physical systems," in *European Control Conference*, 2019, pp. 968–978.

[14] C. Dwork, "Differential privacy," in *Automata, Languages and Programming*, ser. Lecture Notes in Computer Science, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds. Berlin, Heidelberg: Springer, 2006, pp. 1–12.

[15] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Advances in Cryptology - EUROCRYPT 2006*, ser. Lecture Notes in Computer Science, S. Vaudenay, Ed. Berlin, Heidelberg: Springer, 2006, pp. 486–503.

[16] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*, ser. Lecture Notes in Computer Science, S. Halevi and T. Rabin, Eds. Berlin, Heidelberg: Springer, 2006, pp. 265–284.

[17] C. Dwork, "Differential privacy: A survey of results," in *Theory and Applications of Models of Computation*, ser. Lecture Notes in Computer Science, M. Agrawal, D. Du, Z. Duan, and A. Li, Eds. Berlin, Heidelberg: Springer, 2008, pp. 1–19.

[18] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2013.

[19] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What can we learn privately?" in *IEEE Symposium on Foundations of Computer Science*, 2008, pp. 531–540.

[20] R. Chen, H. Li, A. K. Qin, S. P. Kasiviswanathan, and H. Jin, "Private spatial data aggregation in the local setting," in *IEEE International Conference on Data Engineering*, 2016, pp. 289–300.

[21] M. Bun, J. Nelson, and U. Stemmer, "Heavy hitters and the structure of local privacy," *ACM Transactions on Algorithms*, vol. 15, no. 4, pp. 1–40, 2019.

[22] A. Friedman and A. Schuster, "Data mining with differential privacy," in *ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 493–502.

[23] N. Mohammed, R. Chen, B. C. Fung, and P. S. Yu, "Differentially private data release for data mining," in *ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 493–501.

[24] U. Erlingsson, V. Pihur, and A. Korolova, "RAPPOR: Randomized aggregatable privacy-preserving ordinal response," in *ACM SIGSAC Conference on Computer and Communications Security*, 2014, pp. 1054–1067.

[25] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 1310–1321.

[26] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 308–318.

[27] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.

[28] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, and H. Vincent Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[29] M. Gong, Y. Xie, K. Pan, K. Feng, and A. Qin, "A survey on differentially private machine learning," *IEEE Computational Intelligence Magazine*, vol. 15, no. 2, pp. 49–64, 2020.

[30] J. Le Ny and G. J. Pappas, "Differentially private filtering," *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 341–354, 2014.

[31] J. Cortés, G. E. Dullerud, S. Han, J. Le Ny, S. Mitra, and G. J. Pappas, "Differential privacy in control and network systems," in *IEEE Conference on Decision and Control*, 2016, pp. 4252–4272.

[32] F. Koufogiannis and G. J. Pappas, "Differential privacy for dynamical sensitive data," in *IEEE Conference on Decision and Control*, 2017, pp. 1118–1125.

[33] Y. Wang, Z. Huang, S. Mitra, and G. E. Dullerud, "Differential privacy in linear distributed control systems: Entropy minimizing mechanisms and performance tradeoffs," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 1, pp. 118–130, 2017.

[34] S. Han and G. J. Pappas, "Privacy in control and dynamical systems," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 309–332, 2018.

[35] Y. Kawano and M. Cao, "Design of privacy-preserving dynamic controllers," *IEEE Transactions on Automatic Control*, vol. 65, no. 9, pp. 3863–3878, 2020.

[36] Y. Kawano, K. Kashima, and M. Cao, "Modular control under privacy protection: Fundamental trade-offs," *Automatica*, vol. 127, p. 109518, 2021.

[37] G. Sugiura, K. Ito, and K. Kashima, "Bayesian differential privacy for linear dynamical systems," *IEEE Control Systems Letters*, vol. 6, pp. 896–901, 2022.

[38] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Foundations of secure computation*, vol. 4, no. 11, pp. 169–180, 1978.

[39] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Computing Surveys*, vol. 51, no. 4, 2019.

[40] C. Marcolla, V. Sucasas, M. Manzano, R. Bassoli, F. H. P. Fitzek, and N. Aaraj, "Survey on fully homomorphic encryption, theory, and applications," *Proceedings of the IEEE*, vol. 110, no. 10, pp. 1572–1609, 2022.

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

[42] S. Goldwasser and S. Micali, "Probabilistic encryption & how to play mental poker keeping secret all partial information," in *ACM symposium on Theory of computing*, 1982, pp. 365–377.

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

[44] J. Benaloh, "Dense probabilistic encryption," in *Workshop on Selected Areas of Cryptography*, 1994, pp. 120–128.

[45] D. Naccache and J. Stern, "A new public key cryptosystem based on higher residues," in *ACM conference on Computer and communications security*, 1998, pp. 59–66.

[46] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology – EUROCRYPT '99*, ser. Lecture Notes in Computer Science, J. Stern, Ed. Berlin, Heidelberg: Springer, 1999, pp. 223–238.

[47] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Theory of Cryptography*, ser. Lecture Notes in Computer Science, J. Kilian, Ed. Berlin, Heidelberg: Springer, 2005, pp. 325–341.

[48] Y. Ishai and A. Paskin, "Evaluating branching programs on encrypted data," in *Theory of Cryptography*, ser. Lecture Notes in Computer Science, S. P. Vadhan, Ed. Berlin, Heidelberg: Springer, 2007, pp. 575–594.

[49] J. Dyer, M. Dyer, and J. Xu, "Practical homomorphic encryption over the integers for secure computation in the cloud," *International Journal of Information Security*, vol. 18, no. 5, pp. 549–579, 2019.

[50] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *ACM symposium on Theory of computing*, 2009, pp. 169–178.

[51] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in *Advances in Cryptology – EUROCRYPT 2010*, ser. Lecture Notes in Computer Science, H. Gilbert, Ed. Berlin, Heidelberg: Springer, 2010, pp. 24–43.

