Towards Encrypted Cloud-Based Control-as-a-Service

Der Fakultät Maschinenbau der Universität Dortmund
vorgelegte Abhandlung zur Erlangung des
akademischen Grades eines Dr.-Ing.

von

Nils Schlüter

aus Mettingen

Erstgutachter: Prof. Dr.-Ing. Moritz Schulze Darup
Zweitgutachter: Prof. Dr. Riccardo Ferrari
Dissertation eingereicht am: 29.04.2024
Tag der mündlichen Prüfung: 07.11.2024

Lehrstuhl für Regelungstechnik und cyberphysische Systeme

Universität Dortmund

2024

Vorwort

Diese Arbeit ist während meiner Zeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Regelungstechnik und cyberphysische Systeme der TU Dortmund entstanden. Von Herzen möchte ich meinem Doktorvater, Prof. Dr. Moritz Schulze Darup, nicht nur für die hervorragende Zusammenarbeit an spannenden interdisziplinären Themen danken. Auch Herrn Prof. Dr. Ferrari möchte ich meinen herzlichen Dank für das Interesse an meiner Arbeit und der Teilnahme am Promotionsausschuss aussprechen. Herrn Prof. Dr. Daniel Quevedo danke ich für die (leider viel zu kurze) Zeit an seinem Lehrstuhl und den ausgezeichneten Kaffee. Ebenfalls danke ich Prof. Dr. Junsoo Kim, Prof. Dr. Henrik Sandberg und Prof. Dr. Karl Johansson, mir einen Aufenthalt an der KTH in Stockholm ermöglicht zu haben.

Obschon Cybersicherheit in den vergangenen Jahren einen größer werdenden Stellenwert in der Regelungstechnik eingenommen hat, ist man als "Kryptograph" teilweise fachlich isoliert. Demgegenüber stand jedoch die Warmherzigkeit unzähliger Menschen, die an dieser Stelle nicht alle genannt werden können und meine Zeit als Doktorand unvergesslich gemacht haben. Ich bin überaus dankbar für die Diskussionen und die unterhaltsamen Konferenzaufenthalte mit meinen Kollegen: Dieter Teichrieb, Manuel Klädtke, Matthias Neuhaus, Katrine Tjell, Johannes van Randenborgh, Janis Adamek und Teimour Hosseinalizadeh. Ein besonderer Dank geht an Philipp Binfet für die Zusammenarbeit bei vielen Beiträgen und an Sebastian Schlor für unsere wöchentlichen Besprechungen, die stets zu neuen Ideen geführt haben. In diesem Zusammenhang möchte ich auch Prof. Dr. Tibor Jager, Jonas von der Heyden und Martin Asman für neue Einflüsse und die erfolgreiche Kooperation danken.

Zu guter Letzt danke ich meiner Familie, meiner wunderbaren Frau Alina und meinen Söhnen Oskar und Juri für die fortwährende Unterstützung jenseits von fachlichen Herausforderungen.

Kurzfassung

Die zunehmende Vernetzung von Regelungssystemen eröffnet neue Möglichkeiten, wie etwa cloudbasierte Regelungsservices. Gleichzeitig sind vernetzte Systeme anfällig für Cyberangriffe und Datenlecks. Diese Dissertation zielt daher darauf ab, die Sicherheit cyber-physischer Systeme zu erhöhen, indem die Vertraulichkeit von Prozessdaten während ihrer Auswertung gewährleistet wird. Zu diesem Zweck werden fortschrittliche kryptographische Techniken wie homomorphe Verschlüsselung und Secure Multi-Party Computation eingesetzt, welche Berechnungen auf verschlüsselten Daten ermöglichen. Bei der Entwicklung neuartiger verschlüsselter Regelungen zeigen sich jedoch einzigartige interdisziplinäre Herausforderungen, die Kenntnisse aus der Kryptographie und der Regelungstheorie erfordern.

Der erste Abschnitt dieser Arbeit widmet sich iterativen Regelungsalgorithmen. Zunächst werden cloudbasierte dynamische Regler betrachtet. Diese nutzen eine lineare Iteration, welche bei verschlüsselter Auswertung zu einem Zahlenüberlauf führen kann. Wie sich mittels systemtheoretischer Analyse zeigt, ist eine Untergruppe von dynamischen Reglern, die sich effizient verschlüsseln lässt, nicht von diesem Problem betroffen. Zusätzlich werden Konsensprobleme im Bereich verteilter Systeme untersucht. Innerhalb der zugehörigen Algorithmen werden sowohl Iterationen als auch die Zugriffskontrolle auf Daten adressiert. Im nächsten Teil der Arbeit stehen Regelungen im Fokus, die nicht-polynomielle Funktionen nutzen, wobei modellprädiktive Regelungen von besonderem Interesse sind. Verschlüsselte Implementierungen solcher Funktionen stellen sich oftmals als ineffizient heraus. Ein Ausweg bieten neuronale max-out Netze, die in der Lage sind, beliebige kontinuierliche Funktionen zu approximieren. Außerdem lassen sich diese Netze effizient mittels Secure Multi-Party Computation auswerten. Abschließend behandelt der letzte Teil der Arbeit eine Sicherheitsanalyse von zufälligen affinen Transformationen. Diese Transformationen erfreuen sich seit Kurzem großer Popularität, bieten aber derzeit keine technisch fundierte Sicherheitsgarantie.

Die zuvor genannten Resultate werden stets mithilfe numerischer Analysen validiert und anhand relevanter Beispiele veranschaulicht. Obwohl bereits bedeutende Fortschritte erzielt werden konnten, sind weitere Effizienzsteigerungen für die praktische Anwendung von verschlüsselter Regelungstechnik von größter Bedeutung.

Abstract

The increasing interconnectivity of control systems paves the way for exciting new opportunities, such as cloud-based control services. At the same time, connected control systems are susceptible to cyberattacks and data leakage. Therefore, this dissertation aims to enhance the security of cyberphysical systems by ensuring the confidentiality of control data during its evaluation. To achieve this, advanced cryptographic techniques such as homomorphic encryption and secure multi-party computation are utilized, which enable computations on encrypted data. However, the development of novel encrypted controllers results in unique interdisciplinary challenges that necessitate a nuanced approach, blending elements of cryptography and control theory.

The first section of this thesis is devoted to iterative control algorithms. First, cloud-based dynamic controllers are considered. These are based on a linear iteration, which can lead to a numerical overflow in an encrypted evaluation. Our system-theoretic analysis reveals that a specific subset of dynamic controllers, which can be efficiently encrypted, is not affected by this problem. Second, privacy in consensus problems of multi-agent systems is explored. In the related algorithms, we address iterations and a mechanism to control the data access. The subsequent section of this thesis focuses on controllers that use non-polynomial functions, where model predictive control is a prominent example. Encrypted implementations of these functions are often inefficient. Our novel approach uses max-out neural networks, which form a versatile basis to approximate any continuous function. Moreover, they are well-aligned with primitives from secure multi-party computation, resulting in an efficient implementation. The final section conducts a security analysis of random affine transformations. Although these transformations have gained significant popularity, they currently do not offer a technically sound security guarantee.

The aforementioned results are numerically validated and illustrated with relevant examples. Although significant progress has been made, further efficiency improvements are paramount for the practical application of encrypted control.

Contents

I Preliminaries and Results

II Articles

Notation

In the following, we list the symbols and acronyms used in Part I of this thesis. The articles in Part II are self-contained and may use slightly different notation, which is explained within each article.

Acronyms and abbreviations

ADMM	alternating direction method of multipliers	LWE	learning with errors
CCA	chosen-ciphertext attack	MPC	model predictive control
CKKS	Cheon-Kim-Kim-Song	OT	oblivious transfer
COA	ciphertext-only attack	PHE	partially homomorphic encryption
CPA	chosen-plaintext attack	PPT	probabilistic polynomial time
CPS	cyberphysical system	PWA	piecewise affine
DP	differential privacy	RAT	random affine transformation
FHE	fully homomorphic encryption	RLWE	ring learning with errors
FIR	finite impulse response	SMPC	secure multi-party computation
GC	garbled circuit	SS	secret sharing
HE	homomorphic encryption	s.t.	such that
IND	indistinguishability	TTP	trusted third party
KPA	known-plaintext attack	ZKP	zero-knowledge proof

Control-related symbols

Roman letters

h	characteristic polynomial coefficient
k	discrete time-step
l	output dimension
m	input dimension or global decision variable dimension
n	state dimension
p	number of preactivations
J	local cost function
N_p	prediction horizon
T	controller reset period

$\boldsymbol{b}$	feedback gain vector
$\boldsymbol{c}$	gradient vector
$\boldsymbol{d}, \boldsymbol{e}$	max-out neural network parameter vectors
$\boldsymbol{g}$	affine averaging parameter vector
$\boldsymbol{p}$	distributed optimization parameter vector
$\boldsymbol{s}$	inequality constraint vector
$\boldsymbol{u}$	control input vector
$\boldsymbol{x}$	system state, controller state, or local decision variable vector
$\boldsymbol{y}$	measurement vector
$\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$	system state, input, and output matrix
$\boldsymbol{E}, \boldsymbol{F}, \boldsymbol{G}, \boldsymbol{H}$	controller output, feedthrough, input, and state matrix
$\boldsymbol{I}$	identity matrix
$\boldsymbol{K}$	feedback gain matrix
$\boldsymbol{L}, \boldsymbol{M}$	max-out neural network parameter matrices
$\boldsymbol{P}$	inequality constraint matrix
$\boldsymbol{S}, \boldsymbol{T}$	equality constraint matrices
$\boldsymbol{Q}$	Hessian matrix
$\boldsymbol{W}$	weight matrix
$\boldsymbol{Y}$	matrix with rows representing bit decompositions

Greek letters

θ	number of segments in a piecewise affine function
λ	eigenvalue or characteristic polynomial parameter
ρ	optimization step size in ADMM
$\boldsymbol{\nu}$	vector of Lagrange multipliers
$\boldsymbol{\zeta}$	global decision variable vector
$\boldsymbol{\Gamma}$	parameter matrix in ADMM

Calligraphic letters

$\mathcal{B}$	set of neighboring polyhedra
$\mathcal{I}, \mathcal{K}, \mathcal{L}$	index sets for agents or polynomials
$\mathcal{N}$	set of neighboring agents
$\mathcal{P}$	a polyhedron
$\mathcal{T}$	set of terminal constraints
$\mathcal{U}$	set of input constraints
$\mathcal{V}$	set of convex folds in a convex decomposition
$\mathcal{X}$	set of state constraints
$\mathcal{Y}$	set of concave folds in a convex decomposition
$\boldsymbol{\mathcal{F}}$	FIR controller feedback gain matrix

Cryptographic symbols

Roman letters

a	random polynomial $a(X)$ in an RLWE instance
b	bit or ciphertext component in an LWE (RLWE) instance
e	error e (error polynomial $e(X)$) in a LWE (RLWE) instance
f	function over the message space
g	function over the plaintext space
ℓ	level in CKKS cryptosystem or random label in GC
m	input dimension
p	prime number
q	modulus
$r, r_{\max}$	random number, uniform distribution bound in the RAT scheme
s	scaling factor for a fixed-point encoding
t	reconstruction threshold in SMPC
v	encoded value
x	plaintext value
y	protocol output in SMPC
z	message (often an encoded state or measurement)
C	random variable for security definitions
I	number of wrap-arounds modulo q
N	key length (ring dimension) in an LWE (RLWE) instance
M	number of parties participating in a SMPC protocol
P	large positive integer in the CKKS cryptosystem
P	party in SMPC
Q	large modulus for a CKKS bootstrapping
$R_{\max}$	uniform distribution bound in the RAT scheme
X	polynomial variable
Y, Z	random variables for security definitions
$\boldsymbol{a}$	random vector in an LWE instance
$\boldsymbol{r}$	random vector in the RAT scheme
$\boldsymbol{v}$	message vector (often an encoded control input)
$\boldsymbol{x}$	plaintext vector (often a state)
$\boldsymbol{y}$	ciphertext in the RAT scheme
$\boldsymbol{R}$	random matrix in the RAT scheme
$\boldsymbol{V}$	Vandermonde matrix in the CKKS encoding

Greek letters

ζ	root of unity in the CKKS encoding
α, β, γ	multiplication triples in SMPC
δ	multiplication protocol message in SMPC
ϵ	rounding error or a message in the multiplication protocol
κ	security parameter
ρ	automorphism for rotations in the CKKS cryptosystem
Φ	cyclotomic polynomial
$\boldsymbol{\xi}, \boldsymbol{\omega}$	input vectors for a GC

Calligraphic letters

$\mathcal{A}$	adversary
$\mathcal{C}$	ciphertext space
$\mathcal{D}$	set or distribution to be specified
$\mathcal{H}$	set of adversaries in SMPC
$\mathcal{P}$	plaintext space
$\mathcal{S}$	simulator in SMPC
$\mathcal{Z}$	message space

Multi-letter symbols

1^κ	input string of length κ
$\mathsf{ct}(z)$	ciphertext of z
ek	encryption key
evk	evaluation key in the CKKS cryptosystem
$\gcd(a, b)$	greatest common divisor of integers a and b
$\mathsf{negl}(\kappa)$	negligible function of κ
msg	message in SMPC
pk, sk	public, secret key
swk	switching key in the CKKS cryptosystem
view	view of a party in SMPC
$\mathsf{Add}, \oplus$	addition algorithm or protocol
Dcd	decoding algorithm
Dec	decryption algorithm
Ecd	encoding algorithm
Enc	encryption algorithm
Eval	evaluation algorithm or protocol
$\mathsf{Mult}, \otimes, \times$	multiplication algorithm or protocol
OT_2^1	1-out-of-2 oblivious transfer protocol
$\mathsf{PAdd}, \boxplus$	public addition algorithm or protocol
$\mathsf{PMult}, \boxtimes$	public multiplication algorithm or protocol

Mathematical notation

Algebra

$\mathbb{C}$	complex numbers
$\mathbb{N}$	natural numbers
$\mathbb{Q}$	rational numbers
$\mathbb{R}$	real numbers
$\mathbb{Z}$	integer numbers
$\mathbb{Z}_q$	set, ring, or field of integers modulo q
$\mathbb{Z}_q^*$	multiplicative group of integers modulo q
$\mathcal{R}$	cyclotomic ring of polynomials $\mathbb{Z}[X]/(X^N+1)$ with elements $a(X) = \sum_{i=0}^{N-1} \tilde{a}_i X^i \pmod{X^N+1}$
$\mathcal{R}_q$	quotient ring $\mathcal{R}/q\mathcal{R} = \mathbb{Z}_q[X]/(X^N+1)$

Symbols and operations

$\perp$	indicates no output
$\lfloor\cdot\rceil, \lfloor\cdot\rfloor, \lceil\cdot\rceil$	component-wise rounding, flooring, and ceiling
i, j	integer index variables
$a \leftarrow \mathcal{D}$	sampling uniformly at random from set or distribution $\mathcal{D}$
$\lvert a\rvert$	absolute value of a
$a \lvert\rvert b$	concatenation of a and b
$\bmod q$	reduction of an integer modulo q
$a = b \pmod q$	abbreviation for $a \bmod q = b \bmod q$
$a^{-1} \bmod q$	multiplicative inverse of $a \in \mathbb{Z}_q$
$z^{(i)}$	i-th share of z
$[z]$	additive shares of z
$D(Y_1, Y_2)$	statistical distance between random variables Y_1 and Y_2
$Y_1 \overset{c}{\equiv} Y_2$	Y_1 and Y_2 are computationally indistinguishable
$\Pr(Y = r)$	probability that random variable Y takes value r
$\mathbf{1}$	vector of all ones, i.e., $(1, 1, \ldots, 1)^\top$
$\bar{\boldsymbol{a}}$	component-wise complex conjugation
$\boldsymbol{a} \leqslant \boldsymbol{b}$	component-wise inequality
$\boldsymbol{a} \circ \boldsymbol{b}$	component-wise multiplication (Hadamard product)
$\lVert\boldsymbol{a}\rVert_{\boldsymbol{W}}$	weighted 2-norm, i.e., $\sqrt{\boldsymbol{a}^\top \boldsymbol{W} \boldsymbol{a}}$
$\lVert a(X)\rVert_\infty, \lVert\boldsymbol{a}\rVert_\infty$	∞-norm, i.e., $\lVert a(X)\rVert_\infty = \max_i \lvert\tilde{a}_i\rvert$, $\lVert\boldsymbol{a}\rVert_\infty = \max_i \lvert a_i\rvert$
$\mu(a)$	partial inverse of the modular reduction
$\sigma(a)$	sign bit of a, i.e., 0 if $0 \leqslant a$, 1 otherwise
$\mathcal{A}(Y)$	adversary receives samples of Y
$\mathcal{O}(N)$	asymptotic time complexity linear in N
$\max\{\cdot\}$	largest element of a set

List of Figures

List of Tables

Part I

Preliminaries and Results

Chapter 1

Introduction

In this chapter, we will motivate the thesis, provide an extensive review of the literature, and outline the content.

1.1 Motivation

Control systems form the backbone of automation and smart technologies such as the Internet of Things [35]. At its core, control embodies a decision-making process that provides inputs to a system such that it follows a desired behavior. Central to this approach is the use of data that is most predominantly used in a feedback loop. The power of control lies in its abstraction, which facilitates its ubiquitous application across diverse fields. Due to the seamless operation of a system, control often goes unnoticed despite its pivotal role as an enabling technology.

The growing interconnectivity in control systems due to robust wireless communication and cloud usage [198] paves the way for leveraging large amounts of data, pooling and outsourcing resources, achieving rapid scalability, and simplifying maintenance. Moreover, service-based control solutions that can provide state-of-the-art controllers without the need for expert knowledge become conceivable. This lowers costs and facilitates a future where data-driven and distributed control unfold their full potential by enabling more complex tasks while providing efficiency and resilience. The combination of these benefits can profoundly impact various domains, such as industrial automation, transportation, and energy [81].

In control systems, safety is not an option; it is a necessity. This is because the line between suitable system behavior and catastrophic malfunction can be thin. As a result, safety became one of the most influential concepts in control and presents itself in the form of stability, robustness, fault tolerance, reliability, or resilience. However, with increasing connectivity, considering safety only in terms of these classical control notions falls short. More precisely, connected control systems are embedded in an IT network, which inevitably results in vulnerabilities to cybersecurity threats. For example, a characteristic property of interconnected/networked control systems is that data is processed externally on not necessarily trustworthy platforms, such as cloud servers or

agents. Furthermore, even seemingly isolated control systems, e.g., for critical infrastructure, became targets of tailored malware attacks, where data theft, data erasure, and unauthorized control access are common functionalities (see [11, 224] for detailed overviews). While most of such attacks are sophisticated, also incorrectly configured devices can become a gateway for an attacker [47]. Possible outcomes of attacks include loss of safety, data, life, production, and competitive advantage, which can affect society and the environment. Due to their real-world implications, these threats are paramount in cyberphysical systems (CPSs) such that connected control systems' safety crucially depends on their security.

1.2 Methods overview

Cybersecurity threats are commonly categorized into *confidentiality*, *integrity*, and *availability* threats [29]. Confidentiality focuses on data privacy, integrity ensures that data remain unchanged without authorization or detection, and availability pertains to the accessibility of data and services. Different techniques can be pursued to contend with these imminent threats.

Control-related methods

One appeal of control-based methods is a simple integration into existing systems and no communication overhead because control signals are directly altered. In this context, legacy control systems are of particular interest. These are vulnerable due to a lack of basic security features, stemming from their long-lived and infrequently updated hardware. Therefore, it is commonly assumed that the communication network (often a fieldbus) is insecure, allowing attackers to eavesdrop on and maliciously interfere with sensor and/or actuator signals. To avoid easy detection, sophisticated attackers aim for stealthiness, as a detector can quickly identify those merely causing damage. This gave rise to the notions of *zero-dynamic* [185], *covert* [220], or *bias injection attacks* [231] which differ in several aspects, such as impact, signal access, and required knowledge. For instance, covert attacks use sensor and actuator signals, giving an attacker complete control over the plant, but they require profound system knowledge. Detection and impact limitation strategies for such stealthy attacks are discussed in [231, 239]. Another type of attack is *replay attacks* [169, 170]. These evade detection systems by storing authentic data and replaying it later, which opens the feedback loop. The detection of replay attacks (and other stealthy attacks) is addressed in [84, 87] by (multiplicative) watermarking without degrading the closed-loop performance. Finally, loss of availability via *denial-of-service attacks* has been investigated earlier in networked control (see, e.g., [14, 44, 68]). However, preventing attacks on the availability instead of dealing with their ramifications is attributed to the field of computer science.

Cryptographic methods

Although studying system-theoretic perspectives on cyberattacks and defenses is insightful, standard cryptography can guarantee confidentiality and integrity during communications. More precisely, this is achieved by establishing *shared secrets* [197] that are subsequently used to *encrypt* [62] and *authenticate* [21] messages. Additional cryptographic nonces or timestamps can be used for the detection of replay attacks [125, p.112]. Moreover, these methods are highly efficient, except for the one-time establishment of shared secrets. Still, integrating cryptography into an existing plant can pose practical challenges such as incompatibility with the currently used communication protocol.

Nevertheless, assuming secure communications narrows the potential targets for an attack down to devices and platforms where data is processed, as classical cryptography requires decryption (cf. Figure 1.1 left). Therefore,

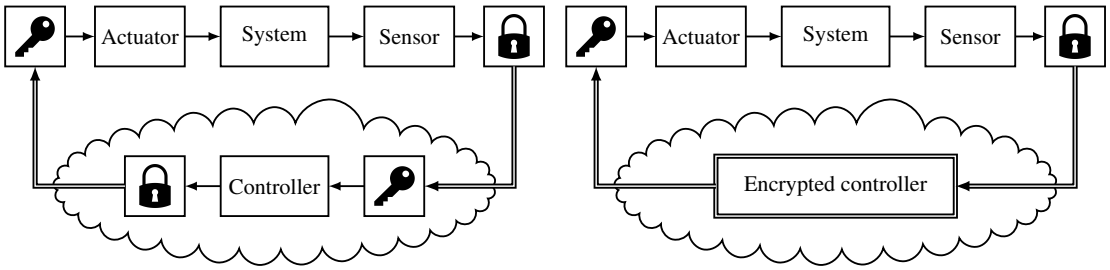

Figure 1.1. Illustration ([210, Figure 1]) of a cloud-based control scheme with encrypted communications but insecure controller evaluation (left) and encrypted communications *and* controller (right). Double-arrows and double-framing emphasize encrypted data transmission and encrypted data processing, respectively.

executing control algorithms securely on external platforms, which may be untrustworthy or susceptible to attacks, is a major challenge in interconnected control systems. In this context, encrypted control addresses the confidentiality/privacy of control data in the entire closed-loop (cf. Figure 1.1 right). This enables *control-as-a-service* between a client (plant) and a provider (cloud) without relying on trust, and it enables decision-making based on sensitive data such as personal, corporate, health, or financial data. Furthermore, secured data prevents attackers from gathering knowledge of a CPS and thwarts stealthy attacks, whereas non-stealthy attacks are detectable by standard methods. While this does not guarantee the integrity of data, confidentiality providing methods often lay the groundwork for those that provide integrity. Now, the key technologies for encrypted control are cryptographic schemes that enable computations on secured data, presented in Table 1.1. We will briefly discuss them next and provide details in Section 2.

Table 1.1. Overview of approaches for privacy-preserving computations in CPSs [P12, Table 1]. The symbols "✓" and "✗" stand for yes/available and no/unavailable, respectively.

	Homomorphic encryption (RLWE)	**SMPC (secret sharing, garbled circuits, ...)**	**Differential privacy**	**Random affine transformations**
Single party computation	✓	✗	✗	✓
High achievable security	✓	✓[1]	✓	✗
Low computational complexity	✗[2]	✓	✓	✓
Low communication overhead	✓[3]	✗	✓	✓
Memory-efficient "ciphertexts"	✗[2]	✓[4]	✓	✓
High achievable accuracy	✓[5]	✓	✗	✓
Any function computable	✓	✓	✓	✗
Memory-efficient auxiliary data	✗[6]	✗[7]	✓	✓
Easy entry to the field	✗	✗	✗	✓

[1] If the number of colluding parties does not exceed the scheme's threshold

[2] Efficiency typically decreases with the depth of the computation circuit

[3] After transmitting the input ciphertexts

[4] Can increase linearly with the number of parties

[5] Possible but increases the computational complexity significantly

[6] Switching keys (evaluation and automorphism keys) must be precomputed and stored

[7] Correlated randomness must be precomputed and stored

Homomorphic encryption. This special type of encryption provides confidentiality of data, while preserving the algebraic structure so that simple computations can be performed on encrypted data/ciphertexts. The idea of homomorphic encryption (HE) dates back to [200]. HE stands out for its reliable security guarantees, high achievable accuracy, and flexibility, as it will become clear in Section 2.2. Nonetheless, HE suffers from high computational complexity, large memory requirements, and a steep learning curve. A characteristic use case for HE is, e.g., computation outsourcing to a cloud.

Secure multi-party computation. In secure multi-party computation (SMPC), information is split between parties such that joint computations are possible while privacy is guaranteed. The concept originated from [249] and comprises a wide variety of protocols, executed among parties, based on primitives for secure computations such as secret sharing (SS; Section 2.3.3), garbled circuits (GCs; Section 2.3.5), or even HE. SMPC provides extraordinary security guarantees under certain assumptions and provides, apart from highly accurate computations, the generality to compute any desired function, which will be detailed in Section 2.3. The bottleneck in SMPC is often its communication overhead.

Differential privacy. Differential privacy (DP) has been proposed in [75] for static data set analyses when an attacker has access to background information. Confidentiality is guaranteed by adding carefully constructed noise before releasing data, where larger noise provides greater security. As a result, there exists a trade-off between security and accuracy, while, on the other hand, computations do not impose an overhead as in HE or SMPC. While single-party computations are in principle possible, DP is often used in a multi-party setup.

Random affine transformations. Although not well-established in cryptography, random affine transformations (RATs) recently gained attention from the control community. RATs are based on the simple idea of encrypting a vector by multiplying it with a random matrix and adding a random vector (see Section 3.3 for details). First, this enables high-accuracy computations without overhead or ciphertext size increase. Second, RATs are well-suited for quadratic programs, while they were initially proposed for matrix products [73, 145, 217].

Trusted execution environments. For completeness, we note that in addition to the aforementioned software-based approaches, there are also hardware-based trusted execution environments. These refer to an area within a computer's main processor designed to isolate data from external threats. To this end, data is encrypted and can only be decrypted in the trusted environment

where it is processed and subsequently re-encrypted before it is stored. From a broader perspective, this creates the impression of computing on encrypted data. This way, confidentiality (and with an extension also integrity) can be ensured on compromised systems [203]. Despite being currently one of the most practical solutions, there are several reasons why we do not consider trusted execution environments further. Namely, they require trust in the vendor and the provider, suffer from *side-channel attacks*, and necessitate specialized hardware [174].

1.3 Privacy-preserving controllers

In comparison to control-related methods and as apparent from Table 1.1, HE and SMPC solutions stand out for their security guarantees, (specified in Sections 2.2.2 and 2.3.2) high accuracy, and generality. From our perspective, this combination of beneficial features makes HE and SMPC particularly suited for sustainable privacy-preserving controller realizations. On the other hand, DP and RAT are generally cheaper in terms of computational effort but require trading off security against accuracy or lack a sound security analysis, respectively. As a consequence, we focus mainly on HE and SMPC next.

Static output and state feedback. Pioneering works in encrypted control focus on HE and realize partial controller encryption. Keeping in mind that the set of operations supported by HE schemes is very limited (cf. Section 2.2.1), affine (or even linear) structures are considered. A prominent example is static state or output feedback in the form

$$\boldsymbol{u}(k) = \boldsymbol{K}\boldsymbol{y}(k). \tag{1.1}$$

Here, $k \in \mathbb{N}$ denotes the discrete time-step and $\boldsymbol{K} \in \mathbb{R}^{m \times l}$ the controller gains, whereas $\boldsymbol{u}(k)$ and $\boldsymbol{y}(k)$ stand for the control input and the system's output, respectively. The first realization of an encrypted controller is found in [138] where (1.1) is evaluated in a cloud environment. Similarly, [86] and [83] consider (1.1) where the latter also discusses robustness guarantees. After these pioneering works, encrypted versions of more complex controller types and full encryptions have been addressed.

Dynamic output feedback. Linear dynamic (output feedback) controllers are important for several reasons. On one hand, practically relevant controller classes such as PID or observer-based feedback can be cast into their representation

$$\boldsymbol{x}_c(k+1) = \boldsymbol{H}\boldsymbol{x}_c(k) + \boldsymbol{G}\boldsymbol{y}(k) \tag{1.2a}$$
$$\boldsymbol{u}(k) = \boldsymbol{E}\boldsymbol{x}_c(k) + \boldsymbol{F}\boldsymbol{y}(k), \tag{1.2b}$$

where $\boldsymbol{x}_c(k) \in \mathbb{R}^n$ and $\boldsymbol{H}, \boldsymbol{G}, \boldsymbol{E}, \boldsymbol{F}$ are the controller's state and gains, respectively. On the other hand, linear dynamic controllers represent a fundamental problem regarding iterations in private computations. At this point, we simply note that HE (and SMPC) require special routines to enable an unlimited number of iterations. In the context of HE, this is called *bootstrapping*, which is computationally costly and will be clarified in Section 3.1.1.

For most control applications, current bootstrapping algorithms are impractical. Thus, [131] uses an orchestration of bootstrapped controllers to evaluate (1.2) where the computational burden is outsourced to a cloud. Remedies that entirely circumvent the need for bootstrapping are regularly resetting $\boldsymbol{x}_c(k)$ to a predefined state [175] or using $\boldsymbol{H} \in \mathbb{Z}^{n \times n}$ [48]. Based on a suitable transformation, [134] exploits integer $\boldsymbol{H}$ while relying on a *trusted third party* (TTP), i.e., an additional party to which computations can be safely outsourced. In practice, this is typically the client (plant), leading to a suboptimal distribution of computational effort. Approximating unstable eigenvalues of a given controller by algebraic integers (roots of an integer polynomial) and using a suitable time-varying transformation is another option that creates such $\boldsymbol{H}$ [135].

Luenberger observers and asymptotic Kalman filters are considered in [133] and [6], respectively. The former work designs an integer controller matrix and builds entirely on HE. The latter utilizes an SMPC approach with two parties. There, an HE scheme is used for basic computations, while re-encryptions, that avoid bootstrapping, are enabled via *additive masking/blinding* and a second party. Additive masking is a common approach to guarantee privacy (see Appendix B.1.3) without relying on a TTP for computational tasks.

Model predictive control. Next, we focus on privacy-preserving model predictive control (MPC) for discrete-time linear systems

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) \tag{1.3a}$$
$$\boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{x}(k), \tag{1.3b}$$

where $\boldsymbol{x}(k)$ is the system state and $\boldsymbol{A}, \boldsymbol{B}$, and $\boldsymbol{C}$ are the system state, input, and output matrix. Central to classical MPC is the recurring solution of the optimal control problem

$$\begin{aligned} \min_{\substack{\tilde{\boldsymbol{x}}(0),\dots,\tilde{\boldsymbol{x}}(N_p) \\ \tilde{\boldsymbol{u}}(0),\dots,\tilde{\boldsymbol{u}}(N_p-1)}} \; & \|\tilde{\boldsymbol{x}}(N_p)\|^2_{\boldsymbol{W}_{N_p}} + \sum_{\tau=0}^{N_p-1} \|\tilde{\boldsymbol{x}}(\tau)\|^2_{\boldsymbol{W}_x} + \|\tilde{\boldsymbol{u}}(\tau)\|^2_{\boldsymbol{W}_u} \\ \text{s.t. } & \tilde{\boldsymbol{x}}(0) = \boldsymbol{x}(k),\ \tilde{\boldsymbol{x}}(N_p) \in \mathcal{T}, \\ & \tilde{\boldsymbol{x}}(\tau+1) = \boldsymbol{A}\tilde{\boldsymbol{x}}(\tau) + \boldsymbol{B}\tilde{\boldsymbol{u}}(\tau), \qquad \forall \tau \in \{0,\dots,N_p-1\}, \\ & (\tilde{\boldsymbol{x}}(\tau), \tilde{\boldsymbol{u}}(\tau)) \in \mathcal{X} \times \mathcal{U}, \qquad \forall \tau \in \{0,\dots,N_p-1\} \end{aligned} \tag{1.4}$$

in every time-step for the current system state $\boldsymbol{x}(k)$. Here, $N_p \in \mathbb{N}$ is the prediction horizon, $\boldsymbol{W}_{N_p}, \boldsymbol{W}_x, \boldsymbol{W}_u$ are weighting matrices, and the sets $\mathcal{U}, \mathcal{X}, \mathcal{T}$

describe input, state, and terminal constraints, respectively. The solution to (1.4) determines the control input via $\boldsymbol{u}(k) = \tilde{\boldsymbol{u}}(0)$ [193]. An early work in this regard is [218], where quadratic optimization is realized with partial privacy via HE. By restricting (1.4) to input constraints, real-time iterations based on projected gradient descent have been proposed in [211], where a TTP is used for nonlinearities and re-encryptions. Building on the flexibility of SMPC in comparison to pure HE solutions, [5] realizes a gradient ascent method on the dual problem through additive masking. For the solution of quadratic programs capable of solving (1.4), RATs have also been used recently in [177, 227] while approaches [247] and [242] address not only confidentiality, but also the integrity of the data.

