Universidade Vigo

HOMOMORPHIC LATTICE CRYPTOSYSTEMS FOR SECURE SIGNAL PROCESSING

Alberto Pedrouzo Ulloa | PhD Thesis

Supervised by:
Juan Ramón Troncoso Pastoriza
Fernando Pérez González

UniversidadeVigo

International Doctoral School

Alberto Pedrouzo Ulloa

DOCTORAL DISSERTATION

Homomorphic Lattice Cryptosystems for Secure Signal Processing

Supervised by:

Juán Ramón Troncoso Pastoriza

Fernando Pérez González

2020

International mention

This work was supported in part by the Spanish Ministry of Economy and Competitiveness and the European Regional Development Fund (ERDF) under projects TACTICA, COMPASS (TEC2013-47020-C2-1-R), WINTER (TEC2016-76409-C2-2-R) and COMONSENS (TEC2015-69648-REDC), in part by the FPI grant (BES-2014-069018), in part by the Galician Regional Government and ERDF under projects Agrupación Estratéxica Consolidada accreditation 2016-2019 and Grupo de Referencia ED431C2017/53 and AtlantTIC, and in part by the EU H2020 Framework Programme through the Project WITDOM, with Project No. 644371.

European Union
European Regional
Development Fund

GOBIERNO
DE ESPAÑA

MINISTERIO
DE EDUCACIÓN, CULTURA
Y DEPORTE

XUNTA
DE GALICIA

Abstract

Signal processing has become ubiquitous in our daily lives, being present in everyday digital appliances and applications. Actually, although apparently it is hard to be aware of its presence, it does have a great influence on our everyday life and the list of applications and science fields which make use of signal processing tools is accordingly huge: it encompasses communication and entertainment technologies, from speech and audio processing to image and video analysis (e.g., biometric processing of faces, fingerprints, iris, etc.), with a strong impact on emerging applications such as smart grids, autonomous driving, tele-diagnosis and analysis of medical signals (like Electrocardiograms or DNA), among others.

This is especially relevant because, although in our everyday life we are not really conscious of the possible risks, many of the previous scenarios involve the use of very privacy-sensitive data (e.g., a service which depends on the use of personal information). This becomes even worse in many of the most prominent signal processing applications, where the involved signals have to be processed by untrusted parties (i.e., the service provider requires the use of the personal information for the correct operation of the service), and the user must trust the service provider.

The field of Secure Signal Processing (SSP) was born to address these challenges, by devising efficient solutions stemming from the collaborative efforts of cryptography and signal processing. Due to its inherent multidisciplinary grounds, it can effectively combine and take advantage of the advances and technologies from the two disciplines. Additionally, numerous applications have been already proposed based on the use of different cryptographic techniques.

This thesis proposes novel methods for privacy protection when dealing with highly privacy-sensitive signals in untrustworthy environments. With this goal in mind, the thesis was originally motivated by the following two research lines: (1) privacy protection when dealing with multidimensional signals, and (2) design of new primitives and protocols for encrypted signal processing.

In particular, the work presented in this thesis introduces a secure framework for outsourced and unattended (multidimensional) signal processing; that is, the proposed solutions do not need the intervention of the secret key owner in the middle of the process.

The contributions are numerous and their outcomes range from low-level cryptographic primitives to more concrete high-level practical applications. Next, we briefly enumerate the main contributions:

(1) We formalize a lattice hard problem denoted as multivariate RLWE (Ring Learning with Errors). Due to its particular structure, it is especially useful to work with multidimensional signals. It also brings about efficiency improvements on current RLWE-based cryptographic primitives.

(2) Building on modern lattice-based primitives, we present a toolbox for unattended secure

signal processing (e.g., filtering, generalized convolutions, error correcting codes or matrix-based processing, among others).

(3) We exemplify the use of the proposed tools on several concrete signal processing scenarios (going from genomics to multimedia applications), where we face the additional difficulties which arise in each specific application due to the nature of the used signals.

Acknowledgements

Me encuentro ante la última etapa para terminar la escritura de la tesis. Aunque todos los capítulos están ya acabados desde hace un tiempo, esta parte todavía se me resiste, y a pesar de pretender escribirla durante las vacaciones, tras el paso de estas he vuelto y todavía está en blanco. Así que, aunque suene a tópico, no es hasta ahora que soy realmente consciente de la dificultad de *rellenar* estas hojas.

No voy a decir que voy a ser lo más conciso posible porque el número de personas a las que tengo que dar las gracias es muy grande. Así que pido perdón por adelantado si me olvido de alguien.

A los primeros que quiero agradecer su ayuda es a mis directores de tesis, Fernando y Juan. Ambos me han guiado durante todo el proceso que ya abarca ahora un poco más de un total de cinco años, que se dice pronto. Durante este periodo, su confianza y apoyo ha sido fundamental para empezar a investigar, y desde luego, para afrontar no sólo las primeras publicaciones rechazadas, sino también las segundas, terceras y las que todavía llegan a día de hoy. De Fernando quiero agradecer sobre todo sus muchos y acertados consejos, y su habilidad para poner en perspectiva todas las situaciones[1].

Respecto a Juan necesito un párrafo entero aparte. Por suerte (o por desgracia, que nunca se sabe[2]) él ha sido la figura fundamental que me introdujo en la temática en la que encaja la tesis (por favor, remítanle a él las quejas). De él tengo que remarcar no sólo su gran profesionalidad, sino también las innumerables horas que ha dedicado a lo largo de estos años a escuchar, aconsejar, asesorar, corregir, proponer, revisar, corregir, revisar, corregir... (léase como un bucle hasta el agotamiento del lector), y en definitiva ayudar a la exitosa realización de esta tesis (al principio en persona, y hacia el final con montones y montones de reuniones por skype, slack o la aplicación de moda que hiciera falta[3]).

I also want to thank the people who accepted to be part of the committee of this thesis. This includes Alessandro, Mariya, Mauro, Nicolas, Pedro and Rebeca.[4]

También tengo mucho que agradecer a Carmela, Dani y Gabi por la rápida elaboración de los informes pese a la “poca” (por usar un bonito eufemismo) antelación por mi parte en avisarles.

En un lugar del campus, de cuyo nombre no quiero acordarme, no ha mucho tiempo que se encuentra la eido. No puedo más que agradecer enormemente cada una de las quijotescas gestiones que tan acertadamente me ha requerido y que han sido “necesarias” para el correcto depósito.

Durante el transcurso de la tesis he tenido la oportunidad de poder impartir docencia en algunos

[1] Merece la pena mencionar que es uno de los mayores expertos en el campo tropical.

[2] Sobre todo para las pobres personas que lean este documento entero.

[3] Esto incluye por supuesto una cantidad no despreciable de horas en las que sólo me quejo ~~de los revisores~~.

[4] I am sorry in advance for the size of this document.

laboratorios de TTRS. Esta experiencia ha resultado ser muy enriquecedora (seguramente no tanto para los alumnos) y quiero agradecer a los profesores de la asignatura por su ayuda: Eduardo, Isasi, José Luis Rodríguez, Óscar, Pedro, Roberto y Yolanda. Mención especial merece Roberto por mis periódicas consultas al acercarse las vacaciones, y desde luego, Eduardo y Óscar por acogerme tan amablemente en sus laboratorios y por la paciencia que han tenido conmigo. No puedo olvidarme tampoco de todos los estudiantes que me han tenido que sufrir en las clases.

Todo este proceso no se limita a la tesis, y puedo decir que empezó mucho antes al venir a Vigo a estudiar (allá por el año 2008 con tan sólo diecisiete añitos). Por lo tanto, quiero mencionar también a otros profesores de la escuela que a día de hoy todavía se interesan en mis avances desde el momento en el que entré en la carrera. Esto incluye a Carlos, al que le pido perdón por llegar siempre tarde a los laboratorios de TDIX. También a Artemio y José Luis Alba, mis tutores de PFC y también de proyecto en el máster. Aunque ahora queda muy lejano, los inicios compaginando el trabajo en el laboratorio junto con el máster no fueron nada sencillos. No me quiero olvidar de Eduardo Liz, al que considero uno de los mejores docentes de la escuela.

A realización desta tese de doutoramento resultou converterse nunha viaxe moi instrutiva, tanto no plano persoal como nun aspecto máis terreal andando polo mundo adiante, xa que tiven a oportunidade de coñecer outras culturas e visitar moitos lugares ós que probablemente nunca tería ido. Din que o importante dunha viaxe non é nin o principio nin o final (que tamén eh!) senón o camiño percorrido.

Podo dicir sen ningún medo a equivocarme que tiven a gran sorte de compartir esta viaxe moi ben rodeado no día a día. Falo por suposto dos membros do laboratorio A515 (antigamente coñecido como TSC-5) que axudaron (e axudan) a converter este traballo en algo moito máis divertido e ameno, e dos que a día de hoxe vexo como unha pequeniña familia. David, compañeiro anual de carreiras no cuvi, agradézoche o teu constante interese, e dende logo é moito de valorar que é a única persoa que a día de hoxe pon algo de cordura no noso pequeno circo. Miguel, agradézoche moito a túa sinceridade sen censuras (ti é-la voz do pobo), dende logo desfruto moito dos teus *interesantísimos* debates con Simón, con David, con Gabi e en definitiva con calquera que se atreva a enfrontarse a ti (incluso co canadiense no tapiz); non deixes que ninguén infravalore os teus *femtopolvos*[5]. Simón, agradézoche terme ensinado tanto acerca da fauna urbana, da constante vixilancia da orde no lab, e por suposto, por ternos ó día dos novos éxitos musicais de recoñecidos artistas internacionais como Leticia Sabater (quédanos o muro e skype como testigo das túas dotes artísticas). Tamén me quero lembrar daqueles membros que agora non están no laboratorio, pero que formaron parte de momentos moi memorables nel. Gabi de novo, desfrutei moito das túas numerosas e variadas ensinanzas en todo o que respecta á historia e novos vocábulos, como o tal "presunto"; teño que recoñocer que o laboratorio notou moito a túa ausencia (sobre todo cando había que poñer a MM no seu sitio). Iria, é de remarcar que é a única integrante do TSC-5 que conseguiu acadar a nota máxima na escala masciopíntica, pese a todas as empanadas que levei ó laboratorio, ningunha conseguiu acadar unha nota comparable o teu sushi. Jareño, que dicir deste home, coido que é imposible incluir en tan pouco espacio moitas das anécdotas vividas con el, pero non podo evitar recordar perfectamente a súa persecución incansable en Tenerife[6]. Dende logo agradézolle moito o ambiente ameno de traballo (coido que ás veces chegando a ser superlativamente ameno), aínda que todavía lle gardo rencor por posuír certa camiseta. Juan de novo, outro membro fundamental do TSC-5, no laboratorio aínda se recorda a hora cero, esa singularidade no tempo na que deixaches de vixiar os servidores; nunca tal caos presenciara na miña vida. Por último, aínda que eu non cheguei a convivir con el no lab tamén quero mencionar a

[5] Dise dos cefalópodos moi pequenos.

[6] Algo semellante a un guepardo na sabana, o que quizais poderíamos denominar como un gran felino. Para describilo diría: "chóvechelle un pouco no tellado".

Pedro, que ademais de todo o xa dito, proporcionoume moi bos consellos para adestrar o maratón.

Dentro do GPSC, non me podo esquecer de Carmen Touriño, sempre disposta a axudar. Tamén están as máis recentes adquisicións ó laboratorio como Lois, Pablo, Samuel e Betty, xunto con outros compañeiros que tivemos como Laura, Mina, Patricia e Vlad. Ademais, tamén tivemos moitos compañeiros que fixeron estadías con nós como Michael, Omar, Reza, Benedetta, Martijn, Aurélien, Matthieu, Andrea e Andrea (a orde non altera o resultado final do produto). Inclúo tamén os demáis compañeiros da periferia do TSC-5 (sexa gradiant ou doutros laboratorios coñecidos) que participaron nos *crossover* realizados nas ceas de "traballo": Dani, Eduardo, Fátima, Gonzalo, Luis, Magui, Marta, Paula, Pedro, Rocío (estou moi seguro de que me esquezo dalgún...). É salientábel que anos despois do "incidente", aínda *temo* atoparme cun peluco ó rebuscar nas miñas follas do lab.