[52] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," in *Innovations in Theoretical Computer Science Conference*, 2012, pp. 309–325.

[53] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption," 2012. [Online]. Available: https://eprint.iacr.org/2012/144

[54] A. López-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption," in *ACM symposium on Theory of computing*, 2012, pp. 1219–1234.

[55] J. W. Bos, K. Lauter, J. Loftus, and M. Naehrig, "Improved security for a ring-based fully homomorphic encryption scheme," in *Cryptography and Coding*, ser. Lecture Notes in Computer Science, M. Stam, Ed. Berlin, Heidelberg: Springer, 2013, pp. 45–64.

[56] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," in *Advances in Cryptology – CRYPTO 2013*, ser. Lecture Notes in Computer Science, R. Canetti and J. A. Garay, Eds. Berlin, Heidelberg: Springer, 2013, pp. 75–92.

[57] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Advances in Cryptology – ASIACRYPT 2017*, ser. Lecture Notes in Computer Science, T. Takagi and T. Peyrin, Eds. Cham: Springer International Publishing, 2017, pp. 409–437.

[58] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, "TFHE: Fast fully homomorphic encryption over the torus," *Journal of Cryptology*, vol. 33, no. 1, pp. 34–91, 2020.

[59] C. Bonte, I. Iliashenko, J. Park, H. V. L. Pereira, and N. P. Smart, "FINAL: Faster FHE Instantiated with NTRU and LWE," in *Advances in Cryptology – ASIACRYPT 2022*, ser. Lecture Notes in Computer Science, S. Agrawal and D. Lin, Eds. Cham: Springer Nature Switzerland, 2022, pp. 188–215.

[60] M. Zhao E and Y. Geng, "Homomorphic encryption technology for cloud computing," *Procedia Computer Science*, vol. 154, pp. 73–83, 2019.

[61] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy," in *International Conference on Machine Learning*, 2016, pp. 201–210.

[62] X. Jiang, M. Kim, K. Lauter, and Y. Song, "Secure outsourced matrix computation and application to neural networks," in *ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 1209–1222.

[63] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan, "GAZELLE: A low latency framework for secure neural network inference," in *USENIX Security Symposium*, 2018, pp. 1651–1669.

[64] J.-W. Lee, H. Kang, Y. Lee, W. Choi, J. Eom, M. Deryabin, E. Lee, J. Lee, D. Yoo, Y.-S. Kim, and J.-S. No, "Privacy-preserving machine learning with fully homomorphic encryption for deep neural network," *IEEE Access*, vol. 10, pp. 30 039–30 054, 2022.

[65] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[66] G. R. Blakley, "Safeguarding cryptographic keys," in *International Workshop on Managing Requirements Knowledge*, 1979, pp. 313–313.

[67] A. Beimel, "Secret-sharing schemes: A survey," in *Coding and Cryptology*, ser. Lecture Notes in Computer Science, Y. M. Chee, Z. Guo, S. Ling, F. Shao, Y. Tang, H. Wang, and C. Xing, Eds. Berlin, Heidelberg: Springer, 2011, pp. 11–46.

[68] V. Attasena, J. Darmont, and N. Harbi, "Secret sharing for cloud data security: A survey," *The VLDB Journal*, vol. 26, no. 5, pp. 657–681, 2017.

[69] M. Ben-Or and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," in *ACM symposium on Theory of computing*, 1988, pp. 1–10.

[70] D. Chaum, C. Crépeau, and I. Damgard, "Multiparty unconditionally secure protocols," in *ACM symposium on Theory of computing*, 1988, pp. 11–19.

[71] T. Rabin and M. Ben-Or, "Verifiable secret sharing and multiparty protocols with honest majority," in *ACM symposium on Theory of computing*, 1989, pp. 73–85.

[72] R. Cramer, I. Damgård, and U. Maurer, "General secure multi-party computation from any linear secret-sharing scheme," in *Advances in Cryptology – EUROCRYPT 2000*, ser. Lecture Notes in Computer Science, B. Preneel, Ed. Berlin, Heidelberg: Springer, 2000, pp. 316–334.

[73] A. C.-C. Yao, "How to generate and exchange secrets," in *Annual Symposium on Foundations of Computer Science*, 1986, pp. 162–167.

[74] O. Goldreich, S. Micali, and A. Wigderson, "How to play ANY mental game," in *ACM conference on Theory of computing*, 1987, pp. 218–229.

[75] D. Beaver, S. Micali, and P. Rogaway, "The round complexity of secure protocols," in *ACM symposium on Theory of Computing*, 1990, pp. 503–513.

[76] Y. Lindell and B. Pinkas, "A proof of security of Yao's protocol for two-party computation," *Journal of Cryptology*, vol. 22, no. 2, pp. 161–188, 2009.

[77] M. Bellare, V. T. Hoang, and P. Rogaway, "Foundations of garbled circuits," in *ACM conference on Computer and communications security*, 2012, pp. 784–796.

[78] S. Even, O. Goldreich, and A. Lempel, "A randomized protocol for signing contracts," *Communications of the ACM*, vol. 28, no. 6, pp. 637–647, 1985.

[79] M. O. Rabin, "How to exchange secrets with oblivious transfer," 2005. [Online]. Available: https://eprint.iacr.org/2005/187

[80] V. Kolesnikov and T. Schneider, "Improved garbled circuit: Free XOR gates and applications," in *Automata, Languages and Programming*, ser. Lecture

Notes in Computer Science, L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, and I. Walukiewicz, Eds. Berlin, Heidelberg: Springer, 2008, pp. 486–498.

[81] V. Kolesnikov, A.-R. Sadeghi, and T. Schneider, "Improved garbled circuit building blocks and applications to auctions and computing minima," in *Cryptology and Network Security*, ser. Lecture Notes in Computer Science, J. A. Garay, A. Miyaji, and A. Otsuka, Eds. Berlin, Heidelberg: Springer, 2009, pp. 1–20.