Under common assumptions, the (parametric) solution of (1.4) has a piecewise affine (PWA) structure

$$\boldsymbol{u}(\boldsymbol{x}) = \begin{cases} \boldsymbol{K}_1\boldsymbol{x} + \boldsymbol{b}_1 & \text{if} \quad \boldsymbol{x} \in \mathcal{P}_1, \\ \vdots & \vdots \\ \boldsymbol{K}_\theta\boldsymbol{x} + \boldsymbol{b}_\theta & \text{if} \quad \boldsymbol{x} \in \mathcal{P}_\theta \end{cases} \tag{1.5}$$

with the gains $\boldsymbol{K}_i, \boldsymbol{b}_i$ and polyhedral regions $\mathcal{P}_i \subseteq \mathcal{X}$ [23]. Based on (1.5), [212] provides a partially encrypted implementation of (1.4) where $\boldsymbol{x} \in \mathcal{P}_i$ is ascertained using a TTP.

Distributed control. Solving a cooperative task (such as flocking) in a decentralized manner with linear agent dynamics and quadratic objectives leads to controllers that are structurally almost identical to (1.1) [149]. Nonetheless, the design, communication, and security models of such distributed controllers are more complex. Encrypted implementations are proposed in [9, 215] that aim to protect the privacy of the agents' internal states, which must be communicated to evaluate the control inputs. Iterative algorithms in multi-agent systems, that are structurally similar to (1.2), can be used to solve consensus problems. In this context, [101, 102, 136, 202] make use of TTPs for an encrypted controller evaluation, while [253] considers a decomposition of each agent's state into a private and exchangeable part.

Another research direction is that of privacy-preserving distributed optimization-based control. The case of a joint optimization problem

$$\min_{\boldsymbol{x}_1,\dots,\boldsymbol{x}_M} \sum_{i=1}^{M} J_i(\boldsymbol{x}_i) \quad \text{s.t.} \quad \sum_{i=1}^{M} \boldsymbol{T}_i\boldsymbol{x}_i = \boldsymbol{p} \tag{1.6}$$

with private costs J_i and decision variables $\boldsymbol{x}_i$ for M agents, connected through linear constraints, is considered in [237]. There, a set of computing parties solves (1.6) via SMPC. To this end, simple iteration formulas resulting from the alternating direction method of multipliers (ADMM) and a dual decomposition are employed.

In multi-agent systems, solutions via DP can also be found because the noise requirements for security are less restrictive (see [105, 108] for an introduction and overview) and aggregations can be performed to limit the accuracy loss. Addressed are, for instance, linear distributed controllers [120, 180, 243], distributed optimization [103, 106], and federated learning [245].

Data-driven control. Furthermore, data-driven schemes have been considered. These are of special interest because they fit naturally in the context of service-based control. The first encrypted data-driven scheme is found in [8], where a data-driven linear quadratic regulator (LQR) from [58] is considered. While computations are fully encrypted via HE, one final computation step is performed on the client's side. Next, [7] shows an implementation of an encrypted least-square method with L_1-regularization (also known as Lasso) that can be used for data-driven LQR. The solution is based on SMPC using multiparty HE [172] which provides a computationally less demanding bootstrapping. The first private controller design is presented in [P15], where encrypted PID controller tuning is realized via extremum seeking for black-box systems. In reinforcement learning, where policies are synthesized from large amounts of data, privacy can become a major issue. The works [225, 226] implement the linear updates for TD(0) and SARSA(0) (see [28] for details) by HE, and address the remaining steps by employing a TTP. Recently, an SMPC-based approach to Markov decision processes was published, where the problem is formulated in terms of a linear program [12].

Miscellaneous. Using a two-party setup, where HE is used as computation primitives, [214] implements privacy-preserving polynomial control. An optimization of this scheme is presented in [205]. Another perspective on nonlinear encrypted control is presented in [132], where past input and output measurements are used to construct a nonlinear control law which is partially computed by a TTP. An extended Kalman filter is implemented in [96] using an SMPC approach by combining partially encrypted computations with tailored two-party protocols for more complex operations. This allows the authors to keep everything except for the nonlinear prediction private. Using similar techniques, [235] implements a recursive least-squares method, while [236] develops a special SMPC scheme that is applied to adaptive Kalman filtering. Remarkably, control methods can also be applied in cryptography. Recently, [223] showed how to attack the Diffie-Hellman cryptosystem with Koopman theory.

1.4 Thesis outline and contributions

This thesis builds on the publications of the author listed at the end of the document.[1] It explores the intersection of control and cryptography by addressing the confidentiality of data in CPSs through HE and SMPC, which results in unique challenges. For example, control data is created and processed dynamically while the robustness of algorithms is mandatory for the safe operation of a control loop. This is in stark contrast to other domains such as machine learning, where data is typically collected and processed offline without safety implications for the real world. Thus, an interdisciplinary approach is required in encrypted control, in which efficiency and reliability demands can often only be met through a non-trivial co-design of controllers and cryptosystems.

Chapter 2. To this end, we lay the foundations in Chapter 2, which is based on our articles [P12, P14], by gathering, unifying, and explaining relevant cryptography from various sources. More precisely, we detail concepts, computation models, and overarching security notions before we dive deeper into state-of-the-art HE and SMPC schemes. The provided framework shall also facilitate future research.

Chapter 3. This chapter is devoted to fundamental challenges related to privacy-preserving controller realizations such as iterations, non-polynomial functions, and security guarantees. After specifying threat models, each section follows the same structure, that is, problem specification, article summaries, and discussion.

Section 3.1 deals with iterative controllers. These are of special interest, as they capture the problem of limited privacy-preserving computations (we touched on this in Section 1.3). In this context, we analyze iterative controllers from a system-theoretic perspective. Namely, the implications of integer controller matrices as they circumvent the iteration problem. Based on this, we find a suitable controller type that aligns well with homomorphic cryptosystems. Then, we realize two different iterative algorithms for multi-agent systems, i.e., distributed optimization via ADMM and affine averaging. In the former, we build on key-switching for an efficient access control mechanism, while the latter is realized through a novel reset technique that builds on information aggregation. We verify our approaches by applying them to a benchmark problem in networked control and practical examples in multi-agent systems, respectively. The results of this section have been previously presented in [P3, P8, P11, P16].

Section 3.2 considers non-polynomial controllers. As it will become clear in Chapter 2, non-polynomial functions are not directly supported by HE (or secret sharing) primitives. Consequently, even seemingly simple computations

[1] Note that we refer to our publications via [P1] whereas other literature is cited via [1].

can become a major hurdle. In light of this, we propose the use of max-out neural networks, which are capable of representing PWA functions exactly, for the approximation of arbitrary continuous functions. Representing computations in this tailored form facilitates the use of SMPC methods. This way, non-polynomial functions can be evaluated efficiently and privately. Our novel method is tested on various explicit MPC control laws. The results of this section have been published in [P9, P10, P17].

Finally, Section 3.3 addresses the security of RATs, which is an open problem despite their popularity in the control community. Here, we analyze two variants of the cipher under different attack scenarios. Furthermore, we specify a digital implementation and its implications, which are often overlooked yet critical aspects. The results of this section appeared in [P13].

Chapter 4. We conclude this thesis in Section 4.1 and provide an outlook, which is based on our insights, on fruitful research directions in Section 4.2. Novel challenges include steps towards more complex control algorithms and extensions for integrity.

Part II. The second part of this thesis contains reprints of the articles summarized in Chapter 3 in order of appearance.

Appendices. Lastly, we provide additional material that enriches Chapters 2 and 3 in the appendices. Appendix A discusses algebra for encrypted control, such as computations over finite integer and polynomial sets, while Appendix B deals with security and further material on homomorphic encryption. Then, Appendix C details a convex training algorithm and the optimal quantization for max-out neural networks.

Chapter 2

Preliminaries

This thesis combines elements from control and cryptography. While many readers are likely familiar with control theory, they may not have a background in cryptography. To bridge this gap, we offer an introduction to cryptography in this chapter, building on our survey and tutorial articles [P12, P14] specifically tailored for engineers. The foundational knowledge provided here sets the stage for the rest of this thesis and lays the foundation for future research. To this end, we gather and unify relevant cryptography from various sources. Rather than providing an exhaustive and technical introduction, which can be found in the existing literature [32, 60, 125], our goal is to present relevant cryptography concisely and accessibly. However, it is essential to acknowledge that cryptography is a vast and diverse research field in its own right. Therefore, our exposition offers only a glimpse of ideas that have already transformed our everyday lives and may continue their success story in control engineering applications.

2.1 Fundamentals

In this section, we abstractly introduce the basics of cryptosystems, relevant computational circuits, plaintext encodings, and security notions, which are later specified for different cryptographic schemes.

2.1.1 Cryptosystems

The main purpose of cryptosystems is to protect information, typically a *message* $z \in \mathcal{Z}$, from unauthorized access by encrypting it into a *ciphertext* $\mathsf{ct}(z) \in \mathcal{C}$. Here, $\mathcal{Z}$ and $\mathcal{C}$ denote the message and ciphertext space, which vary for different cryptosystems but are typically finite. To this end, cryptosystems provide three algorithms:

$\underline{\mathsf{KeyGen}(1^\kappa)}$. Output keys, e.g., $(\mathsf{ek}, \mathsf{sk})$ based on the security parameter κ.

$\underline{\mathsf{Enc}_{\mathsf{ek}}(z)}$. Output a ciphertext $\mathsf{ct}(z) \in \mathcal{C}$ based on the message $z \in \mathcal{Z}$ by using the encryption key ek.

$\underline{\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct}(z))}$. Output the message $z \in \mathcal{Z}$ based on the ciphertext $\mathsf{ct}(z) \in \mathcal{C}$ by using the secret key sk.

The security of a cryptosystem is based on the encryption algorithm Enc, which ensures that accessing z is only feasible with sk through the decryption algorithm Dec. To this end, the key generation KeyGen and Enc are often probabilistic algorithms such that encryptions of the same z almost certainly do not result in the same $\mathsf{ct}(z)$. On the other hand, Dec is deterministic and must (at least approximately) ensure

$$\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct}(z)) = z \quad \text{for all } \mathsf{ct}(z) \in \mathcal{C}. \tag{2.1}$$

The property (2.1) is called *correctness*, and it is paramount for the usefulness of a cryptosystem.

Based on ek, we distinguish two types of cryptosystems. If $\mathsf{ek} = \mathsf{sk}$, the scheme is called *symmetric* (or private key), whereas $\mathsf{ek} = \mathsf{pk}$ refers to *asymmetric* (or public key) schemes. Although symmetric cryptosystems are typically much faster than asymmetric ones, they require a key exchange of sk to establish a secure communication between parties. Asymmetric schemes allow anyone in possession of the public key pk to encrypt a message, while decryption requires sk. By construction, pk can be safely distributed.

2.1.2 Computation circuits

Later, we will see that beyond mere information protection, homomorphic cryptosystems and SMPC support functionalities such as Add and Mult which enable computations on secured data. Taking HE as an example, the evaluation of a function $f : \mathcal{Z}^m \to \mathcal{Z}$ on the messages $z_1, \ldots, z_m$ with $m \in \mathbb{N}$ can equivalently be performed on the encrypted messages $\mathsf{ct}(z_1), \ldots, \mathsf{ct}(z_m)$ via

$$f(z_1, \ldots, z_m) = \mathsf{Dec}_{\mathsf{sk}}\left(\mathsf{Eval}\left(\mathsf{ct}(z_1), \ldots, \mathsf{ct}(z_m)\right)\right)$$

as long as f can be expressed by Eval, that is, a composition of Add and Mult.

At this point, the representation of f can make a significant difference. There are two main approaches. First, *arithmetic circuits* where Add and Mult are used to evaluate compositions of multivariate polynomials. While arithmetic circuits form a valuable building block for control algorithms, they are not well-suited for non-polynomial f. Typically, such f require high-order approximations. Second, *Boolean circuits* where $f : \{0,1\}^m \to \{0,1\}$ are realized by reducing Add and Mult modulo 2, which results in XOR and AND. This computation domain enables any computable f through functionally complete logic gates, for example, NAND. However, Boolean circuits are inefficient for arithmetic.

Lastly, we want to point out a particular characteristic of encrypted computations. Namely, conditional statements are not useful for reducing the overall computational effort, regardless of the computation circuit. To clarify this point, let us consider a scenario where $z_1 \leqslant z_2$ is the condition that decides between two potential computations. Then, if only one of these computations is executed based on the condition, an observer can easily deduce the result of $z_1 \leqslant z_2$, thereby compromising privacy. In order to avoid that, both computations have to be evaluated.

2.1.3 Plaintext encoding

Generally speaking, different cryptosystems rely on different realizations of message and ciphertext spaces $\mathcal{Z}$ and $\mathcal{C}$, respectively. Consequently, data x from a *plaintext* space $\mathcal{P}$ might not be immediately supported. Therefore, an encoding $\mathsf{Ecd} : \mathcal{P} \to \mathcal{Z}$ and a decoding $\mathsf{Dcd} : \mathcal{Z} \to \mathcal{P}$ are required that, in the case of encrypted computations, must be compatible with Add and Mult. More precisely, for $x_1, x_2 \in \mathcal{P}$ they satisfy (at least approximately)

$$x_1 + x_2 = \mathsf{Dcd}(\mathsf{Ecd}(x_1) + \mathsf{Ecd}(x_2)) \quad \text{and} \quad x_1\, x_2 = \mathsf{Dcd}(\mathsf{Ecd}(x_1)\, \mathsf{Ecd}(x_2)) \tag{2.2}$$

which makes corresponding plaintext functions g available via

$$\begin{array}{ccccc} \mathcal{P}^m & \xrightarrow{\mathsf{Ecd}} & \mathcal{Z}^m & \xrightarrow{\mathsf{Enc}} & \mathcal{C}^m \\ \downarrow{\scriptstyle g} & & \downarrow{\scriptstyle f} & & \downarrow{\scriptstyle \mathsf{Eval}} \\ \mathcal{P} & \xleftarrow{\mathsf{Dcd}} & \mathcal{Z} & \xleftarrow{\mathsf{Dec}} & \mathcal{C} \end{array} \tag{2.3}$$

where Ecd and Enc are applied element-wise. Often, $\mathcal{Z}$ and $\mathcal{C}$ are finite sets. This offers several advantages, among which security stands out. In particular, $\mathcal{Z} = \mathbb{Z}_q$ is commonly used with the standard representatives $\{0, 1, \ldots, q-1\}$ for a *modulus* $q > 1$ with $q \in \mathbb{N}$. The modulo reduction $z \bmod q := z - q\lfloor z/q \rfloor$ then maps from $\mathbb{Z}$ to $\mathbb{Z}_q$. In control and many other applications, real-valued data is omnipresent, such that $\mathcal{P} = \mathbb{R}$. Consequently, two approaches become evident to obtain a suitable $\mathsf{Ecd}(x) = z$, where $x \in \mathbb{R}$ and $z \in \mathbb{Z}_q$. Namely, x is either approximated by a fixed-point or floating-point number. Due to their finite size, these numbers can then be associated with integers in $\mathbb{Z}_q$. Floating-point numbers provide greater robustness and flexibility, making them the unspoken standard for numerical computations. However, their data-dependent exponent makes them less efficient than their fixed-point counterparts. In order to maximize efficiency, most applications of HE and SMPC utilize fixed-point encodings, and we follow this trend here.

A straightforward way to realize a generalized fixed-point encoding suitable for cryptosystems in encrypted control is:

$$\underline{\mathsf{Ecd}_s(x)}.\ \text{Output } z = \lfloor sx \rceil \bmod q \in \mathbb{Z}_q, \text{ with } x \in \mathbb{R} \text{ and } s \geqslant 1. \tag{2.4a}$$

$$\underline{\mathsf{Dcd}_s(z)}.\ \text{Output } x \approx \mu(z)/s \in \mathbb{R}, \text{ where } \mu(z) = \begin{cases} z - q & \text{if } z \geqslant q/2 \\ z & \text{otherwise.} \end{cases} \tag{2.4b}$$

Note that the scaling factor in Dcd_s must be adapted if computations are performed on z. Verifying that Ecd and Dcd satisfy (2.2) arbitrarily well by increasing the public scaling factor s is straightforward. More interesting is the error of $\mathsf{Dcd}_s(\mathsf{Ecd}_s(x))$ that satisfies

$$|x - \mu(z)/s| \leqslant \frac{1}{2s} + \frac{q}{s}\left|\left| \frac{\lfloor sx \rceil + q/2}{q} \right|\right|. \tag{2.5}$$

In other words, for $\lfloor sx \rceil \in [-q/2, q/2) \cap \mathbb{Z}$ a quantization error $1/(2s)$ occurs while any other $\lfloor sx \rceil$ causes an overflow of $\mathbb{Z}_q$. This is captured in the second error term, which is either 0 or very large, i.e., a multiple of q/s. Thus, there is a trade-off between accuracy and risk of overflow. Observe that arithmetic circuits can be realized immediately by (2.4a). For Boolean circuits, one can use a bit-decomposition of z.

2.1.4 Security principles

Next, we will discuss design and security principles for secure ciphers. First, the design of cryptosystems should adhere to *Kerckhoffs' principles*, which involve disclosing a cryptosystem defined by $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$. Although it may initially seem counterintuitive, publishing the details of a cryptosystem has two important advantages. It enables a public security analysis, likely leading to more robust results, and the security of the cryptosystem cannot depend on the secrecy of its design. In fact, if the security of a cryptosystem relied on concealing information about its operation (*security through obscurity*), it would become susceptible to reverse engineering and must be replaced once this information is exposed [125, Section 1.2].

Second, the security of a cryptosystem should be proven according to a precise definition. An important metric in this context is the *statistical distance*, which allows quantifying security.

Definition 1 ([60, Definition 2.1])**.** Let Y_1 and Y_2 be random variables with the same probability space and the same range $\mathcal{R}$. Then, the statistical distance between Y_1 and Y_2 is

$$D(Y_1, Y_2) = \frac{1}{2} \sum_{r \in \mathcal{R}} |\Pr(Y_1 = r) - \Pr(Y_2 = r)| \in [0, 1]. \tag{2.6}$$

The case $D = 0$ corresponds to $\Pr(Y_1 = r) = \Pr(Y_2 = r)$ for all $r \in \mathcal{R}$ while $D = 1$ occurs when $\Pr(Y_1)$ and $\Pr(Y_2)$ are disjoint. Furthermore, the statistical distance is an information-theoretic property and thus cannot be increased by any amount of computation [60, Chapter 2].

In the context of cryptosystems, consider an adversary $\mathcal{A}$ that tries to infer (new) information about z given $\mathsf{ct}(z)$, but does not know sk. Note that ct is typically probabilistic and $\mathcal{A}$' knowledge about z may in general be uncertain such that it may guess z. Thus, let C and Z denote the random variables for ct and z, respectively, such that $\Pr(C = \mathsf{ct} | Z = z)$ is the probability of observing ct given z, which is determined by Enc. This, in conjunction with (1), allows us to define the following security notion.

Definition 2 ([125, Lemma 2.3])**.** A cryptosystem is *perfectly secure* if and only if

$$\sum_{\mathsf{ct} \in \mathcal{C}} |\Pr(C = \mathsf{ct} | Z = z_1) - \Pr(C = \mathsf{ct} | Z = z_2)| = 0 \tag{2.7}$$

holds for every probability distribution over $\mathcal{Z}$ and every $z_1, z_2 \in \mathcal{Z}$.

Equivalently, any ct must be equally likely an encryption of any z. In this case, $\mathcal{A}$ cannot infer information about z based on ct, even with unlimited computational resources. From $\mathcal{A}$'s perspective, the ciphertexts are *perfectly indistinguishable*. In order to achieve that, a uniform ciphertext distribution is sufficient.

Definition 2 turns out to be very useful as it serves as a basis for a plethora of other definitions. Nevertheless, a fundamental limitation of perfectly secure ciphers is related to their space complexity. More precisely, the key must have at least the length of the message. To ensure correctness, the key and ciphertext require twice the memory of the plaintext (see [125, Theorem 2.7] and Section 2.3.3). Fortunately, slightly weaker security notions circumvent this issue, while withstanding any feasible (explained below) adversarial behavior. In a first step, the security will be parameterized by the *(statistical) security parameter* $\kappa \in \mathbb{N}$. To this end, we introduce the notion of a *negligible function*.

Definition 3 ([125, Definition 3.5])**.** A negligible function $\mathsf{negl}(\kappa)$ satisfies the relation $\mathsf{negl}(\kappa) < 1/g(\kappa)$ for all positive polynomials $g(\kappa)$ and all $\kappa > \kappa_0 \in \mathbb{N}$.

Now, instead of requiring random variables[1] to have the same distribution, as in Definition 2, we allow for a negligible deviation.

Definition 4 ([60, Definition 2.7])**.** The random variables Y_1 and Y_2 are called *statistically indistinguishable* if they satisfy $D(Y_1, Y_2) \leqslant \mathsf{negl}(\kappa)$.

Relaxing Definition (2) via (4) results in a statistically secure cryptosystem. In principle, $\mathcal{A}$ can then exploit that a ciphertext may not result from any message with equal probability to gain information. However, such ciphertexts are statistically insignificant and adjustable by κ.

The second relaxation step will be with respect to the algorithmic complexity that is feasible for $\mathcal{A}$. Importantly, there is no limitation imposed on the attack strategy of $\mathcal{A}$. Now, in the same way, we consider inverse polynomial probabilities significant (see Definition 4), we consider *probabilistic polynomial time* (PPT) algorithms *efficient* enough. This leads us to the notion of *computational indistinguishability*.

Definition 5 ([125, Definition 6.31])**.** The random variables Y_1 and Y_2 are computationally indistinguishable, also denoted by $Y_1 \overset{c}{\equiv} Y_2$, if for every PPT $\mathcal{A}$, there exists a negligible function such that

$$|\Pr(\mathcal{A}(Y_1) = 1) - \Pr(\mathcal{A}(Y_2) = 1)| \leqslant \mathsf{negl}(\kappa). \tag{2.8}$$

In this setup, $\mathcal{A}$ receives samples and tries to distinguish whether they are a realization of the random variable Y_1 or Y_2 based on a computational analysis. In particular, $\mathcal{A}(Y_1) = 1$ denotes that $\mathcal{A}$ receives samples from Y_1 and

[1] Definitions 4 and 5 are typically considering probability ensembles, e.g., $Y = \{Y_\kappa\}_{\kappa \in \mathbb{N}}$, such that $D(Y_{1,\kappa}, Y_{2,\kappa}) \leqslant \mathsf{negl}(\kappa)$ for $\kappa > \kappa_0$ is required. We omit stating probability ensembles explicitly here.

correctly identifies Y_1, whereas in $\mathcal{A}(Y_2) = 1$ it incorrectly identifies Y_1 based on samples from Y_2. When ciphertext distributions are computationally indistinguishable, a computationally unbounded $\mathcal{A}$ may be able to distinguish them with non-negligible success, which can be leveraged for an attack. However, the exponential complexity required for this can be made intractable in practice by increasing κ. Of course, only $\mathcal{A}$ faces a large computational burden when attacking a cryptosystem, while $\mathsf{Dec}_{\mathsf{sk}}$ is efficient. Exactly this complexity gap is the mechanism that regulates the secrecy of data in public key cryptosystems.

2.2 Homomorphic encryption

Expanding on the previous section, we will introduce a blueprint for homomorphic cryptosystems in Section 2.2.1. Then, Section 2.2.2 outlines the security guarantees of HE schemes, followed by a description of the Paillier and CKKS cryptosystems in the remaining sections.

2.2.1 Overview

HE extends a cryptosystem $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ by computations on encrypted data that are enabled by an algebraic structure preservation (*homomorphism*) of Enc. Based on two ciphertexts $\mathsf{ct}(z_1), \mathsf{ct}(z_2)$, one or both of the following encrypted operations[2] are available:

$$\underline{\mathsf{Add}(\mathsf{ct}(z_1), \mathsf{ct}(z_2))}.\ \text{Output } \mathsf{ct}(z_1 + z_2) = \mathsf{ct}(z_1) \oplus \mathsf{ct}(z_2). \tag{2.9a}$$

$$\underline{\mathsf{Mult}(\mathsf{ct}(z_1), \mathsf{ct}(z_2))}.\ \text{Output } \mathsf{ct}(z_1 z_2) = \mathsf{ct}(z_1) \otimes \mathsf{ct}(z_2). \tag{2.9b}$$

HE schemes that offer the evaluation of encrypted additions as in (2.9a) via the operation "$\oplus$" are called *additively homomorphic*. Analogously, schemes that enable (2.9b) via "$\otimes$" are referred to as *multiplicatively homomorphic*. Additionally, public operations "$\boxplus$" and "$\boxtimes$", where for instance z_1 is not encrypted, are often possible:

$$\underline{\mathsf{PAdd}(z_1, \mathsf{ct}(z_2))}.\ \text{Output } \mathsf{ct}(z_1 + z_2) = z_1 \boxplus \mathsf{ct}(z_2). \tag{2.10a}$$

$$\underline{\mathsf{PMult}(z_1, \mathsf{ct}(z_2))}.\ \text{Output } \mathsf{ct}(z_1 z_2) = z_1 \boxtimes \mathsf{ct}(z_2). \tag{2.10b}$$

Note that (2.10a) and (2.10b) result in a ciphertext which then protects the confidentiality of z_1. Depending on their supported operations, HE schemes are categorized as detailed next:

- *Partially homomorphic encryption* (PHE) allows either for (2.9a) or (2.9b), but not both.
- *Leveled FHE* allows for a finite amount of (2.9a) and (2.9b).

[2]Strictly speaking, many ciphertexts exist that encrypt $z_1 + z_2$ in (2.9a) and $z_1 z_2$ in (2.9b).

Table 2.1. Multiplicative depth of four polynomials.

Polynomial	$(z_1 + z_2)(z_1 + z_2)$	$(z_1 + z_2)(z_1 z_2)$	$(z_1 z_2)(z_1 z_2)$	$z_1(z_2(z_1 z_2))$
Mult. depth	1	2	2	3

- *Fully homomorphic encryption* (FHE) allows for an unlimited amount of operations (2.9a) and (2.9b).

When it comes to leveled FHE, the *multiplicative depth* (or likewise the supported "levels") is typically used as a complexity metric because it defines critical parameters to a large extent. Simply put, multiplicative depth is the number of successive multiplications performed on a quantity. An illustration is provided in Table 2.1. Interestingly, a leveled scheme can be transformed into a FHE scheme via periodic execution of *bootstrapping* that is famously introduced in [91]. Unfortunately, bootstrapping is computationally costly and unsuitable for many control applications (see Section 3.1 for a detailed discussion).

Of course, the capability to compute on encrypted data generally comes at a cost. Although significant improvements have been made in HE over the last decade, its computational overhead as well as the memory footprint of ciphertexts and key data in comparison to plaintexts are restrictive, i.e., they can differ by several orders of magnitude. This makes HE in resource-constrained environments challenging. Thinking about service-based applications, the computational and storage capabilities of a cloud alleviate these issues. Still, for computation queries, the plant must encrypt its inputs and forward them to the cloud service, which can constitute a bottleneck.

2.2.2 Security of homomorphic cryptosystems

The relevant security guarantee for HE schemes is based on computational indistinguishability (see Definition 5). In this context, we already stated that PPT $\mathcal{A}$ are of interest. The overall specification of $\mathcal{A}$ will be completed next by categorizing different attack scenarios in terms of $\mathcal{A}$'s knowledge:

- *Ciphertext-only attack* (COA): $\mathcal{A}$ has ciphertexts $\{\mathsf{ct}(z_1), \mathsf{ct}(z_2), \dots\}$ and attempts to obtain one of the messages z_i.
- *Known-plaintext attack* (KPA): $\mathcal{A}$ has pairs $\{(z_1, \mathsf{ct}(z_1)), (z_2, \mathsf{ct}(z_2)), \dots\}$ which are encrypted under the same key and attempts to determine the message of another ciphertext.
- *Chosen-plaintext attack* (CPA): $\mathcal{A}$ has the ability to obtain encryptions of arbitrary messages of its choice. Then it attempts to determine the message of another ciphertext.

- *Chosen-ciphertext attack* (CCA): $\mathcal{A}$ has, in addition to the CPA scenario, the ability to obtain the decryption of ciphertexts of its choice, except for a target ciphertext. Then it attempts to determine the message of the target ciphertext.

To motivate Definition 2, we have already introduced a COA without explicitly naming it. Another important attacker type is an *eavesdropper*, which can potentially engage in a COA or KPA, depending on the available information. Importantly, security against a KPA protects against potential information $\mathcal{A}$ may have, which we cannot influence. Public key cryptosystems, including HE schemes, should at least be secure against CPAs. Otherwise, $\mathcal{A}$ could use $\mathsf{Enc}_{\mathsf{pk}}$ to break the scheme. Security against CCAs requires *non-malleability* [32, Section 12.2.1], i.e., ciphertexts cannot be manipulated in a logical way. Given that homomorphisms enable exactly that, it is a truism that HE schemes cannot attain this security certificate.

We have now established all building blocks to understand the computational security guarantee that HE schemes should provide, i.e., *indistinguishability under chosen-plaintext attacks* (IND-CPA).[3]

Definition 6 ([125, Definition 3.10 + p.323])**.** Based on the experiment

$\mathsf{Exp}_{\mathrm{CPA},\kappa}(b)$:

1: $\mathsf{KeyGen}(1^\kappa)$ outputs the pair $(\mathsf{pk}, \mathsf{sk})$ based on κ.
2: $\mathcal{A}$ is given pk and it outputs the messages z_0, z_1.
3: The ciphertext $\mathsf{ct}(z_b)$ is given to $\mathcal{A}$, where $b \leftarrow \{0,1\}$.
4: $\mathcal{A}$ outputs $b' \in \{0,1\}$ indicating which message is encrypted in $\mathsf{ct}(z_b)$.

a public key cryptosystem has indistinguishable encryptions under chosen-plaintext attacks if for all PPT $\mathcal{A}$

$$|\Pr(\mathsf{Exp}_{\mathrm{CPA},\kappa}(0) = 1) - \Pr(\mathsf{Exp}_{\mathrm{CPA},\kappa}(1) = 1)| \leqslant \mathsf{negl}(\kappa). \tag{2.11}$$

In line 3, $b \leftarrow \{0,1\}$ denotes sampling uniformly at random from the set $\{0,1\}$.[4] Now, for large enough κ an IND-CPA secure cipher ensures that $\mathcal{A}$ cannot distinguish between $\mathsf{ct}(z_1)$ and $\mathsf{ct}(z_2)$ despite choosing z_1, z_2 and having access to $\mathsf{Enc}_{\mathsf{pk}}$. Such *experiment* or *game-based* definitions are popular in the context of cryptosystems, since they are comparatively simple to work with. However, the relationship between an IND-CPA guarantee and the information $\mathcal{A}$ can obtain may not be immediate, although $\mathsf{Exp}_{\mathrm{CPA},\kappa}(b)$ is very carefully designed to address practical security needs. Fortunately, it can be shown that IND-CPA allows no PPT $\mathcal{A}$ to learn information about the message based on ciphertexts by relating it to a *simulation-based* notion (see Section 2.3.2 and [95]).

[3] Indistinguishability under an eavesdropper implies indistinguishability under CPA [125, Proposition 10.5] and the security of multiple messages [125, Theorem 10.10].

[4] We provide additional information on the generation of randomness in Appendix B.1.1.

2.2.3 The Paillier cryptosystem

The Paillier cryptosystem (Paillier) [182] is an asymmetric additively homomorphic encryption scheme. Although not fully homomorphic, it enables a range of applications and captivates due to its comparative simplicity, which has led to widespread adoption in encrypted control. Paillier relies on the hardness of factoring the composite number q, which is the product of two sufficiently large primes p_1 and p_2. The scheme then builds its encryption and decryption on an isomorphism between $\mathbb{Z}_q \times \mathbb{Z}_q^*$ and $\mathbb{Z}_{q^2}^*$, where $\mathbb{Z}_q^* = \{a \in \mathbb{Z}_q | \gcd(a, q) = 1\}$ is the multiplicative group of integers. In other words, an equivalent representation for elements in these spaces for which a bijective mapping exists. More precisely, the Paillier cryptosystem is defined as follows:

$\underline{\mathsf{KeyGen}(1^\kappa)}$. Output $\mathsf{pk} = p_1 p_2 = q$ and $\mathsf{sk} = (p_1 - 1)(p_2 - 1)$, where $p_1, p_2 \in \mathbb{N}$ are large and distinct primes with bit-lengths of κ.
$\underline{\mathsf{Enc}_{\mathsf{pk}}(z)}$. Output $\mathsf{ct}(z) = (q+1)^z r^q \bmod q^2$, where $z \in \mathbb{Z}_q$ and $r \leftarrow \mathbb{Z}_q^*$.
$\underline{\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct}(z))}$. Output

$$z = \frac{\mathsf{ct}(z)^{\mathsf{sk}} \bmod q^2 - 1}{q} \mathsf{sk}^{-1} \bmod q, \text{ with } \mathsf{sk}^{-1}\mathsf{sk} \bmod q = 1.$$

The security of Paillier is grounded in the decisional composite residuosity assumption. Namely, $r^q \bmod q^2 \overset{c}{\equiv} r'$ holds for a suitable q, where $r \leftarrow \mathbb{Z}_q^*$ and $r' \leftarrow \mathbb{Z}_{q^2}^*$ [125, Definition 11.30]. If this is true, then the randomization of Enc makes ciphertexts indistinguishable for $\mathcal{A}$, which yields the following.[5]

Theorem 1 ([182, Theorem 15])**.** *The Paillier cryptosystem is IND-CPA secure under the decisional composite residuosity assumption.*

Remarkably, the encryption provides an additive homomorphism over $\mathbb{Z}_q$ and a public multiplication.