I would also like to thank Claudio Orlandi and Ivan Damgård, who very kindly welcome me at the Crypto group in Aarhus. Of course this also includes the rest of members and visitors of the group: Anders, Chaya, Daniel Escudero, Daniel Tschudi, Helene, Ignacio Maciej, Luisa, Mark, Michael, Peter, Prastudy, Rafael, Rebekah, Sabine, Sat, Tobias, Xinze, Yuanxi and Zahra (it is very probable I have forgotten someone, I am very sorry if that is the case). I learnt a lot and really enjoyed my stay there. In particular, I would like to thank Daniel Escudero for some very interesting discussions, and also Anders[7] and Yuanxi for our very long and funny talks about many random topics after lunch. Finally, I cannot forget Malene who always very efficiently helped me with any administrative issue.

No me puedo tampoco olvidar de todos los amigos y compañeros que he conocido desde que entré en teleco (con algunos he compartido una para nada desdeñable cantidad de horas). Toni, la primera persona que conocí al entrar en la escuela y que ha sido siempre tan buen compañero (aunque es importante no tocarle el coche). Pepe, espero que te quedaras contento, esa empanada conseguida a base de insistencia no te la quita nadie. También está David, con el que he compartido muchas horas de ping pong en la residencia, y luego, ya como compañero de piso, muchas enseñanzas acerca del manejo de drones con dos hélices y cursos intensivos de soldado de telefonillo[8]. Gabriel, con el que he compartido numerosas conversaciones acerca de las alpacas y demás mamíferos rumiantes, sus comportamientos sociales y sus derivados. Javi, ese Don Quijote del siglo XXI sin Sancho Panza (o tal vez sí), que en vez de molinos se enfrenta a *millimeter wave MIMO systems* . Mateo, con el que siempre es muy divertido subir en bus. Jorge, de vez en cuando aún me recuerdan nuestras discusiones gritando sentados en las últimas filas de clase. Natalia, de la que siempre puedes aprender algo nuevo acerca de Japón (¡y de más temas, por supuesto!); aunque para ello tengamos que coincidir[9]. Anxo, que sempre está disposto a axudar, e ó igual ca min, tamén está pronto a acabar. Rosalía, a la que agradezco sinceramente que siempre haya sido tan atenta y tan cercana. Grego, siempre es un honor escuchar tus enseñanzas. Pablo, al que le debo el conocer lo que es un diagrama de vela. Víctor, con el que he compartido muchísimas horas en clases de inglés; recuerdo especialmente una terrible tarde de Viernes y las peleas constantes con la sevillana. Por supuesto Fran, con el que he compartido la mayor parte de mi experiencia en teleco: primero en la carrera, después en el máster y ahora finalmente durante el doctorado. Siempre ha sido un gran amigo dispuesto a ayudar y escuchar en todo lo posible (aunque verlo con Grego me da miedo en ocasiones[10]). Tengo también que agradecerle su importante contribución a la parte

[7] I regret to say that "Hvordan har du det?" is the only thing in Danish I can remember and pronounce more or less "well".

[8] Esto me recuerda a mis otros excompañeros de piso: Víctor, siempre estuviste un poco ausente. Alejandro, a ver si quedamos para jugar al squash alguna vez.

[9] Agradezamos ó patrón dos cegos, por iguala-los días e as noites.

[10] De verdad espero que consigáis acercaros, aunque sea sólo un poco, a terminar algún tipo de proyecto alguna vez.

buena de la portada. Tampoco me puedo olvidar de Noelia, que fue la que me animó realmente a meterme en esta aventura y a la que tengo que agradecer muchas cosas.

No me quiero olvidar de otros compañeros pertenecientes al sector energía-industrial-minas-sur de ingeniería que han amenizado las horas del almuerzo. Ana, que espero destine los *presupuestos* a comprar algún tipo de aire acondicionado. Los Javis, a uno de ellos lo considero una persona increíblemente optimista. Sandra, siempre me ha parecido muy injusto que tus amigas te obligaran a ir al Castro (estando *un poco* nebuloso). No te preocupes, la hemeroteca dejará constancia de ello.

Tamén teño moito que agradecer ós colegas de toda a vida de Carballiño. É de destacar que a pesar dos meses que botáramos sen vernos, cada vez que nos xuntamos é como se non pasara o tempo. Non os incluirei a todos pero si que quero agradecer especialmente ós que me aguantaron "algunhas" veces falando ou queixándome da tese, espero non esquecer a ninguén. Iván e Marcos, cos que cheguei a convivir uns cantos anos en Vigo dos que dende logo gardo moi bo recordo. Tamén estivo un ano con nós Álvaro, aínda me lembro do día do bingo e algunhas cantas máis (unha ten que ver co cerraxeiro); o que me recorda tamén a Carla, Ciri, Laura, Noemi e Zaka, e tamén Samuel por unha cama rota. Toupa, co que tiven moitísimas charlas e do que recibín moitos consellos. Tizón, ó que lle agradezo moito o seu interese polas miñas aventuras. Fran, ó que pola distancia vexo moi de cando en vez, pero co que sempre remato tendo moi longas conversacións nos lugares e nas datas máis inoportunas. Por suposto, tamén a Rubén e Juán Marcos por moitas horas invertidas no *bar*.

Deixo para o final as persoas máis importantes. Teño que por suposto agradecer a tódolos membros da miña familia que tanto se preocupan por min e que tanto me teñen axudado ó longo destes anos. Ós meus pais, sen a vosa axuda nunca tería chegado ata aquí, sempre estivestedes atentos a calquera necesidade que tivera dende pequeno, e sei que sodes os que máis orgullosos e preocupados estaredes por min. Moitas gracias por todo. Á miña irmá, que é case coma a miña segunda nai e que tanto me ten axudado. Tamén por suposto ó seu marido, compañeiro de carreiras os findes. Non me podo esquecer de miña sobriña Silvia, paréceme mentira que xa sexas tan maior. Quero agradecer tamén ó meu tío, que me foi buscar ó cuvi un número dende logo non pequeno de veces. Por último, á miña avoa, que tan bos momentos teño pasado con ela. Pouca xente pode presumir de ter unha avoa tan activa con noventa e oito anos! Oxalá se me pegara un pouquiño da túa xenética.

Y termino con lo siguiente:

É de ben nacidos ser agradecido.

Contents

List of Figures

1 Introduction

2 Multivariate Ring Learning with Errors

3 Applications of Multivariate RLWE on Lattice-based Cryptography

4 Number Theoretic Transforms

5 Revisiting Multivariate Lattices

6 Genomic Susceptibility Testing

7 Image Denoising

List of Tables

Acronyms and abbreviations

BDD	Bounded Distance Decoding
BGN	Boneh-Goh-Nissim cryptosystem
BGV	Brakerski-Gentry-Vaikuntanathan cryptosystem
CRC	Cyclic Redundancy Check
CRT	Chinese Remainder Theorem
CKKS	Cheon-Kim-Kim-Song cryptosystem
DC	Direct Current
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DGS	Discrete Gaussian Sampling
DWT	Discret Wavelet Transform
FFT	Fast Fourier Transform
FHE	Fully Homomorphic Encryption
FHEW	Fastest Homomorphic Encryption in the West
FV	Fan-Vercauteren cryptosytem
FWHT	Fast Walsh-Hadamard Transform
GC	Garbled Circuits
GDD	Guaranteed Distance Decoding
GLWE	General Learning with Errors
GSW	Gentry-Sahai-Waters cryptosystem
GWAS	Genome-Wide Association Studies
HE	Homomorphic Encryption
IDFT	Inverse Discrete Fourier Transform
INTT	Inverse Number Theoretic Transform
LWE	Learning with Errors
MPC	Secure Multiparty Computation
NFLlib	NTT-based Fast Lattice library
NGS	Next Generation Sequencing
NTRU	Number-Theory-Research-Unit (Number-Theorists-R-Us) cryptosystem

NTT	Number Theoretic Transform
OT	Oblivious Transfer
PRNU	Photoresponse Non-Uniformity
RLWE	Ring Learning with Errors
RNS	Residue Number System
ROC	Receiver Operating Characteristic
RSA	Rivest-Shamir-Adleman cryptosystem
SEAL	Simple Encrypted Arithmetic Library
SHE	Somewhat Homomorphic Encryption
SIMD	Single-Instruction-Multiple-Data
SIVP	Shortest Independent Vector Problem
SPDZ	Smart-Pastro-Damgård-Zakarias
SPED	Signal Processing in the Encrypted Domain
SSP	Secure Signal Processing
SVP	Shortest Vector Problem
TFHE	Fast Fully Homomorphic Encryption over the Torus
WHT	Walsh-Hadamard Transform
YASHE	Yet-Another-Somewhat-Homomorphic-Encryption scheme

Notation

We represent vectors and matrices by boldface lowercase and uppercase letters, respectively. We define the l_p norm of a vector $\boldsymbol{x} \in \mathbb{C}^n$ as $||\boldsymbol{x}||_p = \left(\sum_{i\in[n]} |x_i|^p\right)^{1/p}$, where $1 \leq p < \infty$ with $p \in \mathbb{R}$, and $||\boldsymbol{x}||_\infty = \max_{i\in[n]}|x_i|$. If p is omitted, we consider the Euclidean norm.

The set $[n]$ is defined as $\{1, 2, \ldots, n\}$. We also work with some additional operators as the tensor product $\bigotimes$ and the direct sum $\bigoplus$. When dealing with number fields (or the corresponding ring of integers), as the tensor product is always defined over the rational numbers (integer numbers) we ignore the subscript if there is no ambiguity. $\phi(m)$ and $\Phi_m(x)$ denote, respectively, Euler's totient function and the m-th cyclotomic polynomial.

Polynomials are denoted with regular lowercase letters, ignoring the polynomial variable (e.g., a instead of $a(x)$) whenever there is no ambiguity. We follow a recursive definition of multivariate modular rings: $R_q[x_1] = \mathbb{Z}_q[x_1]/f_1(x_1)$ denotes the polynomial ring in the variable x_1 modulo $f_1(x_1)$ with coefficients belonging to $\mathbb{Z}_q$. Analogously, $R_q[x_1, x_2] = (R_q[x_1])[x_2]/(f_2(x_2))$ is the bivariate polynomial ring with coefficients belonging to $\mathbb{Z}_q$ reduced modulo univariate $f_1(x_1)$ and $f_2(x_2)$. In general, $R_q[x_1, \ldots, x_l]$ (resp. $R[x_1, \ldots, x_l]$) represents the multivariate polynomial ring with coefficients in $\mathbb{Z}_q$ (resp. $\mathbb{Z}$) and the l modular functions $f_i(x_i)$ with $1 \leq i \leq l$.

For a ring R, $a \leftarrow R$ is a uniformly random $a \in R$. If χ is a distribution defined over R, $a \leftarrow \chi$ means that a is drawn from χ.

The polynomial a can also be denoted by a column vector $\boldsymbol{a}$ whose components are the corresponding polynomial coefficients. When needed, we also represent polynomials as column vectors of their coefficients $\boldsymbol{a}$; $\boldsymbol{a} \cdot \boldsymbol{s}$ represents the scalar product between the vectors $\boldsymbol{a}$ and $\boldsymbol{s}$, whose components may belong to the integers or to a polynomial ring, $\boldsymbol{a} \circ \boldsymbol{b}$ is the Hadamard product between vectors, and $\boldsymbol{a} \circledast \boldsymbol{b}$ (resp. $\boldsymbol{a} * \boldsymbol{b}$) is the circular (resp. linear) convolution. Finally, $\boldsymbol{A} \otimes \boldsymbol{B}$ is the Kronecker product between the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$.

Chapter 1

Introduction

Signal processing has become ubiquitous in our daily lives, being present in everyday digital appliances and applications. Actually, although apparently it is hard to be aware of its presence, it does have a great influence on our everyday life and the list of applications and science fields which make use of signal processing tools is accordingly huge: it encompasses communication and entertainment technologies, from speech and audio processing to image and video analysis (e.g., biometric processing of faces, fingerprints, iris, etc.), with a strong impact on emerging applications such as smart grids, autonomous driving, tele-diagnosis and analysis of medical signals (like Electrocardiograms or DNA), among others.