[82] Y. Huang, D. Evans, J. Katz, and L. Malka, "Faster secure two-party computation using garbled circuits," in *USENIX Security Symposium*, 2011.

[83] S. Zahur, M. Rosulek, and D. Evans, "Two halves make a whole," in *Advances in Cryptology - EUROCRYPT 2015*, ser. Lecture Notes in Computer Science, E. Oswald and M. Fischlin, Eds. Berlin, Heidelberg: Springer, 2015, pp. 220–250.

[84] W. Du and M. J. Atallah, "Secure multi-party computation problems and their applications: A review and open problems," in *New Security Paradigms Workshop*, 2001, pp. 13–22.

[85] P. Bogetoft, D. L. Christensen, I. Damgård, M. Geisler, T. Jakobsen, M. Krøigaard, J. D. Nielsen, J. B. Nielsen, K. Nielsen, J. Pagter, M. Schwartzbach, and T. Toft, "Secure multiparty computation goes live," in *Financial Cryptography and Data Security*, ser. Lecture Notes in Computer Science, R. Dingledine and P. Golle, Eds. Berlin, Heidelberg: Springer, 2009, pp. 325–343.

[86] Y. Lindell, "Secure multiparty computation," *Communications of the ACM*, vol. 64, no. 1, pp. 86–96, 2021.

[87] R. Bendlin, I. Damgård, C. Orlandi, and S. Zakarias, "Semi-homomorphic encryption and multiparty computation," in *Advances in Cryptology – EUROCRYPT 2011*, ser. Lecture Notes in Computer Science, K. G. Paterson, Ed. Berlin, Heidelberg: Springer, 2011, pp. 169–188.

[88] I. Damgård, V. Pastro, N. Smart, and S. Zakarias, "Multiparty computation from somewhat homomorphic encryption," in *Advances in Cryptology –*

CRYPTO 2012, ser. Lecture Notes in Computer Science, R. Safavi-Naini and R. Canetti, Eds. Berlin, Heidelberg: Springer, 2012, pp. 643–662.

[89] R. Cramer, I. B. Damgrd, and J. B. Nielsen, *Secure multiparty computation and secret sharing*, 1st ed. USA: Cambridge University Press, 2015.

[90] Y. Huang, C.-h. Shen, D. Evans, J. Katz, and A. Shelat, "Efficient secure computation with garbled circuits," in *Information Systems Security*, ser. Lecture Notes in Computer Science, S. Jajodia and C. Mazumdar, Eds. Berlin, Heidelberg: Springer, 2011, pp. 28–48.

[91] P. Mohassel, M. Rosulek, and Y. Zhang, "Fast and secure three-party computation: The garbled circuit approach," in *ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 591–602.

[92] Y. Ishai, M. Prabhakaran, and A. Sahai, "Founding cryptography on oblivious transfer – efficiently," in *Advances in Cryptology – CRYPTO 2008*, ser. Lecture Notes in Computer Science, D. Wagner, Ed. Berlin, Heidelberg: Springer, 2008, pp. 572–591.

[93] M. Keller, E. Orsini, and P. Scholl, "MASCOT: Faster malicious arithmetic secure computation with oblivious transfer," in *ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 830–842.

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

[95] F. Farokhi, I. Shames, and N. Batterham, "Secure and private cloud-based control using semi-homomorphic encryption," *IFAC-PapersOnLine*, vol. 49, no. 22, pp. 163–168, 2016.

[96] ——, "Secure and private control using semi-homomorphic encryption," *Control Engineering Practice*, vol. 67, pp. 13–20, 2017.

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

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

[99] A. B. Alexandru and G. J. Pappas, "Encrypted LQG using labeled homomorphic encryption," in *ACM/IEEE International Conference on Cyber-Physical Systems*, 2019, pp. 129–140.

[100] 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.

[101] J. Kim, H. Shim, and K. Han, "Design procedure for dynamic controllers based on LWE-based homomorphic encryption to operate for infinite time horizon," in *IEEE Conference on Decision and Control*, 2020, pp. 5463–5468.

[102] J. Kim, H. Shim, H. Sandberg, and K. H. Johansson, "Method for running dynamic systems over encrypted data for infinite time horizon without bootstrapping and re-encryption," in *IEEE Conference on Decision and Control*, 2021, pp. 5614–5619.

[103] N. Schlüter, M. Neuhaus, and M. S. Darup, "Encrypted dynamic control with unlimited operating time via FIR filters," in *European Control Conference*, 2021, pp. 952–957.

[104] J. Kim, H. Shim, and K. Han, "Dynamic controller that operates over homomorphically encrypted data for infinite time horizon," *IEEE Transactions on Automatic Control*, vol. 68, no. 2, pp. 660–672, 2023.

[105] K. Teranishi, T. Sadamoto, and K. Kogiso, "Input-output history feedback controller for encrypted control with leveled fully homomorphic encryption," *IEEE Transactions on Control of Network Systems*, 2023, early access.

[106] M. S. Darup, "Encrypted polynomial control based on tailored two-party computation," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 11, pp. 4168–4187, 2020.

[107] S. Schlor, M. Hertneck, S. Wildhagen, and F. Allgöwer, "Multi-party computation enables secure polynomial control based solely on secret-sharing," in *IEEE Conference on Decision and Control*, 2021, pp. 4882–4887.

[108] M. Kishida, "Encrypted control system with quantiser," *IET Control Theory & Applications*, vol. 13, no. 1, pp. 146–151, 2019.

[109] J. Kim, F. Farokhi, I. Shames, and H. Shim, "Toward nonlinear dynamic control over encrypted data for infinite time horizon," in *IFAC World Congress*, 2020.