$\underline{\mathsf{Add}(\mathsf{ct}(z_1), \mathsf{ct}(z_2))}$. Output $\mathsf{ct}(z_1)\mathsf{ct}(z_2) = (q+1)^{z_1+z_2}(r_1 r_2)^q \pmod{q^2}$.
$\underline{\mathsf{PMult}(\mathsf{ct}(z_1), z_2)}$. Output $\mathsf{ct}(z_1)^{z_2} = (q+1)^{z_1 z_2} r_1^{q z_2} \pmod{q^2}$, where $z_2 \in \mathbb{Z}_q$.

Next, let us illustrate an encrypted control task using the Paillier cryptosystem.

Example 1 (Encrypted output feedback)**.** Consider the output feedback controller $u = \boldsymbol{k}^\top \boldsymbol{y}$ and let the feedback gain and current measurement be $\boldsymbol{k}^\top = \begin{pmatrix} 0.1 & 0.2 \end{pmatrix}$ and $\boldsymbol{y} = \begin{pmatrix} 0.3 & 0.4 \end{pmatrix}^\top$, respectively, which results in $u = 0.11$. Paillier only supports public multiplications. Because $\boldsymbol{y}$ is dynamic and contains plant information, we encrypt $\boldsymbol{y}$ instead of $\boldsymbol{k}$. With the toy parameter $\kappa = 16$ (at least 1024 is recommended), we sampled the keys $\mathsf{pk} = 2560505147 = q$ and

[5] A question that may occur is how the hardness of a mathematical problem, indistinguishability, and the security of a cryptosystem are linked. To this end, see Appendix B.1.2.

$\mathsf{sk} = 2560403520$. Then, we apply the encoding (2.4a) with $s = 10$ componentwise, i.e., $\lfloor s\boldsymbol{k} \rceil = \mathsf{Ecd}_s(\boldsymbol{k})$ and $\boldsymbol{z} = \lfloor s\boldsymbol{y} \rceil = \mathsf{Ecd}_s(\boldsymbol{y})$, where $\bmod q$ has no effect. For the encryption, we sample $r_1 = 2281162363$ and obtain

$$\mathsf{ct}(z_1) = (q+1)^{y_1} r_1^q = 1542788270409383814 \,(\mathrm{mod}\, q^2).$$

Similarly, we find $\mathsf{ct}(z_2) = 2230041660558490075 \,(\mathrm{mod}\, q^2)$. Then, the control input is privately computed via

$$\begin{aligned}\mathsf{ct}(v) &= \lfloor sk_1 \rceil \boxtimes \mathsf{ct}(z_1) \oplus \lfloor sk_2 \rceil \boxtimes \mathsf{ct}(z_2) = \mathsf{ct}(z_1)^{\lfloor sk_1 \rceil} \mathsf{ct}(z_2)^{\lfloor sk_2 \rceil} \\ &= 63745278380961139 72 \,(\mathrm{mod}\, q^2).\end{aligned}$$

To decrypt $\mathsf{ct}(v)$, we obtain $\mathsf{sk}^{-1} = 121843633 \,(\mathrm{mod}\, q)$ with the extended Euclidean algorithm (see [219, Section 4.3] and find $\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct}(v)) = 11$ which uses $(q+1)^v \bmod q^2 = qv + 1$ for $v \in \mathbb{Z}_q$. Finally, observe that $\lfloor sk_i \rceil \lfloor sy_i \rceil$ has a scaling factor of s^2 such that $u = \mathsf{Dcd}_{s^2}(v) = \mu(11)/s^2 = 0.11$ as desired.[6]

2.2.4 The CKKS cryptosystem

The Cheon-Kim-Kim-Song (CKKS) scheme [51] is an approximate asymmetric leveled FHE scheme that is often considered the preferred choice for arithmetic circuits. As most competitive HE schemes and the new standard for quantum-resistant key exchanges [33], it builds on the *learning with errors problem* (LWE) introduced in [194]. In the context of LWE and the cryptosystems that are based on it, we identify $\mathbb{Z}_q$ equivalently by *centered representatives* $[-q/2, q/2) \cap \mathbb{Z}$. The corresponding modulo reduction is $z \bmod q := z - q\lfloor z/q \rceil$.

Learning with errors. Because of its pivotal role in HE and as a preparation, let us consider the LWE problem in a bit more detail, where we use prime q.

Definition 7. A random vector, a secret vector, and an error are sampled according to $\boldsymbol{a}_i \leftarrow \mathbb{Z}_q^N$, $\mathbf{sk} \leftarrow \mathcal{D}_s^N$, $e \leftarrow \mathcal{D}_e$, respectively. Then, a polynomial amount of LWE tuples $(b_i, \boldsymbol{a}_i^\top) = (-\boldsymbol{a}_i^\top \mathbf{sk} + e_i \bmod q, \boldsymbol{a}_i^\top) \in \mathbb{Z}_q^{1\times(N+1)}$ are generated in a black-box manner. *Search* LWE *problem:* $\mathcal{A}$ tries to find $\mathbf{sk}$ based on the LWE tuples. *Decisional* LWE *problem:* $\mathcal{A}$ tries to distinguish LWE tuples $(b_i, \boldsymbol{a}_i^\top)$ from $(b_i', \boldsymbol{a}_i^\top)$, where $b_i' \leftarrow \mathbb{Z}_q$.

Note that the secret is denoted $\mathbf{sk}$ here because it will later serve as the secret key. Originally, the distribution $\mathcal{D}_s$ results from sampling $\mathbb{Z}_q^N$ uniformly at random. Because "small" secrets turn out to be beneficial, $\mathbf{sk}$ is often sampled from $\{-1, 0, 1\}^N$ and may even be sparse[7]. The error distribution $\mathcal{D}_e$ is typically a discrete Gaussian such that $|e| \ll |\boldsymbol{a}_i^\top \mathbf{sk}|$. Interestingly, solving the search LWE

[6] A brief overview of relevant algebra is provided in Appendix A.1.

[7] Sparse secrets should be treated with care because specialized attacks exist. For secure parameter choices, see [61].

problem also solves the decisional LWE problem and vice versa [194, Lemma 4.2]. Moreover, if LWE is computationally hard, then $(b_i, \boldsymbol{a}_i^\top) \overset{c}{\equiv} (b_i', \boldsymbol{a}_i^\top)$ holds, which provides a solid foundation to construct a cryptosystem. Intuitively, $\boldsymbol{a}_i^\top \mathbf{sk} \bmod q$ provides a uniformly random output, while computations over $\mathbb{Z}_q^N$ and the error e harden the problem.[8]

Before shifting our focus to the CKKS cryptosystem, we consider the polynomial ring variant of LWE (RLWE). To this end, the $2N$-th cyclotomic ring is $\mathcal{R} = \mathbb{Z}[X]/(X^N + 1)$ and the corresponding quotient ring is $\mathcal{R}_q = \mathcal{R}/q\mathcal{R} = \mathbb{Z}_q[X]/(X^N + 1)$.[9] Associated ring elements $a(X) = \sum_{i=0}^{N-1} \tilde{a}_i X^i \in \mathcal{R}_q$ are the remainders $\bmod X^N + 1$ and $\bmod q$. Thus, they have a degree of strictly less than N and they can naturally be identified by their coefficient vector $\tilde{\boldsymbol{a}} = (\tilde{a}_0, \ldots, \tilde{a}_{N-1})^\top \in \mathbb{Z}_q^N$. Whenever clear from context, we write a instead of $a(X)$ and omit $\bmod X^N + 1$ for brevity. The RLWE variant of Definition 7 and its decisional version are obtained by replacing $\mathbb{Z}_q^N$ (and analogously the distributions) with $\mathcal{R}_q$, i.e., $(b_i, a_i) = (-a_i \mathsf{sk} + e_i \bmod q, a_i) \in \mathcal{R}_q^2$ [158]. The reason for this algebraic abstraction lies in its higher efficiency. More precisely, the time and space complexity shrink from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$ and $\mathcal{O}(N \log(N))$, respectively, which is enabled by using the polynomial coefficients to encode multiple plaintexts and the number-theoretic transformation (see [188] and Appendix A.2) that enables fast multiplications over $\mathcal{R}_q$. Although using $\mathcal{R}_q$ introduces additional structure in an RLWE instance, it is not known how to exploit this for an attack such that LWE and RLWE provide the same security [163].

CKKS encoding. In CKKS, $\mathcal{R}_q$ serves as the message and ciphertext space. Thus, transitioning from $\mathbb{Z}_q$ to $\mathcal{R}_q$ requires another encoding strategy. Two obvious choices are a *scalar encoding* and a *packed encoding*. Given $z \in \mathcal{R}_q$, the former encodes an integer in the coefficients, e.g., $\tilde{\boldsymbol{z}} = (0, \ldots, 0, \lfloor sx \rceil)^\top \pmod q$, whereas the latter encodes multiple integers $\tilde{\boldsymbol{z}} = (\lfloor sx_0 \rceil, \ldots, \lfloor sx_{N-1} \rceil)^\top \pmod q$. Obviously, a packed encoding is more efficient because it uses the dimension of z. Now, for $z_1, z_2 \in \mathcal{R}_q$ the addition $z_1 + z_2$ is component-wise, however, $z_1 z_2$ is a (negative wrapped) convolution of the coefficient vectors $\tilde{\boldsymbol{z}}_1$ and $\tilde{\boldsymbol{z}}_2$, which is rarely a desired basic operation. Fortunately, there exists a *canonical embedding* from elements in $\mathbb{Q}[X]/(X^N + 1)$ to $\mathbb{C}^N$ (see [157, Section 2.5.2] for technical details). With this, addition and multiplication in $\mathbb{Q}[X]/(X^N + 1)$ correspond to their component-wise counterparts in $\mathbb{C}^N$. A natural extension to $\mathbb{R}[X]/(X^N + 1)$ then looks as follows. Let $\zeta = \exp(\mathrm{i}\pi/N)$ be the $2N$-th root of unity, where N is a power of 2. Then, the Vandermonde matrix

$$\boldsymbol{V} = \begin{pmatrix} 1 & \zeta^1 & \zeta^2 & \ldots & \zeta^{\frac{N}{2}-1} \\ 1 & \zeta^5 & \zeta^{2\cdot 5} & \ldots & \zeta^{(\frac{N}{2}-1)\cdot 5} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \zeta^{4\frac{N}{2}-3} & \zeta^{(4\frac{N}{2}-3)\cdot 5} & \ldots & \zeta^{(4\frac{N}{2}-3)(\frac{N}{2}-1)} \end{pmatrix} \in \mathbb{C}^{\frac{N}{2} \times N}$$

[8] Informal arguments for the hardness of the LWE problem are gathered in Appendix B.2.1.
[9] An introduction to $\mathcal{R}_q$ is provided in Appendix A.2.

specifies the following encoding, where a scaling factor $s \geqslant 1$ is used.

$$\underline{\mathsf{Ecd}_s(\boldsymbol{x})}.\ \text{Output } z \in \mathcal{R}_q, \text{ with } \tilde{\boldsymbol{z}} = \left\lfloor \frac{s}{N}(\overline{\boldsymbol{V}}^\top \boldsymbol{x} + \boldsymbol{V}^\top \overline{\boldsymbol{x}}) \right\rceil \bmod q, \boldsymbol{x} \in \mathbb{C}^{N/2}. \tag{2.12a}$$

$$\underline{\mathsf{Dcd}_s(z)}.\ \text{Output } \boldsymbol{x} \approx \overline{\boldsymbol{V}}\tilde{\boldsymbol{z}}/s \in \mathbb{C}^{N/2}, \text{ where } \tilde{\boldsymbol{z}} = (\tilde{z}_0, \tilde{z}_1, \ldots, \tilde{z}_{N-1})^\top. \tag{2.12b}$$

Here, $\mathsf{Dcd}_s(\mathsf{Ecd}_s(\boldsymbol{x})) \approx \boldsymbol{x}$ due to $\overline{\boldsymbol{V}}\boldsymbol{V}^\top = N$ and $\overline{\boldsymbol{V}}\boldsymbol{V}^\top = \boldsymbol{0}$ with the decoding error $\|\boldsymbol{V}\|_\infty/(2s) = N/s$. This way, CKKS enables single-instruction multiple-data (*SIMD*) operations on up to $N/2$ plaintexts via

$$\mathsf{Ecd}(\boldsymbol{x}_1) + \mathsf{Ecd}(\boldsymbol{x}_2) = \mathsf{Ecd}(\boldsymbol{x}_1 + \boldsymbol{x}_2) \quad \text{and} \quad \mathsf{Ecd}(\boldsymbol{x}_1)\mathsf{Ecd}(\boldsymbol{x}_2) = \mathsf{Ecd}(\boldsymbol{x}_1 \circ \boldsymbol{x}_2)$$

where $\boldsymbol{x}_1 \circ \boldsymbol{x}_2$ denotes component-wise multiplication.

Basic functionalities. In the next steps, we will introduce the functionalities of the CKKS scheme. In encrypted control, its leveled variant is most commonly used, which supports finite-depth computations. There, a base modulus q_0 is selected appropriately to avoid an overflow of the computation result. Then, the modulus of a ciphertext is $q_\ell = q_0 s^\ell$, where $\ell \in \{0, 1, \ldots, L\}$ is called "level", which can change during computations, and the maximum level L corresponds to the multiplicative depth.

$\underline{\mathsf{Setup}(1^\kappa, q_L)}$. Select the distributions $\mathcal{D}_s, \mathcal{D}_e, \mathcal{D}_r$, N as power of 2, and a large positive P such that an RLWE problem achieves κ-bit security for q_L.
$\underline{\mathsf{KeyGen}()}$. Output $\mathsf{sk} \leftarrow \mathcal{D}_s$ and $\mathsf{pk} = (b_{\mathsf{pk}}, a_{\mathsf{pk}})$, with $b_{\mathsf{pk}} = -a_{\mathsf{pk}}\mathsf{sk} + e_{\mathsf{pk}} \pmod{q_L}$, where $a_{\mathsf{pk}} \leftarrow \mathcal{R}_{q_L}$, and $e_{\mathsf{pk}} \leftarrow \mathcal{D}_e$.
$\underline{\mathsf{Enc}_{\mathsf{pk}}(z)}$. Output $\mathsf{ct}_{q_L}(z) = (b, a) = r\mathsf{pk} + (z + e_0, e_1) \pmod{q_L} \in \mathcal{R}_{q_L}^2$, with $z = \mathsf{Ecd}_s(\boldsymbol{x}) \in \mathcal{R}_q$, $r \leftarrow \mathcal{D}_r$, and $e_0, e_1 \leftarrow \mathcal{D}_e$.
$\underline{\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct}_{q_\ell}(z))}$. Output $z \approx b + a\,\mathsf{sk} \pmod{q_\ell}$, where $\mathsf{ct}_{q_\ell}(z) = (b, a)$.

Despite its apparent complexity at first glance, the encryption is remarkably simple. In fact, it embeds z into an RLWE instance

$$\mathsf{ct}_{q_L}(z) = (-ra_{\mathsf{pk}}\mathsf{sk} + re_{\mathsf{pk}} + z + e_0, ra_{\mathsf{pk}} + e_1) \approx (-a\,\mathsf{sk} + z + e, a) \pmod{q_L}$$

which allows stating the following.

Theorem 2 ([51]). *The CKKS cryptosystem is IND-CPA secure under the hardness assumption of the decisional* RLWE *problem.*

Interestingly, $\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct}_{q_L}(z)) = z + e$ contains a small error e, which is acceptable in most applications and characteristic for CKKS[10].

[10] If $z + e$ falls in the hands of $\mathcal{A}$, the RLWE hardness is lost and sk can be recovered [146]. To avoid this, one may exploit that $\|z\|_\infty > \|e\|_\infty$ among other techniques.

Homomorphisms. A major advantage of RLWE is its linearity, which enables:

$\underline{\mathsf{Add}(\mathsf{ct}_{q_\ell}(z_1), \mathsf{ct}_{q_\ell}(z_2))}$. Output $\mathsf{ct}_{q_\ell}(z_1 + z_2) = \mathsf{ct}_{q_\ell}(z_1) + \mathsf{ct}_{q_\ell}(z_2) \pmod{q_\ell}$.

$\underline{\mathsf{Mult}_{\mathsf{evk}}(\mathsf{ct}_{q_\ell}(z_1), \mathsf{ct}_{q_\ell}(z_2))}$. Output $\mathsf{ct}_{q_\ell}(z_1 z_2) = (d_0, d_1) + \lfloor d_2 \mathsf{evk}/P \rceil \pmod{q_\ell}$, where $(d_0, d_1, d_2) = (b_1 b_2, a_1 b_2 + a_2 b_1, a_1 a_2) \pmod{q_\ell}$, $\mathsf{ct}_{q_\ell}(z_1) = (b_1, a_1)$, and $\mathsf{ct}_{q_\ell}(z_2) = (b_2, a_2)$.

$\underline{\mathsf{KeyGen}_{\mathsf{evk}}(\mathsf{sk})}$. Output $\mathsf{evk} = (b_{\mathsf{evk}}, a_{\mathsf{evk}})$, where $a_{\mathsf{evk}} \leftarrow \mathcal{R}_{Pq_L}$ and $b_{\mathsf{evk}} = -a_{\mathsf{evk}}\mathsf{sk} + e_{\mathsf{evk}} + P\,\mathsf{sk}^2 \pmod{Pq_L}$.

$\underline{\mathsf{PAdd}(z_1, \mathsf{ct}_{q_\ell}(z_2))}$. Output $\mathsf{ct}_{q_\ell}(z_1 + z_2) = (z_1 + b_2, a_2) \pmod{q_\ell}$, where $z_1 = \mathsf{Ecd}_s(\boldsymbol{x}_1)$ and s is the scaling of z_2.

$\underline{\mathsf{PMult}(z_1, \mathsf{ct}_{q_\ell}(z_2))}$. Output $\mathsf{ct}_{q_\ell}(z_1 z_2) = z_1(b_2, a_2) \pmod{q_\ell}$, where $z_1 = \mathsf{Ecd}_s(\boldsymbol{x}_1)$.

While it is straightforward to verify that $\mathsf{ct}_{q_\ell}(z_1) \oplus \mathsf{ct}_{q_\ell}(z_2)$, $z_1 \boxplus \mathsf{ct}_{q_\ell}(z_2)$, and $z_1 \boxtimes \mathsf{ct}_{q_\ell}(z_2)$ provide correctness, $\mathsf{ct}_{q_\ell}(z_1) \otimes \mathsf{ct}_{q_\ell}(z_2)$ is somewhat more complex. To this end, CKKS adopts the ideas presented in [80, Section 4]. First, note that $z_1 z_2 \approx (b_1 + a_1\mathsf{sk})(b_2 + a_2\mathsf{sk}) = d_0 + d_1\mathsf{sk} + d_2\mathsf{sk}^2 \pmod{q_\ell}$, with (d_0, d_1, d_2) as in $\mathsf{Mult}_{\mathsf{evk}}$, is simple to compute but has one additional element, and sk^2 is required for decryption. Obviously, this is not sustainable for many multiplications. Therefore, $d_2\mathsf{sk}^2$ must be replaced. To this end, one uses the evaluation key $\mathsf{evk} = (b_{\mathsf{evk}}, a_{\mathsf{evk}})$ with the property $\mathsf{Dec}_{\mathsf{sk}}(d_2\mathsf{evk}/P) = d_2\mathsf{sk}^2 + d_2 e_{\mathsf{evk}}/P \pmod{q_\ell}$. The crucial role of $1/P$ is to keep the error term $d_2 e_{\mathsf{evk}}/P$ small so that it does not interfere with $z_1 z_2$. With this at hand, we see that $\mathsf{ct}_{q_\ell}(z_1) \otimes \mathsf{ct}_{q_\ell}(z_2) = (d_0, d_1) + \lfloor d_2\mathsf{evk}/P \rceil \pmod{q_\ell}$ is indeed correct, where the rounding ensures integer coefficients. Although not strictly homomorphic, the following additional functionalities are available.

$\underline{\mathsf{Rescale}_s(\mathsf{ct}_{q_\ell}(z))}$. Output $\mathsf{ct}_{q_{\ell-1}}(\lfloor z/s \rceil) = \lfloor \mathsf{ct}_{q_\ell}(z)/s \rceil \pmod{q_{\ell-1}}$.

$\underline{\mathsf{KeySwitch}_{\mathsf{swk}}(\mathsf{ct}_{q_\ell}(z))}$. Output $\mathsf{ct}'_{q_\ell}(z) = (b, 0) + \lfloor a\,\mathsf{swk}/P \rceil \pmod{q_\ell}$.

$\underline{\mathsf{KeyGen}_{\mathsf{swk}}(\mathsf{sk})}$. Output $\mathsf{swk} = (b_{\mathsf{swk}}, a_{\mathsf{swk}})$, where $a_{\mathsf{swk}} \leftarrow \mathcal{R}_{Pq_L}$, $\mathsf{sk}' \leftarrow \mathcal{D}_s$, $e_{\mathsf{swk}} \leftarrow \mathcal{D}_e$, $b_{\mathsf{swk}} = -a_{\mathsf{swk}}\,\mathsf{sk}' + e_{\mathsf{swk}} + P\,\mathsf{sk} \pmod{Pq_L}$.

Here, $\mathsf{Rescale}_s(\mathsf{ct}_{q_\ell}(z))$ outputs $\mathsf{ct}_{q_{\ell-1}}(\lfloor z/s \rceil) = (b, a)$ with a reduced level and is typically evaluated after every $\mathsf{Mult}_{\mathsf{evk}}$. For a better understanding, observe that $\lfloor b/s \rceil = \lfloor (-a\,\mathsf{sk} + e + z - Iq_\ell)/s \rceil$ with $I = \lfloor q_\ell^{-1}(-a\,\mathsf{sk} + z + e) \rceil$ which can be rearranged using the rounding errors $\epsilon_a = \lfloor a/s \rceil - a/s$ and $\epsilon_z = \lfloor (z+e)/s \rceil - (z+e)/s$ into $-\lfloor a/s \rceil\,\mathsf{sk} + \lfloor (z+e)/s \rceil + \lfloor \epsilon_a\,\mathsf{sk} - \epsilon_z \rceil \pmod{q_{\ell-1}}$. The term of interest is $\lfloor (z+e)/s \rceil$ which shows that the scaling factor of z and the error e can be reduced. This way, $\mathsf{Rescale}_s(\mathsf{ct}_{q_\ell}(z))$ enables L sequential multiplications, which is exponentially more than what is feasible without rescaling. The price of this is the error $\lfloor \epsilon_a\,\mathsf{sk} - \epsilon_z \rceil$, which is small for small $\|\mathsf{sk}\|_\infty$, and the reduced modulus that ultimately limits the computations. Next, as the name suggests, $\mathsf{KeySwitch}_{\mathsf{swk}}(\mathsf{ct}_{q_\ell}(z))$ allows switching from $\mathsf{ct}_{q_\ell}(z)$, encrypted under sk, to a ciphertext with the same z but encrypted under sk'. Based on our explanations above, this is straightforwardly verifiable.

For completeness, we note that there is also a Rotation and Conjugation functionality. Based on $\mathsf{ct}(z)$, the former cyclically rotates the message $z = \mathsf{Ecd}_s(\boldsymbol{x})$ utilizing the automorphism $\rho_j : z(X) \mapsto z(X^{5^j \bmod N}) \bmod X^N + 1$. Consequently, $\rho_j(\mathsf{ct}(z))$ is a valid encryption of $(x_j, \ldots, x_{N/2-1}, x_0, \ldots, x_{j-1})^\top$ but under the key $\rho_j(\mathsf{sk})$. Thus, a subsequent KeySwtich to sk is evaluated. Similarly, Conjugation is realized with ρ_{-1} and enables the computation of $\bar{\boldsymbol{x}}$ based on ciphertexts [45].

Example 2 (Encrypted polynomial feedback)**.** Due to its fully homomorphic properties, CKKS enables encrypted polynomial feedback. Let us consider the control inputs $u_i = g_{i,0} + g_{i,2}\, y^2$ with $i \in \{1, 2\}$, $(g_{1,0}, g_{1,2}, g_{2,0}, g_{2,2}) = (0.1, 0.2, 0.3, 0.4)$, and $y = 0.5$ which result in $u_1 = 0.15$ and $u_2 = 0.4$. The cryptosystem is parametrized by $L = 2$, $s = 10^4$, and the toy parameters $q_0 = 10^{12} + 39$, $N = 4$, and $P = 10^{15}$. SIMD operations are enabled through $z_y = \mathsf{Ecd}_s((y, y)^\top)$, $z_{g0} = \mathsf{Ecd}_s((g_{1,0}, g_{2,0})^\top)$, and $z_{g2} = \mathsf{Ecd}_s((g_{1,2}, g_{2,2})^\top) = 3000 - 707\, X + 707\, X^3 \,(\mathrm{mod}\, q_2)$. Next, via $\mathsf{Enc}_{\mathsf{pk}}$ we obtain the ciphertexts $\mathsf{ct}_{q_2}(z_y), \mathsf{ct}_{q_2}(z_{g0}), \mathsf{ct}_{q_2}(z_{g2})$ of the form (b, a), where, for example, $b = -41457... - 40721...\, X - 43622...\, X^2 + 17389...\, X^3$ and $a = -29055... - 42539...\, X - 11659...\, X^2 - 10825...\, X^3$. With this at hand, we prepare $\mathsf{ct}_{q_1}(z_y^2) = \mathsf{Rescale}_s(\mathsf{ct}_{q_2}(z_y) \otimes \mathsf{ct}_{q_2}(z_y))$ and $\mathsf{ct}_{q_0}(z_{g0}) = \mathsf{ct}_{q_2}(z_{g0}) \bmod q_0$, which ensures a common modulus, and compute the encrypted control inputs by

$$\mathsf{ct}_{q_0}(v) = \mathsf{ct}_{q_0}(z_{g0}) \oplus \mathsf{Rescale}_s\left(\mathsf{ct}_{q_1}(z_{g2}) \otimes \mathsf{ct}_{q_1}(z_y^2)\right).$$

Finally, $\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct}_{q_0}(v)) = 2750 - 884\, X + X^2 + 883\, X^3$ for which Dcd_s yields $(0.150, 0.399)^\top \approx (0.15, 0.4)^\top$ as desired.

Other HE schemes. Apart from the Paillier and CKKS cryptosystem, there exist several other HE schemes. Since a complete overview is too lengthy at this point, the interested reader can find it in Appendix B.2.2.

2.3 Secure multi-party computation

Shifting focus, we delve now into secure multi-party computation (SMPC) while adhering to the presentation pattern of the previous section. Finally, we introduce the SMPC instantiations: *additive secret sharing*, *oblivious transfer*, and *garbled circuits* in Sections 2.3.3–2.3.5.

2.3.1 Overview

SMPC provides a wide variety of protocols that allow $M \in \mathbb{N}$ parties $\mathsf{P}_1, \ldots, \mathsf{P}_M$ to jointly compute a functionality $(f_1, \ldots, f_M) = \boldsymbol{f}(z_1, \ldots, z_M)$, where the i-th party provides a secret input z_i and receives an output y_i. A protocol is *correct* if $y_i = f_i$ for all i and *private* if no party can learn anything about the inputs

of other parties except what can be inferred from their output y_i. To this end, SMPC provides a diverse set of protocols in which the idea is to divide information among parties.

Instead of a high computational overhead occurring in HE, communication speed in terms of latency and bandwidth is the bottleneck in SMPC. In particular, additive secret sharing relies on cheap local computations, but requires frequent communication between parties where small amounts of data are exchanged. Thus, the network's latency typically dominates the execution time. In this case, the number of *communication rounds* is a useful complexity metric. On the other hand, garbled circuits support a circuit-independent number of communication rounds. Here, typically large amounts of data must be exchanged, making the bandwidth an important consideration. In addition to these network aspects, computational overhead and memory footprint can become issues. This is especially true if large-depth computations are required or protocols invoke costly primitives such as HE.

2.3.2 Security of secure multi-party computation

Threat models in SMPC have two orthogonal dimensions: the behavior of corrupted parties and their number. Honest parties follow an SMPC protocol faithfully and do not try to infer information, whereas corrupted parties are mainly categorized into *semi-honest* (also known as *honest-but-curious*) and *malicious*. Note that the party's corruption can either be intrinsic or come from an adversary that got access to it, and corrupted parties are generally assumed to collaborate.

Definition 8 (semi-honest adversaries)**.** A party is semi-honest if it faithfully adheres to the protocol, yet retains and analyzes the record of received data to gain additional information beyond what the protocol stipulates.

Malicious parties, on the other hand, can in addition to semi-honest behavior deviate arbitrarily from a protocol, e.g., by sending false data. Although this thesis focuses on semi-honest parties, we briefly comment on malicious behavior in Section 4.2. Regarding the number of corrupted parties, if secret data in a protocol can be recovered by $t \in \{1, \dots, M\}$ collaborating parties, we write (t, M). In other words, such a protocol relies on a *non-collusion assumption* of t corrupted parties. Clearly, larger t makes this assumption more likely to be true.

The security analysis of SMPC protocols often builds on the *real/ideal paradigm*. There, one assumes the existence of a *trusted third party* that is incorruptible and faithful. This trusted third party then receives the inputs $(z_1, \dots, z_M)$ separately from each party via a secure channel, computes $\boldsymbol{f}(z_1, \dots, z_M)$, and provides each party with its output $y_i = f_i$. This imaginary setup is *ideal* since it trivially provides correctness and privacy. In comparison, these properties are not immediate in a *real* protocol execution, where parties interact. The idea

is now to construct a *simulation* of the real protocol based on what is available in the ideal setup. If it is impossible to distinguish between the simulation and the real execution, the real protocol inherits the properties of the ideal setup. A formalization of this idea can be based on the *view* of P_i during the real protocol execution, i.e., $\mathsf{view}_i = (z_i, r_i; \mathsf{msg}_1, \ldots, \mathsf{msg}_\vartheta)$, where z_i, r, and $\mathsf{msg}_1, \ldots, \mathsf{msg}_\vartheta$ denote P_i's input, randomness, and the received messages, respectively. In the ideal setup, the simulator $\mathcal{S}_i$ is provided with z_i and the correct output f_i, which we denote by $\mathcal{S}_i(z_i, f_i(z_1, \ldots, z_M))$. The task of $\mathcal{S}_i$ is now to generate "something" for a corrupted P_i that is computationally indistinguishable from its view, i.e., $\mathcal{S}_i(z_i, f_i(z_1, \ldots, z_M)) \stackrel{c}{\equiv} \mathsf{view}_i$ [150]. The next definition extends this to multiple parties.

Definition 9 (Privacy with respect to semi-honest parties [93])**.** Given is a correct protocol Π that computes deterministic functionalities $\boldsymbol{f}$ in the presence of $|\mathcal{H}|$ corrupted parties, where the index set is $\mathcal{H} = \{a_1, \ldots, a_{|\mathcal{H}|}\} \subset \{1, \ldots, M\}$. We say that Π computes $\boldsymbol{f}$ privately, if there exists a PPT simulator $\mathcal{S}$ such that for every $\mathcal{H}$ and all inputs, it holds that $\mathcal{S}(\mathcal{H}, \boldsymbol{z}_\mathcal{S}, \boldsymbol{f}_\mathcal{S}) \stackrel{c}{\equiv} \mathsf{view}_\mathcal{A}$. Here, the adversaries' view is $\mathsf{view}_\mathcal{A} = (\mathcal{H}, \mathsf{view}_{a_1}, \ldots, \mathsf{view}_{a_{|\mathcal{H}|}})$, while the simulator's inputs are $\boldsymbol{z}_\mathcal{S} = (z_{a_1}, \ldots, z_{a_{|\mathcal{H}|}})$ and $\boldsymbol{f}_\mathcal{S} = (f_{a_1}(z_1, \ldots, z_M), \ldots, f_{a_{|\mathcal{H}|}}(z_1, \ldots, z_M))$.

The modus operandi for $\mathcal{S}$ is often to construct "garbage" variables indistinguishable from the real ones. The appeal of *simulation-based* proofs using Definition 9 is two-fold. First, their privacy guarantee is easily interpretable due to the idealized world. Second, they enable a *composition* of protocols, which enables a modular approach toward more complex $\boldsymbol{f}$ (see [93, Theorem 7.3.3], [40] for sequential composition of protocols or [41] for concurrent composition with arbitrary other protocols).