This is especially relevant because, although in our everyday life we are not really conscious of the possible risks, many of the previous scenarios involve the use of very privacy-sensitive data (e.g., a service which depends on the use of personal information). It seems that this is not a problem as, in order to mitigate these privacy concerns, conventional cryptographic techniques can be applied on top of transmission and storage modules. This gives us mechanisms for the protection of data both at rest and in transit, and hence provides both secure storage and communication.

However, many of the most prominent signal processing applications deal with very privacy-sensitive signals which have to be processed by untrusted parties (i.e., the service provider requires the use of the personal information to provide the corresponding service), and the user must trust the service provider to make use of it.

Thus, it is clear that with the advent and widespread use of outsourced computation paradigms (e.g., Cloud computing services), the challenge of protecting the signals while they are processed becomes much harder.

1.1. Secure Signal Processing

The field of Secure Signal Processing (SSP), also known as Signal Processing in the Encrypted Domain (SPED), was born to address these challenges, by devising efficient solutions stemming from the collaborative efforts of cryptography and signal processing [6]. Since then, numerous applications have been proposed based on the use of different cryptographic techniques. Next, we briefly revise some of the most representative methods and ideas.

Adversary model: One of the differences of Secure Multiparty Computation (MPC) schemes with respect to more conventional cryptographic techniques is that the parties involved in the protocol can also behave as an adversary. Although there are many types of possible behaviours to model the adversary, the two most representative which are typically considered in the literature are:

- Passive security: A semi-honest adversary which tries to gather as much information of the honest parties' inputs as possible, but does not deviate from the protocol.
- Active security: A malicious adversary which can arbitrarily deviate from the protocol when trying to cheat.

Secure Multiparty Computation: Secure Multiparty Computation (MPC) searches for methods which allow a set of parties to securely evaluate a function over their inputs, while at the same time protecting their input privacy. It was originally introduced as a solution to the millionaires' problem by Yao [7], which can be seen as a particular instance of secure two-party computation, and it was later generalized to secure multiparty computation [8].

Consequently, MPC is a generic solution to the problem of privacy-preserving processing. In general, there are two basic primitives which are used as building blocks on more general MPC protocols: (1) those based on Oblivious Transfer (OT) [9], in fact it is known that OT is complete for secure multiparty computation, and (2) those based on splitting the data in several shares (secret sharing) [10].

A common feature of most of the MPC protocols is the need of an interactive protocol between the different parties. This step usually causes a high communication overhead.

On the one hand, OT-based methods such as Garbled Circuits (GC) are a specific solution for two-party computation which works with a very small number of interactive rounds (two rounds). However, GC needs to move the bulk of the computation to the client as it requires the client to generate (typically offline) the circuit to be securely evaluated.

On the other hand, many current MPC solutions (e.g., SPDZ-based protocols) can be applied in practice [11], but the efficiency of the involved operations in SPDZ-based schemes [12] contrasts with the typically high required number of communication rounds. These solutions are based on secret-sharing the input data and make use of both the additive homomorphism of the shares, and Beaver's multiplication protocol [13] to perform multiplications.

The previous MPC protocols force us to choose between either a secure interactive protocol with many rounds or a high computational cost for the client. As we will remark later, in this thesis our *main objective* is to provide *unanttended solutions where most of the computation overhead is moved to the untrusted third party*. This objective severely limits the use of the aforementioned cryptographic tools.

The most promising alternative relies on homomorphic encryption (HE). Next, we review the state of the art of HE in Secure Signal Processing, also motivating our results and contributions inside the field.

(Partially) Homomorphic Encryption: In addition to general MPC protocols, Secure Signal Processing has made an extensive use of partially homomorphic encryption schemes such as Paillier [14] and El Gamal [15], to name just a few. These schemes present a group homomorphism between the plaintext and ciphertext spaces which can be conveniently used to perform operations over the encrypted plaintexts (homomorphic operations) without the need of previously decrypting them.

Interestingly, homomorphic encryption schemes can be used to enable a third party to securely evaluate functions under a semi-honest adversary model. In fact, even for the case of active security it does provide input privacy, although it is not simulatable security [16]. This shows the interest on homomorphic cryptography for many signal processing applications where the computation is outsourced to an untrusted third party.

Most of the existing approaches rely on the additively homomorphic Paillier cryptosystem [14] as the basic block for performing encrypted additions between ciphertexts, and multiplications between a ciphertext and a cleartext. These approaches can mainly cope with encrypted linear transforms [17, 18] with known (cleartext) coefficients.

As these solutions are usually limited to additive homomorphisms like Paillier [14], they also need interactive protocols in which the client (or an authority in which the client delegates trust) must communicate with the outsourced processing party in order to produce a result [19]. This imposes many restrictions on the client side, and presents an insurmountable barrier to the development of secure outsourced services. Hence, *the goal of unattended Secure Signal Processing*, where the client only has to pre-process the inputs and post-process the outputs, is still an open problem.

Additionally, solutions resorting to the Paillier cryptosystem present a very high cipher expansion, and despite the proposal of techniques like packing and unpacking to mitigate this effect [20, 21], the cipher expansion becomes a serious problem when dealing with multimedia content as images.

Thesis Contribution: We have introduced new solutions based on lattice cryptography which can efficiently deal with multidimensional signals, and in particular, images [4, 5, 22, 23].

1.2. Lattice-based Cryptography

In recent years, we have witnessed an increasing interest in the research of schemes enabling operations with encrypted data. All these solutions are based on MPC techniques, which aim at achieving privacy-preserving solutions for secure processing of sensitive signals [6]. As we have previously discussed, many of the approaches are particularly based on homomorphic encryption, and rely on the Paillier cryptosystem [14] as the basic block for performing encrypted additions between ciphertexts and multiplications between a ciphertext and a cleartext.

However, approaches based on Paillier present two serious limitations: (a) high overhead and cipher expansion, even when mitigated by packing and unpacking techniques [17, 20]; and (b) they require the involvement of the client (secret key holder) engaging in interactive protocols with the outsourced party [19].

Due to this lack of flexibility, *lattice cryptosystems (which present a ring homomorphism)* are being progressively adopted by researchers in the field [24, 25, 26, 27, 28]. In particular, cryptosystems based on RLWE (Ring Learning with Errors) present a clear advantage when dealing with signals, as its underlying polynomial structure allows for very efficient filtering and convolution operations [29]; hence, most of the applications involving correlations and filtering can benefit from recent RLWE-based schemes, which keep constantly evolving [30, 31, 32].

Thesis Contribution: Among the signal processing applications, those working with images or higher dimensional signals are much more demanding, as the computational cost and cipher expansion of typical SHE cryptosystems becomes unaffordable for them. To address this problem we: (1) introduce a hard lattice problem called m-RLWE (multivariate Ring Learning with Errors) which gives support to efficient encrypted processing of multidimensional signals, and (2) also present a secure toolbox for secure unattended signal processing based on RLWE-based cryptosystems [29].

Somewhat/Fully Homomorphic Encryption Schemes: Gentry's seminal work [33, 34] introduces a new family of cryptosystems enabling FHE (Fully Homomorphic Encryption) schemes that can perform both additions and multiplications in the encrypted domain, while being resilient against quantum cryptanalysis. Despite the relevance of their theoretical contribution, current FHE schemes are not entirely practical for real scenarios [35], so the most promising alternative relies on SHE (Somewhat Homomorphic Encryption) schemes, which have been shown [25] to be able to efficiently work with encrypted signals and encrypted transform coefficients. As a counterpart, while FHE schemes can perform an unbounded number of encrypted operations, SHE schemes can cope only with a limited number of consecutive encrypted operations over the same ciphertext; nevertheless, in most real scenarios, the maximum number of operations that have to be performed on the encrypted data can be previously known, so SHE naturally fits.

Post-Quantum Cryptography: As we have highlighted, Somewhat and Fully Homomorphic Cryptosystems appear as a promising solution enabling both encrypted additions and multiplications, but this is not their only advantage. As a byproduct of being based on hard problems over lattices, they can be proven *secure against classical and quantum computers* [36].

Since the introduction of Shor's algorithm [37], it is known that some problems which were considered secure against classical adversaries can be efficiently solved by means of a quantum computer [36]. Among these problems, we can mention integer factorization and (elliptic-curve) discrete logarithm, which are the basis of the current most widespread cryptosystems (RSA, Paillier [14] or El Gamal [15]). Lattice-based cryptography yields the most suited solution to achieve both resilience against quantum attacks and, at the same time, operate on encrypted information.

The quantum-resistance property is another driver for our goal of providing more efficient schemes which can deal with real problems and, additionally, can *stand as future-proof against quantum computers*.

1.3. A Brief Summary of the Thesis Objectives

The main objective of this thesis is to advance the state of the art for privacy protection when dealing with sensitive (multidimensional) signals in untrustworthy environments. For this purpose, we have applied and developed novel cryptographic tools in the field of Secure Signal Processing.

With this main objective in mind, this thesis has produced novel contributions in the following two research lines:

- **Privacy protection when dealing with multidimensional signals.** Images are multimedia signals that can carry especially sensitive information, so the scenarios where they are used pose serious privacy constraints. Some representative examples comprise biometric recognition, medical image processing, media sharing in social networks or videosurveillance.

Thesis Contribution: In this thesis, we propose novel efficient methods from homomorphic encryption especially targeted at the protection and encrypted processing of multidimensional signals like images or video.

- **Design of new primitives and protocols for encrypted signal processing.** If the scenarios dealing with sensitive signals involve outsourcing the data or processes (i.e., cloud servers or grid, the use and share by means of web services, etc.), the privacy problems are aggravated, as currently the privacy guarantees for the data owner are mainly based on her trust on the outsourced environment.

  Thesis Contribution: In this thesis, novel encrypted efficient operations within the field of signal processing are studied along with security and efficiency improvements to the already existing solutions (e.g., filtering, different types of signal transforms, matrix operations, etc.). Hence, a complete set of tools and encrypted signal processing primitives is provided, therefore reducing the needed confidence between the owner of the private information and the party operating on it.

In general, these two high-level objectives are not specifically addressed in any concrete chapter of the thesis. However, their achievement is guaranteed by a set of more specialized contributions which are deeply studied in each chapter. These more technical contributions are detailed next together with the structure of the thesis.

Trust model: As a working hypothesis in the signal processing applications exemplified in this thesis, we consider a semi-honest setting where the adversary can try to infer as much information as possible, but does not deviate from the protocol.

It is important to remark that for most of the proposed solutions, we consider a two-party scenario where we want to move the bulk of the computation to the server (untrusted party). As our focus is on unattended processing (one-round), our solutions provide input privacy for a malicious adversary, although not simulatable security [16].

1.4. Structure of the Thesis

This thesis is divided in three main parts: (1) The first part includes our contributions to the underlying cryptographic layers which work as building blocks of higher-level practical secure applications. It introduces the main definitions involved in the *multivariate RLWE problem* and discusses its hardness together with the possible advantages it brings about. (2) The second part introduces a *secure framework for signal processing*. We present a toolbox which enables to perform in an unattended way the main operations which are present in most signal processing applications. (3) The third part *exemplifies the use of our toolbox* for Secure Signal Processing with several practical applications: going from genomics to multimedia. Finally, we conclude with a discussion section where we draw some conclusions and also sketch out some of the possible future research lines that this thesis opens up in the field of Secure Signal Processing.

Roadmap: Although the thesis is clearly divided in three separate parts, as we have already discussed, the different contributions are highly interdependent (see Figure 1.1).[1] This means that there is no unique path to read this thesis, and many of the concepts introduced in one chapter are built over tools introduced in the other parts (not necessarily in a sequential order).