[110] K. Teranishi, K. Kogiso, and J. Ueda, "Encrypted feedback linearization and motion control for manipulator with somewhat homomorphic encryption," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2020, pp. 613–618.

[111] K. Teranishi and K. Kogiso, "Encrypted gain scheduling with quantizers for stability guarantee," in *IEEE Conference on Decision and Control*, 2021, pp. 5628–5633.

[112] K. Teranishi, J. Ueda, and K. Kogiso, "Event-triggered approach to increasing sampling period of encrypted control systems," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 3502–3507, 2020.

[113] R. Fritz, M. Fauser, and P. Zhang, "Controller encryption for discrete event systems," in *American Control Conference*, 2019, pp. 5633–5638.

[114] Y. Qiu and J. Ueda, "Encrypted motion control of a teleoperation system with security-enhanced controller by deception," in *ASME Dynamic System and Control Conference*, 2019.

[115] K. Teranishi, M. Kusaka, N. Shimada, J. Ueda, and K. Kogiso, "Secure observer-based motion control based on controller encryption," in *American Control Conference*, 2019, pp. 2978–2983.

[116] Q. Hu, Y. Shi, and E. Nekouei, "Secure motion control of micro-spacecraft using semi-homomorphic encryption," *Security and Safety*, 2023.

[117] H. Takanashi, K. Teranishi, and K. Kogiso, "Experimental validation of reaction force estimation for secure robot teleoperation," in *IEEE/SICE International Symposium on System Integration*, 2023.

[118] S. Kosieradzki, X. Zhao, H. Kawase, Y. Qiu, K. Kogiso, and J. Ueda, "Secure teleoperation control using somewhat homomorphic encryption," *IFAC-PapersOnLine*, vol. 55, no. 37, pp. 593–600, 2022.

[119] N. Shono, T. Miyazaki, K. Teranishi, T. Kanno, T. Kawase, K. Kogiso, and K. Kawashima, "Implementation of encrypted control of pneumatic bilateral control system using wave variables," in *International Symposium on Artificial Life and Robotics, International Symposium on BioComplexity, International Symposium on Swarm Behavior and Bio-Inspired Robotics*, 2022, pp. 1169–1174.

[120] A. B. Alexandru, A. Tsiamis, and G. J. Pappas, "Towards private data-driven control," in *IEEE Conference on Decision and Control*, 2020, pp. 5449–5456.

[121] N. Schlüter, M. Neuhaus, and M. S. Darup, "Encrypted extremum seeking for privacy-preserving PID tuning as-a-Service," in *European Control Conference*, 2022, pp. 1288–1293.

[122] Y. Chen, D. Huang, and S. Tan, "Privacy-enabled secure iterative learning control of networked systems subject to disclosure attacks," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2023, early Access.

[123] A. B. Alexandru, M. Morari, and G. J. Pappas, "Cloud-based MPC with encrypted data," in *IEEE Conference on Decision and Control*, 2018, pp. 5014–5019.

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

[125] M. S. Darup, A. Redder, I. Shames, F. Farokhi, and D. E. Quevedo, "Towards encrypted MPC for linear constrained systems," *IEEE Control Systems Letters*, vol. 2, no. 2, pp. 195–200, 2018.

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

[127] N. Schlüter and M. S. Darup, "Encrypted explicit MPC based on two-party computation and convex controller decomposition," in *IEEE Conference on Decision and Control*, 2020, pp. 5469–5476.

[128] K. Tjell, N. Schlüter, P. Binfet, and M. S. Darup, "Secure learning-based MPC via garbled circuit," in *IEEE Conference on Decision and Control*, 2021, pp. 4907–4914.

[129] A. M. Naseri, W. Lucia, and A. Youssef, "Encrypted cloud-based set-theoretic model predictive control," *IEEE Control Systems Letters*, vol. 6, pp. 3032–3037, 2022.

[130] M. Zamani, L. Sadeghikhorrami, A. A. Safavi, and F. Farokhi, "Private state estimation for cyber-physical systems using semi-homomorphic encryption," in *International Symposium on Mathematical Theory of Networks and Systems*, 2018, pp. 399–404.

[131] M. Ristic and B. Noack, "Privileged estimate fusion with correlated gaussian keystreams," in *IEEE Conference on Decision and Control*, 2022, pp. 7732–7739.

[132] M. Ristic, B. Noack, and U. D. Hanebeck, "Cryptographically privileged state estimation with Gaussian keystreams," *IEEE Control Systems Letters*, vol. 6, pp. 602–607, 2022.

[133] M. Aristov, B. Noack, U. D. Hanebeck, and J. Müller-Quade, "Encrypted multisensor information filtering," in *International Conference on Information Fusion*, 2018, pp. 1631–1637.

[134] Y. Shoukry, K. Gatsis, A. Alanwar, G. J. Pappas, S. A. Seshia, M. Srivastava, and P. Tabuada, "Privacy-aware quadratic optimization using partially homomorphic encryption," in *IEEE Conference on Decision and Control*, 2016, pp. 5053–5058.

[135] A. B. Alexandru, K. Gatsis, and G. J. Pappas, "Privacy preserving cloud-based quadratic optimization," in *Annual Allerton Conference on Communication, Control, and Computing*, 2017, pp. 1168–1175.

[136] A. B. Alexandru, K. Gatsis, Y. Shoukry, S. A. Seshia, P. Tabuada, and G. J. Pappas, "Cloud-based quadratic optimization with partially homomorphic encryption," *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2357–2364, 2021.

[137] J. Suh and T. Tanaka, "Encrypted value iteration and temporal difference learning over leveled homomorphic encryption," in *American Control Conference*, 2021, pp. 2555–2561.

[138] ——, "SARSA(0) reinforcement learning over fully homomorphic encryption," in *SICE International Symposium on Control Systems*, 2021, pp. 1–7.