2.3.3 Additive secret sharing

As an instantiation of SMPC, additive secret sharing (SS) is a multi-party protocol that builds on additively decomposing secrets over $\mathbb{Z}_q$. Consider M parties $\mathsf{P}_1, \ldots, \mathsf{P}_M$ with secrets $z_i \in \mathbb{Z}_q$. Then, every P_i can generate M shares $z_i^{(j)}$ of its secret z_i and reconstruct the secret in the following way:

$\underline{\mathsf{Setup}(M, q)}$. Every P_i selects $z_i^{(j)} \leftarrow \mathbb{Z}_q$ for $j \in \{1, \ldots, M\} \backslash \{k\}$, computes $z_i^{(k)} = z_i - \sum_{j=1, j\neq k}^{M} z_i^{(j)} \pmod q$, and distributes its shares $[z_i]$.
$\underline{\mathsf{Reconstruct}([z_i])}$. Collect all shares of z_i and compute $z_i = \sum_{j=1}^{M} z_i^{(j)} \pmod q$.
$\underline{\mathsf{Reveal}([z_i])}$. Invoke $\mathsf{Reconstruct}([z_i])$ and distribute z_i.

For a compact presentation, the shorthand notation $[z_i] = (z_i^{(1)}, z_i^{(2)}, \ldots, z_i^{(M)})$ is used. Above, distributing the shares means that every P_i sends its share $z_i^{(j)}$ to the corresponding P_j, which then owns $(z_1^{(j)}, z_2^{(j)}, \ldots, z_M^{(j)})$. Distributing z_i means sending it to all parties. Importantly, shares (and combinations of up to

$M-1$ shares) are uniformly distributed over $\mathbb{Z}_q$ for any z_i [219, Theorem 8.13]. Consequently, one can safely distribute the shares, which makes this scheme (M, M), and state the following.

Theorem 3. *Additive secret sharing provides perfect security.*

Additive secret sharing also supports arithmetic. In this context, operations on $[z_i]$ are understood component-wise, where each component is computed by one of the parties. With this at hand, the protocols are as follows:

$\underline{\mathsf{Add}([z_1],[z_2])}$. Locally compute $[z_1]+[z_2] \pmod q$.
$\underline{\mathsf{Mult}([z_1],[z_2])}$. Invoke Protocol 1 (below). We also denote this by $[z_1]\times[z_2]$.
$\underline{\mathsf{PAdd}(z_1,[z_2])}$. Locally compute $z_1+[z_2] := (z_1+z_2^{(1)}, z_2^{(2)}, \ldots, z_2^{(M)}) \pmod q$, where $z_1 \in \mathbb{Z}_q$.
$\underline{\mathsf{PMult}(z_1,[z_2])}$. Locally compute $z_1[z_2] \pmod q$, where $z_1 \in \mathbb{Z}_q$.

Verifying that the linear operations above provide correctness is straightforward using Reconstruct. However, it remains to specify Protocol 1, which follows next. Due to their nonlinearity, multiplications are more complex and require communication. To see this, note that $z_1z_2 = (z_1^{(1)}+\ldots+z_1^{(M)})(z_2^{(1)}+\ldots+z_2^{(M)}) \pmod q$ creates coupling terms $z_1^{(i)}z_2^{(j)}$ with $i \neq j$ that can neither be computed by P_i nor by P_j alone. In other words, this information must somehow be communicated without revealing it. A generic approach to this end is through *multiplication triples* α, β, γ, which are used in Protocol 1 [18, 72].

Protocol 1: $\mathsf{Mult}([z_1],[z_2])$ via triples

Preprocessing phase

1: $\alpha \leftarrow \mathbb{Z}_q$, $\beta \leftarrow \mathbb{Z}_q$, $\gamma = \alpha\beta \pmod q$.
2: Create and distribute $[\alpha]$, $[\beta]$, $[\gamma]$ //1 communication round

Execution phase

1: $[\delta] = [z_1]-[\alpha] \pmod q$ and $[\epsilon] = [z_2]-[\beta] \pmod q$
2: $\mathsf{Reveal}([\delta])$ and $\mathsf{Reveal}([\epsilon])$ // 1 communication round
3: $[z_1z_2] = [\gamma]+\delta[\beta]+\epsilon[\alpha]+\delta\epsilon \pmod q$

Clearly, $\mathsf{Reconstruct}([z_1z_2]) = \gamma+\delta\beta+\epsilon\alpha+\delta\epsilon = (z_1-\alpha+\alpha)(z_2-\beta+\beta) \pmod q$ which shows correctness. For privacy, note that δ and ϵ are uniformly random in $\mathbb{Z}_q$ by construction. In fact, all messages received in the above protocols can be simulated by sampling uniformly at random from $\mathbb{Z}_q$. Hence, for $|\mathcal{H}| < M$ semi-honest parties, they provide a basis for (perfectly) private functionalities.

In order to ensure privacy, triples cannot be reused or revealed. Thus, the preprocessing phase of Protocol 1 is an important step which can be realized, e.g., by a *dealer* or interactively among the parties. In the former case, the dealer is solely responsible for generating and distributing $[\alpha], [\beta], [\gamma]$, which

leads to a high performance. For privacy reasons, the dealer is not allowed to participate in the computations. In the latter case, the advantage lies within the generality of the setup. However, the required protocols use costly primitives such as oblivious transfer or HE (see [60, Chapter 8] or [126, 127]). Fortunately, due to the triples' independence of the input data, they can be precomputed offline, which trades memory usage for reduced online computation times.

Example 3 (Privacy-preserving polynomial feedback)**.** Let us reconsider $u = 0.1 + 0.2y^2$ with $y = 0.5$ as a functionality and evaluate it privately with $M = 2$ parties. To this end, the plant selects $s = 10$, $q = 2^{16}$ and encodes its inputs via $z_y = \mathsf{Ecd}_s(y) = 5$, $z_{p0} = \mathsf{Ecd}_s(0.1) = 1$, $z_{p2} = \mathsf{Ecd}_s(0.2) = 2$. Next, shares $\{[z_y], [z_{p0}], [z_{p2}]\} = \{(63234, 2307), (10329, 55208), (63608, 1930)\}$ and two sets of triples $[\alpha], [\beta], [\gamma]$ are created and distributed between the parties. Then, the parties compute

$$[v] = s^2[z_{p0}] + [z_{p2}] \times [z_y] \times [z_y] \,(\mathrm{mod}\, q),$$

which requires two communication rounds (assuming preprocessing is done beforehand) due to the two non-parallelizable multiplications. Note that s^2 ensures the same scaling between both terms. Finally, the parties forward their shares of v to the plant, which can then reconstruct v and decode it to $\mathsf{Dcd}_{s^3}(v) = 0.15$.

Other secret sharing schemes. Apart from additive secret sharing, *Shamir's* [216] and *replicated secret sharing* [24] are noteworthy in this context. The former is based on polynomials $\eta(x) = z + \eta_1 x + \eta_2 x^2 + \cdots + \eta_t x^t \,(\mathrm{mod}\, q)$ where $t \in \mathbb{N}$, $\eta_1, \ldots, \eta_t \leftarrow \mathbb{Z}_q$ and q is prime. Then, $[z] = (\eta(1), \ldots, \eta(M))$, which enables reconstructing the polynomial and $z = \eta(0)$ by at least $t + 1$ shares [60, p. 244]. Thus, this scheme is (t, M) and provides more flexibility. The latter method is based on additive secret sharing, but typically $M = 3$ parties are used, where every party receives 2 shares of each secret, resulting in a $(1, 3)$ scheme that requires encrypted communications. The benefits of these schemes encompass multiplication without preprocessing and more efficient scalar products. For both of these schemes, *self-trust* of a party is insufficient to ensure the confidentiality of its secret because the remaining parties can collude for a reconstruction.

2.3.4 Oblivious transfer

Oblivious transfer (OT) [191] occurs on several occasions as an important primitive in many SMPC protocols. To understand it, let us consider the following data exchange between Claude and Plantin, which represent a cloud server and a plant, respectively. Claude has two values z_0, z_1, from which Plantin selects one value z_b with $b \in \{0, 1\}$, which is revealed to Plantin while Claude receives nothing, i.e., $(z_b, \perp) = \boldsymbol{f}(b, (z_0, z_1))$. OT then realizes this functionality,

but Claude remains oblivious to Plantin's choice b. If Plantin can select 1-out-of-2 values, the protocol is denoted by OT_2^1, which is illustrated in Protocol 2.

Protocol 2: OT_2^1 via homomorphic encryption

Plantin selects $b \in \{0,1\}$		**Claude has** z_0, z_1
Generates keys $(\mathsf{pk}, \mathsf{sk})$		
Encrypts b	$(\mathsf{pk}, \mathsf{ct}(b))$ $\longrightarrow$	
		$\mathsf{ct}(z_b) = ((1 \boxminus \mathsf{ct}(b)) \boxtimes z_0) \oplus (\mathsf{ct}(b) \boxtimes z_1)$
	$\longleftarrow$ $\mathsf{ct}(z_b)$	
$z_b = \mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct}(z_b))$		

Clearly, Protocol 2 is expensive because it uses HE for didactic purposes. Although homomorphisms are not strictly required here, OT is unfortunately only known to be constructed with public key cryptography [121]. Furthermore, OT can be realized securely.

Theorem 4 ([152]). *In the presence of a semi-honest adversary, the OT_2^1-functionality can be computed privately.*

Intuitively, Claude's view can be simulated due to the guarantees of the cryptosystem. Since OT uses public key cryptography as primitives, its security guarantee is computational.

A natural extension is OT_k^1, in which Claude holds k values, and Plantin chooses $b \in \{0, 1, \ldots, k-1\}$. With this at hand, any function f with k possible outputs can be queried in an oblivious manner. Generic functions then follow from the observation that OT_4^1 enables oblivious logic gates. This is used in the Goldreich-Micali-Wigderson (GMW) protocol [93]. However, relying on OT and a circuit-dependent number of communication rounds makes GMW expensive.

2.3.5 Garbled circuits

Garbled circuits (GCs) [20] is a two-party protocol. Here, we assume that P_1 and P_2 provide inputs $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_m)$ and $\boldsymbol{w} = (\omega_1, \omega_2, \ldots, \omega_m)$, respectively, and that they want to evaluate $\boldsymbol{f}(\boldsymbol{\xi}, \boldsymbol{\omega})$. To this end, the idea is that the *garbler* represents $\boldsymbol{f}$ in terms of a sequence of lookup tables, but it replaces the inputs and outputs with randomly selected labels ℓ. The result is called "garbled circuit", which is then sent to the *evaluator* who does not know the labels' correspondences. Still, based on the input labels, the evaluator can compute the output label of each lookup table until it reaches the final output. In practice, logic gates are typically used such that $\boldsymbol{f}$ becomes a Boolean circuit. Next, we shed more light on these steps based on the simple example $f(\xi, \omega) = \mathsf{AND}(\xi, \omega)$, where $(\xi, \omega) \in \{0,1\}^2$.

Table 2.2. Garbling of an AND gate.

AND gate with labels			encrypt	outputs	permute	garbled AND gate
ℓ_0^ξ	ℓ_0^ω	ℓ_0^y	$\longrightarrow$	$\mathsf{Enc}_{\ell_0^\xi \| \ell_0^\omega}(\ell_0^y)$	$\longrightarrow$	$\mathsf{Enc}_{\ell_0^\xi \| \ell_1^\omega}(\ell_0^y)$
ℓ_0^ξ	ℓ_1^ω	ℓ_0^y		$\mathsf{Enc}_{\ell_0^\xi \| \ell_1^\omega}(\ell_0^y)$		$\mathsf{Enc}_{\ell_1^\xi \| \ell_0^\omega}(\ell_0^y)$
ℓ_1^ξ	ℓ_0^ω	ℓ_0^y		$\mathsf{Enc}_{\ell_1^\xi \| \ell_0^\omega}(\ell_0^y)$		$\mathsf{Enc}_{\ell_1^\xi \| \ell_1^\omega}(\ell_1^y)$
ℓ_1^ξ	ℓ_1^ω	ℓ_1^y		$\mathsf{Enc}_{\ell_1^\xi \| \ell_1^\omega}(\ell_1^y)$		$\mathsf{Enc}_{\ell_0^\xi \| \ell_0^\omega}(\ell_0^y)$

Garbling. The process of garbling is illustrated in Table 2.2. First, the garbler replaces all possible inputs and outputs in the AND gate with the random labels $\ell_0^\xi, \ell_1^\xi, \ell_0^\omega, \ell_1^\omega, \ell_0^y$, and ℓ_1^y where the indices are purely for presentation. The outputs are then encrypted using the corresponding concatenated input labels as secret keys, e.g., $\ell_0^\xi || \ell_0^\omega$. This way, permission to the gate's output is regulated. Finally, the outputs are randomly permuted, which prevents the evaluator from exploiting positional information later on. We denote this result by f_{GC}. Then, the garbler sends f_{GC} and its input label ℓ_ξ^ξ to the evaluator. In order to send the correct label ℓ_ω^ω to the evaluator, OT_2^1 is executed.[11] This way, neither ω nor both labels $\ell_0^\omega, \ell_1^\omega$ are revealed.

Evaluating. Based on the labels $\ell_\xi^\xi, \ell_\omega^\omega$ and f_{GC}, the evaluator can only correctly decrypt the related output label ℓ_y^y and none of the others. Note, however, that it has no means to decide which row of f_{GC} to decrypt because an unsuccessful decryption looks like a random label. To resolve this, the garbler can add markers to the labels, e.g., prepending two bits that indicate the correct row in f_{GC} [19]. In a more complex circuit, output labels, such as ℓ_y^y, serve as input labels for the next gate. For such f_{GC}, Protocol 3 summarizes the process. The final results $\ell_{f_i}^{f_i}$ and their correspondences can then be exchanged between the garbler and the evaluator.

Protocol 3: Garbled circuit evaluation for $f(\xi, \omega)$

Garbler has inputs ξ		**Evaluator has inputs ω**
Garbles the circuit $f(\xi, \omega)$	$f_{\mathrm{GC}}, (\ell_{\xi_1}^{\xi_1}, \ldots, \ell_{\xi_m}^{\xi_m})$ $\longrightarrow$	
	OT_2^1 for each $\ell_{\omega_i}^{\omega_i}$ $\longleftrightarrow$	
		Evaluate f_{GC} gate-by-gate until $\ell_{f_i}^{f_i}$.

The security of Protocol 3 is captured by the following theorem.

[11] For efficiency, the OT extension technique should be used [123]. There, after a small number of OTs, many more can be generated with fast symmetric primitives.

Theorem 5 ([152]). *In the presence of a semi-honest adversary, garbled circuits compute f privately.*

Here, the simulation proof essentially relies on a secure implementation of OT and a fake correspondence table. Due to OT, the computational security guarantee is inherited. A crucial limitation of GCs is that they offer one-time usage only. For instance, if the evaluator obtains another set of input labels for the same garbled circuit, reconstructing gate inputs can become feasible.

To improve the performance of GCs, several optimizations are available such as row-reduction [176], Free-XOR [140], and Half-gates [250]. These techniques reduce the number of gates to be communicated or the number of encryption and decryption calls. The main cost for GCs in most scenarios is network bandwidth, while the computation cost can also become substantial and is dominated by encryption (mainly for the garbler). Therefore, the encryption is typically replaced with a more efficient hash function (or even AES-NI instructions [22]), which significantly increases the evaluation speed and reduces the memory footprint. Moreover, pipelined execution [118], where the evaluation occurs during the garbling, eliminates the need to store the entire circuit. A multi-party extension of GCs is BMR [19], where all parties simultaneously contribute to garbling the circuit and then share their parts.

Chapter 3

Article Summaries and Discussions

Building on the preparations outlined in the previous chapter, we recognize profound challenges inherent to privacy-preserving computations. Key obstacles include efficient iterations and non-polynomial functions. Consequently, often seemingly simple computations result in non-trivial challenges shaping the landscape of encrypted control solutions and necessitate novel design strategies. This chapter addresses these issues by contextualizing our contributions in the overarching field of privacy-preserving computations. In particular, we illustrate the problems resulting from iterations and non-polynomial functions with regard to control, explore their implications, and propose tailored solutions in Sections 3.1 and 3.2. Although controller realizations based on HE or SMPC offer strict security guarantees, they require significant computation or communication resources. As a more efficient alternative, RATs have gained traction, though they lack a rigorous cryptographic security analysis. Section 3.3 addresses this gap by discussing several pertinent issues. For more details and the full versions of the summarized papers, we refer to Part II of this thesis.

Threat models. In the following, we consider two threat models, which define the context for our results. First, in cloud-based setups, we consider an honest plant in possession of confidential control-related data and a semi-honest cloud. This setup is special in the sense that it assumes asymmetric trust. The plant is interested in outsourcing a computation by using a cloud service. Consequently, the plant has no incentive to behave maliciously. On the other hand, the cloud could, in principle, be malicious. However, this implies an erroneous computation and jeopardizes the cloud's reputation and business model. Therefore, we assume that the cloud will execute program code and protocols as specified but may try to infer as much information as possible, i.e., it is semi-honest as in Definition 8. Second, in multi-party setups, we also consider semi-honest parties. While this can be motivated in scenarios where, e.g., the parties own confidential input data that can be combined in a fruitful collaboration, it also serves as a stepping stone for protocols that provide security against malicious parties. We comment on malicious behavior in more detail in Section 4.2.

3.1 Iterative controllers

3.1.1 Problem overview

Let us begin by understanding the problem that iterations entail when evaluated with privacy. To this end, we reconsider the dynamic controller (1.2) for which a suitable encoding in $\mathbb{Z}_q$ is given by

$$\boldsymbol{z}(k+1) = \lfloor s\boldsymbol{H} \rceil\, \boldsymbol{z}(k) + \lfloor s\boldsymbol{G} \rceil\, \lfloor s^{k+1}\boldsymbol{y}(k) \rceil \,(\mathrm{mod}\, q) \tag{3.1a}$$

$$\boldsymbol{v}(k) = \lfloor s\boldsymbol{E} \rceil\, \boldsymbol{z}(k) + \lfloor s\boldsymbol{F} \rceil\, \lfloor s^{k+1}\boldsymbol{y}(k) \rceil \,(\mathrm{mod}\, q), \tag{3.1b}$$

with the integer state $\boldsymbol{z}(k) = \lfloor s\boldsymbol{x}_c(k) \rceil \,(\mathrm{mod}\, q)$ and the control input $\boldsymbol{v}(k)$. Note that for a correct decoding, (3.1) is constructed such that sums possess the same scaling factor. A privacy-preserving variant of (3.1) can be realized with HE or SMPC primitives, since Add and Mult suffice. Due to the equivalence between (3.1) and its privacy-preserving version, as stated in (2.3), we analyze the former and omit stating the latter.

Now, observe that the autonomous part $\boldsymbol{z}(k+1) = \lfloor s\boldsymbol{H} \rceil \boldsymbol{z}(k)$ increases the scaling with every iteration, which is reflected by $\lfloor s^{k+1}\boldsymbol{y}(k) \rceil$. Consequently, $\mathsf{Dcd}_{s^{k+2}}(\boldsymbol{v}(k)) \approx \boldsymbol{u}(k)$ is correct as long as $\| \lfloor s\boldsymbol{E} \rceil \boldsymbol{z}(k) + \lfloor s\boldsymbol{F} \rceil \lfloor s^{k+1}\boldsymbol{y}(k) \rceil \|_\infty \in [-q/2, q/2) \cap \mathbb{Z}$. However, if the number of iterations is unlimited or unknown beforehand, a suitable choice for (q, s) is unattainable. Thus, without countermeasures, an overflow of $\mathbb{Z}_q$ will almost certainly occur after a finite number of iterations. In this case, the decoded control input $\boldsymbol{u}(k)$ would be highly erroneous, putting the control performance and closed-loop stability at risk. A prominent control example, where an overflow of integer numbers led to a catastrophic failure, is the Ariane 5 rocket [154]. Thus, rescaling $\boldsymbol{z}(k)$ regularly, e.g., via $\lfloor \boldsymbol{z}(k)/s \rceil$, to limit its scaling factor growth is essential for an unlimited number of iterations.

Bootstrapping. One way to address this problem is by using FHE. There, an unlimited number of iterations is realized via regular execution of bootstrapping which generates a "refreshed ciphertext" enabling further encrypted computations [91]. In this context, we first note that an implementation of the dynamic controller (1.2) with CKKS slightly differs from (3.1) but results in an analogous problem. In particular, the encoding and decoding (2.12) must be adapted, i.e., $z = \mathsf{Ecd}_s(\boldsymbol{x}_c) \in \mathcal{R}_q$, and instead of an increasing scaling factor, $\mathsf{Rescale}_s$ keeps the scaling factors constant throughout the computation at the cost of reducing q_ℓ to $q_{\ell-1}$ (see Section 2.2.4). This prohibits further computations when q_0 is reached. Thus, in order to avoid an overflow at this point, each component of the encoded form of (1.2b), which corresponds to $\lfloor s\boldsymbol{E} \rceil \boldsymbol{z}(k) + \lfloor s\boldsymbol{F} \rceil \lfloor s\boldsymbol{y}(k) \rceil$, must be smaller in absolute value than $q_0/2$. For further computations, bootstrapping then lifts $\mathsf{ct}_{q_0}(z)$ to $\mathsf{ct}_Q(z)$ with $Q \gg q_0$. This transforms leveled CKKS into an FHE scheme. More precisely, considering $\mathsf{ct}_{q_0}(z) = (b, a)$ under the modulus Q yields a ciphertext $\mathsf{ct}_Q(z')$, where

$z' = z - Iq_0$ with $I = \lfloor (-a\,\mathsf{sk} + z + e)\,/q_0 \rceil \in \mathcal{R}$. The challenge then lies in eliminating $-Iq_0$, which can be achieved through a homomorphic modulo q_0 reduction. Rather than using the discontinuous modulo function, an approximation by $q_0/(2\pi)\sin(2\pi z'/q_0)$ over a bounded interval is typically employed, which requires $z \ll q_0$. Finally, a homomorphic encoding is used to ensure a reduction in each coefficient of z'. Unfortunately, these steps make bootstrapping computationally costly, i.e., it can take several seconds to minutes depending on the parameter choices. See [50] for the initial procedure and [16, 34] for the state-of-the-art. Hence, further improvements are required to make bootstrapping practical for control applications.

Rescaling protocols. The integer controller (3.1) can be used directly in secret sharing. There, rescaling protocols are available which provide a scaling reduction of z via $[\![\lfloor z/s \rceil]\!]$. Instead of building on perfectly secure protocols [64], competitive methods [42, 43, 63, 171] use a *mask-and-open* approach. Therein, the parties reveal the additively masked share $[\![z+r]\!] = [\![z]\!] + [\![r]\!] \pmod{q'}$. This way, protocols can be significantly simplified. For $z \in \mathbb{Z}_q$, security then requires that $r \leftarrow [0, 2^\kappa(q-1)] \cap \mathbb{Z}$. Thus, the scheme's modulus must be increased to $q' > (q-1)(2^\kappa + 1)$ (see Appendix B.1.3 for details). Suitable $[\![r]\!]$ and other *correlated randomness* can be precomputed (similar to multiplication triples), in a preprocessing phase [79]. However, for a large number of iterations or memory-constrained devices, preprocessing has to be re-executed during the execution phase, which typically leads to impractical runtimes.

3.1.2 Summaries of articles [P3, P8, P11, P16]

Cloud-based. As noted above, arithmetic circuits with arbitrary depth can in principle be evaluated with HE and SMPC. Unfortunately, current approaches do not satisfy the needs of control in terms of efficiency. This begs the question whether a system theoretic approach can be a remedy for implementing (3.1). To this end, observe that $\boldsymbol{H} \in \mathbb{Z}^{n\times n}$ allows for $s = 1$ in the encoding and thus circumvents an accumulation of scaling factors. Consequently, q can be designed to robustly prevent overflows while no additional cost occurs during the privacy-preserving evaluation of (3.1).

In [P16], we therefore analyze the resulting limitations for such $\boldsymbol{H}$ by means of its characteristic polynomial

$$\lambda^n + h_{n-1}\lambda^{n-1} + \cdots + h_1\lambda + h_0, \quad \text{where} \quad h_{n-1}, \ldots, h_0 \in \mathbb{Z}. \tag{3.2}$$

Since encrypted controllers operate in a networked setup, especially Schur stable and marginally stable[1] $\boldsymbol{H}$ are of interest. The rationale for this is that packet delays and dropouts would otherwise quickly lead to a diverging controller,

[1] While the eigenvalues of a Schur stable $\boldsymbol{H}$ satisfy $|\lambda_i| < 1$, the eigenvalues of marginally stable $\boldsymbol{H}$ fulfill $|\lambda_i| \leqslant 1$ and $\lambda_i \neq \lambda_j$ for $|\lambda_i| = |\lambda_j| = 1$ with $i, j \in \{1, \ldots, n\}$.

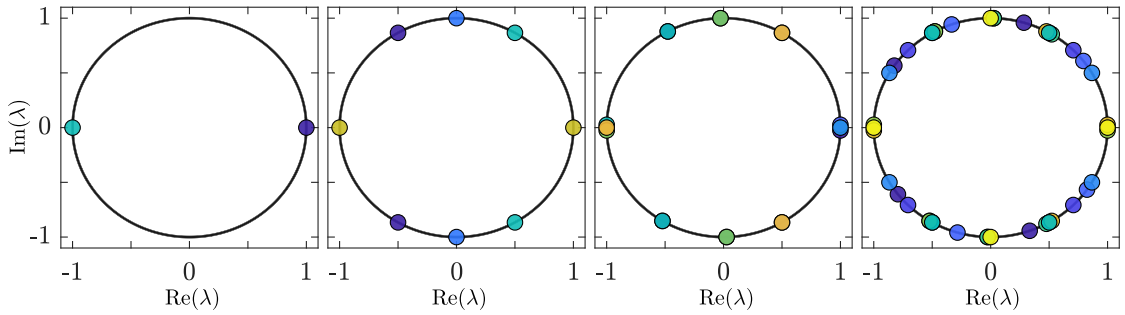

Figure 3.1. Marginally stable roots with degree $n \in \{1, 2, 3, 4\}$ from left to right. Roots corresponding to the same polynomial share the same color, and overlapping roots are slightly shifted.

which is moreover impractical to detect homomorphically. In the case of Schur stability, we then find the following.

Corollary 1 ([P16, Corollary 3]). *The roots of* (3.2) *are Schur stable if and only if* $h_{n-1} = \cdots = h_0 = 0$.

Thus, the only Schur stable choice is a finite impulse response (FIR) controller of the form

$$\boldsymbol{u}(k) = \sum_{i=0}^{n} \boldsymbol{\mathcal{F}}_i \boldsymbol{y}(k-i). \tag{3.3}$$

Next, we analyze marginally stable roots based on cyclotomic polynomials

$$\Phi_p(\lambda) = \prod_{\substack{1 \leqslant j \leqslant p \\ \gcd(j,p)=1}} \left(\lambda - \exp\left(\frac{2\pi \mathrm{i} j}{p}\right)\right) \quad \text{with} \quad p \in \mathbb{N} \quad \text{but} \quad p \neq 0. \tag{3.4}$$

Here, the p-th cyclotomic polynomial is the unique irreducible monic polynomial with integer coefficients that is a divisor of $\lambda^p - 1$ and not a divisor of $\lambda^j - 1$ for any $j \in \{1, \ldots, p-1\}$. Thus, by increasing p, only new roots are considered. Remarkably, the roots of Φ_p are the marginally stable roots of (3.2).

Corollary 2 ([P16, Corollary 4]). *The roots of* (3.2) *are marginally stable if and only if there exists a nonempty set* $\mathcal{L}$ *such that* $n_0 \leqslant n$ *and*

$$\lambda^n + h_{n-1}\lambda^{n-1} + \cdots + h_1\lambda + h_0 = \lambda^{n-n_0} \prod_{p \in \mathcal{L}} \Phi_p(\lambda). \tag{3.5}$$

For a specification of $\mathcal{L}$, see [P16, Section 4]. The number of possible marginally stable controller dynamics is then given by $|\mathcal{L}|$ and increases strictly with n. An illustration is depicted in Figure 3.1. As apparent from the figure, the number of different root configurations increases quickly with n, but specific locations may require high controller orders, e.g., a root at 14.4 degrees requires at least $n = 20$. Integer variants of discrete-time proportional-integral-derivative controllers are also affected by these results. In fact, the integral part

is always marginally stable, whereas for common discretization methods, stable derivative parts are not of practical use. Finally, we note that non-minimal realizations of (3.1) imply unstable pole-zero cancellations for stable controllers and thus cannot enhance the flexibility in practice [229, 230].

In [P8], we discuss the advantages of encrypted FIR controllers in comparison to other privacy-preserving realizations of (1.2). That is, encrypted FIR controllers do not require additional communication to a TTP, their inherent stability makes them robust against packet dropouts and delays, and they can be efficiently evaluated because the multiplicative depth of (3.3) is 1. Moreover, for $\boldsymbol{u} \in \mathbb{R}^m$, m parallel Mult suffice to evaluate (3.3) in an RLWE-based cryptosystem. To this end, one uses the rows of $(\lfloor s\boldsymbol{\mathcal{F}}_0 \rceil, \lfloor s\boldsymbol{\mathcal{F}}_1 \rceil, \ldots, \lfloor s\boldsymbol{\mathcal{F}}_n \rceil) \bmod q$ and $(\lfloor s\boldsymbol{y}(k)^\top \rceil, \lfloor s\boldsymbol{y}(k-1)^\top \rceil, \ldots, \lfloor s\boldsymbol{y}(k-n)^\top \rceil)^\top \bmod q$ as coefficients in the message polynomials and observes that the (standard) convolution, required for (3.3), is evaluated in the X^{N-1} coefficient of the ciphertext. For the design of FIR controllers, we propose to approximate a given Schur stable controller in the form (1.2). Using a finite window of the controller's response, we find $\boldsymbol{\mathcal{F}}_0 = \boldsymbol{F}$ and $\boldsymbol{\mathcal{F}}_j = \boldsymbol{E}\boldsymbol{H}^{j-1}\boldsymbol{G}$ for all $j \in \{1, \ldots, n\}$. Additionally, we rely on an H_∞-optimal design and nonlinear programming to minimize the FIR's order.

With this at hand, the performance of FIR controllers in comparison to reset controllers is evaluated. To this end, we consider a benchmark system in networked control which is unstable and non-minimum phase (see [P8, Section 6] for details) and approximate two reset controllers from the literature. These periodically reset their state according to

$$\boldsymbol{z}(k+1) = \begin{cases} \boldsymbol{z}_{\text{reset}} & \text{if } k+1 \bmod T = 0, \\ \lfloor s\boldsymbol{H} \rceil \, \boldsymbol{z}(k) + \lfloor s\boldsymbol{G} \rceil \, \lfloor s^{k+1}\boldsymbol{y}(k) \rceil \, (\bmod q) & \text{otherwise,} \end{cases} \tag{3.6}$$

where $\boldsymbol{z}_{\text{reset}}$ is a predefined value and T is the reset interval. This way, they provide an alternative method for a privacy-preserving evaluation of (3.1). In our experiments, FIR controllers provide better performance in terms of convergence speed and magnitude of the controller state than their reset counterparts [P8, Figure 6.3], while they can be evaluated more efficiently.

Multi-agent systems. In [P3] we focus on a distributed solution of a generalized consensus problem in multi-agent systems. The setup considers bidirectional communications and a system operator who specifies the cooperative control task by parameterizing a separable cost function to be minimized by the agents. More precisely, at each sampling instance, the agents solve

$$\min_{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M, \boldsymbol{\zeta}} \; \sum_{i=1}^{M} J_i(\boldsymbol{x}_i, \boldsymbol{p}_i) \quad \text{s.t.} \quad \boldsymbol{x}_i = \boldsymbol{\zeta}_{\mathcal{K}_i}, \tag{3.7}$$

which has multiple applications in cooperative control but requires the exchange of possibly sensitive data between agents. Thus, we propose a privacy-preserving distributed solution of (3.7). Here, $i \in \{1, \ldots, M\}$ and M is the

number of agents. Furthermore, $\boldsymbol{x}_i \in \mathbb{R}^{n_i}$, $\boldsymbol{p}_i \in \mathbb{R}^{\psi_i}$, and $J_i : \mathbb{R}^{n_i} \times \mathbb{R}^{\psi_i} \to \mathbb{R}$ denote local decision variables, parameters, and the cost function of the i-th agent, respectively, whereas $\boldsymbol{\zeta} \in \mathbb{R}^m$ refers to a global decision variable. Lastly, the ordered index sets $\mathcal{K}_i \subseteq \{1, \dots, m\}$ specify which entries in the global $\boldsymbol{\zeta}$ are associated with the local $\boldsymbol{x}_i$. We assume $J_i(\boldsymbol{x}_i, \boldsymbol{p}_i) = \boldsymbol{x}_i^\top \boldsymbol{Q}_i \boldsymbol{x}_i/2 + \boldsymbol{p}_i^\top \boldsymbol{V}_i \boldsymbol{x}_i$, where $\boldsymbol{Q}_i$ is a positive definite Hessian, and add equality constraints of the form $\boldsymbol{T}_i \boldsymbol{x}_i = \boldsymbol{S}_i \boldsymbol{p}_i$ that can, e.g., incorporate the dynamics of the agents. By applying the alternating direction method of multipliers (ADMM), a suitable initialization, and rearrangements of the resulting formulas (see [P3, Sections 3 and 4.2] for details), we obtain the iterations

$$\boldsymbol{x}_i(k+1) = \rho \boldsymbol{\Gamma}_{i,\zeta} \boldsymbol{\zeta}_{\mathcal{K}_i}(k) - \boldsymbol{\Gamma}_{i,\nu} \boldsymbol{\nu}_i(k) + \boldsymbol{\Gamma}_{i,p} \boldsymbol{p}_i \tag{3.8a}$$

$$\boldsymbol{\nu}_i(k+1) = \boldsymbol{\nu}_i(k) + \rho \left(\boldsymbol{x}_i(k+1) - \boldsymbol{\zeta}_{\mathcal{K}_i}(k+1) \right) \tag{3.8b}$$

$$\zeta_\phi(k+1) = \frac{1}{|\mathcal{I}_\phi|} \sum_{i \in \mathcal{I}_\phi} x_{i,\phi_i}(k+1), \tag{3.8c}$$

where $\rho, \boldsymbol{\Gamma}_{i,\zeta}, \boldsymbol{\Gamma}_{i,\lambda}, \boldsymbol{\Gamma}_{i,p}$ are parameters and $\boldsymbol{\nu}$ is the Lagrange multiplier. The distributed evaluation forces us to introduce ϕ_i as the local position of the global entry ϕ (x_{i,ϕ_i} is the ϕ_i-th entry of $\boldsymbol{x}_i$) and the sets $\mathcal{I}_\phi = \{i \in \{1, \dots, M\} \mid \phi \in \mathcal{K}_i\}$, where $\mathcal{I}_\phi$ collects all agents that make use of ζ_ϕ. In this context, we assume that for each ζ_ϕ, there exists at least one agent i who is able to evaluate (3.8c) by collecting information from its neighbors.