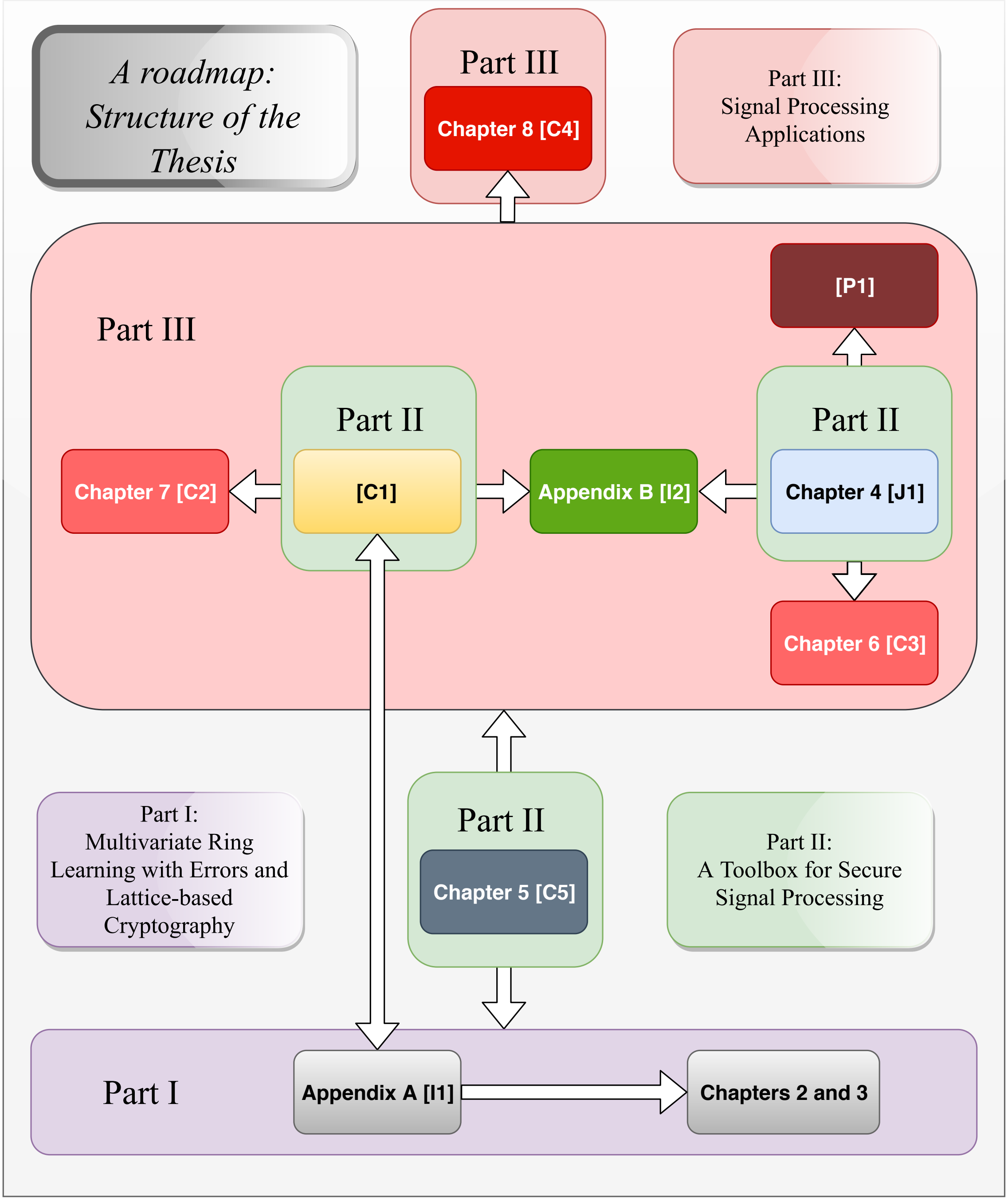

A roadmap: Structure of the Thesis
Part III
Chapter 8 [C4]
Part III: Signal Processing Applications
Part III
[P1]
Part II
Part II
Chapter 7 [C2]
[C1]
Appendix B [I2]
Chapter 4 [J1]
Chapter 6 [C3]
Part II
Chapter 5 [C5]
Part I: Multivariate Ring Learning with Errors and Lattice-based Cryptography
Part II: A Toolbox for Secure Signal Processing
Part I
Appendix A [I1]
Chapters 2 and 3

Figure 1.1: Relation between the different contributions.

With this in mind, we have tried to make each chapter as self-contained as possible. However, due to space constraints and also to avoid repetitions, we refer to the specific chapter where the

[1] Figure 1.1 details the existing interconnections between the different Chapters and publications (this includes journal papers, conference papers, other publications and patent applications) produced in this Thesis.

required concept is discussed in depth.

In view of the above restrictions, we have decided to follow a *bottom-up* strategy; that is, we first formalize and introduce the main cryptographic definitions and hard problems (see Part I), afterwards we present a set of higher-level operations (built over the previous definitions) for signal processing (see Part II), and we end up by exemplifying our results with some more practical applications (see Part III).

1.5. Summary of Part I: Multivariate Ring Learning with Errors and Lattice-based Cryptography

Lattices have become a very promising tool for the development and improvement of new cryptographic constructions, notably those belonging to the field of homomorphic encryption. Instead of directly working with lattice assumptions, it is frequent to deal with assumptions whose underlying security can be based on the hardness of lattice problems. Among them, the family of LWE (Learning with Errors) [38, 39] has become the preferred one due to its versatility. Lyubashevsky *et al.* [40, 41] proposed a variant of LWE called Ring-LWE (or RLWE), which can be reduced from hardness problems over ideal lattices (instead of the general ones used in the LWE version). RLWE has proven to be more practical than LWE, as the underlying primitives can be usually more efficient; e.g., RLWE enables a notable reduction in the size of the public and private keys in public key cryptosystems.

Summary: In this part, we formalize a multivariate version of RLWE, denoted multivariate RLWE, m-RLWE. Due to its particular structure, the security of this multivariate version is especially sensitive to the chosen parameterization, so we will perform a careful and detailed security analysis, showing how secure parameteres can be chosen. Additionally, we also discuss the efficiency improvements that this multivariate version can introduce into the basic cryptographic blocks.

Other Cryptographic Primitives: Although in this thesis we focus on primitives for homomorphic cryptography, ideal lattices have also been used to develop algorithms for key exchange [42] and signatures [43], among others. Hence, we want to remark that those primitives based on RLWE could be also extended to the m-RLWE problem.

Thesis Contributions: The main contributions of this part are divided in two chapters and one appendix:

- **Chapter 2: Multivariate Ring Learning with Errors.** The "Multivariate Ring Learning with Errors" problem is introduced as a generalization of Ring Learning with Errors (RLWE), introducing efficiency improvements with respect to the RLWE counterpart thanks to its multivariate structure [4]. Nevertheless, the recent attack presented by Bootland *et al.* [44] has important consequences on the security of the multivariate RLWE problem with "non-coprime" modular functions; this attack transforms instances of m-RLWE with power-of-two cyclotomic modular functions of degree $n = \prod_i n_i$ into a set of RLWE samples with dimension $\max_i \{n_i\}$. This is especially devastating for low-degree modular functions (e.g., $\Phi_4(x) = 1 + x^2$). In this chapter, we revisit the security of multivariate RLWE and propose new alternative instantiations of the problem that avoid the attack while still preserving the advantages of the multivariate structure, especially when using low-degree modular functions. Additionally, we show how to parameterize these instances in a secure and practical way.

- **Chapter 3: Applications of Multivariate RLWE on Lattice-based Cryptography.** In this chapter, we propose novel constructions and strategies based on m-RLWE that bring notable space and time efficiency improvements over current RLWE-based constructions, including, but not limited to: (a) faster degree-n polynomial multiplication leveraging an $\boldsymbol{\alpha}$-generalized fast Walsh Hadamard Transform, reducing the required elemental products by a factor $\log n$; (b) more efficient homomorphic packing/unpacking strategies with a single switch key operation, independently of the used number of slots; (c) better space-time trade-offs for relinearization operations, needing only $\log_2 n + 1$ matrices with a worst-case chain of $\lceil \frac{\log_2 n}{2} \rceil$ relinearizations; (d) full utilization of packing slots for complex coefficients embedding. These contributions enable vastly more efficient primitives for (fully) homomorphic encryption based on the proposed m-RLWE.

- **Appendix A: A Reduction to Multivariate RLWE.** In this Appendix we formalize the generalization of RLWE to multivariate rings, denoted multivariate RLWE or m-RLWE, and introduce its security reduction to hard problems over the tensor product of ideal lattices, as an extension of the original RLWE proof by Lyubashevsky *et al.* [41, 45]. It is important to remark that due to the Bootland *et al.*'s attack [44], we know we are reducing to an easy problem. In Chapter 2 we search for those instantiations of multivariate RLWE which are not easy (or that are isomorphic to RLWE over a general number field).

  This appendix is adapted with permission from ArXiv: Alberto Pedrouzo-Ulloa, Juan Ramón Troncoso-Pastoriza, and Fernando Pérez-González. On Ring Learning with Errors over the Tensor Product of Number Fields. ArXiv e-prints, CoRR abs/1607.05244v3, February 2018.

1.6. Summary of Part II: A Toolbox for Secure Signal Processing

Many signal processing applications deal with privacy-sensitive signals that must be protected whenever they are outsourced to an untrusted environment. Approaches based on Secure Signal Processing (SSP) [6] address this challenge by proposing novel mechanisms for signal processing in the encrypted domain and interactive secure protocols [19] to achieve the goal of protecting signals without disclosing the sensitive information they convey.

Summary: In this part, we search for unattended secure solutions, that is, those solutions which do not need the intervention of the secret key owner in the middle of the process. To this aim, we propose a toolbox of Secure Signal Processing primitives mainly based on lattice-based cryptography. We also cover the case of multimedia contents (in general any multidimensional signal), showing how the obtained toolbox can be adapted to work with multidimensional signals.

Our Contributions: The main contributions of this part are divided in two chapters:

- **Chapter 4: Number Theoretic Transforms.** This chapter presents a novel and comprehensive set of approaches and primitives to efficiently process signals in an encrypted form, by using Number Theoretic Transforms (NTTs) in innovative ways. This usage of NTTs paired with appropriate signal pre- and post-coding enables a whole range of easily composable signal processing operations comprising, among others, filtering, generalized convolutions, matrix-based processing or error correcting codes. Our main focus in this chapter is on unattended processing, in which no interaction from the client is needed. To implement our solution, we make use of efficient lattice-based somewhat homomorphic cryptosystems. We also exemplify these approaches and evaluate their performance and accuracy, proving that the proposed framework opens up a wide variety of new applications for secured outsourced-processing of multimedia contents.

This chapter is adapted with permission from IEEE: Alberto Pedrouzo-Ulloa, Juan Ramón Troncoso-Pastoriza, and Fernando Pérez-González. Number Theoretic Transforms for Secure Signal Processing. IEEE Transactions on Information Forensics and Security, vol. 12, no. 5, pp. 1125-1140, May 2017.

- **Chapter 5: Revisiting Multivariate Lattices.** This chapter introduces a new pre-/post-coding block that addresses the Bootland *et al.*'s attack and achieves the efficient results of our initial approach while basing its security directly on RLWE with dimension $\prod_i n_i$, hence preserving the security and efficiency originally claimed. Additionally, in this chapter we provide a detailed comparison between a conventional use of RLWE, m-RLWE and our new pre-/post-coding procedure, which we denote "packed"-RLWE. Finally, we discuss a set of encrypted signal processing applications which clearly benefit from the proposed framework, either alone or in a combination of baseline RLWE, m-RLWE and "packed"-RLWE.

  This chapter is adapted with permission from ACM: Alberto Pedrouzo-Ulloa, Juan Ramón Troncoso-Pastoriza, and Fernando Pérez-González. Revisiting Multivariate Lattices for Encrypted Signal Processing. 7th ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec), July 2019.

1.7. Summary of Part III: Signal Processing Applications

In the previous parts, we have searched for general solutions for signal processing applications. To this aim, in Part II we present a toolbox of secure solutions which take advantage of common operations in signal processing. However, it is worth noting that, in general, each specific application can pose additional difficulties depending on the nature of the used signals. We consider two very different types of signals:

- Multidimensional signals like 2-D and 3-D images or videos are inherently sensitive signals which require privacy-preserving solutions when processed in untrustworthy environments, but their efficient encrypted processing is particularly challenging due to their structure, dimensionality and size.