[139] H. Kawase, W. Meinhold, C. Zeagler, T. P. Miles, and J. Ueda, "Encrypted classification for prevention of adversarial perturbation and individual identification in health-monitoring," in *International Conference on Advanced Intelligent Mechatronics*, 2023, pp. 1153–1158.

[140] K. Kogiso, "Upper-bound analysis of performance degradation in encrypted control system," in *American Control Conference*, 2018, pp. 1250–1255.

[141] R. Baba and K. Kogiso, "Consideration on robust stability of control systems with encryption-induced quantization errors," in *SICE Annual Conference*, 2019, pp. 1067–1069.

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

[143] K. Teranishi and K. Kogiso, "Dynamic quantizer for encrypted observer-based control," in *IEEE Conference on Decision and Control*, 2020, pp. 5477–5482.

[144] ——, "ElGamal-type encryption for optimal dynamic quantizer in encrypted control systems," *SICE Journal of Control, Measurement, and System Integration*, vol. 14, no. 1, pp. 59–66, 2021.

[145] H. Kawase, K. Teranishi, and K. Kogiso, "Dynamic quantizer synthesis for encrypted state-feedback control systems with partially homomorphic encryption," in *American Control Conference*, 2022, pp. 75–81.

[146] J. Kim, M. S. Darup, H. Sandberg, and K. H. Johansson, "Asymptotic stabilization over encrypted data with limited controller capacity and time-varying quantizer," in *IEEE Conference on Decision and Control*, 2022, pp. 7762–7767.

[147] J. H. Cheon, K. Han, H. Kim, J. Kim, and H. Shim, "Need for controllers having integer coefficients in homomorphically encrypted dynamic system," in *IEEE Conference on Decision and Control*, 2018, pp. 5020–5025.

[148] N. Schlüter and M. S. Darup, "On the stability of linear dynamic controllers with integer coefficients," *IEEE Transactions on Automatic Control*, vol. 67, no. 10, pp. 5610–5613, 2022.

[149] M. S. Tavazoei, "Pisot number-based discrete-time controllers with integer state matrices to ensure monotonic closed-loop step responses," *IEEE Transactions on Automatic Control*, 2023.

[150] K. Teranishi and K. Kogiso, "Control-theoretic approach to malleability cancellation by attacked signal normalization," *IFAC-PapersOnLine*, vol. 52, no. 20, pp. 297–302, 2019.

[151] J. Lee, J. Kim, and H. Shim, "Zero-dynamics attack on homomorphically encrypted control system," in *International Conference on Control, Automation and Systems*, 2020, pp. 385–390.

[152] A. M. Naseri, W. Lucia, and A. Youssef, "Confidentiality attacks against encrypted control systems," *Cyber-Physical Systems*, 2022.

[153] R. Alisic, J. Kim, and H. Sandberg, "Model-free undetectable attacks on linear systems using LWE-based encryption," *IEEE Control Systems Letters*, vol. 7, pp. 1249–1254, 2023.

[154] N. Shono, T. Miyazaki, K. Teranishi, K. Kogiso, and K. Kawashima, "A false data injection attack model targeting passivity of encrypted wave variable based bilateral control system," in *IEEE/SICE International Symposium on System Integration*, 2023.

[155] R. Baba, K. Kogiso, and M. Kishida, "Detection method of controller falsification attacks against encrypted control system," in *SICE Annual Conference*, 2018, pp. 244–248.

[156] D. Martynova and P. Zhang, "An approach to encrypted fault detection of cyber-physical systems," in *Asian Control Conference*, 2019, pp. 1501–1506.

[157] T. Shin, K. Teranishi, and K. Kogiso, "Cyber-secure pneumatic actuator system equipped with encrypted controller and attack detectors," *Advanced Robotics*, vol. 36, no. 9, pp. 438–449, 2022.

[158] A. B. Alexandru, L. Burbano, M. F. Çeliktuğ, J. Gomez, A. A. Cardenas, M. Kantarcioglu, and J. Katz, "Private anomaly detection in linear controllers: Garbled circuits vs. homomorphic encryption," in *IEEE Conference on Decision and Control*, 2022, pp. 7746–7753.

[159] K. Kogiso, "Attack detection and prevention for encrypted control systems by application of switching-key management," in *IEEE Conference on Decision and Control*, 2018, pp. 5032–5037.

[160] ——, "Toward dynamic key cryptography for cybersecurity enhancement in networked control systems," in *SICE Annual Conference*, 2019, pp. 669–672.

[161] J. H. Cheon, D. Kim, J. Kim, S. Lee, and H. Shim, "Authenticated computation of control signal from dynamic controllers," in *IEEE Conference on Decision and Control*, 2020, pp. 3249–3254.

[162] M. Fauser and P. Zhang, "Resilience of cyber-physical systems to covert attacks by exploiting an improved encryption scheme," in *IEEE Conference on Decision and Control*, 2020, pp. 5489–5494.

[163] ——, "Resilient homomorphic encryption scheme for cyber-physical systems," in *IEEE Conference on Decision and Control*, 2021, pp. 5634–5639.

[164] ——, "Detection of cyber attacks in encrypted control systems," *IEEE Control Systems Letters*, vol. 6, pp. 2365–2370, 2022.

[165] M. Miyamoto, K. Teranishi, K. Emura, and K. Kogiso, "Cybersecurity-enhanced encrypted control system using keyed-homomorphic public key encryption," *IEEE Access*, vol. 11, pp. 45 749–45 760, 2023.

[166] K. Ishikawa, K. Nagasawa, K. Kogiso, and K. Sawada, "Experimental validation of encrypted controller implemented on Raspberry Pi," in *IEEE International Conference on Cyber-Physical Systems, Networks, and Applications*, 2016, pp. 1–6.

[167] K. Kogiso, R. Baba, and M. Kusaka, "Development and examination of encrypted control systems," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2018, pp. 1338–1343.