Importantly, the distributed iterations (3.8) converge to a solution of (3.7) which is compatible with the agents, and they can be realized using Add and Mult which enables privacy via HE or SS. In contrast to (3.1), the multiplicative depth of (3.8) is not arbitrary because time constraints on the evaluation apply. Thus, we use the maximum realizable number of iterations in (3.8), which can be viewed as a finite approximation similar to the cloud-related analysis above. The remaining challenge for a private realization of (3.8) is to specify how variables and parameters are provided to each agent without leaking information.

While a SS solution cannot exploit the benefits of the distributed evaluation (3.8) well, HE operates communication-wise in the same way as a plaintext solution. Thus, our realization builds on HE, where we deploy multiple instantiations of the CKKS cryptosystem. The core idea, which greatly simplifies the solution, is to let all agents compute in a global cryptosystem "0" while none of them has access to the corresponding sk_0. To this end, the operator sets up this cryptosystem and distributes pk_0. Then, every agent can operate on a fully encrypted version of (3.8) and exchange data with its neighbors without the risk of disclosing information. A small caveat is that the operator can, in principle, eavesdrop on the communications between agents and decrypt them using sk_0. However, we resolve this by an additional layer of standard encryption for the communication. The final step is to give each agent access to its computation result and none of the other agents' results. To achieve this, we exploit the

KeySwitch functionality as follows. During setup, each agent i receives a secret key sk_i. Then, the i-th agent's result is first sent to a different agent, where a KeySwitch from cryptosystem 0 to i is executed. The resulting ciphertext is then forwarded to agent i, who can use sk_i for decryption. This intermediate step is necessary because $\mathsf{swk} = (b_{\mathsf{swk}}, a_{\mathsf{swk}}) = (-a_{\mathsf{swk}}\,\mathsf{sk}_i + e_{\mathsf{swk}} + P\,\mathsf{sk}_0 \bmod Pq_L, a_{\mathsf{swk}})$ can be decrypted with sk_i, which reveals sk_0.

We apply our scheme to a formation task of mobile robots under three different communication graphs (see [P3, Figure 7.1]). The operator specifies a desired displacement between the robots and one agent follows a time-varying reference position. We observed a quick convergence to the desired formation and satisfying tracking behavior after five iterations, as shown in [P3, Figure 7.2].

In [P11], we consider distributed affine averaging which is used for state estimation and is relevant for robot swarms, clock synchronization, processor networks, and data fusion. A compact presentation of the algorithm in its integer form is

$$\boldsymbol{z}(k+1) = \lfloor s\boldsymbol{H} \rceil\, \boldsymbol{z}(k) + \lfloor s^{k+2}\boldsymbol{g} \rceil \,(\mathrm{mod}\, q), \tag{3.9}$$

where we note the similarity to (3.1). However, a crucial difference is that $z_i(k+1)$ depends on $\sum_{j\in\mathcal{N}_i} \lfloor sH_{ij} \rceil z_j(k)$, where $\mathcal{N}_i$ are the neighbors of agent i. Thus, instead of protecting (3.9) entirely, our goal is to provide privacy for the neighbor's states $z_j(k)$ at each agent i. Unlike before, where finite approximations were sufficient, the required number of iterations in (3.9) typically results in impractical values for (q, s). Thus, another approach that enables sufficiently many iterations in (3.9) must be contemplated. In the context of affine averaging, it is important to note that $\boldsymbol{H}$ has additional structure, i.e., $\boldsymbol{H}\mathbf{1}_n = \mathbf{1}_n$, $\mathbf{1}_n^\top \boldsymbol{H} = \mathbf{1}_n^\top$, and $\mathbf{1}_n^\top \boldsymbol{z}(k) = 0$ are required for convergence. Thus, a suitable $\boldsymbol{H} \in \mathbb{Z}^{n\times n}$ must fulfill these restrictions and additionally[2] $\det(\boldsymbol{H} - \lambda \boldsymbol{I}) = \lambda^{n-1}(\lambda - 1)$ while being compatible with the communication graph. Because finding such $\boldsymbol{H}$ is NP-hard (see [186]), we analyze tailored reset strategies (similar to (3.6)) where, after T computation steps, a suitable value is assigned to $\boldsymbol{z}(k)$. An admissible pair (q, s) that guarantees a computation interval T without overflows can be estimated based on the accumulated quantization errors of (3.9) which are derived in [P11, Section 4.1].

Again, our approach favors HE. Setup-wise, one agent is chosen to obtain its state estimate at the end of the computation. Without restriction of generality, we assume that this is agent $i = 1$. We refer to it as the "leader", while the remaining agents are "followers". First, the leader sets up a homomorphic cryptosystem and distributes pk among the followers. Thus, the followers can evaluate T encrypted iterations of (3.9) without the risk of breaching their state's privacy despite data exchanges. An exception is the leader who, in possession of sk, can decrypt messages from its neighbors. However, the leader's ability to decrypt is also key to transferring information from $\boldsymbol{z}(T)$ to $\boldsymbol{z}(T+1)$

[2] One integrator and $n-1$ decaying modes are required for consensus. To this end, select $\Phi_p = \lambda - 1$ in Corollary 2.

during the reset. Yet, resetting $z(k)$ to a value that can be determined offline would lead to a limit cycle with period T after the first iteration phase. Thus, for further convergence, the use of online data is essential, while $\mathbf{1}_n^\top z(kT) = 0$ has to be taken into account. To this end, we use a tree sub-graph of the communication graph for the reset (illustrated in Figure 3.2), where the leader is the root node and follower agents are the leaf nodes. Without going into the specific details of affine averaging, the tree graph enables aggregating encrypted data at parent nodes by forwarding it from their child nodes. The result finally reaches the leader, who can decrypt and use the data to provide suitable updates for its own and the followers' states, which are then distributed. Errors related to resets are analyzed in [P11, Section 4]. The algorithm then alternates between T iterations and a reset.

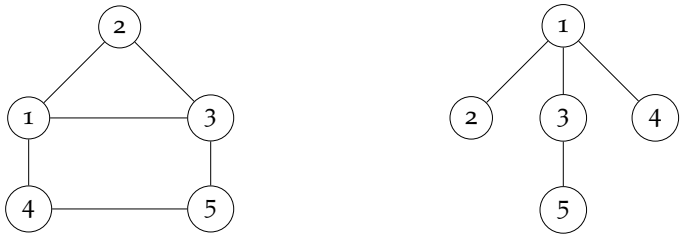

Figure 3.2. Communication graph (left) and tree subgraph with the leader as the root node (right).

We tested the approach on randomly generated graphs with $M \in \{10, 11, \ldots, 100\}$ agents (see [P11, Section 5]) and observed that the convergence continues after the resets until the fixed-point precision of the implementation is reached. For affine averaging, the information aggregation during the reset can even support the convergence.

3.1.3 Discussion

Cloud-based. As it turned out, FIR controllers (3.3) are the only suitable choice that adhere to the form (1.2) and can be evaluated with HE or SMPC without bootstrapping or truncation protocols, respectively, while harmonizing well with a networked setup due to their stability. Nonetheless, two problems arise in this case. First, their approximation-based design is only feasible for predesigned controllers with stable $\boldsymbol{H}$ and it can require high controller orders. Thus, a direct FIR controller design would be advantageous. Unfortunately, FIR controller design is equivalent to output feedback design, which is an old but (to some extent) open problem in control theory. Moreover, FIR controllers cannot stabilize every class of linear systems [P2].[3] Still, based on the idea of finite information reuse, other algorithms can also be realized. For

[3] A practical alternative, leveraging state feedback designs, is through state estimates $\hat{x}(k)$ via moving horizon estimators with quadratic cost, linear system dynamics, and *no initial guess* [166, cf. Section 1.4.4]. The resulting $\hat{x}(k)$ is linear in the input-output sequences.

instance, in [P1], we propose a cloud-based system identification service that solves least squares problems of the form

$$\boldsymbol{N}^\dagger \boldsymbol{\Omega} = \arg\min_{\boldsymbol{X}} \|\boldsymbol{N}\boldsymbol{X} - \boldsymbol{\Omega}\|_F^2 \quad \text{via} \quad \boldsymbol{Z}(k+1) = (2\boldsymbol{I} - \boldsymbol{Z}(k)\boldsymbol{N})\boldsymbol{Z}(k),$$

where the decision variable $\boldsymbol{X}$ and the parameter $\boldsymbol{\Omega}$ are matrices in general. The iteration in $\boldsymbol{Z}(k)$, which was conceived in [206], is guaranteed to converge to the pseudoinverse $\boldsymbol{N}^\dagger$ for suitable initial guesses $\boldsymbol{Z}(0)$.

We found reliable solutions in the context of specific control applications that satisfy our demands in terms of performance and robustness. However, these do not directly address the iteration problem, but rather circumvent it by building on finite approximations. Thus, further research that addresses the problem and unlocks larger computational depths is required. In the context of SMPC, we note that there exist less critical setups in which a periodic preprocessing phase can be justified.

Multi-agent systems. Our approaches in multi-agent systems are built on finite iterations in control algorithms. In both scenarios, distributed algorithms are used, for which we prefer an HE-based solution. This way, distributing a public key among the agents enables encrypted iterations until an overflow becomes imminent. A major drawback of HE when it comes to resource-constrained devices is the computational overhead. With lightweight security parameters and a small communication graph, we barely achieved real-time capabilities on a standard computer (see [P3, Section 5.3]). First, in the context of encrypted ADMM iterations, we manage the access of each agent to its computation result via keyswitches. In the presented form, the scheme is $(1, M)$, i.e., it can only provide security if the agents i and j, where agent j performs the keyswitches for agent i, do not collude. For a (M, M)-scheme, a solution based on additive secret sharing and a centralized algorithm seems to be the best choice. Also, the additional encryption layer during communication can be bypassed by SMPC. To this end, pk and the required switching keys are generated privately using SMPC and then revealed. This way, no agent has access to data that is encrypted with cryptosystem 0. Finally, a re-randomization as part of the keyswitch should be considered to increase the security [57].

In the context of affine averaging, we essentially leverage information aggregation to compute a suitable reset and decryption to prevent an overflow. While the setup permits the leader to learn its state, sk enables it to decrypt other messages as well. During the computation, this reveals the states of its neighbors, which constitutes an information leak. From this perspective, the leader should have a small number of neighbors. Using additive secret sharing, it would be possible to reconstruct computation results for the leader without granting access to its neighbors' states. Preventing an information leak entirely does, however, require bootstrapping or a rescaling protocol.

3.2 Non-polynomial controllers

3.2.1 Problem overview

Arithmetic HE schemes and SS provide the computation primitives Add and Mult. For a suitable modulus, this allows evaluating (a composition of) multivariate polynomials

$$g(x_1, x_2, \dots, x_n) = \sum_{i_1, i_2, \dots, i_n} g_{i_1, i_2, \dots, i_n} x_1^{i_1} x_2^{i_2} \dots x_n^{i_n}, \tag{3.10}$$

with $g_{i_1,i_2,\dots,i_n} \in \mathbb{R}$ and $i_j \geqslant 0$ on the inputs. However, even seemingly simple non-polynomial functions result in a non-trivial representation by (3.10). Moreover, it is well-known that certain functions are not well-representable by (3.10), where discontinuous functions stand out. These lead to large polynomial degrees (multiplicative depth) or low accuracy.

In control, non-polynomial functions arise prominently in MPC. More precisely, the explicit solution to the optimal control problem (1.4) is PWA, as highlighted in (1.5). Robust MPC, which allows taking quantization errors from (2.4a) into account, or data-driven MPC [P7] can also result in PWA solutions (see the illustrations in [P9, P10, P17]). While nonlinear MPC yields more complex control laws[4], PWA functions are a popular choice for their approximation [99, 115]. Moreover, the capability to evaluate certain non-polynomial functions unlocks wide-ranging computation capabilities, e.g., through elementary mathematical functions [78, 173].

3.2.2 Summaries of articles [P9, P10, P17]

Efficient and accurate non-polynomial functions are relevant beyond control, while the current state of affairs, as characterized by (3.10), does not satisfy the demands in many applications. In the context of SMPC, GCs can help to alleviate this problem by enriching the set of computations with Boolean circuits. However, a formulation that is well-aligned with the cryptographic primitives is paramount for practicality. We discuss our findings in the context of nonlinear control laws $\boldsymbol{u}(\boldsymbol{x}) \in \mathbb{R}^m$ that arise in MPC. Without loss of generality, we focus on $m = 1$ to simplify the presentation.

In [P9], we propose to use max-out neural networks of the form

$$\hat{u}(\boldsymbol{x}) = \hat{u}_1(\boldsymbol{x}) - \hat{u}_2(\boldsymbol{x}) = \max\{\boldsymbol{L}\boldsymbol{x} + \boldsymbol{d}\} - \max\{\boldsymbol{M}\boldsymbol{x} + \boldsymbol{e}\} \tag{3.11}$$

to realize cloud-based explicit MPC with privacy. Here, $\boldsymbol{L}, \boldsymbol{M} \in \mathbb{R}^{p \times n}$ and $\boldsymbol{d}, \boldsymbol{e} \in \mathbb{R}^p$ correspond to a single hidden layer with two neurons that pool $p \in \mathbb{N}$ affine preactivations from which the maximum element is selected.

[4] Control laws in nonlinear MPC can be discontinuous and may have bifurcations [26, Maximum theorem]. In comparison to (1.5), piecewise functions may be nonlinear.

Due to the convexity of the expressions $\max\{\boldsymbol{L}\boldsymbol{x}+\boldsymbol{d}\}$ and $\max\{\boldsymbol{M}\boldsymbol{x}+\boldsymbol{e}\}$, (3.11) is also called convex decomposition. An illustration is depicted in Figure 3.3. Importantly, (3.11) is a PWA function and allows to approximate any contin-

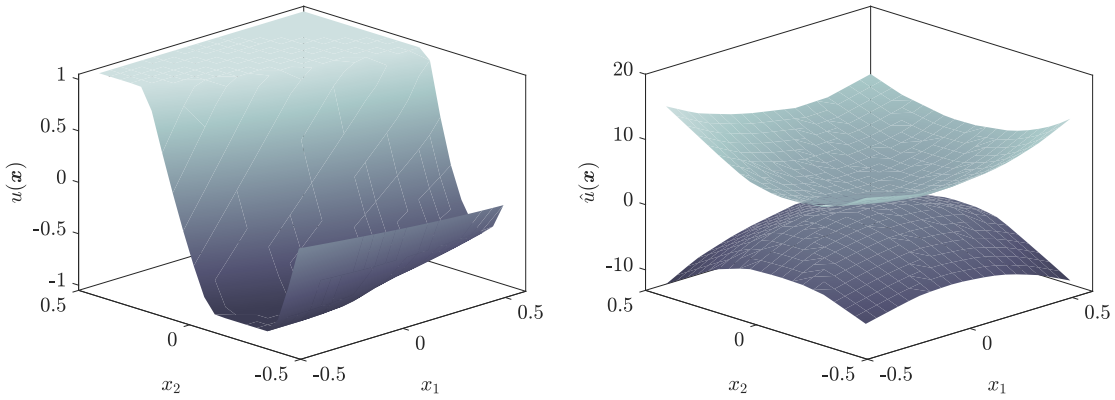

Figure 3.3. Left: Explicit solution $u(\boldsymbol{x})$ to a nonlinear MPC problem with continuously stirred reactor dynamics (normalized and continuous version of [100]). The solution $u(\boldsymbol{x})$ is computed at discrete sample points and interpolated. Right: Exact convex decomposition $\hat{u}(\boldsymbol{x})$ of the interpolated solution with $\max\{\boldsymbol{L}\boldsymbol{x}+\boldsymbol{d}\}$ (top; light) and $\max\{\boldsymbol{M}\boldsymbol{x}+\boldsymbol{e}\}$ (bottom; dark).

uous function $u(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ arbitrarily well [97, Theorem 4.3]. As it will turn out, (3.11) is suitable for a private evaluation. To this end, the encoding of (3.11) results in $\hat{v}(\boldsymbol{z}) = \max\{\lfloor s_1\boldsymbol{L}\rceil\boldsymbol{z} + \lfloor s_3\boldsymbol{d}\rceil\} - \max\{\lfloor s_1\boldsymbol{M}\rceil\boldsymbol{z} + \lfloor s_3\boldsymbol{e}\rceil\} \pmod q$, where $\boldsymbol{z} = \lfloor s_2\boldsymbol{x}\rceil \pmod q$. Here, the potentially different scaling factors s_1, s_2, and $s_3 = s_1 s_2$ enable some design freedom regarding accuracy.[5] Remarkably, despite the quantization errors, $\max\{\}$ always selects the correct element, as implied by the following upper bound.

Proposition 1 ([P9, Proposition 3])**.** *Select q such that an overflow is avoided, and let $\eta \in \mathbb{R}$ be such that $\|\boldsymbol{x}\|_\infty \leqslant \eta/(2s_1)$, $\|\boldsymbol{L}\|_{\max} \leqslant \eta/(2s_2)$, and $\|\boldsymbol{M}\|_{\max} \leqslant \eta/(2s_2)$. Then, the error is upper-bounded via*

$$|u(\boldsymbol{x}) - \mathsf{Dcd}_{s_3}(\hat{v}(\boldsymbol{z}))| \leqslant \frac{1}{s_3}\left(n\eta + \frac{n}{2} + 1\right).$$

With the error bound at hand, a robust MPC formulation is used to account for quantization errors and to ensure robust constraint satisfaction.

In [P10], we consider the computation of a convex decomposition. In particular, we determine $\boldsymbol{L}, \boldsymbol{M}, \boldsymbol{d}, \boldsymbol{e}$ for a given PWA function defined over a partition $\{\mathcal{P}_i\}_{i=1}^{\theta}$ as in (1.5). In this context, two methods exist that result in $u(\boldsymbol{x}) = \hat{u}(\boldsymbol{x})$. The first method uses index pairs of neighboring polyhedra $\mathcal{B} = \left\{(i,j) \in \{1,\ldots,\theta\}^2 \,\middle|\, \dim(\mathcal{P}_i \cap \mathcal{P}_j) = n-1,\ i<j\right\}$ and index pairs associated with "convex folds" in $u(\boldsymbol{x})$, i.e.,

$$\mathcal{V} = \left\{(i,j) \in \mathcal{B} \,\middle|\, \boldsymbol{k}_i^\top\boldsymbol{x} + b_i > \boldsymbol{k}_j^\top\boldsymbol{x} + b_j,\ \forall \boldsymbol{x} \in \mathcal{P}_i \backslash \mathcal{P}_j\right\}.$$

[5] An optimal quantization is addressed in Appendix C.1.

Here, $\boldsymbol{k}_i^\top$ and b_i denote the m-th row of $\boldsymbol{K}_i$ and $\boldsymbol{b}_i$ in (1.5), respectively. Based on $\mathcal{V}$, one can construct $\hat{u}_1(\boldsymbol{x}) = \sum_{(i,j)\in\mathcal{V}} \max\{\boldsymbol{k}_i^\top \boldsymbol{x} + b_i, \boldsymbol{k}_j^\top \boldsymbol{x} + b_j\}$ and $\hat{u}_2(\boldsymbol{x}) = \hat{u}(\boldsymbol{x}) - \hat{u}_1(\boldsymbol{x})$ which can be brought into the form (3.11). The second method uses an optimization-based approach which offers more flexibility. To this end, convexity and continuity are enforced for all $(i,j) \in \mathcal{B}$ through

$$\boldsymbol{l}_i^\top \boldsymbol{x} + d_i \geqslant \boldsymbol{l}_j^\top \boldsymbol{x} + d_j, \quad \boldsymbol{m}_i^\top \boldsymbol{x} + e_i \geqslant \boldsymbol{m}_j^\top \boldsymbol{x} + e_j \quad \text{for every } \boldsymbol{x} \in \mathcal{P}_i \tag{3.12}$$

$$\boldsymbol{l}_j^\top \boldsymbol{x} + d_j \geqslant \boldsymbol{l}_i^\top \boldsymbol{x} + d_i, \quad \boldsymbol{m}_j^\top \boldsymbol{x} + e_j \geqslant \boldsymbol{m}_i^\top \boldsymbol{x} + e_i \quad \text{for every } \boldsymbol{x} \in \mathcal{P}_j, \tag{3.13}$$

while $\boldsymbol{k}_i = \boldsymbol{l}_i - \boldsymbol{m}_i$, $b_i = d_i - e_i$ must hold for all $i \in \{1, \ldots, \theta\}$ to satisfy (3.11). While the first method results in $\hat{u}_2(\boldsymbol{x})$ with significantly more segments in comparison to $\hat{u}_1(\boldsymbol{x})$, the second method suffers from feasibility issues. In fact, the optimization is only feasible if $\{\mathcal{P}_i\}_{i=1}^\theta$ has a regularity property (see [67, p.53]) which is often not fulfilled in practice. However, we found that the feasibility of an optimization-based approach is ensured by using the induced partition of

$$\sum_{(i,j)\in\mathcal{V}} \max\{\boldsymbol{k}_i^\top \boldsymbol{x} + b_i, \boldsymbol{k}_j^\top \boldsymbol{x} + b_j\} \quad \text{and} \quad \sum_{(i,j)\in\mathcal{Y}} \min\{\boldsymbol{k}_i^\top \boldsymbol{x} + b_i, \boldsymbol{k}_j^\top \boldsymbol{x} + b_j\},$$

where $\mathcal{Y} = \{(i,j) \in \mathcal{B} \,|\, \boldsymbol{k}_i^\top \boldsymbol{x} + d_i < \boldsymbol{k}_j^\top \boldsymbol{x} + d_j, \boldsymbol{x} \in \mathcal{P}_i \backslash \mathcal{P}_j\}$ is the set of concave folds. In fact, the resulting partition is regular by definition and often contains fewer polyhedra than the initial partition $\{\mathcal{P}_i\}_{i=1}^\theta$ (see [P10, Table 10.1]).

In [P17], we introduce a two-party protocol for the privacy-preserving evaluation of (3.11) by combining and extending the ideas in [P9, P10]. First, we note that, for efficiency reasons, a convex decomposition with a minimal number of segments is desirable. Fortunately, $u(\boldsymbol{x}) \approx \hat{u}(\boldsymbol{x})$ allows reducing the complexity significantly. To this end, we consider a least-squares training approach for (3.11) using samples $(\boldsymbol{x}_i, u_i(\boldsymbol{x}_i))$.[6] Second, observe that $\boldsymbol{\xi} = \lfloor s_1 \boldsymbol{L} \rceil \boldsymbol{z} + \lfloor s_3 \boldsymbol{d} \rceil \,(\mathrm{mod}\, q)$ and $\boldsymbol{\omega} = \lfloor s_1 \boldsymbol{M} \rceil \boldsymbol{z} + \lfloor s_3 \boldsymbol{e} \rceil \,(\mathrm{mod}\, q)$ can be evaluated with HE or SS primitives. Challenging are, however, the non-polynomial functions $\max\{\boldsymbol{\xi}\}$ and $\max\{\boldsymbol{\omega}\}$. Therefore, we enrich our protocol by means of GCs that are well-suited for Boolean circuits and hence to select the maxima. Consequently, we obtain a $M = 2$ party scheme and select SS rather than HE. At this point, it becomes clear that our reformulation of $u(\boldsymbol{x})$ via $\hat{u}(\boldsymbol{x})$ aligns well with the strengths of SS and GCs, while also acknowledging the non-reusability of GCs.

Due to symmetry in our privacy-preserving computation, we focus in the following overview only on $\max\{\boldsymbol{\xi}\}$. Namely, evaluating $\boldsymbol{\xi}$ using SS, we obtain

$$[\xi_i] = \sum_{j=1}^{n} [\lfloor s_1 L_{ij} \rceil] \times [z_j] + [\lfloor s_3 d_i \rceil] \,(\mathrm{mod}\, q).$$

[6] Instead of directly solving the nonlinear optimization, a convex reformulation can be found in Appendix C.2.

Since $\mathsf{Reconstruct}([\xi_i])$ would reveal private information, while the correct result of $\max\{\boldsymbol{\xi}\}$ depends on the message $\boldsymbol{\xi}$, the reconstruction must be performed within the GC. To avoid a bottleneck, we use $q = 2^\varphi$ for a suitable $\varphi \in \mathbb{N}$ such that the required modulo reduction amounts to discarding bits. Then, the first part of the Boolean circuit is a ripple carry adder which computes

$$\mathsf{Reconstruct}([\xi_i]) = -2^{\varphi-1}Y_{i1} + \sum_{j=2}^{\varphi} 2^{\varphi-j}Y_{ij},$$

where $\boldsymbol{Y} \in \{0,1\}^{p\times\varphi}$ contains a bit-decomposition of ξ_i in its i-th row. The second part then consists of $\max\{\boldsymbol{\xi}\}$ which is evaluated in $\lceil \log_2 p \rceil$ "tournaments", e.g., for $p = 4$ via $\max\{\max\{\xi_1,\xi_2\},\max\{\xi_3,\xi_4\}\}$. Here, $\max\{\xi_i,\xi_j\} = (1-\sigma)\xi_i + \sigma\xi_j$, where σ is the sign bit of $\xi_i - \xi_j$. The k-th bit of $\max\{\xi_i,\xi_j\}$ is then

$$\mathsf{XOR}\left(\mathsf{AND}\left(Y_{ik},\mathsf{NOT}(\sigma)\right),\mathsf{AND}\left(Y_{jk},\sigma\right)\right). \tag{3.14}$$

Repeating these steps for the remaining entries in $\boldsymbol{\xi}$ and the remaining rounds of the tournament completes the circuit-based evaluation of $\max\{\boldsymbol{\xi}\}$. Based on this, two GCs can be constructed where the output labels are bits of $[\max\{\boldsymbol{\xi}\}]$ and $[\max\{\boldsymbol{\omega}\}]$. The final subtraction is evaluated using SS. Then, $u(\boldsymbol{x}) \approx \mathsf{Dcd}_{s_3}(\hat{v}(\boldsymbol{x})) = \mathsf{Dcd}_{s_3}(\max\{\boldsymbol{\xi}\} - \max\{\boldsymbol{\omega}\})$ can be applied. Protocol 4 summarizes the approach.

Protocol 4: Evaluate $u(\boldsymbol{x}) \approx \max\{\boldsymbol{L}\boldsymbol{x}+\boldsymbol{d}\} - \max\{\boldsymbol{M}\boldsymbol{x}+\boldsymbol{e}\}$

Preprocessing phase

1: Distribute $[\lfloor s_1 L_{ij}\rceil]$, $[\lfloor s_3 d_i\rceil]$, $[\lfloor s_1 M_{ij}\rceil]$, $[\lfloor s_3 e_i\rceil]$.
2: Generate and distribute sufficiently many multiplication triples $[\alpha]$, $[\beta]$, $[\gamma]$.
3: Generate a garbled circuit for $[\max\{\boldsymbol{\xi}\}]$ and $[\max\{\boldsymbol{\omega}\}]$.

Execution phase

1: Distribute $[z_1],\ldots,[z_n]$
2: $[\xi_i] = \sum_{j=1}^n [\lfloor s_1 L_{ij}\rceil] \times [z_j] + [\lfloor s_3 d_i\rceil] \pmod q$
$[\omega_i] = \sum_{j=1}^n [\lfloor s_1 M_{ij}\rceil] \times [z_j] + [\lfloor s_3 e_i\rceil] \pmod q$
3: Exchange the garbled circuits and labels. Perform OT_2^1 for each $\ell_{\xi_i^{(2)}}^{\xi_i^{(2)}}$ and $\ell_{\omega_i^{(1)}}^{\omega_i^{(1)}}$.
4: Evaluate the garbled circuits.
5: $[\hat{v}(\boldsymbol{x})] = [\max\{\boldsymbol{\xi}\}] - [\max\{\boldsymbol{\omega}\}] \pmod q$.

The security of Protocol 4 follows from the modular composition property of SMPC protocols (see Section 2.3.2). Since OT_2^1 depends on $[\xi_i],[\omega_i]$, it requires two communication rounds (excluding the preprocessing). Note that we maximized the number of XOR in the Boolean circuit to exploit the Free-XOR optimization as noted in Section 2.3.5. In total, the remaining evaluation

cost amounts to $\varphi(4p-3)$ AND gates. Besides, we tested the scheme numerically, where $p \in \{8, 16\}$ segments and $\varphi \in \{16, 32\}$ bits accuracy led to execution times in $[80, 350]$ ms (see [P17, Table 11.3] for details).

3.2.3 Discussion

By enriching SS with GCs, we enabled the privacy-preserving evaluation of PWA functions that can be used to approximate continuous functions arbitrarily well. At this point, we note that GCs can also offer efficient rescaling, as needed for iterations, such that an alternation between SS and a GC can enable a large class of algorithms. However, there are two disadvantages related to Protocol 4. First, its security relies on a non-collusion assumption between two parties, i.e., it is $(1, 2)$ secure. While this can be realistic in certain scenarios (computation services with multiple clouds, for instance), a higher threshold is desirable. In principle, replacing GCs with its multi-party variant BMR or using a bit-decomposition protocol in SS to evaluate $\max\{\boldsymbol{\xi}\}$ and $\max\{\boldsymbol{\omega}\}$ at the cost of more preprocessing and communication rounds can achieve this. Second, for certain functions, the least-squares training and p can become computationally costly and large, respectively. If this occurs, an enlarged memory footprint and less efficiency due to the need for more multiplication triples and larger GCs are the results.

Interestingly, when considering special functions instead of generic approximators as above, there exist tailored protocols in SS such as for $[\lfloor z/s \rfloor]$ and $[z_1/z_2]$ (see, for instance, [13, 43]). With this at hand, we consider privacy-preserving computations between prosumers in [P5] for smart grids, as shown in Figure 3.4. In particular, we solve the well-known nonlinear power flow

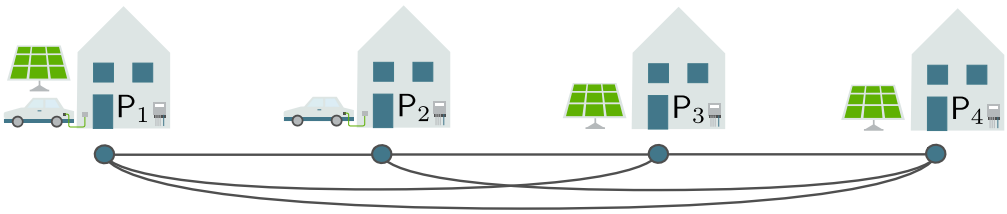

Figure 3.4. Secret sharing in a smart grid with $M = 4$ prosumers P. Each prosumer is connected via a point-to-point connection (gray lines) to all other prosumers.

problem based on Newton's method. Our formulation is specifically designed to leverage the strengths of SMPC to minimize communications. The resulting algorithm is then benchmarked under different grid sizes and network conditions.

3.3 Security of random affine transformations

3.3.1 Problem overview

While well-established cryptographic methods such as HE and SS suffer from large overheads, the promise of RATs is to alleviate this problem and to facilitate new applications. To this end, the symmetric and deterministic RAT cipher is defined as follows:

$\underline{\mathsf{KeyGen}()}$. Output $\mathbf{sk} = (\boldsymbol{R}, \boldsymbol{r})$, where $\boldsymbol{R} \in \mathbb{R}^{m \times m}$ and $\boldsymbol{r} \in \mathbb{R}^m$ are selected (3.15a)

at random, while $\boldsymbol{R}^{-1}$ must exist.