  Many image processing applications require to cope with polynomial operations and comparisons at the same time, this is not an easy task with current homomorphic encryption schemes.

- Genomic data is a paradigmatic example of highly sensitive information. Due to the recent advances in Next Generation Sequencing (NGS), we are seeing an increase in the availability of genomic data for more precise analyses (e.g., testing for the genetic susceptibility to a particular disease).

Current laboratories' facilities cannot cope with this data growth, and genomic processing has to be outsourced.

We see that both scenarios must deal with high volumes of data and they also suffer from severe privacy risks.

Summary: This part exemplifies how to use the tools presented in Parts I and II for several signal processing scenarios dealing with sensitive signals. We briefly enumerate these scenarios and their respective particularities:

- We choose a representative example from the genomic domain, due to the inherent sensitivity of the managed signals: genomic disease susceptibility testing. To this aim, we present a secure protocol for genomic susceptibility testing, where we explain how our techniques can be combined with lattice-based cryptosystems.
- The rest of the described applications focus on multimedia contents: (1) We show how efficient block-processing operations can be performed with multidimensional signals. (2) We work with a fundamental operation in image processing as image denoising. (3) As an example of a more complex application, we make use of the previous results to construct a secure camera analyzer.

Our Contributions: The main contributions of this part are divided in three chapters and one appendix:

- **Chapter 6: Genomic Susceptibility Testing.** This chapter proposes an encrypted genomic susceptibility test protocol based on lattice homomorphic cryptosystems, and introduces optimizations like data packing and transformed processing to achieve considerable gains in performance, bandwidth and storage.

  This chapter is adapted with permission from IEEE: Juan Ramón Troncoso-Pastoriza, Alberto Pedrouzo-Ulloa, and Fernando Pérez-González. Secure Genomic Susceptibility Testing based on Lattice Encryption. The 42nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP17), March 2017.

- **Chapter 7: Image Denoising.** This chapter proposes methods based on 2-RLWE (Bivariate Ring Learning with Errors) to efficiently perform the whole image denoising operation on encrypted images in a fully non-interactive way; we show how to combine homomorphic polynomial operations and thresholding without involving decryption or interaction, therefore enabling fully unattended encrypted image processing. We evaluate our solutions for real image sizes and strict security parameters, showing their practicality and feasibility.

  This chapter is adapted with permission from IEEE: Alberto Pedrouzo-Ulloa, Juan Ramón Troncoso-Pastoriza, and Fernando Pérez-González. Image Denoising in the Encrypted Domain. The 8th IEEE International Workshop on Information Forensics and Security (WIFS16), December 2016.

- **Chapter 8: Camera Attribution Forensic Analyzer.** This chapter proposes a new framework to efficiently perform outsourced PRNU (Photoresponse Non-Uniformity) fingerprint extraction and detection on encrypted images in a fully unattended way. For this purpose, we rely on lattice-based homomorphic cryptosystems paired with advanced optimization strategies. We evaluate our solutions in terms of efficiency, security and performance for real image datasets, showing the feasibility of camera attribution in the encrypted domain.

This chapter is adapted with permission from IEEE: Alberto Pedrouzo-Ulloa, Miguel Masciopinto, Juan Ramón Troncoso-Pastoriza, and Fernando Pérez-González. Camera Attribution Forensic Analyzer in the Encrypted Domain. The 10th IEEE International Workshop on Information Forensics and Security (WIFS18), December 2018.

- **Appendix B: Block-Processing.** This appendix proposes several relinearization-based techniques to efficiently convert signals with different structures and dimensionalities. To this aim, we make use of multivariate RLWE (multivariate Ring Learning with Errors) which generalizes RLWE. The proposed hard problem and the developed techniques give support to lattice cryptosystems that enable encrypted processing of multidimensional signals and efficient conversion between different structures. We show an example cryptosystem and exemplify some of the proposed transformation techniques in critical and ubiquitous block-based processing applications.

  This appendix is adapted with permission from ArXiv: Alberto Pedrouzo-Ulloa, Juan Ramón Troncoso-Pastoriza, and Fernando Pérez-González. Multivariate Cryptosystems for Secure Processing of Multidimensional Signals. ArXiv e-prints, CoRR abs/1712.00848, December 2017.

During the thesis period, we have made other contributions in this topic, but due to limitations of space we have not included them in the thesis. We briefly describe them below:

- **Conference Publication [C1]: Multivariate Lattices for Encrypted Image Processing [4].** This work introduces the use of multivariate RLWE for the efficient processing of encrypted multidimensional signals. To this aim, we extend an RLWE-based cryptosystem to this multivariate setting and showcase its convenience for encrypted image filtering.
- **Patent Application [P1]: Secure Outsourced Prediction.** This work proposes a method for secure prediction in an untrusted environment. Given encrypted training and testing datasets, the method can securely train a prediction model and also predict new samples when a model for prediction is available.
- **Patent Application [P2]: Secure Outsourced Annotation of Datasets.** This work proposes an unattended method for secure annotation of privacy-sensitive datasets in an outsourced environment. Taking as input a private annotation reference, the annotation method searches for tokens and embeds extra information in a sensitive dataset.

1.8. Publications

We list next the main publications which have been produced as the result of this thesis.

1.8.1. Journal papers

J1 Alberto Pedrouzo-Ulloa, Juan Ramón Troncoso-Pastoriza and Fernando Pérez-González. *Number Theoretic Transforms for Secure Signal Processing*. IEEE Transactions on Information Forensics and Security, 2017.

1.8.2. Conference papers

C1 Alberto Pedrouzo-Ulloa, Juan Ramón Troncoso-Pastoriza and Fernando Pérez-González. *Multivariate Lattices for Encrypted Image Processing*. In IEEE ICASSP, 1707-1711, 2015.

C2 Alberto Pedrouzo-Ulloa, Juan Ramón Troncoso-Pastoriza and Fernando Pérez-González. *Image Denoising in the Encrypted Domain*. In IEEE WIFS, 1-6, 2016.

C3 Juan Ramón Troncoso-Pastoriza, Alberto Pedrouzo-Ulloa and Fernando Pérez-González. *Secure Genomic Susceptibility Testing based on Lattice Encryption*. In IEEE ICASSP, 2067-2071, 2017.

C4 Alberto Pedrouzo-Ulloa, Miguel Mascriopinto, Juan Ramón Troncoso-Pastoriza and Fernando Pérez-González. *Camera Attribution Forensic Analyzer in the Encrypted Domain*. In IEEE WIFS, 2018 (*Best paper award*).

C5 Alberto Pedrouzo-Ulloa, Juan Ramón Troncoso-Pastoriza and Fernando Pérez-González. *Revisiting Multivariate Lattices for Encrypted Signal Processing*. In ACM IH&MMSec, 2019.

1.8.3. Patent applications

P1 *Title:* SYSTEM AND METHOD FOR SECURE OUTSOURCED PREDICTION
EPO Patent Application No: EP17382623
Filing Date: 20/09/2017
Inventors: Juan Ramón Troncoso-Pastoriza, Alberto Pedrouzo-Ulloa, Fernando Pérez-González
Assignee: University of Vigo

P2 *Title:* SYSTEM AND METHOD FOR SECURE OUTSOURCED ANNOTATION OF DATASETS
EPO Patent Application No: EP17382624
Filing Date: 20/09/2017
Inventors: Juan Ramón Troncoso-Pastoriza, Alberto Pedrouzo-Ulloa, Fernando Pérez-González
Assignee: Gradiant, University of Vigo

1.8.4. Other Publications

I1 Alberto Pedrouzo-Ulloa, Juan Ramón Troncoso-Pastoriza and Fernando Pérez-González. *On Ring Learning with Errors over the Tensor Product of Number Fields*. In ArXiv e-prints, CoRR abs/1607.05244, 2016.

I2 Alberto Pedrouzo-Ulloa, Juan Ramón Troncoso-Pastoriza and Fernando Pérez-González. *Multivariate Cryptosystems for Secure Processing of Multidimensional Signals*. In ArXiv e-prints, CoRR abs/1712.00848, 2017.

Part I

Multivariate Ring Learning with Errors and Lattice-based Cryptography

Chapter 2

Multivariate Ring Learning with Errors

2.1. Introduction

Lattices have become a very promising tool for the development and improvement of new cryptographic constructions, notably those belonging to the field of homomorphic encryption. Instead of directly working with lattice assumptions, it is frequent to deal with assumptions whose underlying security can be based on the hardness of lattice problems. Among them, the family of LWE (Learning with Errors) [38, 39] has become the preferred one due to its versatility. Lyubashevsky *et al.* [40, 41] proposed a variant of LWE called Ring-LWE (or RLWE), whose hardness can be reduced from hardness problems over ideal lattices (instead of the general ones used in the LWE version). RLWE has proven to be more practical than LWE, as the underlying primitives can be usually more efficient; e.g., RLWE enables a notable reduction in the size of the public and secret keys in public key cryptosystems.

The ring structure of RLWE enables homomorphic cryptography with a ring homomorphism supporting both addition and multiplication of ciphertexts. Among the possible polynomial rings used for this purpose, the most practical ones are those where the modular function is a cyclotomic polynomial of the form $1 + z^n$, with n a power of two. They present two advantages: (a) they enable efficient implementations of polynomial operations through fast radix algorithms of the NTT (Number Theoretic Transforms), and (b) the polynomial operations over the used ring correspond to basic blocks in practical applications in Computer Vision and Signal Processing [29, 5, 46], comprising, among others, linear convolutions, filtering, and linear transforms.

A multivariate version of RLWE (m-RLWE) was proposed as a means to efficiently deal with encrypted multidimensional structures, such as videos or images [4, 46] (see Appendix A). In this scenario, the use of a tensorial decomposition in "coprime" cyclotomic rings (see [45, 41, 40]) is not applicable a priori, as these structures require that the modular functions have the same form (e.g., $1 + z^n$). This is the context in which m-RLWE [4] was originally introduced (see Appendix A).

Related uses of the Tensor product in the literature: The use of the tensor of lattices and/or adding a multivariate structure to the involved rings has been the subject matter of several previous works, but with very different targets. We briefly survey here the closest ones: (a) In [47], the authors applied the standard tensor product of lattices to improve the hardness factor of the SVP problem under different assumptions. (b) In [45], the authors define an isomorphism between

An example of an RLWE sample

To fix ideas, it might help to consider we are working in a polynomial ring like $R_q = \mathbb{Z}_q[x]/(1 - x^n)$. Polynomials belonging to the mentioned ring R_q can be alternatively seen as Z-transforms of a conventional unidimensional signal of length n, but whose coefficients are integers reduced modulo q. Multiplication between polynomials is equivalent to the cyclic convolution between the corresponding signals.

Informally, the RLWE assumption states that given a pair $(a, b = as + e)$ where $a \leftarrow R_q$ is uniformly random and $e \leftarrow \chi$ is drawn from the error distribution (consider a discrete Gaussian distribution for simplicity), this sample is very hard to distinguish from the pair (a, u) where $u \leftarrow R_q$ is also uniformly random.

By assuming this indistinguishability assumption, it is very easy to define a simple cryptosystem based on RLWE. To this aim, plaintext information can be encoded in the noise term by working with signals belonging to the ring $R_t = \mathbb{Z}_t[x]/(1 - x^n)$.

Let a plaintext $m \in R_t$, we could encrypt it by doing $(a' = ta, b' = a's + te + m)$, in such a way that the plaintext is encoded in the lower bits of the error term.

Interestingly, the previous RLWE sample $(a, b = as + e)$ can be alternatively expressed as a pair of signals $(a[l], b[l] = a[l] \circledast s[l] + e[l])$ for $l = 0, \dots, n-1$ (being $\circledast$ the cyclic convolution operation). Hence, it can be seen that *the RLWE sample is "equivalent" to filtering a known and uniformly random signal* $a[l]$ *with a secret filter* $s[l]$*, and afterwards adding a gaussian noise* $e[l]$.