[168] J. Tran, F. Farokhi, M. Cantoni, and I. Shames, "Digital implementation of homomorphically encrypted feedback control for cyber-physical systems," in *Workshop on Cyber-Physical Systems Security and Resilience*, 2019.

[169] ——, "Implementing homomorphic encryption based secure feedback control for physical systems," *Control Engineering Practice*, vol. 97, pp. 1–12, 2020.

[170] J. H. Cheon, K. Han, S. M. Hong, H. J. Kim, J. Kim, S. Kim, H. Seo, H. Shim, and Y. Song, "Toward a secure drone system: Flying with real-time homomorphic authenticated encryption," *IEEE Access*, vol. 6, pp. 24 325–24 339, 2018.

[171] K. Teranishi, N. Shimada, and K. Kogiso, "Development and examination of fog computing-based encrypted control system," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4642–4648, 2020.

[172] H. B. Kwon, S. Kosieradzki, J. Blevins, and J. Ueda, "Encrypted coordinate transformation via parallelized somewhat homomorphic encryption for robotic teleoperation," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2023, pp. 228–233.

[173] H. Gao, C. Zhang, M. Ahmad, and Y. Wang, "Privacy-preserving average consensus on directed graphs using push-sum," in *IEEE Conference on Communications and Network Security*, 2018.

[174] M. Kishida, "Encrypted average consensus with quantized control law," in *IEEE Conference on Decision and Control*, 2018, pp. 5850–5856.

[175] M. Ruan, H. Gao, and Y. Wang, "Secure and privacy-preserving consensus," *IEEE Transactions on Automatic Control*, vol. 64, no. 10, pp. 4035–4049, 2019.

[176] C. N. Hadjicostis and A. D. Domínguez-García, "Privacy-preserving distributed averaging via homomorphically encrypted ratio consensus," *IEEE Transactions on Automatic Control*, vol. 65, no. 9, pp. 3887–3894, 2020.

[177] H. Kawase, K. Sakurama, and K. Kogiso, "Implementation of consensus control using encrypted distributed controller," in *International Symposium on Artificial Life and Robotics, International Symposium on BioComplexity, International Symposium on Swarm Behavior and Bio-Inspired Robotics*, 2022, pp. 1602–1607.

[178] P. M. Chaher, B. Jayawardhana, and J. Kim, "Homomorphic encryption-enabled distance-based distributed formation control with distance mismatch estimators," in *IEEE Conference on Decision and Control*, 2021, pp. 4915–4922.

[179] M. Marcantoni, B. Jayawardhana, M. P. Chaher, and K. Bunte, "Secure formation control via edge computing enabled by fully homomorphic encryption and mixed uniform-logarithmic quantization," *IEEE Control Systems Letters*, vol. 7, pp. 395–400, 2023.

[180] A. B. Alexandru, M. S. Darup, and G. J. Pappas, "Encrypted cooperative control revisited," in *IEEE Conference on Decision and Control*, 2019, pp. 7196–7202.

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

[182] J. Kim and H. Shim, "Encrypted state estimation in networked control systems," in *IEEE Conference on Decision and Control*, 2019, pp. 7190–7195.

[183] N. Schlüter, P. Binfet, J. Kim, and M. S. Darup, "Encrypted distributed state estimation via affine averaging," in *IEEE Conference on Decision and Control*, 2022, pp. 7754–7761.

[184] W. Ding, W. Yang, J. Zhou, L. Shi, and G. Chen, "Privacy preserving via secure summation in distributed Kalman filtering," *IEEE Transactions on Control of Network Systems*, vol. 9, no. 3, pp. 1481–1492, 2022.

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

[186] K. Tjell, I. Cascudo, and R. Wisniewski, "Privacy preserving recursive least squares solutions," in *European Control Conference*, 2019, pp. 3490–3495.

[187] K. Tjell and R. Wisniewski, "Privacy preservation in distributed optimization via dual decomposition and ADMM," in *IEEE Conference on Decision and Control*, 2019, pp. 7203–7208.

[188] C. Zhang, M. Ahmad, and Y. Wang, "ADMM based privacy-preserving decentralized optimization," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 3, pp. 565–580, 2019.

[189] C. Zhang and Y. Wang, "Enabling privacy-preservation in decentralized optimization," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 2, pp. 679–689, 2019.

[190] K. Tjell and R. Wisniewski, "Private aggregation with application to distributed optimization," *IEEE Control Systems Letters*, vol. 5, no. 5, pp. 1591–1596, 2021.

[191] A. B. Alexandru, A. Tsiamis, and G. J. Pappas, "Encrypted distributed lasso for sparse data predictive control," in *IEEE Conference on Decision and Control*, 2021, pp. 4901–4906.

[192] A. B. Alexandru and G. J. Pappas, "Private weighted sum aggregation for distributed control systems," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 11 081–11 088, 2020.

[193] D. Lee, J. Kim, and H. Shim, "Distributed aggregation over homomorphically encrypted data under switching networks," in *IEEE Conference on Decision and Control*, 2020, pp. 5495–5500.

[194] K. Tjell and R. Wisniewski, "Privacy preserving distributed summation in a connected graph," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 3445–3450, 2020.

[195] A. B. Alexandru and G. J. Pappas, "Private weighted sum aggregation," *IEEE Transactions on Control of Network Systems*, vol. 9, no. 1, pp. 219–230, 2022.

[196] D. Boneh, K. Lewi, H. Montgomery, and A. Raghunathan, "Key homomorphic PRFs and their applications," in *Advances in Cryptology – CRYPTO 2013*, ser. Lecture Notes in Computer Science, R. Canetti and J. A. Garay, Eds. Berlin, Heidelberg: Springer, 2013, pp. 410–428.