$\underline{\mathsf{Enc}_{\mathbf{sk}}(z)}$. Output $\mathbf{ct}(z) = \boldsymbol{R}z + \boldsymbol{r}$, with $z \in \mathbb{R}^m$. (3.15b)

$\underline{\mathsf{Dec}_{\mathbf{sk}}(\mathbf{ct}(z))}$. Output $z = \boldsymbol{R}^{-1}(\mathbf{ct}(z) - \boldsymbol{r})$. (3.15c)

Remarkably, since RATs utilize a real-number message and ciphertext space $\mathbb{R}^m = \mathcal{Z}^m = \mathcal{C}^m$, an encoding is not required for most applications. Furthermore, computations over $\mathbb{R}$ allow the use of standard methods to handle both iterations and non-polynomial functions.

A prominent application of RATs is the solution of a quadratic program

$$\min_z \frac{1}{2} z^\top \boldsymbol{Q} z + \boldsymbol{c}^\top z \quad \text{s.t.} \quad \boldsymbol{P}z \leqslant \boldsymbol{s}, \tag{3.16}$$

which can be transformed into an equivalent quadratic program in $\mathbf{ct}(z)$ with the randomized parameters

$$\tilde{\boldsymbol{Q}} = \boldsymbol{R}^{-\top} \boldsymbol{Q} \boldsymbol{R}^{-1},\ \tilde{\boldsymbol{P}} = \boldsymbol{P}\boldsymbol{R}^{-1},\ \tilde{\boldsymbol{c}}^\top = (\boldsymbol{c}^\top - \boldsymbol{r}^\top \boldsymbol{R}^{-\top} \boldsymbol{Q})\boldsymbol{R}^{-1},\ \tilde{\boldsymbol{s}} = \boldsymbol{s} + \boldsymbol{P}\boldsymbol{R}^{-1}\boldsymbol{r} \tag{3.17}$$

by substituting $z = \boldsymbol{R}^{-1}(\mathbf{ct}(z) - \boldsymbol{r})$ in (3.16). The transformed variant is then outsourced to a semi-honest cloud and solved. Subsequently, the returned minimizer is decrypted. In control, this can be used for implicit linear MPC, where (3.16) is solved at each time step to obtain a solution for (1.4).

Although computationally cheap, RATs lack a security analysis under established definitions. Important aspects in this context are as follows. First, while RATs were initially proposed for one-time usage, control applications require periodic usage where data is highly correlated. Second, in some literature, non-standard security definitions are used to prove security only under the weakest attacker model, i.e., a ciphertext-only attacker. Third, KeyGen is only vaguely specified (as above). Therefore, the use of RATs in practice firmly asks for a systematic security analysis.

3.3.2 Summary of article [P13]

In [P13], we address this issue by analyzing the security of RATs under cryptographic notions, which we introduced in Sections 2.1.4 and 2.2.2. More precisely, we compute the statistical distance

$$D = \frac{1}{2} \sum_{\mathbf{ct} \in \mathcal{C}^m} |\Pr(\mathsf{Enc}_{\mathbf{sk}}(z_1) = \mathbf{ct}) - \Pr(\mathsf{Enc}_{\mathbf{sk}}(z_2) = \mathbf{ct})| \quad \forall z_1, z_2$$

as a quantitative measure for the ciphertext security of different RAT variants.[7] Here, $\Pr(\mathsf{Enc}_{\mathbf{sk}}(z_i) = \mathbf{ct})$ denotes the probability of observing $\mathbf{ct}$ given z_i. Note that for negligible D, a security proof is possible, e.g., by simulating the view of an adversary $\mathcal{A}$. Our analysis focuses on the encryption (3.15b) and keeps potentially valuable additional data aside. Furthermore, our setup considers attackers $\mathcal{A}$ with plaintext and plant knowledge. Applying Kerckhoffs' principle to CPSs, the latter assumes $\mathcal{A}$ knows that the RAT is used in the context of linear discrete-time dynamics (1.3). Such background knowledge is important to consider because it cannot be controlled and thus should not influence the cipher's security.

The deterministic RAT cipher is specified in (3.15). In case only ciphertexts $\{\mathbf{ct}(z(k))\}_{k\in\mathbb{N}}$ are available, guesses for $z(k)$ or $\mathbf{sk}$ are not verifiable. This often serves as a security argument in the encrypted control literature. However, $\mathsf{Enc}_{\mathbf{sk}}(z(k))$ always results in the same $\mathbf{ct}(z(k))$ for the same $z(k)$, allowing us to conclude $D = 1$. In practice, this means that a public key RAT variant would be easily broken, i.e., $\mathcal{A}$ could build a library of pairs $(z(k), \mathbf{ct}(z(k)))$ using pk and perform a known-plaintext attack. In a known-plaintext attack, $\mathcal{A}$ has access to the pairs $\{(z(k), \mathbf{ct}(z(k)))\}_{k\in\mathbb{N}}$. In this case, the deterministic RAT cipher breaks because $\mathbf{ct}(z(k)) = \boldsymbol{R}z(k) + \boldsymbol{r}$ becomes linear in $\mathbf{sk}$. A *known-plant attack* grants $\mathcal{A}$ the structural knowledge of the linear system (1.3) and $\{(\mathbf{ct}(\boldsymbol{u}(k)), \mathbf{ct}(z(k)))\}_{k\in\mathbb{N}}$, where $\boldsymbol{u}(k)$ is the control input. This allows $\mathcal{A}$ to identify $\boldsymbol{R}\boldsymbol{A}\boldsymbol{R}^{-1}$, i.e., a similarity transformation of the system matrix $\boldsymbol{A}$, which reveals the eigenvalues of $\boldsymbol{A}$. Furthermore, if $\mathcal{A}$ obtains $\boldsymbol{A}$, it can recover $\mathbf{sk}$.

Next, we define a probabilistic RAT, where $\mathbf{sk}(k)$ varies in each encryption. This requires re-computing and communicating the parameters (3.17), which constitutes significant effort. Nevertheless, we consider the cases $(\boldsymbol{R}, \boldsymbol{r}(k))$ and $(\boldsymbol{R}(k), \boldsymbol{r}(k))$ and assume an independent and uniform distribution for the elements of $\boldsymbol{R}(k)$ and $\boldsymbol{r}(k)$ over the intervals $[-R_{\max}, R_{\max}]$ and $[-r_{\max}, r_{\max}]$ with finite $R_{\max}, r_{\max} > 0$, respectively. Then, the resulting distribution of $\mathbf{ct}(z(k)) = \boldsymbol{R}z(k) + \boldsymbol{r}(k)$ allows for negligible D if $r_{\max} = 2^{\kappa}$ and $R_{\max}$ is fixed. Similarly, $(\boldsymbol{R}(k), \boldsymbol{r}(k))$ results in $D \leqslant \mathsf{negl}(\kappa)$ for suitable parameter choices. Thus, attacking probabilistic RATs based on their ciphertext distributions (illustrated in Figure 3.5) can be made intractable.

However, it is often overlooked that defining a RAT over $\mathbb{R}$ requires infinite precision, which is not realizable on a digital machine. Thus, KeyGen in (3.15a) must be specified while taking this constraint into account. In this context, we propose to emulate a uniform distribution with floating-point numbers in order to maximize the cipher's security. Due to rounding that occurs during floating-point computations, analytical expressions are hard to obtain for this case. Therefore, we analyze the statistical distance D numerically based on the (still tractable) set of 16 bit floating-point numbers. It turns out that D strongly depends on the messages and varies in a complex manner between 0 and 1,

[7] In fact, we consider a continuous version of the statistical distance in most of the analyzed cases.

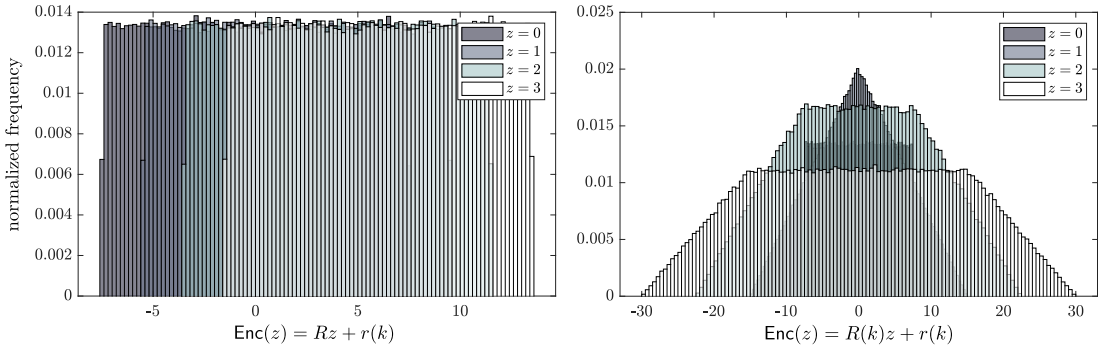

Figure 3.5. Ciphertext distribution of RATs in the scalar case for different messages (left: $R = 2$, $r_{\max} = 7.5$, right: $R_{\max} = r_{\max} = 7.5$). Overlap of the distributions creates security and can be increased by making $r_{\max}$ larger.

which is a result of finite precision. This indicates vulnerabilities and makes a reliable application difficult. Finally, Table 3.1 provides an overview of the results stated above.

Table 3.1. Security guarantees of the RAT ciphers under different scenarios

	ciphertext-only	known-plaintext	known-plant
deterministic variant	$D = 1$	broken	broken
probabilistic variant	$D \leqslant \mathsf{negl}(\kappa)$	secure	secure
floating-point variant	$D \in [0, 1]$	unreliable	unreliable

3.3.3 Discussion

Our findings regarding the security of RATs suggest that their deterministic variant should never be used while probabilistic variants provide security guarantees. Digital implementations of RATs are not reliable without addressing the highlighted issues. Thus, RATs are not recommendable from our point of view. Moreover, in the analysis above, we only considered the encryption of vectors, e.g., $\boldsymbol{z}$. In the context of MPC, it is, however, required to disclose the transformed parameters (3.17) as well. Thus, $\mathcal{A}$ obtains additional data encrypted under the same key. Against this background, we analyze the probabilistic cipher with time-varying $(\boldsymbol{R}(k), \boldsymbol{r}(k))$ in [P4] for the real-number case. It turns out that the original quadratic program (3.16) and its transformed variant (3.17) share the same dual problem, which yields the invariants

$$\boldsymbol{P}\boldsymbol{Q}^{-1}\boldsymbol{P}^{\top} = \widetilde{\boldsymbol{P}}(k)\widetilde{\boldsymbol{Q}}(k)^{-1}\widetilde{\boldsymbol{P}}(k)^{\top} \quad \text{and} \quad \boldsymbol{P}\boldsymbol{Q}^{-1}\boldsymbol{c} + \boldsymbol{s} = \widetilde{\boldsymbol{P}}(k)\widetilde{\boldsymbol{Q}}(k)^{-1}\tilde{\boldsymbol{c}}(k) + \tilde{\boldsymbol{s}}(k).$$

With these at hand, the uncertainty regarding the parameters and keys can be reduced. Then, additional side-knowledge suffices to break the cipher. Another problem of RATs is their flexibility. Although techniques based on system

immersion have been proposed that widen their capabilities [111], they introduce additional exploitable structure. In [P6], this approach and several MPC formulations are attacked. In many scenarios, parameters are recovered up to a similarity transformation, which constitutes the remaining ambiguity.

Chapter 4

Conclusions and Outlook

4.1 Conclusions

In control, safety is classically ensured by concepts such as stability or robustness. Considering the rising deployment of interconnected control systems, addressing cybersecurity threats through data confidentiality and integrity becomes crucial. To this end, we provide a foundation for encrypted control in Chapter 2 (and the Appendices A-B), mitigating the steep learning curve that advanced cryptography presents to beginners entering this field. Chapter 3 focuses first on key challenges of privacy-preserving computations in control: iterations, non-polynomial functions, and security. The former two are pivotal as they are found in a broad spectrum of privacy-preserving control algorithms. Notably, many optimization algorithms, which are vital for decision-making, identification, and control design, involve a simple iteration and a subsequent non-polynomial function. The latter is of utmost importance when it comes to novel constructions such as RATs. Based on the publications underlying this thesis, we significantly contributed to and pushed the current state of the art in encrypted control in the following ways.

First, building on integers is a computationally very efficient approach to unlimited privacy-preserving iterations. In Section 3.1, we take a system-theoretic viewpoint on encrypted dynamic controllers. It turns out that most linear iterations with integer matrices are unstable. Hence, such controllers are fragile in networked setups. An exception are finite impulse response and marginally stable controllers, for which homomorphic cryptosystems are well-suited. An alternative is a reset strategy which provides a controller operation without overflows at the cost of accuracy. While these solutions are practical for various control applications, such as dynamic networked control and affine averaging, they unfortunately do not resolve the iteration problem in general. Nonetheless, several algorithms (distributed optimization or matrix inverses) can be satisfactorily realized even with a finite iteration amount.

Second, in Section 3.2, we present our pioneering approach towards non-polynomial functions. In particular, we use a max-out neural network, which provides a piecewise affine approximation of non-polynomial functions. Remarkably, these networks align perfectly with two-party computation methods and allow the exploitation of their strengths while avoiding their weaknesses.

The effectiveness of our scheme is demonstrated based on model predictive control laws. Still, a weakness is that the security relies on a non-collusion assumption between the parties. Another approach to complex computations is through tailoring algorithms and protocols, which can be used, e.g., to solve nonlinear equation systems.

Finally, Section 3.3 analyzes RATs and provides novel insights into their security. While RATs are a tempting solution to increase the privacy of quadratic programs, their current form is not recommendable due to security issues that arise during a digital implementation. The fact that several attacks can be constructed based on additional encrypted data, e.g., transformed parameters in the context of quadratic programs, further substantiates our perspective. At this point, it seems that more reliability of RATs is unfortunately closely linked to the loss of their benefits.

Although substantial steps towards more practical encrypted control solutions have been made in this thesis, embarking on the path to novel constructions and improving the efficiency of existing ones is still paramount. Overcoming current limitations is essential for advancing encrypted control from a theoretical concept to a practical cybersecurity solution such that control becomes a gateway for technical improvement instead of novel threats. To this end, we will subsequently identify future research directions for the next chapter in encrypted control.

4.2 The next chapter

While in some fields, such as health, security can be an enabling technology for certain applications [139, 153], it will always entail additional costs. Thus, the practical usage of encrypted control is closely related to its usefulness in terms of capabilities and efficiency.

More complex decisions. To a large extent, solutions in encrypted control are targeting very specific controllers or applications. Although this can be fruitful, contributions often have limited impact. For faster progress, it seems promising to address computational primitives where the efficiency of iterations and non-polynomial functions (or lookup tables [52, 116]) have a major influence on the overall performance due to their frequent use. Furthermore, they are a stepping stone to state-of-the-art elementary functions [78, 173] and thus to numerical analysis [189] and advanced control algorithms. Among other specialties, control applications have a nuanced set of threat models, e.g., an asymmetric trust model with honest majority, that can allow for optimizations in SMPC. Also, the accuracy of HE [128] comes to mind, which can be reduced for higher efficiency utilizing the robustness of controllers.

Recently, non-leveled HE schemes began to support basic elementary functions and provide a practical bootstrapping routine [56]. However, only small

integers (less than 8 bit) are currently supported, which limits their applicability (current research [27, Sects. 2.3 and 4] may change that). In this context, it can become interesting to homomorphically switch between cryptosystems to combine their strengths [36] or to build on specialized hardware [222].

Towards integrity. So far, we have dealt with privacy protection by means of HE and SMPC. However, ciphertexts and shares are malleable by design, allowing a malicious attacker to alter the messages in a logical way. This is critical for outsourcing computations, but can also be exploited for attacks [10]. Therefore, additional means are needed to ensure integrity. A useful requirement is that the computational cost of integrity should be strictly less than the computational cost of the algorithm it is used for. For example, an optimization may be computationally costly and thus outsourced. Then, with the help of the corresponding KKT conditions, one can verify the correctness of the solution with little effort.

In the context of cryptography, SMPC provides several useful tools for security against malicious parties. Note that correctness in the presence of maliciously acting parties requires integrity. The simplest case is perhaps (t, M) secret sharing, which allows detecting a minority of erroneous shares, as they do not belong to the correct polynomial. More generally, the early work [94] showed how zero-knowledge proofs (ZKPs) can be used to compile a protocol that is secure against semi-honest adversaries into a protocol that is secure against malicious adversaries. ZKPs allow a prover to convince a verifier that a statement is true without revealing additional information [228]. However, the compiler typically does not result in a protocol with practical efficiency. In contrast to this, the works [25, 65] use additive secret sharing and extend the shares by additive message authentication codes. Computations are then performed on the shares of the secret and the message authentication codes. By checking the message authentication codes, a malicious adversary is bound to the correct computations or will be exposed. Based on [65], practicality is possible in some scenarios, as we show in [P5].

Similarly to HE, there exist homomorphic authentication techniques, where homomorphic signatures are attached to the data [89, 98]. A valid signature corresponds to the correct execution of the computation and can be verified. In comparison to SMPC, these methods are far less mature and suffer from large computational and memory overhead. With a restriction to linear operations, an authentication fast enough for a linear controller can be achieved [54]. Verifiable computations are often more efficient. In contrast to before, the correctness of computations is now proven using a ZKP. The main challenge is to precisely define the circuit that characterizes the ZKP. A tailored method via group-based cryptography for linear systems can be found in [53]. Moreover, non-interactive ZKPs, such as [184], stand out due to their computational and communication efficiency. In the context of RLWE, there exists a recent variant [88] which is tailored for computations over polynomial rings.

Following these research directions addresses core limitations as well as additional threats, thereby opening up new applications and enhancing the robustness of encrypted control. Ultimately, this will lead to broader adoption across various industries and drive innovations in areas previously constrained by security concerns.

Part II

Articles

The print version of this thesis, contains our articles [P3, P8, P9, P10, P11, P13, P16, P17] which are summarized in Chapter 3 and listed in the Bibliography at the end of this document.[1] Their layout has been revised, but the notation is left as is and can differ from the notation introduced in Part I.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles. Interested readers are encouraged to consult the original articles or the print version.

[1] Note that the articles [P12, P14] and [P1, P2, P4, P5, P6, P7, P15] served as a basis for Chapter 2 and are part of the discussions in Chapter 3, respectively. Thus, they are excluded here.

Article I

On the stability of linear dynamic controllers with integer coefficients

Nils Schlüter and Moritz Schulze Darup

IEEE Transactions on Automatic Control 67.10 (2021), pp. 5610–5613.
Available at: https://doi.org/10.1109/TAC.2021.3131126

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Article II

Encrypted dynamic control with unlimited operating time via FIR filters

Nils Schlüter, Matthias Neuhaus, and Moritz Schulze Darup

Proceedings of the 19th European Control Conference. IEEE. 2021, pp. 947–952. Available at: https://doi.org/10.23919/ECC54610.2021.9655161

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Article III

Towards privacy-preserving cooperative control via encrypted distributed optimization

Philipp Binfet[†], Janis Adamek[†], Nils Schlüter[†], and Moritz Schulze Darup

at - Automatisierungstechnik 71.9 (2023), pp. 736–747. Available at: https://doi.org/10.1515/auto-2023-0082

[†]The first three authors contributed equally to this work.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Article IV

Encrypted distributed state estimation via affine averaging

Nils Schlüter, Philipp Binfet, Junsoo Kim, and Moritz Schulze Darup

Proceedings of the 61st Conference on Decision and Control. IEEE. 2022, pp. 7754–7761. Available at: https://doi.org/10.1109/CDC51059.2022.9992840

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Article V

Encrypted explicit MPC based on two-party computation and convex controller decomposition

Nils Schlüter and Moritz Schulze Darup

Proceedings of the 59th Conference on Decision and Control. IEEE. 2020, pp. 5469–5476. Available at: https://doi.org/10.1109/CDC42340.2020.9304078

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Article VI

Novel convex decomposition of piecewise affine functions

Nils Schlüter and Moritz Schulze Darup

Late breaking results to the 21st IFAC World Congress. 2020. Available at:
https://doi.org/10.48550/arXiv.2108.03950

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Article VII

Secure learning-based MPC via garbled circuit

Katrine Tjell[†], Nils Schlüter[†], Philipp Binfet, and Moritz Schulze Darup

Proceedings of the 60th Conference on Decision and Control. 2021, pp. 4907–4914. Available at: https://doi.org/10.1109/CDC45484.2021.9683540

[†]Katrine Tjell and Nils Schlüter equally share the first authorship.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Article VIII

Cryptanalysis of Random Affine Transformations for Encrypted Control

Nils Schlüter[†], Philipp Binfet[†] and Moritz Schulze Darup

Proceedings of the 22nd IFAC World Congress. 2023, pp. 12031–12038.
Available at: https://doi.org/10.1016/j.ifacol.2023.10.848

[†] Nils Schlüter and Philipp Binfet share the first authorship.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Due to copyright restrictions, the electronic version of this thesis can not include the reprinted articles.

Appendix A

Algebra for encrypted control

This brief overview serves to refresh or establish algebraic concepts relevant to encrypted control. For a more comprehensive presentation, interested readers may consult [219] or [125, Section 7].

A.1 Ring of integers

Let $\mathbb{Z}_q$ (also denoted $\mathbb{Z}/q\mathbb{Z}$) be represented by $\{0, 1, \ldots, q-1\}$ and let $\bmod q$ be the corresponding modulo reduction. Then, additions (subtractions) and multiplications modulo q form a (commutative) ring (with unity). In other words, these operations are closed and the commutative, associative, and distributive laws hold. Thus, computations in $\mathbb{Z}_q$ behave as expected. Moreover, prime q ensure that an inverse for each element in $\mathbb{Z}_q$ exists, making it a finite field. In particular, for any $a \in \mathbb{Z}_q$, one can find a^{-1} such that $a^{-1}\, a \bmod q = 1$. However, $a^{-1}\, b \bmod q = b/a \in \mathbb{R}$ holds only if $b/a \in \mathbb{Z}_q$.

Remarkable properties of the modulo reduction are

$$\begin{aligned}(a+b) \bmod q &= (a \bmod q + b \bmod q) \bmod q \\ (ab) \bmod q &= (a \bmod q \; b \bmod q) \bmod q.\end{aligned} \tag{A.1}$$

To clarify the implications for encrypted computations, consider the univariate plaintext polynomial

$$g(x) = g_0 + g_1 x + \ldots + g_{n-1} x^{n-1}, \tag{A.2}$$

with $g_i \in \mathbb{R}$, which is encoded in $\mathbb{Z}_q$ via $f_i = \lfloor s g_i \rceil \pmod q$, $z = \lfloor sx \rceil \pmod q$, and

$$f(z) = s^{n-1} f_0 + s^{n-2} f_1 z + \ldots + f_{n-1} z^{n-1} \pmod q.$$

By applying (A.1), one obtains $\mathsf{Dcd}_{s^n}(f(z)) = \mu(f(z))/s^n \approx g(x)$ as long as $f(z) \in [-q/2, q/2) \cap \mathbb{Z}$ (and for $s \in \mathbb{N}$). Therefore, overflows during intermediate computations do not influence the result.

A.2 Ring of polynomials

Constructions analogous to $\mathbb{Z}_q$ also exist in the context of polynomials. First, a reduction of the polynomial degree is required for closure regarding mul-

tiplications. To this end, irreducible (cyclotomic) polynomials $X^N + 1$ with N as a power of 2 are a suitable choice that combines efficiency with security properties when used in cryptography. For example, consider the set $\mathcal{R} = \mathbb{Z}[X]/(X^N + 1)$ which contains integer polynomials in X that are the remainders of the division by $X^N + 1$. Thus, $\mathcal{R}$ contains polynomials with degrees less than N. Second, reducing coefficients additionally modulo q results in the (finite) quotient ring $\mathcal{R}_q = \mathcal{R}/q\mathcal{R} = \mathbb{Z}_q[X]/(X^N + 1)$. The associated elements are of the form $a(X) = \sum_{i=0}^{N-1} \tilde{a}_i X^i$, where $\tilde{a}_i \in \mathbb{Z}_q$. With this at hand, one can define an addition and a multiplication between $a(X), b(X) \in \mathcal{R}_q$ via

$$a(X) + b(X) \bmod q = \sum_{i=0}^{N-1} (\tilde{a}_i + \tilde{b}_i) X^i \bmod q \quad \text{and}$$

$$a(X)b(X) \bmod (X^N + 1) \bmod q = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (-1)^{\lfloor (i+j)/N \rfloor} \tilde{a}_i \tilde{b}_j X^{i+j \bmod N} \bmod q.$$

The latter expression is derived using $X^N = -1 \bmod (X^N + 1)$ and compactly evaluates the remainder of $a(X)b(X)/(X^N + 1)$. Furthermore, it is equivalent to a negative-wrapped convolution of the coefficients, denoted by $\tilde{\boldsymbol{a}} \circledast \tilde{\boldsymbol{b}}$. Similarly to $\mathbb{Z}_q$, $\mathcal{R}_q$ also forms a ring together with the addition and multiplication from above.

Example 4. Consider $(q, N) = (10, 3)$, and polynomials with coefficients $\tilde{\boldsymbol{a}} = (1, 2, -3)^\top$ and $\tilde{\boldsymbol{b}} = (4, -5, 6)^\top$. Using centered representatives, for $a(X) + b(X)$ we first obtain $5 - 3X + 3X^2$ which reduces to $-5 - 3X + 3X^2 \in \mathcal{R}_q$ via a reduction modulo q. Similarly, $a(X)b(X)$ first evaluates to $4 + 3X + 4X^2 - 3X^3 + 2X^4$, which results in $-3 + 1X + 4X^2 \in \mathcal{R}_q$ through the reductions.

Number-theoretic transformation. Despite the optimization from above, evaluating $a(X)b(X) \in \mathcal{R}_q$ requires $\mathcal{O}(N^2)$ multiplications. However, this can be reduced to $\mathcal{O}(N \log(N))$ using the number-theoretic transformation (NTT) [155]. There, coefficients $\tilde{a}_i \in \mathbb{Z}_q$ of a polynomial $a(X)$ are transformed via

$$\tilde{A}_i = \sum_{j=0}^{N-1} \tilde{a}_j \omega^{ij} \,(\mathrm{mod}\, q), \quad \text{or compactly} \quad \tilde{\boldsymbol{A}} = \mathsf{NTT}(\boldsymbol{a}) = \boldsymbol{\Omega} \boldsymbol{a} \,(\mathrm{mod}\, q),$$

where ω is a N-th primitive root of unity in $\mathbb{Z}_q$. The inverse transformation is $\mathsf{NTT}^{-1}(\tilde{\boldsymbol{A}}) = N^{-1} \boldsymbol{\Omega}^{-1} \boldsymbol{A} \,(\mathrm{mod}\, q)$. Based on the NTT, one can evaluate a convolution of $\tilde{\boldsymbol{a}}$ and $\tilde{\boldsymbol{b}}$ over $\mathbb{Z}_q$. In order to obtain a negative-wrapped convolution, $\psi^2 = \omega \,(\mathrm{mod}\, q)$ is required, which exists for prime q and N as a power of 2. With this at hand, we define $\boldsymbol{\psi} = (1, \psi, \psi^2, \ldots, \psi^{N-1})^\top$ and $\boldsymbol{\psi}^{-1}$ such that $\boldsymbol{\psi} \odot \boldsymbol{\psi}^{-1} = \mathbf{1} \,(\mathrm{mod}\, q)$, where $\odot$ denotes component-wise multiplications. Then,

$$\tilde{\boldsymbol{a}} \circledast \tilde{\boldsymbol{b}} = \boldsymbol{\psi}^{-1} \odot \mathsf{NTT}^{-1} \left(\mathsf{NTT}(\boldsymbol{\psi} \odot \tilde{\boldsymbol{a}}) \odot \mathsf{NTT}(\boldsymbol{\psi} \odot \tilde{\boldsymbol{b}}) \right)$$

can be computed in $\mathcal{O}(N \log(N))$ as desired using an FFT-algorithm and assuming all necessary ω^{ij} as well as ψ^i are precomputed.

Residue number systems. Another bottleneck constitutes the computation with large numbers that result from large q. Although algorithms (e.g., Karatsuba) for fast multiplications of large numbers exist, more efficiency is gained by exploiting the ring isomorphism $\mathcal{R}_q \cong \mathcal{R}_{\mathsf{q}_1} \times \mathcal{R}_{\mathsf{q}_2} \ldots \times \mathcal{R}_{\mathsf{q}_\ell}$. Here, $q = \prod_{l=1}^{\ell} \mathsf{q}_l$ is a "tower" of (co-)prime q_i that fit into the word-size of the machine. This way, efficient computations in the independent small rings $\mathcal{R}_{\mathsf{q}_l}$ are enabled. The NTT as well as residue number systems are used in competitive implementations of RLWE cryptosystems, e.g., for CKKS see [49].

Appendix B

Supplements on cryptography

B.1 Security

B.1.1 Random number generation

Cryptographic constructions heavily rely on the secure generation of random integers/bit-strings. To this end, randomness can be derived from a high entropy source, such as operating system events (inter-keypress timings, mouse movements, hardware interrupts) or physical phenomena (for instance, thermal/electrical noise, radioactive decay) [32, 125, Chapter 3]. It is crucial for many applications that an adversary cannot reconstruct this randomness.

To meet the substantial demand for random numbers, randomness is typically utilized as a (sufficiently long and secret) seed σ in a *cryptographically secure pseudo-random number generator* (CSPRNG). A CSPRNG can be conceptualized as an efficient and deterministic algorithm G, which expands $\sigma \in \{0,1\}^l$ into a longer sequence $G(\sigma) \in \{0,1\}^L$ where $L > l$. Importantly, $G(\sigma)$ is computationally indistinguishable from true uniform randomness for a PPT $\mathcal{A}$ who does not know σ. Note that random number generators used in numerical analysis lack this (and other) security requirements and should never be used for real cryptographic purposes.

B.1.2 Reduction proofs

When it comes to proving the security of a scheme, one first assumes that an attacker $\mathcal{A}$ has non-negligible success in the IND-CPA experiment (2.11). Then, one shows that this $\mathcal{A}$ can be used to solve a mathematical problem (like integer factorization, discrete logarithms, or the composite residuosity problem) in polynomial time with non-negligible success. However, if that problem has been shown to require non-polynomial time, it contradicts the existence of $\mathcal{A}$. This procedure is called *reduction*.

B.1.3 Mask-and-open

Let $[z]$ and $[r]$ be shares of $z \in \mathbb{Z}_q$ and $r \leftarrow [0, 2^\kappa(q-1)] \cap \mathbb{Z}$ in the modulus $q' > (q-1)(2^\kappa+1)$, respectively. Then,

$$\mathsf{Reconstruct}([z]+[r]) = \sum_{j=1}^{M} z^{(j)} + r^{(j)} \,(\mathrm{mod}\, q') = z + r.$$

This is called *mask-and-open* and, in contrast to $[z]$, allows computing on $z+r$ without a modulo reduction, which can greatly simplify protocols. The security of $z+r$ is as follows.

Theorem 6. *The masked share $z+r$ is statistically secure.*

Proof. Statistical security means that ciphertexts are statistically indistinguishable. Let $\Pr(z+r=z')$ denote the probability of observing z' given z, where r is as above. Then, via Definition 2 and 4

$$\frac{1}{2} \sum_{z'=0}^{(q-1)(2^\kappa+1)} |\Pr(z_1+r=z') - \Pr(z_2+r=z')| \leqslant \mathsf{negl}(\kappa)$$

must hold for every z_1, z_2. To this end, verify that

$$\Pr(z_i+r=z') = \begin{cases} 0 & \text{if } z_i > z' \text{ or } z' > z + 2^\kappa(q-1) \\ \frac{1}{2^\kappa(q-1)} & \text{otherwise.} \end{cases}$$

With this at hand, one finds

$$|\Pr(z_1+r=z') - \Pr(z_2+r=z')| = \frac{2|z_1-z_2|}{2^\kappa(q-1)}$$

where the worst-case occurs for $|z_1-z_2| = |q-1-0| = q-1$. Then, it follows that

$$\frac{1}{2} \sum_{z'=0}^{(q-1)(2^\kappa+1)} |\Pr(z_1+r=z') - \Pr(z_2+r=z')| = \frac{1}{2^\kappa} \leqslant \mathsf{negl}(\kappa).$$

□

In other words, the statistical security of the mask-and-open approach requires making the scheme's modulus larger, i.e., from q to q', which results in larger shares.

B.2 Homomorphic encryption

B.2.1 An intuition about the hardness of LWE

Let us discuss some seemingly promising approaches for the search LWE problem. To this end, LWE tuples $(b_i, \boldsymbol{a}_i^\top)$ are given, which can be concatenated

(row-wise) into $(\boldsymbol{b}, \boldsymbol{A})$. An important aspect, when it comes to *key-recovery attacks* on LWE, is that guesses of $\mathbf{sk}$ can be verified. To see this, note that $\boldsymbol{b} + \boldsymbol{A}\mathbf{sk} \bmod q$ has only small entries for the correct $\mathbf{sk}$ due to the small e_i. However, *brute-forcing* all possibilities for $\mathbf{sk}$ is intractable for large N. Next, we discuss several methods from linear algebra and optimization.