Table 2.1: A Signal Processing perspective: An example of an RLWE sample.

some cyclotomic fields and a tensor product of cyclotomic fields when in $\Phi_m(z)$, if m can be factored into several (different) prime powers. (c) The "tensor" representation also appears in the definition of the GLWE problem (also called Module-LWE [48]) which was originally introduced in [49, 50]. In fact, analogously to LWE versus RLWE, the introduced multivariate RLWE problem can be seen as a ring version of the GLWE problem, by means of adding for a second time a ring structure into the module. (d) Finally, the FHEW fully homomorphic encryption scheme features [51] a ring tensoring for a speed-up of the homomorphic accumulator, and also bivariate rings are used as a means to enhance the efficiency of polynomial products inside the refreshing procedure in [52].

A reduction: It can be shown that the m-RLWE problem can be reduced from worst case discrete Gaussian Sampling (equivalent to SIVP) over the tensor of rings (see Appendix A). Unfortunately, a recent work [44] shows an effective attack against m-RLWE when the univariate subrings share common roots, therefore considerably lowering the security of the underlying problem. Hence, our main contribution in this chapter is to redefine the m-RLWE problem and find secure instantiations that preserve the efficient results on multivariate RLWE, by basing their security on a subset of RLWE on general number fields (see the recent work by Peikert *et al.* [53], that generalizes the RLWE problem to any modulus and any ring over number fields).

We now informally introduce the definition of m-RLWE, the attack by Bootland *et al.* [44], and the rationale of our solution, all exemplified in the bivariate case.

Bivariate RLWE: Let $K_{(T)} = K_x \bigotimes K_y$ be the tensor product of 2 cyclotomic number fields of dimensions $n_x = \phi(m_x)$ and $n_y = \phi(m_y)$, $R = \mathbb{Z}[x, y]/(\Phi_{m_x}(x), \Phi_{m_y}(y))$ the tensor of their corresponding rings of integers, and $R^\vee$ its dual.

RLWE and its convenience for signal processing

Following with the example of the RLWE sample $(a, b = as + e)$ whose elements belong to the ring $R_q = \mathbb{Z}_q[x]/(1 - x^n)$ (see Table 2.1), we know that we can easily encrypt a signal $m[l]$ (being $m(x)$ its Z-transform and whose elements belong to $\mathbb{Z}_t$) by considering $(a', b' = a's + te + m)$.
Additionally, this cryptosytem also allows for homomorphic operations. Consider two *encryptions* $(a_1, b_1 = a_1 s + te_1 + m_1)$ and $(a_2, b_2 = a_2 s + te_2 + m_2)$. If q is high enough compared to the maximum value of the noise terms, we can easily obtain a *homomorphic addition* of the plaintexts by doing

$$(a_{add} = a_1 + a_2, b_{add} = b_1 + b_2 = a_{add}s + t(e_1 + e_2) + (m_1 + m_2)) .$$

Although the process for a *homomorphic multiplication* is slightly more complicated, it can still be done:

$$(a_{mult}, b_{mult}, c_{mult}) = (a_1 a_2, a_1 b_2 + a_2 b_1, b_1 b_2) .$$

Although we skip the details, the triple $(a_{mult}, b_{mult}, c_{mult})$ can be seen as an encryption of the polynomial product $m_1 m_2 \bmod 1 - x^n$, which, considering the signal representation, is equivalent to $m_1[l] \circledast m_2[l]$.
Consequently, it seems that this very simple cryptosystem is very convenient for some basic signal processing blocks: (1) A signal can be encrypted in a whole ciphertext, and (2) we have two homomorphic operations which correspond to cyclic convolution and addition of signals.

Table 2.2: A Signal Processing perspective: RLWE and its convenience for signal processing.

We define a Bivariate Ring LWE sample (see Definition 2 for the general formulation of m-RLWE) as the pair $(a, b = (a \cdot s)/q + e \bmod R^\vee)$, where $a \leftarrow R_q$ is uniformly random and $e \leftarrow \Psi$ comes from the error distribution Ψ.

Bootland *et al.*'s attack: Choices of modular functions $f_x(x) = \Phi_{m_x}(x)$, $f_y(y) = \Phi_{m_y}(y)$ as $f_x(x) = x^{n_x} + 1$, $f_y(y) = y^{n_y} + 1$ have been proposed in [4], as this structure presents computational advantages and can be very beneficial for practical applications.

Bootland *et al.*'s attack is able to exploit common roots on the involved rings to factorize the multivariate RLWE samples into RLWE samples of smaller dimension. For example, consider that $n_x = n_y = n$; by applying the substitution $y \leftarrow x$, we obtain n RLWE samples of dimension n each, hence decreasing the n^2 lattice dimension of the original m-RLWE sample.

Secure multivariate RLWE instantiations: Let $m = m_x m_y$ and $\gcd(m_x, m_y) = 1$; then, the m-th cyclotomic field $K = \mathbb{Q}(\zeta_m) \cong \mathbb{Q}[x]/(\Phi_m(x))$ (with ζ_m the m-th root of unity) is isomorphic to the bivariate field

$$K \cong \mathbb{Q}[x, y]/(\Phi_{m_x}(x), \Phi_{m_y}(y)). \tag{2.1}$$

Consequently, by considering instantiations satisfying $\gcd(m_x, m_y) = 1$, the bivariate RLWE problem becomes equivalent to the equally sized RLWE problem. However, we would like to search for other instantiations where the modular functions can have a similar form and, if possible, the same degree.

By restricting ourselves to the most common scenario of power of two cyclotomics, modular functions of the form $\{x^{n_x} + d_x, y^{n_y} + d_y, z^{n_z} + d_z, \ldots\}$, could avoid Bootland *et al.*'s attack for some parameters $\{n_x, d_x, n_y, d_y, n_z, d_z, \ldots\}$. E.g., the rings $\mathbb{Z}[x]/(x^{64} + 1)$ and $\mathbb{Z}[y]/(y^{27} + 5)$ do not have common roots, so trivial substitutions such as $x \rightarrow y$ cannot be applied. Additionally,

Motivation for a multivariate RLWE sample

In Tables 2.1 and 2.2 we have briefly shown that the RLWE assumption can be used to define a very simple cryptosytem allowing for both homomorphic additions and convolutions of encrypted signals.
This observation gives us the starting point for the main idea introduced in [4]. If an RLWE sample $(a, b = as + e)$ can be equivalently expressed as a pair of signals $(a[l]$, $b[l] = a[l] \circledast s[l] + e[l])$ for $l = 0, \ldots, n-1$, in [4] we can wonder whether this convolution can be extended to work with higher dimensional signals.
With this in mind, we considered a *multivariate* RLWE sample by substituting the previous *unidimensional* convolution by a *multidimensional* convolution.
This redefinition of the cryptosystem allows for homomorphic additions and multidimensional cyclic convolutions of encrypted multidimensional signals.
Exemplifying it with 2-dimensional signals (e.g., images), we have

$$(a(x,y), b(x,y) = a(x,y)s(x,y) + e(x,y)),$$

where the different polynomials belong to the ring $R_q = \mathbb{Z}_q[x,y]/(1 - x^{n_x}, 1 - y^{n_y})$.
A cryptosystem defined with the previous *bivariate* RLWE sample can encrypt 2-dimensional signals of lengths n_x and n_y (i.e., $m[l_x, l_y]$ for $l_x = 0, \ldots, n_x - 1$ and $l_y = 0, \ldots, n_y - 1$). Analogously to the unidimensional case, polynomials represent the bivariate (in general, multivariate) Z-transforms of the signals (e.g., a plaintext for $m[l_x, l_y]$ would be represented by its bivariate Z-transform $m(x,y) \in \mathbb{Z}_t[x,y]/(1 - x^{n_x}, 1 - y^{n_y})$).

Table 2.3: A Signal Processing perspective: Motivation for a multivariate RLWE sample.

whenever we reduce modulo q and work over R_q, we can impose (for the sake of efficiency) that both modular functions $x^{64}+1$ and $y^{27}+5$ factor in linear terms enabling the use of variants of the NTT. Additionally, slot encoding and slot manipulations are still possible in the plaintext ring by means of the pre-/post-processing, as presented in [29]. Analogously to the negayclic convolution, these pre-/post-processing steps preserve the properties of the NTT transform over a ring with an α-generalized convolution [54] (see Chapter 3).

This seems to effectively avoid a substitution attack; however, there might be some small ideal divisor for which, modulo some particular q, the noise does not increase substantially, and we can distinguish the resulting sample from uniform. This attack has been extensively studied by Peikert in [55] and we will discuss it in Section 2.7.1.

The proposed solution: The previous strategy preserves most of the advantages of the multivariate constructions while apparently avoiding the effects of Bootland *et al.*'s attack. However, the security of these instantiations is not based on any specific formulation of the RLWE problem, and there is no trivial way of parameterizing them. This raises the following questions:

1. *Can we find multivariate rings similar to $\mathbb{Z}[x, y, \ldots]/(x^{n_x} + d_x, y^{n_y} + d_y, \ldots)$ while (a) still preserving the aforementioned advantages of this structure, and (b) basing its security on the hardness of the RLWE problem (see Definition 7); i.e., without a decrease in the ring dimension due to Bootland's attack (see Proposition 1)?*

2. *If these multivariate rings exist, how can the values $\{n_x, n_y, \ldots\}$, $\{d_x, d_y, \ldots\}$ be chosen to easily define the ring of integers R, its dual $R^\vee$ and the basis?*

From this point forward, we focus on answering these two questions. To this aim, we identify number fields whose ring of integers (and their dual) have the sought structure (see Section 2.4). In particular, we divide this set of fields in two categories:

1. *Multiquadratic number fields* (see Section 2.5). These structures enable efficient radix-2 transforms for faster polynomial arithmetic (see Section 3.2 in Chapter 3).
2. *More general number fields with modular functions* $\{x^{n_x} + d_x,\, y^{n_y} + d_y,\, \ldots\}$ (see Section 2.6). These structures support all the signal processing applications described in [5], and the matrix operations introduced by the original MHEEAN scheme [56] (not based on coprime cyclotomic polynomials [57]) while preserving the equivalent RLWE security.

Rationale for the security of our solution: The weakness of some m-RLWE instantiations is rooted on the existence of (small norm) zero divisors in the compositum field. For example, $\mathbb{Q}[x, y]/(x^2+1, y^2+1)$ has zero divisors as $x+y$ (e.g., $(x+y)(x-y) = 0$), and hence m-RLWE samples defined on rings $\mathbb{Z}[x, y]/(x^2 + 1, y^2 + 1)$ can be easily factored, as the effective *degree* can be reduced with substitutions $\{x \to y, -x \to y\}$. Additionally, as these roots have *small norm*, the noise in the reduced samples is not increased enough to preserve security.

Instead of the previously proposed $\mathbb{Z}[x, y]/(x^2 + 1, y^2 + 1)$, we work with a bivariate ring with modular functions of the form $\{x^{n_x} + d_x, y^{n_y} + d_y\}$ (we use $\mathbb{Z}[x, y]/(x^2 + 1, y^2 + 3)$ as our example). The use of different modular functions avoids a trivial substitution attack. However, we need to rule out the possibility of (small norm) substitution attacks, such as the one from [44], modulo some q; even if they exist, finding them would require solving a hard subset-sum $\bmod q$ (knapsack) problem.

As there is a security reduction from ideal lattices to RLWE defined on general number fields [53], we search for the ring of integers of *multivariate number fields*. This gives us a way to find secure parameters for the used ring, and also the right error distribution to guarantee that the noise increase after a substitution modulo q is enough to preserve the required security [55]. To exemplify this rationale, we compare the differences between a bivariate cyclotomic ring (which can be seen as a univariate cyclotomic ring), and our proposed solution.

Consider the ring $\mathbb{Z}[z]/\Phi_{12}(z)$ with $\Phi_{12}(z) = z^4 - z^2 + 1$. There is an isomorphism with the bivariate ring $\mathbb{Z}[x, y]/(\Phi_4(x), \Phi_3(y))$ where $\Phi_4(x) = x^2 + 1$ and $\Phi_3(y) = y^2 + y + 1$. Therefore, our intuition is that if we found an effective substitution attack on our example ring $\mathbb{Z}[x, y]/(x^2 + 1, y^2 + 3)$, this would work analogously for the cyclotomic bivariate case $\mathbb{Z}[x, y]/(\Phi_4(x), \Phi_3(y))$. In particular, if we apply the transformation $T(y) = 2y + 1$ in the ring $\mathbb{Z}[y]/(y^2 + 3)$, we obtain $\mathbb{Z}[y]/y^2 + y + 1$, which is the mentioned cyclotomic ring with $\Phi_3(y)$. Consequently, for this particular case, it is clear that the existence of an attack in our example ring implies an attack to the bivariate cyclotomic ring.