[197] E. Barker, "Recommendation for key management: Part 1 – General," National Institute of Standards and Technology, Tech. Rep. NIST Special Publication (SP) 800-57 Part 1 Rev. 5, May 2020. [Online]. Available: https://csrc.nist.gov/pubs/sp/800/57/pt1/r5/final

[198] H. Sandberg, A. Teixeira, and K. H. Johansson, "On security indices for state estimators in power networks," in *First Workshop on Secure Control Systems*, 2010.

[199] M. S. Chong and M. Kuijper, "Characterising the vulnerability of linear control systems under sensor attacks using a system's security index," in *IEEE Conference on Decision and Control*, 2016, pp. 5906–5911.

[200] H. Sandberg and A. M. Teixeira, "From control system security indices to attack identifiability," in *Science of Security for Cyber-Physical Systems Workshop*, 2016, pp. 1–6.

[201] C. Murguia, N. V. D. Wouw, and J. Ruths, "Reachable sets of hidden CPS sensor attacks: Analysis and synthesis tools," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 2088–2094, 2017.

[202] S. Gracy, J. Milošević, and H. Sandberg, "Actuator security index for structured systems," in *American Control Conference*, 2020, pp. 2993–2998.

[203] J. Milošević, A. Teixeira, K. H. Johansson, and H. Sandberg, "Actuator security indices based on perfect undetectability: Computation, robustness, and sensor placement," *IEEE Transactions on Automatic Control*, vol. 65, no. 9, pp. 3816–3831, 2020.

[204] C. Murguia, I. Shames, J. Ruths, and D. Nešić, "Security metrics and synthesis of secure control systems," *Automatica*, vol. 115, p. 108757, 2020.

[205] S. Gracy, J. Milošević, and H. Sandberg, "Security index based on perfectly undetectable attacks: Graph-theoretic conditions," *Automatica*, vol. 134, p. 109925, 2021.

[206] L. Zhai, K. G. Vamvoudakis, and J. Hugues, "A graph-theoretic security index based on undetectability for cyber-physical systems," in *American Control Conference*, 2022, pp. 1479–1484.

[207] J. Katz and Y. Lindell, *Introduction to modern cryptography*. CRC Press, 2020.

[208] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *Journal of the ACM*, vol. 56, no. 6, 2009.

[209] D. Boneh, "The decision Diffie-Hellman problem," in *Algorithmic Number Theory*, ser. Lecture Notes in Computer Science, J. P. Buhler, Ed. Berlin, Heidelberg: Springer, 1998, pp. 48–63.

[210] M. R. Albrecht, R. Player, and S. Scott, "On the concrete hardness of learning with errors," *Journal of Mathematical Cryptology*, vol. 9, no. 3, pp. 169–203, 2015.

[211] K. Emura, G. Hanaoka, G. Ohtake, T. Matsuda, and S. Yamada, "Chosen ciphertext secure keyed-homomorphic public-key encryption," in *Public-Key Cryptography – PKC 2013*, ser. Lecture Notes in Computer Science, K. Kurosawa and G. Hanaoka, Eds. Berlin, Heidelberg: Springer, 2013, pp. 32–50.

[212] A. Everspaugh, K. Paterson, T. Ristenpart, and S. Scott, "Key rotation for authenticated encryption," in *Advances in Cryptology – CRYPTO 2017*, ser. Lecture Notes in Computer Science, J. Katz and H. Shacham, Eds. Cham: Springer International Publishing, 2017, pp. 98–129.

[213] K. Cohn-Gordon, C. Cremers, and L. Garratt, "On post-compromise security," in *IEEE Computer Security Foundations Symposium*, 2016, pp. 164–178.

[214] A. Lehmann and B. Tackmann, "Updatable encryption with post-compromise security," in *Advances in Cryptology – EUROCRYPT 2018*, ser. Lecture Notes in Computer Science, J. B. Nielsen and V. Rijmen, Eds. Cham: Springer International Publishing, 2018, pp. 685–716.

[215] M. Klooß, A. Lehmann, and A. Rupp, "(R)CCA Secure updatable encryption with integrity protection," in *Advances in Cryptology – EUROCRYPT 2019*, ser. Lecture Notes in Computer Science, Y. Ishai and V. Rijmen, Eds. Cham: Springer International Publishing, 2019, pp. 68–99.

[216] C. Boyd, G. T. Davies, K. Gjøsteen, and Y. Jiang, "Fast and secure updatable encryption," in *Advances in Cryptology – CRYPTO 2020*, ser. Lecture Notes

in Computer Science, D. Micciancio and T. Ristenpart, Eds. Cham: Springer International Publishing, 2020, pp. 464–493.

[217] C. Scherer and S. Weiland, "Linear matrix inequalities in control," Delft, The Netherlands, 2015.

[218] T. R. V. Steentjes, M. Lazar, and P. M. J. V. d. Hof, "Scalable distributed and decentralized $\mathscr{H}_2$ controller synthesis for interconnected linear discrete-time systems," Jan. 2021. [Online]. Available: http://arxiv.org/abs/2001.04875

[219] K. Johansson, "The quadruple-tank process: A multivariable laboratory process with an adjustable zero," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 3, pp. 456–465, 2000.

[220] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2909–2913, 2016.

[221] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, "A rewriting system for convex optimization problems," *Journal of Control and Decision*, vol. 5, no. 1, pp. 42–60, 2018.