First, we apply Gaussian elimination. To this end, an assumption regarding the unknown errors e_i in each of the equations $b_i = -\boldsymbol{a}_i^\top \mathbf{sk} + e_i \pmod q$ is required. Here, we apply the certainty equivalence principle (ignoring e_i) and build the linear equation system $-\boldsymbol{A}\mathbf{sk} = \boldsymbol{b} \pmod q$ using N LWE-tuples. However, Gaussian elimination linearly combines the equations, which amplifies the errors. Thus, simply ignoring them is hopeless for sufficiently large N. Besides, one may presume $e_i \in \{-1, 0, 1\}$ (or similar) and enumerate all 3^N combinations, which also fails for large enough N. Secondly, we consider $\arg\min_{\mathbf{sk} \in \mathbb{Z}_q^N} \|\boldsymbol{b} + \boldsymbol{A}\mathbf{sk} \bmod q\|_2^2$. The resulting $\mathbf{sk}$ is, with high probability, correct. However, solving integer least-square problems is NP-hard (see [109] and references therein) such that large N (and q) makes a solution intractable. Third, another attempt builds on constructing "more useful" tuples [30]. For instance, the linear combination $\sum_i \alpha_i \boldsymbol{a}_i = (\gamma, 0, \ldots, 0)^\top$ allows for the approximation $\mathrm{sk}_1 = \gamma^{-1} \sum_i \alpha_i (e_i - b_i) \approx -\gamma^{-1} \sum_i \alpha_i b_i \pmod q$ if $\sum_i \alpha_i b_i \gg \sum_i \alpha_i e_i$. Then, one proceeds similarly for the remaining components of $\mathbf{sk}$. The issues with this approach are two-fold. On the one hand, it requires exponential time to obtain suitable samples. On the other hand, the secret is somewhat sensitive to perturbation.

In fact, LWE has been proven to be as hard as worst-case lattice (a discrete subgroup of $\mathbb{R}^N$) problems [195]. Consequently, progress on LWE implies breakthroughs in other areas, such as coding theory [196]. As the state of affairs presents itself, the best-known algorithms for LWE operate in exponential time (see [163] for an overview), and even quantum algorithms do not offer improvements [4]. This is despite LWE has been subjected to a lot of scrutiny throughout the last decade because of its versatile applications in cryptography.

Parameter choices. From our short exposition, we learn that the main security parameter is N. We also note that the "signal-to-noise ratio" q/e must be small for high security (noise-free LWE is trivial). Moreover, since several $(b_i, \boldsymbol{a}_i^\top)$ can be linearly combined to "new" samples with a slightly larger error, the hardness of LWE is essentially independent of the number of samples. In practice, LWE-based schemes come with a predetermined error distribution. Then, one selects q based on the computations at hand and increases N such that a desired security threshold is exceeded. Typically, this leads to at least $N \geqslant 128$ where small N should be treated with care. In this context, the LWE estimator [4] is helpful because state-of-the-art attacks are considered. More conservative recommendations can be found in [3].

B.2.2 An overview of homomorphic cryptosystems

Next, a brief overview of HE cryptosystems is provided, where we highlight their use in the encrypted control literature. For the interested reader, we refer to the detailed HE survey [163] and the original papers.

One of the first PHE schemes is RSA [199] which provides the simple encryption $\mathsf{ct}(z) = z^e \,(\mathrm{mod}\, q)$, where $\mathsf{pk} = (q, e)$ with the exponent e.[1] It is easy to verify that $\mathsf{ct}(z_1)\mathsf{ct}(z_2) = (z_1 z_2)^e \,(\mathrm{mod}\, q)$ and $\mathsf{ct}(z_1)^{z_2} = (z_1^{z_2})^e \,(\mathrm{mod}\, q)$ are a multiplicative homomorphism and public exponentiation, respectively. RSA is used in practice for public key encryption and digital signatures. Similarly, ElGamal [77] provides encrypted multiplications and public exponentiation. It can be found in early works in encrypted control, such as [138]. Next, the Paillier cryptosystem [182] is additively homomorphic and allows for public multiplications (see Section 2.2.3). This enables, for example, weighted sums with public weights and has led to a wider application in encrypted control [5, 83, 212].

About three decades after the first PHE scheme, [194] proposed the LWE problem and showed the construction of a simple additively homomorphic cryptosystem based on it. Shortly after, the first FHE scheme [91] was invented. The groundbreaking idea was to evaluate the decryption circuit homomorphically in order to provide arbitrary computations, which is called bootstrapping. However, the constructions were very far from practicality. In comparison, current state-of-the-art schemes, which are based almost entirely on RLWE (and LWE), have reduced the computation times by several orders of magnitude while providing encrypted additions and multiplications. These schemes fall mainly into two categories, i.e., *arithmetic* and *fast-bootstrapping schemes*.

Arithmetic schemes. These include the Brakerski-Gentry-Vaikuntantan (BGV; [38]) and the Brakerski/Fan-Vercauteren (BFV; [80]) cryptosystem, which are used for modular arithmetic, and the CKKS cryptosystem (see Section 2.2.4) which is used for real-number computations. Arithmetic schemes have commonalities, such as their realization of Mult. In particular, Mult yields a ciphertext in sk^2 at first, which is then *re-linearized* to a ciphertext that is decryptable by sk. Furthermore, these schemes use packing methods that enable SIMD operations. However, they are not well-suited for non-polynomial computations and provide slow bootstrapping routines. Implementations of arithmetic schemes can be found in [1, 46, 104]. In encrypted control, they are used in, e.g., [8, P3, P8, P15].

Fast-bootstrapping schemes. These emerged from the Gentry-Sahai-Waters (GSW; [92]) cryptosystem. GSW is used in [134, 135, 222] to encrypt dynamic

[1] This variant is called "textbook RSA" and it is insecure. In order to fix that, replace z with $r||z$, where the message string z is concatenated with a random string r.

controllers. RLWE-variants of it are the Ducas-Micciancio (DM; [74]) and the Chillotti-Gama-Georgieva-Izabachene (CGGI; [55]) cryptosystems which follow the blueprint of GSW and realize Mult with a different approach. Namely, instead of using $\mathsf{ct}(z_1) = (b, a) = (-a\,\mathsf{sk} + z_1 + e \bmod q, a)$, we define

$$\mathsf{ct}'(z_1) = \Big(\mathsf{ct}(z_1), \mathsf{ct}(Bz_1), \mathsf{ct}(B^2 z_1), \ldots, \mathsf{ct}(B^{l-1} z_1)\Big)\,,$$

where B is a base (typically $B = 2$) and $l = \log_B(q)$. Then, for a public multiplication with $z_2 \in \mathcal{R}_q$, we compute $z_2 = \sum_{i=0}^{l-1} B^i Y_{z_2,i}$ with the component polynomials $Y_{z_2,i}$ and evaluate

$$\mathsf{ct}(z_1 z_2) = \sum_{i=0}^{l-1} \mathsf{ct}(B^i z_1) Y_{z_2,i} \,(\mathrm{mod}\, q). \tag{B.1}$$

In comparison to $\mathsf{ct}(z_1)z_2$, (B.1) results in a much smaller error growth which enhances the accuracy and computational depth (similar to Rescale). More precisely, $bz_2 = -a\,z_2\,\mathsf{sk} + z_1 z_2 + e z_2 \bmod q$ contains the large error ez_2, whereas in (B.1) an error in $\mathsf{ct}'(z_1)$ is multiplied by $Y_{z_2,i}$ with small coefficients. However, observe that the resulting ciphertext is $\mathsf{ct}(z_1 z_2)$ and not $\mathsf{ct}'(z_1 z_2)$. To enable more than one multiplication, a more complex (tensor-like) ciphertext $\mathsf{ct}''(z)$ can be introduced (see [167] for details). This way,

$$\mathsf{ct}(z_1 z_2)' = \mathsf{Mult}(\mathsf{ct}'(z_1), \mathsf{ct}''(z_2)), \quad \text{and} \quad \mathsf{ct}(z_1 z_2)'' = \mathsf{Mult}(\mathsf{ct}''(z_1), \mathsf{ct}''(z_2))$$

are possible. Additional functionalities in DM and CGGI allow switching between these and more ciphertext types, such as LWE.

Another characteristic of these schemes is that they do not use packing. Thus, a Rotation of $\mathsf{ct}(z)$ by j positions can be realized via $X^j\mathsf{ct}(z)$ which is exploited for *fast bootstrapping* in [55]: Based on LWE ciphertexts and encryptions of sk_i, one computes the decryption $b + \boldsymbol{a}^\top \mathbf{sk} \approx z \,(\mathrm{mod}\, q)$ homomorphically. Then, a RLWE dummy ciphertext V is prepared, which contains fresh encryptions of all possible values for z. With this at hand, observe that $X^{-z}V$ rotates a fresh ciphertext of z to the first entry, which can then be selected. To hide z in X^{-z} a technique called *blind rotations* is used. This way, bootstrapping is evaluated much faster in comparison to its arithmetic counterpart. Interestingly, instead of values for z, also functions $f(z)$ can be stored in V. This is referred to as *programmable bootstrapping*. Implementations can be found in [1, 56]. While these methods resolve problems of arithmetic schemes, fast-bootstrapping schemes only support small integers (currently 2-8 bits) for efficiency reasons, and their ciphertexts as well as their computational complexity are larger. This explains why these schemes have not yet been used in encrypted control and are often used for Boolean circuits.

Scheme switching. In order to combine the best of arithmetic and fast-bootstrapping schemes, homomorphic *scheme-switching* was invented [36]. In this

context, especially switching between CKKS and DM/CGGI is of interest. From the former to the latter, a linear transformation to decode the message in a CKKS ciphertext and an extraction into LWE ciphertexts is required. From the latter to the former, a linear transformation to perform the CKKS encoding followed by a lightweight CKKS bootstrap must be performed [156].

Appendix C

Max-out neural networks

C.1 Optimal fixed-point quantization

Let the integer-variant of (3.11) be

$$\hat{v}(\boldsymbol{x}) = \max\{\lfloor s_1 \boldsymbol{L} \rfloor \boldsymbol{z} + \lfloor s_3 \boldsymbol{d} \rfloor\} - \max\{\lfloor s_1 \boldsymbol{M} \rfloor \boldsymbol{z} + \lfloor s_3 \boldsymbol{e} \rfloor\} \,(\mathrm{mod}\, q).$$

In order to avoid overflows, we determine $\|\boldsymbol{L}\boldsymbol{x} + \boldsymbol{d}\|_\infty$ and $\|\boldsymbol{M}\boldsymbol{x} + \boldsymbol{e}\|_\infty$ for $\boldsymbol{x} \in \cup_{i=1}^{\theta} \mathcal{P}_i$ via linear programs and subsequently select a suitable $q = 2^\varphi$. Then, the error $|u(\boldsymbol{x}) - \mathsf{Dcd}_{s_3}(\hat{v}(\boldsymbol{x}))|$ is minimized for the largest possible s_3, i.e., $s_3 = 2^{\varphi-1} / \max\{\|\boldsymbol{L}\boldsymbol{x} + \boldsymbol{d}\|_\infty, \|\boldsymbol{M}\boldsymbol{x} + \boldsymbol{e}\|_\infty\}$ and it remains to specify s_1, s_2. From the decoding constraint, we obtain $s_2 = s_3/s_1$, while Proposition 1 allows us to deduce that the optimal choice satisfies

$$s_1 \|\boldsymbol{x}\|_\infty = s_2 \max\{\|\boldsymbol{L}\|_{\max}, \|\boldsymbol{M}\|_{\max}\}$$

which determines s_1.

C.2 Convex training algorithm

Let $\{(\boldsymbol{x}_1, u_1), \ldots (\boldsymbol{x}_{N_s}, u_{N_s})\}$ be N_s samples of the function $u(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ which shall be approximated via $\hat{u}(\boldsymbol{x}) = \max\{\boldsymbol{L}\boldsymbol{x} + \boldsymbol{d}\} - \max\{\boldsymbol{M}\boldsymbol{x} + \boldsymbol{e}\}$. In this regard, a straightforward approach is solving the nonlinear program

$$\min_{\boldsymbol{L},\boldsymbol{d},\boldsymbol{M},\boldsymbol{e}} \sum_{i=1}^{N_s} \left(u_i - \max\{\boldsymbol{L}\boldsymbol{x}_i + \boldsymbol{d}\} - \max\{\boldsymbol{M}\boldsymbol{x}_i + \boldsymbol{e}\}\right)^2, \tag{C.1}$$

where $\boldsymbol{L}, \boldsymbol{M} \in \mathbb{R}^{p \times n}$ and $\boldsymbol{d}, \boldsymbol{e} \in \mathbb{R}^p$. However, solving (C.1) is tedious because p is user-defined, the optimization process is slow due to max{}, and good initial guesses are crucial.

Instead, we can solve (C.1) by convex optimization as follows. First, we triangulate $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_s}\}$ with a regular triangulation, e.g., a Delaunay triangulation (see [67]), which we denote by $\mathcal{T}_{\mathrm{reg}}(\boldsymbol{X})$. This step has a complexity of $\mathcal{O}(N_s \log(N_s))$ and results in a set of polyhedrons $\{\mathcal{P}_i\}_{i=1}^{\theta}$ with neighborhood relations $\mathcal{B}$ (see Section 3.2.2). At the same time, $\mathcal{T}_{\mathrm{reg}}(\boldsymbol{X})$ ensures that

an optimization-based convex decomposition (as in [P10]) is always feasible. Thus, we utilize this approach while we do not enforce strict convexity between neighboring affine segments and use, e.g., the weighted cost function

$$J(w) = (1-w)\sum_{i=1}^{N_s}\left(u_i - \boldsymbol{l}_{i_x}^\top \boldsymbol{x}_i - d_{i_x} - \boldsymbol{m}_{i_x}^\top \boldsymbol{x}_i - e_{i_x}\right)^2 + w\sum_{(i,j)\in\mathcal{B}} \|\boldsymbol{l}_i - \boldsymbol{l}_j\|_1 + |e_i - e_j| + \\ |\mathbf{1}^\top(\boldsymbol{l}_i - \boldsymbol{l}_j) + e_i - e_j| + \|\boldsymbol{m}_i - \boldsymbol{m}_j\|_1 + |d_i - d_j| + |\mathbf{1}^\top(\boldsymbol{m}_i - \boldsymbol{m}_j) + d_i - d_j|,$$

where i_x is such that $\boldsymbol{x}_i \in \mathcal{P}_{i_x}$. For $w = 0$, the convex decomposition becomes more accurate (see Figure 3.3), while $w = 1$ promotes affine segments with the same coefficients. This allows us to unite them and reduce p. Finally, for a given accuracy, the minimal p is found by running a bisection on w.

Bibliography

[1] A. Al Badawi et al. "OpenFHE: Open-Source Fully Homomorphic Encryption Library." In: *10th Workshop on Encrypted Computing & Applied Homomorphic Cryptography*. 2022, pp. 53–63.

[2] N. Alamati and C. Peikert. "Three's compromised too: Circular insecurity for any cycle length from (Ring-)LWE." In: *Advances in Cryptology – CRYPTO*. Springer. 2016, pp. 659–680.

[3] M. Albrecht et al. *Homomorphic Encryption Security Standard*. Tech. rep. Toronto, Canada: HomomorphicEncryption.org, 2018.

[4] M. R. Albrecht, R. Player, and S. Scott. "On the concrete hardness of learning with errors." In: *Journal of Mathematical Cryptology* 9.3 (2015), pp. 169–203.

[5] A. B. Alexandru, M. Morari, and G. J. Pappas. "Cloud-based MPC with Encrypted Data." In: *Proceedings of the 57th Conference on Decision and Control*. IEEE. 2018, pp. 5014–5019.

[6] A. B. Alexandru and G. J. Pappas. "Encrypted LQG using labeled homomorphic encryption." In: *Proceedings of the 10th ACM/IEEE Conference on Cyber-Physical Systems*. 2019, pp. 129–140.

[7] A. B. Alexandru, A. Tsiamis, and G. J. Pappas. "Encrypted Distributed Lasso for Sparse Data Predictive Control." In: *Proceedings of the 60th Conference on Decision and Control*. IEEE. 2021, pp. 4901–4906.

[8] A. B. Alexandru, A. Tsiamis, and G. J. Pappas. "Towards Private Data-driven Control." In: *Proceedings of the 59th Conference on Decision and Control*. IEEE. 2020, pp. 5449–5456.

[9] A. B. Alexandru, M. Schulze Darup, and G. J. Pappas. "Encrypted cooperative control revisited." In: *Proceedings of the 58th Conference on Decision and Control*. IEEE. 2019, pp. 7196–7202.

[10] R. Alisic, J. Kim, and H. Sandberg. "Model-Free Undetectable Attacks on Linear Systems Using LWE-Based Encryption." In: *IEEE Control Systems Letters* 7 (2023), pp. 1249–1254.

[11] T. Alladi, V. Chamola, and S. Zeadally. "Industrial control systems: Cyberattack trends and countermeasures." In: *Computer Communications* 155 (2020), pp. 1–8.

[12] M. Alsayegh, J. Fuentes, L. Bobadilla, and D. A. Shell. "Oblivious Markov Decision Processes: Planning and Policy Execution." In: *Proceedings of the 62nd Conference on Decision and Control*. IEEE. 2023, pp. 3850–3857.

[13] A. Aly and N. P. Smart. "Benchmarking privacy preserving scientific operations." In: *Proceedings of the 17th Conference on Applied Cryptography and Network Security*. Springer. 2019, pp. 509–529.

[14] S. Amin, A. A. Cárdenas, and S. S. Sastry. "Safe and Secure Networked Control Systems under Denial-of-Service Attacks." In: *Proceedings of the 12th Conference on Hybrid Systems: Computation and Control*. Springer. 2009, pp. 31–45.

[15] T. M. Apostol. "Resultants of cyclotomic polynomials." In: *Proceedings of the American Mathematical Society* 24.3 (1970), pp. 457–462.

[16] Y. Bae, J. H. Cheon, W. Cho, J. Kim, and T. Kim. "META-BTS: Bootstrapping Precision Beyond the Limit." In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. 2022.

[17] P. Barooah and J. P. Hespanha. "Estimation on Graphs from Relative Measurements." In: *IEEE Control Systems Magazine* 27.4 (2007), pp. 57–74.

[18] D. Beaver. "Efficient Multiparty Protocols Using Circuit Randomization." In: *Proceedings of the International Cryptology Conference*. Springer. 1991, pp. 420–432.

[19] D. Beaver, S. Micali, and P. Rogaway. "The round complexity of secure protocols." In: *Proceedings of the 22nd Symposium on Theory of Computing*. ACM, 1990, pp. 503–513.

[20] M. Bellare, V. Hoang, and P. Rogaway. "Foundations of Garbled Circuits." In: *Proceedings of the Conference on Computer and Communications Security*. 2012, pp. 784–796.

[21] M. Bellare, R. Canetti, and H. Krawczyk. "Keying hash functions for message authentication." In: *Advances in Cryptology — CRYPTO*. Springer. 1996, pp. 1–15.

[22] M. Bellare, V. T. Hoang, S. Keelveedhi, and P. Rogaway. "Efficient garbling from a fixed-key blockcipher." In: *Symposium on Security and Privacy*. IEEE. 2013, pp. 478–492.

[23] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. "The explicit linear quadratic regulator for constrained systems." In: *Automatica* 38.1 (2002), pp. 3–20.

[24] J. Benaloh and J. Leichter. "Generalized Secret Sharing and Monotone Functions." In: *Advances in Cryptology — CRYPTO 1988*. Springer New York, 1990, pp. 27–35.

[25] R. Bendlin, I. Damgård, C. Orlandi, and S. Zakarias. "Semi-homomorphic encryption and multiparty computation." In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2011, pp. 169–188.

[26] C. Berge. *Topological spaces: Including a treatment of multi-valued functions, vector spaces and convexity*. Oliver & Boyd, 1963.

[27] L. Bergerat et al. "Parameter Optimization and Larger Precision for (T)FHE." In: *Journal of Cryptology* 36.3 (2023), p. 28.

[28] D. Bertsekas. *Reinforcement learning and optimal control*. Vol. 1. Athena Scientific, 2019.

[29] M. Bishop et al. *Introduction to computer security*. Vol. 50. Addison-Wesley Boston, 2005.

[30] A. Blum, A. Kalai, and H. Wasserman. "Noise-tolerant learning, the parity problem, and the statistical query model." In: *Journal of the ACM* 50.4 (2003), pp. 506–519.

[31] S. Bolognani, S. D. Favero, L. Schenato, and D. Varagnolo. "Consensus-based distributed sensor calibration and least-square parameter identification in WSNs." In: *International Journal of Robust and Nonlinear Control* 20.2 (2010), pp. 176–193.

[32] D. Boneh and V. Shoup. *A Graduate Course in Applied Cryptography*. http://toc.cryptobook.us/. 2023.

[33] J. Bos et al. "CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM." In: *Proceedings of the European Symposium on Security and Privacy*. 2018, pp. 353–367.

[34] J.-P. Bossuat, C. Mouchet, J. Troncoso-Pastoriza, and J.-P. Hubaux. "Efficient Bootstrapping for Approximate Homomorphic Encryption with Non-sparse Keys." In: *Advances in Cryptology – EUROCRYPT*. Springer. 2021.

[35] A. Botta, W. De Donato, V. Persico, and A. Pescapé. "Integration of cloud computing and internet of things: a survey." In: *Future generation computer systems* 56 (2016), pp. 684–700.

[36] C. Boura, N. Gama, M. Georgieva, and D. Jetchev. "Chimera: Combining Ring-LWE-based Fully Homomorphic Encryption Schemes." In: *Journal of Mathematical Cryptology* 14.1 (2020), pp. 316–338.

[37] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers." In: *Foundations and Trends in Machine Learning* 3.1 (2011), pp. 1–122.

[38] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. "(Leveled) Fully Homomorphic Encryption without Bootstrapping." In: *ACM Transactions on Computation Theory* 6.3 (2014), 13:1–13:36.

[39] F. Bullo. *Lectures on Network Systems*. 1.6. Kindle Direct Publishing, 2022.

[40] R. Canetti. "Security and composition of multiparty cryptographic protocols." In: *Journal of Cryptology* 13 (2000), pp. 143–202.

[41] R. Canetti. "Universally composable security: A new paradigm for cryptographic protocols." In: *Proceedings of the 42nd Symposium on Foundations of Computer Science*. IEEE. 2001, pp. 136–145.

[42] O. Catrina and S. De Hoogh. "Improved primitives for secure multiparty integer computation." In: *Proceedings of the 7th International Conference on Security and Cryptography for Networks*. Springer. 2010, pp. 182–199.

[43] O. Catrina and A. Saxena. "Secure computation with fixed-point numbers." In: *Proceedings of the Conference on Financial Cryptography and Data Security*. Springer. 2010, pp. 35–50.

[44] A. Cetinkaya, H. Ishii, and T. Hayakawa. "An overview on denial-of-service attacks in control systems: Attack models and security analyses." In: *Entropy* 21.2 (2019), p. 210.

[45] H. Chen, I. Chillotti, and Y. Song. "Improved Bootstrapping for Approximate Homomorphic Encryption." In: *Advances in Cryptology – EUROCRYPT*. Springer. 2019.

[46] H. Chen, K. Laine, and R. Player. "Simple Encrypted Arithmetic Library – SEAL v2.1." In: *Financial Cryptography and Data Security*. Springer. 2017, pp. 3–18.

[47] Y. Chen et al. "Exploring Shodan from the perspective of industrial control systems." In: *IEEE Access* 8 (2020), pp. 75359–75369.

[48] J. H. Cheon, K. Han, H. Kim, J. Kim, and H. Shim. "Need for controllers having integer coefficients in homomorphically encrypted dynamic system." In: *Proceedings of the 57th Conference on Decision and Control*. IEEE. 2018, pp. 5020–5025.

[49] J. H. Cheon, K. Han, A. Kim, M. Kim, and Y. Song. "A full RNS variant of approximate homomorphic encryption." In: *Selected Areas in Cryptography – SAC 2018*. Springer. 2019, pp. 347–368.

[50] J. H. Cheon, K. Han, A. Kim, M. Kim, and Y. Song. "Bootstrapping for Approximate Homomorphic Encryption." In: *Advances in Cryptology – EUROCRYPT*. Springer. 2018.

[51] J. H. Cheon, A. Kim, M. Kim, and Y. Song. "Homomorphic Encryption for Arithmetic of Approximate Numbers." In: *Advances in Cryptology – ASIACRYPT*. Springer. 2017.

[52] J. H. Cheon, D. Kim, D. Kim, H. H. Lee, and K. Lee. "Numerical Method for Comparison on Homomorphically Encrypted Numbers." In: *Advances in Cryptology – ASIACRYPT*. Springer. 2019, pp. 415–445.

[53] J. H. Cheon, D. Kim, J. Kim, S. Lee, and H. Shim. "Authenticated computation of control signal from dynamic controllers." In: *Proceedings of the 59th Conference on Decision and Control*. IEEE. 2020, pp. 3249–3254.

[54] J. H. Cheon et al. "Toward a secure drone system: Flying with real-time homomorphic authenticated encryption." In: *IEEE Access* 6 (2018), pp. 24325–24339.

[55] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. "TFHE: Fast fully homomorphic encryption over the torus." In: *Journal of Cryptology* 33.1 (2020), pp. 34–91.

[56] I. Chillotti, M. Joye, D. Ligier, J.-B. Orfila, and S. Tap. "CONCRETE: Concrete operates on ciphertexts rapidly by extending TFHE." In: *8th Workshop on Encrypted Computing & Applied Homomorphic Cryptography*. 2020.

[57] A. Cohen. "What about Bob? The inadequacy of CPA security for proxy reencryption." In: *IACR International Workshop on Public Key Cryptography*. Springer. 2019, pp. 287–316.

[58] J. Coulson, J. Lygeros, and F. Dörfler. "Data-enabled predictive control: In the shallows of the DeePC." In: *Proceedings of the 18th European Control Conference*. IEEE. 2019, pp. 307–312.

[59] T. M. Cover. *Elements of information theory*. John Wiley & Sons, 1999.

[60] R. Cramer, I. B. Damgård, and J. B. Nielsen. *Secure multiparty computation*. Cambridge University Press, 2015.

[61] B. R. Curtis and R. Player. "On the Feasibility and Impact of Standardising Sparse-secret LWE Parameter Sets for Homomorphic Encryption." In: *7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography*. ACM, 2019, pp. 1–10.

[62] J. Daemen and V. Rijmen. "AES Proposal: Rijndael." In: *AES Algorithm Submission*. Gaithersburg, MD, USA, 1999.

[63] A. Dalskov, D. Escudero, and M. Keller. "Secure evaluation of quantized neural networks." In: *arXiv preprint arXiv:1910.12435* (2019).

[64] I. Damgård, M. Fitzi, E. Kiltz, J. B. Nielsen, and T. Toft. "Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation." In: *Proceedings of the Theory of Cryptography Conference*. Springer. 2006, pp. 285–304.

[65] I. Damgård et al. "Practical covertly secure MPC for dishonest majority–or: breaking the SPDZ limits." In: *Computer Security – ESORICS*. Springer. 2013, pp. 1–18.

[66] P. A. Damianou. "Monic polynomials in $\mathbb{Z}[x]$ with roots in the unit disc." In: *The American Mathematical Monthly* 108.3 (2001), pp. 253–257.

[67] J. A. De Loera, J. Rambau, and F. Santos. *Triangulations Structures for algorithms and applications*. Springer, 2010.

[68] C. De Persis and P. Tesi. "Input-to-state stabilizing control under denial-of-service." In: *IEEE Transactions on Automatic Control* 60.11 (2015), pp. 2930–2944.

[69] D. F. Delchamps. "Stabilizing a Linear System with Quantized State Feedback." In: *IEEE Transactions Automatic Control* 35.8 (1990), pp. 916–924.

[70] P. S. Diniz, E. A. Da Silva, and S. L. Netto. *Digital signal processing: system analysis and design*. Cambridge University Press, 2010.

[71] J. Dreier and F. Kerschbaum. "Practical privacy-preserving multiparty linear programming based on problem transformation." In: *Proceedings 3rd International Conference on Privacy, Security, Risk, and Trust*. IEEE. 2011, pp. 916–924.

[72] W. Du and M. J. Atallah. "Secure Multi-party Computation Problems and Their Applications: A Review and Open Problems." In: *Workshop on new security paradigms*. 2001, pp. 13–22.

[73] W. Du and Z. Zhan. "A practical approach to solve secure multi-party computation problems." In: *Workshop on new security paradigms*. 2002, pp. 127–135.

[74] L. Ducas and D. Micciancio. "FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second." In: *Advances in Cryptology – EUROCRYPT*. Springer. 2015, pp. 617–640.

[75] C. Dwork. "Differential privacy." In: *Proceedings of the 33rd Colloquium on Automata, Languages and Programming*. Springer. 2006, pp. 1–12.

[76] C. Dwork, A. Roth, et al. "The algorithmic foundations of differential privacy." In: *Foundations and Trends in Theoretical Computer Science* 9.3–4 (2014), pp. 211–407.

[77] T. ElGamal. "A public key cryptosystem and a signature scheme based on discrete logarithms." In: *IEEE Transactions on Information Theory* 31.4 (1985), pp. 469–472.

[78] M. D. Ercegovac and T. Lang. *Digital arithmetic*. Elsevier, 2004.

[79] D. Escudero, S. Ghosh, M. Keller, R. Rachuri, and P. Scholl. "Improved Primitives for MPC over Mixed Arithmetic-Binary Circuits." In: *Proceedings of the 40th Annual International Cryptology Conference*. Springer, 2020, pp. 823–852.

[80] J. Fan and F. Vercauteren. "Somewhat Practical Fully Homomorphic Encryption." In: *Cryptology ePrint Archive, Paper 2012/144* (2012).

[81] B. Fang et al. "The contributions of cloud technologies to smart grid." In: *Renewable and Sustainable Energy Reviews* 59 (2016), pp. 1326–1331.

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

[83] F. Farokhi, I. Shames, and N. Batterham. "Secure and private cloud-based control using semi-homomorphic encryption." In: *6th IFAC Workshop on Distributed Estimation and Control in Networked Systems* 49.22 (2016), pp. 163–168.

[84] R. M. Ferrari and A. M. Teixeira. "A switching multiplicative watermarking scheme for detection of stealthy cyber-attacks." In: *IEEE Transactions on Automatic Control* 66.6 (2020), pp. 2558–2573.

[85] T. K. Frederiksen, M. Keller, E. Orsini, and P. Scholl. "A Unified Approach to MPC with Preprocessing using OT." In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2015, pp. 711–735.

[86] T. Fujita, K. Kogiso, K. Sawada, and S. Shin. "Security enhancements of networked control systems using RSA public-key cryptosystem." In: *Proceedings of the 10th Asian Control Conference*. IEEE. 2015, pp. 1–6.

[87] A. J. Gallo, S. C. Anand, A. M. Teixeira, and R. M. Ferrari. "Design of multiplicative watermarking against covert attacks." In: *Proceedings of the 60th Conference on Decision and Control*. IEEE. 2021, pp. 4176–4181.

[88] C. Ganesh, A. Nitulescu, and E. Soria-Vazquez. "Rinocchio: SNARKs for ring arithmetic." In: *Cryptology ePrint Archive, Paper 2021/322* (2021).

[89] R. Gennaro and D. Wichs. "Fully homomorphic message authenticators." In: *Proceedings of the Conference on Theory and Application of Cryptology and Information Security*. Springer. 2013, pp. 301–320.

[90] C. Gentry. "Computing Arbitrary Functions of Encrypted Data." In: *Communications of the ACM* 22.11 (2010), pp. 612–613.

[91] C. Gentry. "Fully homomorphic encryption scheme using ideal lattices." In: *Proceedings of the 41st ACM Symposium on Theory of Computing*. Association for Computing Machinery, 2009, pp. 169–178.

[92] 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*. Springer. 2013, pp. 75–92.

[93] O. Goldreich. *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge University Press, 2004.

[94] O. Goldreich, S. Micali, and A. Wigderson. "How to play any mental game, or a completeness theorem for protocols with honest majority." In: *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*. 2019, pp. 307–328.

[95] S. Goldwasser and S. Micali. "Probabilistic encryption and how to play mental poker keeping secret all partial information." In: *Proceedings of the 14th ACM Symposium on Theory of Computing*. 1982, pp. 365–377.

[96] F. J. Gonzalez-Serrano, A. Amor-Martın, and J. Casamayon-Anton. "State estimation using an extended Kalman filter with privacy-protected observed inputs." In: *Workshop on Information Forensics and Security*. IEEE. 2014, pp. 54–59.

[97] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. "Maxout Networks." In: *Proceedings of the 30th International Conference on Machine Learning*. Vol. 28. 2013, pp. 1319–1327.

[98] S. Gorbunov, V. Vaikuntanathan, and D. Wichs. "Leveled fully homomorphic signatures from standard lattices." In: *Proceedings of the 47th Symposium on Theory of Computing*. ACM, 2015, pp. 469–477.

[99] A. Grancharova and T. A. Johansen. "Computation, approximation and stability of explicit feedback min–max nonlinear model predictive control." In: *Automatica* 45.5 (2009), pp. 1134–1143.