For more general multivariate rings, we can apply a similar idea. In general, for a secure bivariate ring such as $\mathbb{Z}[x, y]/(x^{n_x} + d_x, y^{n_y} + d_y)$, we can search for a transformation $y \leftarrow T(y)$ where the new modular function can share at least some roots with $x^{n_x} + d_x$. If this transformation can be effectively applied, we could use it to attack multivariate cyclotomic rings.

Thus, this strengthens the belief that an attack on secure m-RLWE instantiations defined on a general number field should provide us with either an attack to RLWE on the product of prime-powers cyclotomic rings, and/or a better understanding on the weaknesses of general cyclotomic rings.

For a discussion on the practical security of RLWE on the proposed number fields we refer the

Some security issues with our example of an RLWE sample

The provided examples for univariate (and multivariate) RLWE samples in Tables 2.1, 2.2, and 2.3 work by considering polynomial rings with cyclic modular functions (i.e., with the form $1 - x^n$). However, due to security reasons, we must be very careful on the choice of the modular function. For example, for the function $1 - x^n$, an attack can exploit the fact that 1 is a small-norm integer root by evaluating the Z-transform of the RLWE sample in 1 (i.e., *the DC component*). Consequently, a substitution $x \leftarrow 1$ can be applied which simplifies the RLWE sample $(a, b = as + e)$ into a new sample

$$\left(\underbrace{\sum_l a[l]}_{\tilde{a}}, \underbrace{\sum_l b[l]}_{\tilde{b}} = \underbrace{\left(\sum_l a[l]\right)}_{\tilde{a}}\underbrace{\left(\sum_l s[l]\right)}_{\tilde{s}} + \underbrace{\sum_l e[l]}_{\tilde{e}}\right),$$

where the index $l = 0, \ldots, n-1$.
This problem is much easier than expected because we only have to distinguish *the DC component* of the signals from the uniform distribution in $\mathbb{Z}_q$ (going down the dimension from n to 1). As q is usually poly(n), to distinguish $(\tilde{a}, \tilde{b} = \tilde{a}\tilde{s} + \tilde{e})$ from uniform, we can efficiently try the q different possibilities for $\tilde{s}$ in the expression $\tilde{b} - \tilde{a}\tilde{s}$. For the right $\tilde{s}$ (and considering a reasonable value for the variance of the χ distribution, which we initially considered as Gaussian in Table 2.1) it behaves as a Gaussian distribution.

Secure RLWE: The original RLWE problem [41, 45] was defined on rings whose modular function is a cyclotomic polynomial. The m-th cyclotomic polynomial $\Phi_m(x)$ is the unique irreducible polynomial with integer coefficients whose $\phi(m)$ roots are all the m-th primitive roots of unity ($\phi(m)$ is the Euler's totient function).
Up to today, attacks on these RLWE instanstiations (with parameters following the reduction presented in [41, 45]) are not substantially faster than an attack on general lattices. Consequently, current most efficient lattice-based cryptosystems are implemented based on RLWE considering cyclotomic modular functions and, in particular, power-of-two modular functions $\Phi_{2n}(x) = 1 + x^n$ (with n a power-of-two).

$1 + x^n$ *modular functions:* Even though the use of $1 - x^n$ functions is discarded because of the aforementioned attack, we can still use a very similar polynomial ring $R_q = \mathbb{Z}_q[x]/(1 + x^n)$. This ring allows for homomorphic additions and *negacylic convolutions* of the encrypted signals. In this thesis we show how to transform these homomorphic operations into encrypted cyclic convolutions which are more amenable for signal processing applications (see Chapters 3 and 4).

Table 2.4: A Signal Processing perspective: Some security issues for our example of an RLWE sample.

reader to Section 2.7.1.

Contributions: The main contribution of this chapter is the definition and parameterization of secure instantiations of the multivariate Ring Learning With Errors problem [22, 5], supported by the extended reduction [53] of the original proof by Lyubashevsky *et al.* [40, 41]. The proposed instantiations address the vulnerability leveraged on Bootland's attack to m-RLWE [44], while still preserving all the efficiency improvements that m-RLWE brings. Moreover we show that is possible to securely instantiate the m-RLWE problem, because the canonical embedding of R has a polynomial skewness (λ_n/λ_1).

We instantiate a simple cryptosystem based on m-RLWE, and exemplify with it the use of the multivariate structure of m-RLWE to improve on complex number embedding, enabling fully packed complex numbers, compared to the exponentially decreasing packing ratio of current ap-

***Secure* multivariate RLWE samples**

To define a *secure* RLWE sample, we have seen in Table 2.4 that we can replace in the polynomial ring R_q the modular function $1-x^n$ by $1+x^n$. From the point of view of signal processing given in Tables 2.1 and 2.2, this means that in the RLWE sample $(a[l], b[l] = a[l] \circledast s[l] + e[l])$ (for $l = 0, \dots, n-1$) the cyclic convolution is replaced by a negacyclic convolution.
Following our initial motivation to work with multidimensional signals (see Table 2.3), we can consider the following multivariate RLWE sample

$$(a(x,y), b(x,y) = a(x,y)s(x,y) + e(x,y) \bmod 1 + x^{n_x} \bmod 1 + y^{n_y}),$$

where the different polynomials belong to the ring $R_q = \mathbb{Z}_q[x,y]/(1+x^{n_x}, 1+y^{n_y})$. Hence, the $a(x,y) \cdot s(x,y)$ multiplication can be seen as a 2-dimensional negacyclic convolution between $a[l_x, l_y]$ and $s[l_x, l_y]$ for $l_x = 0, \dots, n_x - 1$ and $l_y = 0, \dots, n_y - 1$.

$\{1 + x^{n_x}, 1 + y^{n_y}\}$ *modular functions:* Although initial works [4] assumed that the security of the previous bivariate RLWE (2-RLWE) sample was equivalent to an RLWE sample with $n = n_x n_y$, we know that due to Bootland's *et al.*'s attack [44] its security is mainly equivalent to an RLWE sample with degree $\max\{n_x, n_y\}$ (see Section 2.2). In Chapter 5 we provide a detailed comparison between this and other possible instantiatons of RLWE samples for the case of encrypted multidimensional signal filtering.

Some intuitions: Working with a modular function $1 - x^n$, we can directly compute the *DC component* of the Fourier transform by the substitution $x \leftarrow 1$. In similar manner, when having two modular functions $1 + x^n$ and $1 + y^n$, by applying the substitution $y \leftarrow x$ we obtain one of the frequency components of the Fourier transform of the bivariate RLWE sample for the dimension y. It is worth noting that both substitutions are *low-norm roots.*

Secure bivariate RLWE samples with $\{d_x + x^{n_x}, d_y + y^{n_y}\}$*:* The weakness of the previous bivariate RLWE sample relies on the repetition of the same structure in the two dimensions (allowing for low-norm substitutions). Similarly to the *unsecure* (cyclic) RLWE sample from Tables 2.1 and 2.2, we can compute different coefficients of the Fourier Transform in one of the dimensions without considerably increasing the noise of the sample. This allows us to deal with a transformed sample of much smaller dimension than initially expected. To avoid this attack, this chapter focuses on studying *how to securely replace* the *unidimensional* convolution in RLWE by a *multidimensional* convolution where the structure is different for each dimension. Consequently, we search for modular functions where there are no low-norm substitutions. This is explained in detail in Sections 2.4, 2.5 and 2.6.

Table 2.5: A Signal Processing perspective: *Secure* multivariate RLWE samples.

proaches. This enables applications in homomorphically encrypted approximate arithmetic, complex processing, and efficient multidimensional signal manipulation (see Section 2.7.2). The applications of these secure instantiations are numerous, achieving improved space-time tradeoffs in the most critical lattice operations, and therefore enabling more efficient homomorphic processing and closing the gap to the realization of practical fully homomorphic encryption. As we will show in Chapter 3, m-RLWE can bring significant efficiency improvements in all of them.

Structure: The rest of the chapter is organized as follows: Section 2.2 describes Bootland *et al.*'s attack to multivariate RLWE. For completeness, Section 2.3 reminds some algebraic number theory notions and the main definitions for the m-RLWE problem. Section 2.4 describes the followed strategy to achieve secure instantiations of multivariate RLWE, including the well-known tensor of "coprime" cyclotomic rings. Section 2.5 focuses on the analysis of multiquadratic rings.

Section 2.6 studies a set of more general multivariate rings. Section 2.7 includes a discussion on the achieved resilience against known attacks together with example instantiations that showcase the practicality of multivariate RLWE, and discusses some practical applications. Finally, Section 2.8 draws some conclusions.

2.2. Worst case security of multivariate RLWE

For the sake of clarity, we present the definition of multivariate RLWE with power-of-two cyclotomic polynomials, as originally introduced in [4], but all the results in this section can be generalized to any cyclotomic function:

Definition 1 (multivariate RLWE with power-of-two modular functions as $x_i^{n_i}+1$)**.** *Given a multivariate polynomial ring $R_q[x_1,\ldots,x_l]$ with $f_j(x_j)=1+x_j^{n_j}$ for $j=1,\ldots,l$ where $n=\prod_j n_j$ (with all n_j a power of two) and an error distribution $\chi[x_1,\ldots,x_l]\in R_q[x_1,\ldots,x_l]$ that generates small-norm random multivariate polynomials in $R_q[x_1,\ldots,x_l]$, the multivariate polynomial RLWE problem relies upon the computational indistinguishability between samples $(a_i,b_i=a_i\cdot s+e_i)$ and (a_i,u_i), where $a_i,u_i\leftarrow R_q[x_1,\ldots,x_l]$ are chosen uniformly at random from the ring $R_q[x_1,\ldots,x_l]$; $s,e_i\leftarrow\chi[x_1,\ldots,x_l]$ are drawn from the error distribution.*

The original works of multivariate RLWE [4, 5] assume that the search and decision m-RLWE problems (see Definitions 3 and 4) in dimension $n=\prod_{i=1}^m n_i$ are as hard as the corresponding RLWE problems in dimension n. However, Bootland *et al.* [44] introduced an attack that can exploit modular functions that allow repeated "low-norm" roots in the multivariate ring. As a result, when the subrings of the tensor have common roots, this attack is able to factor the m-RLWE samples into RLWE samples of smaller dimension, hence reducing the security of these m-RLWE samples to that of solving a set of independent RLWE samples which are easier to break. E.g., for the ring $\mathbb{Z}[x,y]/(x^{2n}+1,y^n+1)$, changes of variable $y\leftarrow x^{2i}$ with $i\in\mathbb{Z}_{2n}^*$ factors the m-RLWE sample into n different RLWE samples with rings of modular function $x^{2n}+1$ and an increase in the error variance of n (maximum degree of y^n+1).

The instantiations of (multivariate) RLWE with cyclotomic rings where the different modular functions have "coprime" order are not affected by this attack, as they do not introduce these "common" roots (see Section 2.4.1).

We now give a more formal description of the attack, particularized for bivariate RLWE (2-RLWE) with power of two cyclotomics (Definition 1). Let $(a,b=as+e)\in R_q^2[x,y]$ and $R_q[x,y]=\mathbb{Z}_q[x,y]/\left(x^{n_x}+1,y^{n_y}+1\right)$ with $n_x\geq n_y$ and $k=\frac{n_x}{n_y}$ without loss of generality.