List of Publications

1. **K. Teranishi**, N. Shimada, and K. Kogiso, "Stability-guaranteed dynamic ElGamal cryptosystem for encrypted control systems," *IET Control Theory & Applications*, vol. 14, iss. 16, pp. 2242–2252, 2020. (Chapter 2)

2. **K. Teranishi**, T. Sadamoto, A. Chakrabortty, and K. Kogiso, "Designing optimal key lengths and control laws for encrypted control systems based on sample identifying complexity and deciphering time," *IEEE Transactions on Automatic Control*, vol. 68, no. 4, pp. 2183–2198, 2023. (Chapters 2, 3, 4, and 5)

3. **K. Teranishi** and K. Kogiso, "Optimal controller and security parameter for encrypted control systems under least squares identification," *IEEE Control Systems Letters*, vol. 7, pp. 1482–1487, 2023. (Chapters 2 and 5)

4. **K. Teranishi** and K. Kogiso, "Optimal security parameter for encrypted control systems against eavesdropper and malicious server," *SICE Journal of Control, Measurement, and System Integration*, vol. 16, iss. 1, pp. 203–214, 2023. (Chapters 2 and 5)

Journal papers

1. **K. Teranishi**, N. Shimada, and K. Kogiso, "Development and examination of fog computing-based encrypted control system," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4642–4648, 2020.

2. **K. Teranishi**, N. Shimada, and K. Kogiso, "Stability-guaranteed dynamic ElGamal cryptosystem for encrypted control systems," *IET Control Theory & Applications*, vol. 14, iss. 16, pp. 2242–2252, 2020.

3. **K. Teranishi** and K. Kogiso, "ElGamal-type encryption for optimal dynamic quantizer in encrypted control systems," *SICE Journal of Control, Measurement, and System Integration*, vol. 14, no. 1, pp. 59–66, 2021.

4. T. Shin, **K. Teranishi**, and K. Kogiso, "Cyber-secure pneumatic actuator system equipped with encrypted controller and attack detectors," *Advanced Robotics*, vol. 36, iss. 9, pp. 438–449, 2022.

5. **K. Teranishi**, T. Sadamoto, A. Chakrabortty, and K. Kogiso, "Designing optimal key lengths and control laws for encrypted control systems based on sample identifying complexity and deciphering time," *IEEE Transactions on Automatic Control*, vol. 68, no. 4, pp. 2183–2198, 2023.

6. **K. Teranishi** and K. Kogiso, "Optimal controller and security parameter for encrypted control systems under least squares identification," *IEEE Control Systems Letters*, vol. 7, pp. 1482–1487, 2023.

7. **K. Teranishi** and K. Kogiso, "Optimal security parameter for encrypted control systems against eavesdropper and malicious server," *SICE Journal of Control, Measurement, and System Integration*, vol. 16, iss. 1, pp. 203–214, 2023.

8. **K. Teranishi**, T. Sadamoto, and K. Kogiso, "Input-output history feedback controller for encrypted control with leveled fully homomorphic encryption," *IEEE Transactions on Control of Network Systems*, 2023. (early access)

9. M. Miyamoto, **K. Teranishi**, K. Emura, and K. Kogiso, "Cybersecurity-enhanced encrypted control system using keyed-homomorphic public key encryption," *IEEE Access*, vol. 11, pp. 45749–45760, 2023.

Conference papers

1. **K. Teranishi**, M. Kusaka, N. Shimada, J. Ueda, and K. Kogiso, "Secure observer-based motion control based on controller encryption," *American Control Conference*, 2019, pp. 2978–2983.

2. **K. Teranishi** and K. Kogiso, "Control-theoretic approach to malleability cancellation by attacked signal normalization," *IFAC Workshop on Distributed Estimation and Control in Networked Systems*, IFAC-PapersOnLine, vol. 52, no. 20, pp. 297–302, 2019.

3. **K. Teranishi**, N. Shimada, and K. Kogiso, "Stability analysis and dynamic quantizer for controller encryption," *IEEE Conference on Decision and Control*, 2019, pp. 7184–7189.

4. **K. Teranishi**, K. Kogiso, and J. Ueda, "Encrypted feedback linearization and motion control for manipulator with somewhat homomorphic encryption," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2020, pp. 613–618.

5. **K. Teranishi**, J. Ueda, and K. Kogiso, "Event-triggered approach to increasing sampling period of encrypted control systems," *IFAC World Congress*, IFAC-PapersOnLine, vol. 53, no. 2, pp. 3502–3507, 2020.

6. **K. Teranishi** and K. Kogiso, "Dynamic quantizer for encrypted observer-based control," *IEEE Conference on Decision and Control*, 2020, pp. 5477–5482.

7. **K. Teranishi** and K. Kogiso, "Encrypted gain scheduling with quantizers for stability guarantee," *IEEE Conference on Decision and Control*, 2021, pp. 5621–5626.

8. N. Shono, T. Miyazaki, **K. Teranishi**, T. Kanno, T. Kawase, K. Kogiso, and K. Kawashima, "Implementation of encrypted control of pneumatic bilateral control system using wave variables," *International Symposium on Artificial Life and Robotics, International Symposium on BioComplexity, International Symposium on Swarm Behavior and Bio-Inspired Robotics*, 2022, pp. 1169–1174.

9. H. Kawase, **K. Teranishi**, and K. Kogiso, "Dynamic quantizer synthesis for encrypted state-feedback control systems with partially homomorphic encryption," *American Control Conference*, 2022, pp. 75–81.

10. **K. Teranishi** and K. Kogiso, "Towards provably secure encrypted control using homomorphic encryption," *IEEE Conference on Decision and Control*, 2022, pp. 7740–7745.

11. N. Shono, T. Miyazaki, **K. Teranishi**, K. Kogiso, and K. Kawashima, "A false data injection attack model targeting passivity of encrypted wave variable based bilateral control system," *IEEE/SICE International Symposium on System Integrations*, 2023, pp. 992–997.

12. H. Takanashi, **K. Teranishi**, and K. Kogiso, "Experimental validation of reaction force estimation for secure robot teleoperation," *IEEE/SICE International Symposium on System Integrations*, 2023, pp. 1004–1007.

13. **K. Teranishi** and K. Kogiso, "Optimal controller and security parameter for encrypted control systems under least squares identification," *IEEE Conference on Decision and Control*, 2023. (accepted)