[100] A. Grancharova and T. A. Johansen. "Explicit solution of regulation control problems for nonlinear systems with quantized inputs." In: *Proceedings of the International Conference on Automatics and Informatics*. Citeseer. 2008.

[101] C. N. Hadjicostis. "Privary Preserving Distributed Average Consensus via Homomorphic Encryption." In: *Proceedings of the 57th Conference on Decision and Control*. IEEE. 2018, pp. 1259–1263.

[102] C. N. Hadjicostis and A. D. Domínguez-García. "Privacy-preserving distributed averaging via homomorphically encrypted ratio consensus." In: *IEEE Transactions on Automatic Control* 65.9 (2020), pp. 3887–3894.

[103] M. T. Hale and M. Egerstedt. "Differentially private cloud-based multi-agent optimization with constraints." In: *Proceedings of the American Control Conference*. 2015, pp. 1235–1240.

[104] S. Halevi and V. Shoup. *Design and implementation of HElib: A homomorphic encryption library*. Cryptology ePrint Archive, Paper 2020/1481. 2020.

[105] S. Han and G. J. Pappas. "Privacy in control and dynamical systems." In: *Annual Review of Control, Robotics, and Autonomous Systems* 1 (2018), pp. 309–332.

[106] S. Han, U. Topcu, and G. J. Pappas. "Differentially Private Distributed Constrained Optimization." In: *IEEE Transactions on Automatic Control* 62.1 (2017), pp. 50–64.

[107] G. H. Hardy and E. M. Wright. *An introduction to the theory of numbers*. 6th ed. Oxford University Press, 2008.

[108] M. U. Hassan, M. H. Rehmani, and J. Chen. "Differential privacy techniques for cyber physical systems: A survey." In: *IEEE Communications Surveys & Tutorials* 22.1 (2019), pp. 746–789.

[109] B. Hassibi and H. Vikalo. "On the expected complexity of integer least-squares problems." In: *Proceedings of the Conference on Acoustics, Speech, and Signal Processing*. IEEE. 2002.

[110] H. Hayati, C. Murguia, and N. van de Wouw. "Privacy-Preserving Federated Learning via System Immersion and Random Matrix Encryption." In: *arXiv preprint arXiv:2204.02497* (2022).

[111] H. Hayati, N. van de Wouw, and C. Murguia. "Privacy in Cloud Computing through Immersion-based Coding." In: *arXiv preprint arXiv:2403.04485* (2024).

[112] C. Hazay and Y. Lindell. *Efficient Secure Two-Party Protocols: Techniques and Constructions*. 1st. Berlin, Heidelberg: Springer-Verlag, 2010.

[113] A. B. Hempel, P. J. Goulart, and J. Lygeros. "Inverse Parametric Optimization with an Application to Hybrid System Control." In: *IEEE Transactions on Automatic Control* 60.4 (2015), pp. 1064–1069.

[114] M. Herceg, M. Kvasnica, C. Jones, and M. Morari. "Multi-Parametric Toolbox 3.0." In: *Proceedings of the European Control Conference*. 2013, pp. 502–510.

[115] M. Hertneck, J. Köhler, S. Trimpe, and F. Allgöwer. "Learning an approximate model predictive controller with guarantees." In: *IEEE Control Systems Letters* 2.3 (2018), pp. 543–548.

[116] S. Hong, J. H. Park, W. Cho, H. Choe, and J. H. Cheon. "Secure tumor classification by shallow neural network using homomorphic encryption." In: *BMC genomics* 23.1 (2022), pp. 1–19.

[117] R. Horst and N. V. Thoai. "DC Programming: Overview." In: *Journal of Optimization Theory and Applications* 103.1 (1999), pp. 1–43.

[118] Y. Huang, D. Evans, J. Katz, and L. Malka. "Faster Secure Two-Party Computation Using Garbled Circuits." In: *Proceedings of the 20th Conference on Security*. USENIX, 2011.

[119] Z. Huang, S. Mitra, and G. Dullerud. "Differentially private iterative synchronous consensus." In: *Workshop on privacy in the electronic society*. ACM. 2012, pp. 81–90.

[120] Z. Huang, S. Mitra, and N. Vaidya. "Differentially private distributed optimization." In: *Proceedings of the 16th International Conference on Distributed Computing and Networking*. 2015, pp. 1–10.

[121] R. Impagliazzo and S. Rudich. "Limits on the provable consequences of one-way permutations." In: *Proceedings of the 21st symposium on theory of computing*. 1989, pp. 44–61.

[122] A. Isaksson and S. Graebe. "Derivative filter is an integral part of PID design." In: *IEE Proceedings - Control Theory and Applications* 149.1 (2002), pp. 41–45.

[123] Y. Ishai, J. Kilian, K. Nissim, and E. Petrank. "Extending Oblivious Transactionsers Efficiently." In: *Crypto*. Springer. 2003, pp. 145–161.

[124] E. I. Jury. "On the roots of a real polynomial inside the unit circle and a stability criterion for linear discrete systems." In: *Proceedings of the 2nd International IFAC Congress on Automatic and Remote Control*. 1963, pp. 142–153.

[125] J. Katz and Y. Lindell. *Introduction to modern cryptography*. CRC press, 2020.

[126] M. Keller, E. Orsini, and P. Scholl. "MASCOT: faster malicious arithmetic secure computation with oblivious Transactionser." In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. 2016, pp. 830–842.

[127] M. Keller, V. Pastro, and D. Rotaru. "Overdrive: Making SPDZ Great Again." In: *Advances in Cryptology - EUROCRYPT*. Springer, 2018, pp. 158–189.

[128] A. Kim, A. Papadimitriou, and Y. Polyakov. "Approximate Homomorphic Encryption with Reduced Approximation Error." In: *Topics in Cryptology – CT-RSA*. Springer. 2022, pp. 120–144.

[129] A. Kim, Y. Polyakov, and V. Zucca. "Revisiting homomorphic encryption schemes for finite fields." In: *Advances in Cryptology – ASIACRYPT,* Springer. 2021, pp. 608–639.

[130] J. Kim, H. Shim, and K. Han. "Dynamic controller that operates over homomorphically encrypted data for infinite time horizon." In: *IEEE Transactions on Automatic Control* (2022).

[131] J. Kim et al. "Encrypting Controller using Fully Homomorphic Encryption for Security of Cyber-Physical Systems." In: *6th IFAC Workshop on Distributed Estimation and Control in Networked Systems*. 2016, pp. 175–180.

[132] J. Kim, F. Farokhi, I. Shames, and H. Shim. "Toward nonlinear dynamic control over encrypted data for infinite time horizon." In: *arXiv preprint arXiv:2110.06270* (2021).

[133] J. Kim and H. Shim. "Encrypted state estimation in networked control systems." In: *Proceedings of the 58th Conference on Decision and Control*. IEEE. 2019, pp. 7190–7195.

[134] J. Kim, H. Shim, and K. Han. "Dynamic Controller That Operates Over Homomorphically Encrypted Data for Infinite Time Horizon." In: *IEEE Transactions on Automatic Control* 68.2 (2023), pp. 660–672.

[135] 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: *Proceedings of the 60th Conference on Decision and Control*. IEEE. 2021, pp. 5614–5619.

[136] M. Kishida. "Encrypted Average Consensus with Quantized Control Law." In: *Proceedings of the 57th Conference on Decision and Control*. IEEE. 2018, pp. 5850–5856.

[137] W. Knowles, D. Prince, D. Hutchiso, J. F. P. Disso, and K. Jones. "A survey of cyber security management in industrial control systems." In: *International Journal of Critical Infrastructure Protection* 9 (2015), pp. 52–80.

[138] K. Kogiso and T. Fujita. "Cyber-Security Enhancement of Networked Control Systems using Homomorphic Encryption." In: *Proceedings of the 54th Conference on Decision and Control*. IEEE. 2015, pp. 6836–6843.

[139] J. Köhler et al. "Robust and optimal predictive control of the COVID-19 outbreak." In: *Annual Reviews in Control* 51 (2021), pp. 525–539.

[140] V. Kolesnikov and T. Schneider. "Improved Garbled Circuit: Free XOR Gates and Applications." In: *International Colloquium on Automata, Languages, and Programming*. Springer. 2008, pp. 486–498.

[141] P. J. Kootsookos, R. R. Bitmead, and M. Green. "The Nehari shuffle: FIR(q) filter design with guaranteed error bounds." In: *IEEE Transactions on signal processing* 40.8 (1992), pp. 1876–1883.

[142] A. Kripfganz and R. Schulze. "Piecewise affine functions as a difference of two convex functions." In: *Optimization* 18.1 (1987), pp. 23–29.

[143] L. Kronecker. "Zwei Sätze über Gleichungen mit ganzzahligen Coefficienten." In: *Journal für die reine und angewandte Mathematik* 1857.53 (1857), pp. 173–175.

[144] D. Lee, J. Kim, and H. Shim. "Distributed Aggregation over Homomorphically Encrypted Data under Switching Networks." In: *Proceedings of the 59th Conference on Decision and Control*. 2020, pp. 5495–5500.

[145] X. Lei, X. Liao, T. Huang, and F. Heriniaina. "Achieving security, robust cheating resistance, and high-efficiency for outsourcing large matrix multiplication computation to a malicious cloud." In: *Information sciences* 280 (2014), pp. 205–217.

[146] B. Li and D. Micciancio. "On the security of homomorphic encryption on approximate numbers." In: *Advances in Cryptology – EUROCRYPT 2021*. Springer. 2021, pp. 648–677.

[147] F. Li, B. Luo, and P. Liu. "Secure information aggregation for smart grids using homomorphic encryption." In: *First international conference on smart grid communications*. 2010, pp. 327–332.

[148] N. Li, N. Zhang, S. K. Das, and B. Thuraisingham. "Privacy preservation in wireless sensor networks: a state-of-the-art survey." In: *Ad Hoc Networks* 7.8 (2009), pp. 1501–1514.

[149] F. Lin, M. Fardad, and M. R. Jovanović. "Augmented Lagrangian Approach to Design of Structured Optimal State Feedback Gains." In: *IEEE Transactions on Automatic Control* 56.12 (2011), pp. 2923–2929.

[150] Y. Lindell. "How To Simulate It – A Tutorial On The Simulation Proof Technique." In: *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich* (2017), pp. 277–346.

[151] Y. Lindell. "Secure Multiparty Computation." In: *Communications of the ACM* (2020), pp. 86–96.

[152] Y. Lindell and B. Pinkas. "A proof of security of Yao's protocol for two-party computation." In: *Journal of cryptology* 22 (2009), pp. 161–188.

[153] O. Linschmann, S. Leonhardt, and C. Hoog Antink. "Model-based sensor fusion of multimodal cardiorespiratory signals using an unscented Kalman filter." In: *at-Automatisierungstechnik* 68.11 (2020), pp. 933–940.

[154] J.-L. Lions et al. *Ariane 5 flight 501 failure report by the inquiry board*. 1996.

[155] P. Longa and M. Naehrig. "Speeding up the number theoretic transform for faster ideal lattice-based cryptography." In: *Proceedings of the 15th Conference on Cryptology and Network Security*. Springer. 2016, pp. 124–139.

[156] W.-j. Lu, Z. Huang, C. Hong, Y. Ma, and H. Qu. "PEGASUS: bridging polynomial and non-polynomial evaluations in homomorphic encryption." In: *Symposium on Security and Privacy (SP)*. IEEE. 2021, pp. 1057–1073.

[157] V. Lyubashevsky, C. Peikert, and O. Regev. "A toolkit for ring-LWE cryptography." In: *Advances in Cryptology – EUROCRYPT*. Springer. 2013, pp. 35–54.

[158] V. Lyubashevsky, C. Peikert, and O. Regev. "On ideal lattices and learning with errors over rings." In: *Advances in Cryptology – EUROCRYPT 2010*. Springer. 2010, pp. 1–23.

[159] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. "TAG: A Tiny Aggregation Service for Ad-Hoc Sensor Networks." In: *5th Symposium on Operating Systems Design and Implementation (OSDI 02)*. 2002.

[160] D. Malkhi, N. Nisan, B. Pinkas, and Y. Sella. "Fairplay – A Secure Two-Party Computation System." In: *13th USENIX Security Symposium*. San Diego, CA, 2004.

[161] A. Maneesha and K. S. Swarup. "A survey on applications of Alternating Direction Method of Multipliers in smart power grids." In: *Renewable and Sustainable Energy Reviews* 152 (2021), p. 111687.

[162] 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." In: *IEEE Control Systems Letters* 7 (2023), pp. 395–400.

[163] C. Marcolla et al. "Survey on Fully Homomorphic Encryption, Theory, and Applications." In: *Proceedings of the IEEE* 110.10 (2022), pp. 1572–1609.

[164] D. Q. Mayne, J. B. Rawlings, C. Rao, and P. O. M. Scokaert. "Constrained model predictive control: Stability and optimality." In: *Automatica* 36 (2000), pp. 789–814.

[165] D. Q. Mayne, M. M. Seron, and S. V. Raković. "Robust model predictive control of constrained linear systems with bounded disturbances." In: *Automatica* 41 (2005), pp. 219–224.

[166] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert. "Constrained model predictive control: Stability and optimality." In: *Automatica* 36.6 (2000), pp. 789–814.

[167] D. Micciancio and Y. Polyakov. "Bootstrapping in FHEW-like cryptosystems." In: *9th on Workshop on Encrypted Computing & Applied Homomorphic Cryptography*. 2021, pp. 17–28.

[168] Y. Mo and R. M. Murray. "Privacy preserving average consensus." In: *IEEE Transactions on Automatic Control* 62.2 (2017), pp. 753–765.

[169] Y. Mo and B. Sinopoli. "Secure Control Against Replay Attacks." In: *Proceedings of the 47th Allerton Conference*. 2009, pp. 911–918.

[170] Y. Mo and B. Sinopoli. "Integrity attacks on cyber-physical systems." In: *Proceedings of the 1st International Conference on High Confidence Networked Systems*. 2012, pp. 47–54.

[171] P. Mohassel and Y. Zhang. "SecureML: A system for scalable privacy-preserving machine learning." In: *Symposium on Security and Privacy*. IEEE. 2017, pp. 19–38.

[172] C. Mouchet, J. Troncoso-Pastoriza, J.-P. Bossuat, and J.-P. Hubaux. "Multiparty homomorphic encryption from ring-learning-with-errors." In: *Proceedings on Privacy Enhancing Technologies* 2021 (2021), pp. 291–311.

[173] J.-M. Muller. *Elementary functions*. Springer, 2006.

[174] A. Muñoz, R. Rios, R. Román, and J. López. "A survey on the security of trusted execution environments." In: *Computers & Security* 129 (2023), p. 103180.

[175] C. Murguia, F. Farokhi, and I. Shames. "Secure and Private Implementation of Dynamic Controllers Using Semi-Homomorphic Encryption." In: *IEEE Transactions on Automatic Control* 65.9 (2020), pp. 3950–3957.

[176] M. Naor, B. Pinkas, and R. Sumner. "Privacy preserving auctions and mechanism design." In: *Proceedings of the 1st ACM Conference on Electronic Commerce*. 1999, pp. 129–139.

[177] A. M. Naseri, W. Lucia, and A. Youssef. "A Privacy Preserving Solution for Cloud-Enabled Set-Theoretic Model Predictive Control." In: *Proceedings of the European Control Conference*. IEEE. 2022, pp. 894–899.

[178] N. A. Nguyen, M. Gulan, S. Olaru, and P. Rodriguez-Ayerbe. "Convex lifting: Theory and control applications." In: *IEEE Transactions on Automatic Control* 63.5 (2017), pp. 1243–1258.

[179] E. Nozari, P. Tallapragada, and J. Cortés. "Differentially private average consensus: Obstructions, trade-offs, and optimal algorithm design." In: *Automatica* 81 (2017), pp. 221–231.

[180] E. Nozari, P. Tallapragada, and J. Cortés. "Differentially private distributed convex optimization via objective perturbation." In: *Proceedings of the American Control Conference*. IEEE. 2016, pp. 2061–2066.

[181] K.-K. Oh, M.-C. Park, and H.-S. Ahn. "A survey of multi-agent formation control." In: *Automatica* 53 (2015), pp. 424–440.

[182] P. Paillier. "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes." In: *Advances in Cryptology - Eurocrypt*. Springer, 1999, pp. 223–238.

[183] *PALISADE Lattice Cryptography Library (release 1.7.4)*. `https://palisade-crypto.org/`. Polyakov, Y. and Rohloff, K. and Ryan, G. W. 2020.

[184] B. Parno, J. Howell, C. Gentry, and M. Raykova. "Pinocchio: Nearly Practical Verifiable Computation." In: *Communications of the ACM* 59.2 (2016), pp. 103–112.

[185] F. Pasqualetti, F. Dörfler, and F. Bullo. "Attack Detection and Identification in Cyber-Physical Systems." In: *IEEE Transactions on Automatic Control* 58.11 (2013), pp. 2715–2729.

[186] F. Pasqualetti, D. Borra, and F. Bullo. "Consensus networks over finite fields." In: *Automatica* 50.2 (2014), pp. 349–358.

[187] B. Pinkas, T. Schneider, N. Smart, and S. Williams. "Secure two-party computation is practical." In: *Advances in Cryptology - Asiacrypt*. Springer, 2009, pp. 250–267.

[188] J. M. Pollard. "The fast Fourier Transactionsorm in a finite field." In: *Mathematics of computation* 25.114 (1971), pp. 365–374.

[189] W. H. Press. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.

[190] M. O. Rabin. "How To Exchange Secrets with Oblivious Transfer." In: *IACR Cryptology ePrint Archive* (2005).

[191] M. O. Rabin. "How to exchange secrets with oblivious Transactionser." In: *Cryptology ePrint Archive* (2005).

[192] D. Rathee, T. Schneider, and K. K. Shukla. "Improved Multiplication Triple Generation over Rings via RLWE-Based AHE." In: *International Conference on Cryptology and Network Security*. Springer. 2019, pp. 347–359.

[193] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. 2nd Edition. Vol. 2. Nob Hill Publishing, 2017.

[194] O. Regev. "On Lattices, Learning with Errors, Random Linear Codes, and Cryptography." In: *Proceedings of the 37th Symposium on Theory of Computing*. ACM, 2005, pp. 84–93.

[195] O. Regev. "On Lattices, Learning with Errors, Random Linear Codes, and Cryptography." In: *Journal of the ACM* 56.6 (2009), pp. 1–40.

[196] O. Regev. "The Learning with Errors Problem." In: *Proceedings of the 25th Conference on Computational Complexity*. 2010, pp. 191–204.

[197] E. Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.3*. RFC 8446. 2018.

[198] J. W. Rittinghouse and J. F. Ransome. *Cloud computing: implementation, management, and security*. CRC press, 2017.

[199] R. L. Rivest, A. Shamir, and L. Adleman. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems." In: *Communications of the ACM* 21.2 (1978), pp. 120–126.

[200] R. L. Rivest, L. Adleman, M. L. Dertouzos, et al. "On data banks and privacy homomorphisms." In: *Foundations of secure computation* 4.11 (1978), pp. 169–180.

[201] M. Ruan, H. Gao, and Y. Wang. "Secure and Privacy-Preserving Consensus." In: *IEEE Transactions of Automatic Control* (2019).

[202] M. Ruan, H. Gao, and Y. Wang. "Secure and privacy-preserving consensus." In: *IEEE Transactions on Automatic Control* 64.10 (2019), pp. 4035–4049.

[203] M. Sabt, M. Achemlal, and A. Bouabdallah. "Trusted execution environment: What it is, and what it is not." In: *2015 IEEE Trustcom/BigDataSE/Ispa*. 2015, pp. 57–64.

[204] J. Sándor, D. S. Mitrinovic, and B. Crstici. *Handbook of number theory I*. Springer, 2005.

[205] S. Schlor, M. Hertneck, S. Wildhagen, and F. Allgöwer. "Multi-party computation enables secure polynomial control based solely on secret-sharing." In: *Proceedings of the 60th Conference on Decision and Control*. IEEE. 2022, pp. 4882–4887.

[206] G. Schulz. "Iterative Berechung der reziproken Matrix." In: *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik* 13.1 (1933), pp. 57–59.

[207] M. Schulze Darup. "Encrypted Model Predictive Control in the Cloud." In: *Privacy in Dynamical Systems*. Springer, 2020, pp. 231–265.

[208] M. Schulze Darup. "Encrypted MPC based on ADMM real-time iterations." In: *Proceedings of 21th IFAC World Congress*. 2020.

[209] M. Schulze Darup, A. B. Alexandru, D. E. Quevedo, and G. J. Pappas. "Encrypted control for networked systems - An illustrative introduction and current challenges." In: *IEEE Control Systems Magazine* (2021).

[210] M. Schulze Darup, A. B. Alexandru, D. E. Quevedo, and G. J. Pappas. "Encrypted Control for Networked Systems: An Illustrative Introduction and Current Challenges." In: *IEEE Control Systems Magazine* 41.3 (2021), pp. 58–78.

[211] M. Schulze Darup, A. Redder, and D. E. Quevedo. "Encrypted cloud-based MPC for linear systems with input constraints." In: *Proceedings of the 6th IFAC Conference on Nonlinear Model Predictive Control NMPC 2018*. 2018, pp. 635–642.

[212] M. Schulze Darup, A. Redder, I. Shames, F. Farokhi, and D. Quevedo. "Towards Encrypted MPC for Linear Constrained Systems." In: *IEEE Control Systems Letters* 2.2 (2018), pp. 195–200.

[213] M. Schulze Darup and D. Teichrib. "Efficient computation of RPI sets for tube-based robust MPC." In: *Proceedings of the 18th European Control Conference*. 2019, pp. 325–330.

[214] M. Schulze Darup. "Encrypted polynomial control based on tailored two-party computation." In: *International Journal of Robust and Nonlinear Control* 30.11 (2020), pp. 4168–4187.

[215] M. Schulze Darup, A. Redder, and D. E. Quevedo. "Encrypted cooperative control based on structured feedback." In: *IEEE Control Systems Letters* 3.1 (2019), pp. 37–42.

[216] A. Shamir. "How to share a secret." In: *Communications of the ACM* 22.11 (1979), pp. 612–613.

[217] Z. Shan, K. Ren, M. Blanton, and C. Wang. "Practical secure computation outsourcing: A survey." In: *ACM Computing Surveys* 51.2 (2018), pp. 1–40.

[218] Y. Shoukry et al. "Privacy-aware quadratic optimization using partially homomorphic encryption." In: *Proceedings of the 55th Conference on Decision and Control*. IEEE. 2016, pp. 5053–5058.

[219] V. Shoup. *A computational introduction to number theory and algebra*. Cambridge university press, 2009.

[220] R. S. Smith. "A decoupled feedback structure for covertly appropriating networked control systems." In: *IFAC Proceedings Volumes* 44.1 (2011), pp. 90–95.

[221] N. I. of Standards and Technology. "Recommendation for Key Management." In: *NIST Special Publication 800-57 Part 1*. 2016.

[222] P. Stobbe, T. Keijzer, and R. M. Ferrari. "A Fully Homomorphic Encryption Scheme for Real-Time Safe Control." In: *2022 Proceedings of the Conference on Decision and Control*. 2022, pp. 2911–2916.

[223] R. Strässer, S. Schlor, and F. Allgöwer. "Decrypting nonlinearity: Koopman interpretation and analysis of cryptosystems." In: *arXiv preprint arXiv:2311.12714* (2023).

[224] B. E. Strom et al. "Mitre att&ck: Design and philosophy." In: *Technical report*. The MITRE Corporation, 2018.

[225] J. Suh and T. Tanaka. "Encrypted Value Iteration and Temporal Difference Learning over Leveled Homomorphic Encryption." In: *Proceedings of the American control conference*. IEEE. 2021, pp. 2555–2561.

[226] J. Suh and T. Tanaka. "SARSA(0) Reinforcement Learning over Fully Homomorphic Encryption." In: *Proceedings of the International Symposium on Control Systems*. IEEE. 2021, pp. 1–7.

[227] A. Sultangazin and P. Tabuada. "Symmetries and isomorphisms for privacy in control over the cloud." In: *IEEE Transactions on Automatic Control* 66.2 (2020), pp. 538–549.

[228] X. Sun et al. "A Survey on Zero-Knowledge Proof in Blockchain." In: *IEEE Network* 35.4 (2021), pp. 198–205.

[229] M. S. Tavazoei. "Non-minimality of the realizations and possessing state matrices with integer elements in linear discrete-time controllers." In: *IEEE Transactions on Automatic Control* (2022).

[230] M. S. Tavazoei. "Pisot number-based discrete-time controllers with integer state matrices to ensure monotonic closed-loop step responses." In: *IEEE Transactions on Automatic Control* (2023).

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

[232] N. Tian, Q. Guo, H. Sun, and X. Zhou. "Fully privacy-preserving distributed optimization in power systems based on secret sharing." In: *iEnergy* 1.3 (2022), pp. 351–362.

[233] K. Tjell and R. Wisniewski. "Privacy Preserving Optimization with Functionality Guarantee in the case of Attacks." In: *Proceedings of the 58th Conference on Decision and Control*. 2019.

[234] K. Tjell and R. Wisniewski. "Private aggregation with application to distributed optimization." In: *IEEE Control Systems Letters* 5.5 (2021), pp. 1591–1596.

[235] K. Tjell, I. Cascudo, and R. Wisniewski. "Privacy Preserving Recursive Least Squares Solutions." In: *Proceedings of the 18th European Control Conference*. 2019, pp. 3490–3495.

[236] K. Tjell and R. Wisniewski. "Privacy in distributed computations based on real number secret sharing." In: *arXiv preprint arXiv:2107.00911* (2021).

[237] K. Tjell and R. Wisniewski. "Privacy Preservation in Distributed Optimization via Dual Decomposition and ADMM." In: *Proceedings of the 58th Conference on Decision and Control*. IEEE. 2019, pp. 7203–7208.

[238] P. Tøndel, T. A. Johansen, and A. Bemporad. "Computation and approximation of piecewise affine control laws via binary search trees." In: *Proceedings of the 41st Conference on Decision and Control*. 2002, pp. 3144–3149.

[239] D. I. Urbina et al. "Limiting the impact of stealthy attacks on industrial control systems." In: *Proceedings of the ACM SIGSAC conference on computer and communications security*. 2016, pp. 1092–1105.

[240] J. Vaidya. "Privacy-preserving linear programming." In: *Proceedings of the ACM symposium on Applied Computing*. 2009, pp. 2002–2007.

[241] R. Van Parys and G. Pipeleers. "Distributed MPC for multi-vehicle systems moving in formation." In: *Robotics and Autonomous Systems* 97 (2017), pp. 144–152.

[242] C. Wang, K. Ren, and J. Wang. "Secure and practical outsourcing of linear programming in cloud computing." In: *Proceedings of the Infocom*. IEEE. 2011, pp. 820–828.

[243] Y. Wang, Z. Huang, S. Mitra, and G. E. Dullerud. "Differential privacy in linear distributed control systems: Entropy minimizing mechanisms and performance tradeoffs." In: *IEEE Transactions on Control of Network Systems* 4.1 (2017), pp. 118–130.

[244] P. C. Weeraddana, G. Athanasiou, C. Fischione, and J. S. Baras. "Per-se privacy preserving solution methods based on optimization." In: *Proceedings of the 52nd Conference on Decision and Control*. 2013, pp. 206–211.

[245] K. Wei et al. "Federated learning with differential privacy: Algorithms and performance analysis." In: *IEEE Transactions on Information Forensics and Security* 15 (2020), pp. 3454–3469.

[246] L. Xiao and S. Boyd. "Fast linear iterations for distributed averaging." In: *Systems & Control Letters* 53.1 (2004), pp. 65–78.

[247] Z. Xu and Q. Zhu. "Secure and resilient control design for cloud enabled networked control systems." In: *1st ACM workshop on cyber-physical systems-security and/or privacy*. ACM. 2015, pp. 31–42.

[248] Y. Yamamoto, B. D. Anderson, M. Nagahara, and Y. Koyanagi. "Optimizing FIR approximation for discrete-time IIR filters." In: *IEEE Signal Processing Letters* 10.9 (2003), pp. 273–276.

[249] A. C. Yao. "Protocols for secure computations." In: *23rd annual symposium on foundations of computer science*. IEEE. 1982, pp. 160–164.

[250] S. Zahur, M. Rosulek, and D. Evans. "Two Halves Make a Whole: Reducing Data Transfer in Garbled Circuits using Half Gtes." In: *Advances in Cryptology – EUROCRYP*. Springer. 2015, pp. 220–250.

[251] C. Zhang, M. Ahmad, and Y. Wang. "ADMM Based Privacy-Preserving Decentralized Optimization." In: *IEEE Transactions on Information Forensics and Security* 14.3 (2019), pp. 565–580.

[252] K. Zhang, Z. Li, Y. Wang, and N. Li. "Privacy-preserved nonlinear cloud-based model predictive control via affine masking." In: *arXiv preprint arXiv:2112.10625* (2021).

[253] K. Zhang, Z. Li, Y. Wang, A. Louati, and J. Chen. "Privacy-preserving dynamic average consensus via state decomposition: Case study on multi-robot formation control." In: *Automatica* 139 (2022).

Publications of the Author

[P1] J. Adamek, P. Binfet, N. Schlüter, and M. Schulze Darup. "Encrypted system identification as-a-service via reliable encrypted matrix inversion." In: *Proceedings of the 63rd Conference on Decision and Control*. IEEE. 2024, pp. 4582–4588.

[P2] J. Adamek, N. Schlüter, and M. Schulze Darup. "On the design of stabilizing FIR controllers." In: *Proceedings of the 10th Conference on Control, Decision and Information Technologies*. IEEE. 2024, pp. 2037–2042.

[P3] P. Binfet, J. Adamek, N. Schlüter, and M. Schulze Darup. "Towards privacy-preserving cooperative control via encrypted distributed optimization." In: *at - Automatisierungstechnik* 71.9 (2023), pp. 736–747.

[P4] P. Binfet, N. Schlüter, and M. Schulze Darup. "On the security of randomly transformed quadratic programs for privacy-preserving cloud-based control." In: *Proceedings of the 62nd Conference on Decision and Control*. IEEE. 2023, pp. 3872–3877.

[P5] J. von der Heyden, N. Schlüter, P. Binfet, M. Asman, M. Zdrallek, T. Jager, and M. Schulze Darup. "Privacy-Preserving Power Flow Analysis via Secure Multi-Party Computation." In: *IEEE Transactions on Smart Grid* 16.1 (2025), pp. 344–355.

[P6] T. Hosseinalizadeh, N. Schlüter, M. Schulze Darup, and N. Monshizadeh. *Privacy Analysis of Affine Transformations in Cloud-based MPC: Vulnerability to Side-knowledge.* arXiv:1812.04168v2. 2024.

[P7] M. Klädtke, D. Teichrib, N. Schlüter, and M. Schulze Darup. "A deterministic view on explicit data-driven (M)PC." In: *Proceedings of the 61st Conference on Decision and Control*. 2022, pp. 499–504.

[P8] N. Schlüter, M. Neuhaus, and M. Schulze Darup. "Encrypted dynamic control with unlimited operating time via FIR filters." In: *Proceedings of the 19th European Control Conference*. IEEE. 2021, pp. 947–952.

[P9] N. Schlüter and M. Schulze Darup. "Encrypted explicit MPC based on two-party computation and convex controller decomposition." In: *Proceedings of the 59th Conference on Decision and Control*. IEEE. 2020, pp. 5469–5476.

[P10] N. Schlüter and M. Schulze Darup. "Novel convex decomposition of piecewise affine functions." In: *Late breaking result to the 21st IFAC World Congress*. 2020.

[P11] N. Schlüter, P. Binfet, J. Kim, and M. Schulze Darup. "Encrypted distributed state estimation via affine averaging." In: *Proceedings of the 61st Conference on Decision and Control*. IEEE. 2022, pp. 7754–7761.

[P12] N. Schlüter, P. Binfet, and M. Schulze Darup. "A brief survey on encrypted control: From the first to the second generation and beyond." In: *Annual Reviews in Control* 56 (2023).

[P13] N. Schlüter, P. Binfet, and M. Schulze Darup. "Cryptanalysis of Random Affine Transformations for Encrypted Control." In: *Proceedings of the 22nd IFAC World Congress*. 2023, pp. 12031–12038.

[P14] N. Schlüter, J. Kim, and M. Schulze Darup. "A code-driven tutorial on encrypted control: From pioneering realizations to modern implementations." In: *Proceedings of the 22nd European Control Conference*. IEEE. 2024, pp. 914–920.

[P15] N. Schlüter, M. Neuhaus, and M. Schulze Darup. "Encrypted extremum seeking for privacy-preserving PID tuning as-a-Service." In: *Proceedings of the 20th European Control Conference*. IEEE. 2022, pp. 1288–1293.

[P16] N. Schlüter and M. Schulze Darup. "On the Stability of Linear Dynamic Controllers With Integer Coefficients." In: *IEEE Transactions on Automatic Control* 67.10 (2021), pp. 5610–5613.

[P17] K. Tjell, N. Schlüter, P. Binfet, and M. Schulze Darup. "Secure learning-based MPC via garbled circuit." In: *Proceedings of the 60th Conference on Decision and Control*. 2021, pp. 4907–4914.