Now we define the map $\tilde{\Theta}$:

$$\begin{aligned}\tilde{\Theta}:\mathbb{Z}_q[x,y]/(x^{n_x}+1,y^{n_y}+1)&\rightarrow(\mathbb{Z}_q[x]/(x^{n_x}+1))^{n_y}\\ a(x,y)&\rightarrow\Big(a(x,x^k),a(x,x^{3k}),\ldots,a(x,x^{(2n_y-1)k})\Big)\end{aligned}$$

This map is a ring homomorphism, and if q is odd it is also invertible (see [44]). This transforms the pair $(a,b)\in R_q[x,y]$ into $(\tilde{\Theta}(a),\tilde{\Theta}(b)\in R_q^{n_y}[x]$. If we denote each of the components by $\tilde{\Theta}_i$, for $i=1,\ldots,n_y$, we have

$$\Big(\tilde{\Theta}_i(a),\tilde{\Theta}_i(b)=\tilde{\Theta}_i(a)\tilde{\Theta}_i(s)+\tilde{\Theta}_i(e)\Big)\in R_q^2[x],\tag{2.2}$$

for $i = 1, \ldots n_y$. This results in n_y different RLWE samples of dimension n_x and whose noise has a variance n_y times higher than the original 2-RLWE sample (the result of adding n_y independent variables).

The attack works by trying to break the obtained n_y RLWE samples. Once this is done, as the map is invertible, it is possible to reconstruct the original secret key with the different n_y smaller keys.

This attack can be generalized to an m-RLWE sample (Definition 1) by recursively applying "versions" of this map $(l-1)$ times. This recursion converts an m-RLWE sample into $\frac{n}{n_l}$ RLWE samples (assuming, without loss of generality, that $n_1 \leq n_2 \leq \ldots \leq n_l$) with dimension n_l and an error variance $\frac{n}{n_l}$ times higher.

2.3. Multivariate Ring Learning with Errors

This section revisits the main definitions from algebraic number theory and multivariate RLWE, including a generalized version of the multivariate polynomial RLWE problem which admits any type of cyclotomic polynomial as modular function (see Appendix A). For the sake of clarity, we particularize to power-of-two modular cyclotomic functions (see Definition 1) when exemplifying some of the results, but this does not affect to the generality of the discussion.

2.3.1. Algebraic Number Theory background

This section presents the fundamental concepts of lattices and algebraic number theory and extends them to the more general case of a tensor of number fields on which m-RLWE is based.

The Space $H_{(T)} = \bigotimes_i H_i$

When dealing with cyclotomic fields, it is useful to work with the subspace $H \subseteq \mathbb{R}^{s_1} \times \mathbb{C}^{2s_2}$ with $s_1 + 2s_2 = n$, where the tuple (s_1, s_2) is called the signature of the number field, and H satisfies

$$H = \{(x_1, \ldots, x_n) \in \mathbb{R}^{s_1} \times \mathbb{C}^{2s_2} \text{ such that } x_{s_1+s_2+j} = \bar{x}_{s_1+j}, \forall j \in [s_2]\} \subseteq \mathbb{C}^n\} \qquad (2.3)$$

An orthonormal basis $\{\boldsymbol{h}_j\}_{j\in[n]}$ for H can be defined as

$$\boldsymbol{h}_j = \begin{cases} \boldsymbol{e}_j & \text{if } j \in [s_1] \\ \frac{1}{\sqrt{2}}(\boldsymbol{e}_j + \boldsymbol{e}_{j+s_2}) & \text{if } s_1 < j \leq s_1 + s_2 \\ \frac{\sqrt{-1}}{\sqrt{2}}(\boldsymbol{e}_{j-s_2} - \boldsymbol{e}_j) & \text{if } s_1 + s_2 < j \leq s_1 + 2s_2, \end{cases}$$

where $\boldsymbol{e}_j$ are the vectors of the standard basis in $\mathbb{R}^n$. Each element $a = \sum_{j\in[n]} a_j \boldsymbol{h}_j \in H$ (with $a_j \in \mathbb{R}$) has its own l_p norm. For our purposes, we define the subspace $H_{(T)} = \bigotimes_{i\in[l]} H_i$ as the tensor product of l subspaces H_i (each subspace H_i defined as in Eq. (2.3) but with $s_1 + 2s_2 = n_i$).

In particular, if we see each element belonging to each H_i as a different linear transformation, we are actually working with the Kronecker product of the subspaces H_i. We can therefore express an orthonormal basis for $H_{(T)}$ given by $\{\boldsymbol{h}_j\}_{j\in[n]}$ as the result of the Kronecker product of the original basis of each H_i, by defining any invertible mapping for j and $\{j_1, \ldots, j_l\}$, where $\boldsymbol{h}_j = \bigotimes_{i\in[l]} \boldsymbol{h}_{j_i}^{(i)}$ are the basis vectors for $H_{(T)}$, and $n = \prod_{i\in[l]} n_i$; each $\{\boldsymbol{h}_{j_i}^{(i)}\}_{j_i\in[n_i]}$ is the orthonormal basis of each $H_i \subseteq \mathbb{C}^{n_i}$ for $i \in [l]$.

Lattice background

A lattice in our multivariate setting is defined as an additive subgroup of $H_{(T)}$. We only consider full rank lattices, obtained as the set of all integer linear combinations of a set of n linear independent basis vectors $\boldsymbol{B} = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\} \subset H_{(T)}$

$$\Lambda = \mathcal{L}(B) = \left\{ \sum_{i \in [n]} z_i \boldsymbol{b}_i \text{ such that } \boldsymbol{z} \in \mathbb{Z}^n \right\}$$

The minimum distance $\lambda_1(\Lambda)$ of a lattice Λ for the norm $||.||$ is given by the length of the shortest non-zero lattice vector, that is, $\lambda_1(\Lambda) = \min_{\boldsymbol{x} \in \Lambda / \boldsymbol{x} \neq \boldsymbol{0}} ||\boldsymbol{x}||$.

The dual lattice of $\Lambda \subset H_{(T)}$ is defined as $\Lambda^* = \{\boldsymbol{x} \in H_{(T)} | \langle \Lambda, \boldsymbol{x} \rangle \subseteq \mathbb{Z}\}$ and it satisfies $(\Lambda^*)^* = \Lambda$.

Gaussian Measures

The results on nonspherical Gaussian distributions presented in [41] can be extended to our case. Hence, we revisit here some of the concepts for Gaussian measures, adapted to our tensor setting.

We consider the Gaussian function $\rho_r : H_{(T)} \to (0, 1]$ with $r > 0$ as $\rho_r(\boldsymbol{x}) = \exp(-\pi ||\boldsymbol{x}||^2 / r^2)$. A continuous Gaussian probability distribution D_r can be obtained by normalizing the previous function to obtain a probability density function as $r^{-n} \rho_r(\boldsymbol{x})$. Extending this to the non spherical Gaussian case, we consider the vector $\boldsymbol{r} = \bigotimes_{i \in [l]} \boldsymbol{r}_i$ where $\boldsymbol{r} = (r_1, \ldots, r_n) \in (\mathbb{R}^+)^n$ and $\boldsymbol{r}_i = (r_{i,1}, \ldots, r_{i,n_i}) \in (\mathbb{R}^+)^{n_i}$ and whose components satisfy $r_{i,j+s_1+s_2} = r_{i,j+s_1}$. Finally, a sample from $D_{\boldsymbol{r}}$ is given by $\sum_{j \in [n]} x_j \boldsymbol{h}_j$ where each x_j is drawn independently from a Gaussian distribution D_{r_j} over $\mathbb{R}$; r_j equals $\prod_{i \in [l]} r_{i,j_i}$ (where l is the number of "unidimensional" spaces H_i in the tensor, that is $n = \prod_{i \in [l]} n_i$) and we are using any invertible mapping between $\{j\}_{j \in [n]}$ and $\{j_i\}_{j_i \in [n_i], i \in [l]}$.

2.3.2. Main Definitions for Multivariate Ring-LWE

Let $K_{(T)} = \bigotimes_{i \in [l]} K_i$ be the tensor product of l cyclotomic fields of dimension $n_i = \phi(m_i)$ each, and $R = \bigotimes_{i \in [l]} \mathcal{O}_{K_i}$ ($R^\vee = \bigotimes_{i \in [l]} \mathcal{O}_{K_i}^\vee$) the tensor of their corresponding (respectively, dual of the) ring of integers. We have the following definitions:

Definition 2 (Multivariate Ring LWE distribution)**.** *For $s \in R_q^\vee$ and an error distribution ψ over $K_{(T),\mathbb{R}}$, a sample from the m-RLWE distribution $A_{s,\psi}$ over $R_q \times \mathbb{T}$ is generated by $a \leftarrow R_q$ uniformly at random, $e \leftarrow \psi$, and outputting $(a, b = (a \cdot s)/q + e \bmod R^\vee)$.*

Definition 3 (Multivariate Ring LWE, Search)**.** *Let Ψ be a family of distributions over $K_{(T),\mathbb{R}}$. m-RLWE$_{q,\Psi}$ denotes the search version of the m-RLWE problem. It is defined as follows: given access to arbitrarily many independent samples from $A_{s,\Psi}$ for some arbitrary $s \in R_q^\vee$ and $\psi \in \Psi$, find s.*

Definition 4 (Multivariate Ring LWE, Average-Case Decision)**.** *Let Υ be a distribution over a family of error distributions, each over $K_{(T),\mathbb{R}}$. The average-case decision version of the m-RLWE problem, denoted m-R-DLWE$_{q,\Upsilon}$, is to distinguish with non-negligible advantage between* *arbitrarily many independent samples from $A_{s,\psi}$, for a random choice of $(s,\psi) \leftarrow U(R_q^\vee) \times \Upsilon$,*[1] *and the same number of uniformly random and independent samples from $R_q \times \mathbb{T}$.*

For an asymptotic treatment of the m-RLWE problems, we let $K_{(T)}$ come from an infinite sequence of tensors of number fields $\mathbb{K} = \{K_{(T),n}\}$ of increasing dimension n ($n = \prod_i \phi(m_i)$ is the number of basis elements that form the integral basis), and let q, Ψ, and Υ depend on n as well.

Error distributions We include here two definitions about the error distributions.

Definition 5 (extension of Lyubashevsky *et al.* [41], Definition 3.4)**.** *For a positive real $\alpha > 0$, the family $\Psi_{\leq\alpha}$ is the set of all elliptical Gaussian distributions $D_{\boldsymbol{r}}$ (over $K_{(T),\mathbb{R}}$), where each parameter $r_i \leq \alpha$ with $i \in [n]$.*

Definition 6 (extension of Lyubashevsky *et al.* [41], Definition 3.5)**.** *Let $K_{(T)} = \bigotimes_{i\in[l]} K_i$ where the K_i is the m_i-th cyclotomic number field having degree $n_i = \phi(m_i)$. For a positive real $\alpha > 0$, a distribution sampled from Υ_α is given by an elliptical Gaussian distribution $D_{\boldsymbol{r}}$ (over $K_{(T),\mathbb{R}}$) whose parameters are $r_j \in [n]$ using the unidimensional index (see Section 2.3.1), and each r_j satisfies $r_j^2 = \alpha^2(1+\sqrt{n}x_j)$ where different x_j,x_k that do not correspond to conjugate positions are chosen independently from the distribution $\Gamma(2,1)$.*[2]

Practical applications [4, 29, 46] usually deal with variants of the problem:

- *discrete b*: Instead of working with an error distribution ψ over $K_{(T),\mathbb{R}}$, the m-RLWE distribution $A_{s,\chi}$ can use χ as a discrete error distribution over $R^\vee$, so that the element b belongs to $R_q^\vee$.
- *small key*: Instead of a uniform s, s can be a "short key" equivalently sampled from the error distribution (this is known as “normal form” in [45]), with equivalent security. Given a list of l m-RLWE samples, s can be substituted with the error e of any sample (a,b) whose term a is invertible in R_q, which occurs with constant probability by Claim 1 below.
- *power of 2 cyclotomic*: Instead of sampling a and s from R_q and $R_q^\vee$ respectively, both are usually sampled from R_q (this is usually known as the non-dual variant). In general, the works which consider s in R_q deal with cyclotomic fields where m_i is a power of two. It can be shown that for this particular type of cyclotomic fields both definitions are equivalent.
- *modulus switching*: The original definitions of the problem are presented with a prime modulus q that splits the space into small independent coordinates. With the same hardness guarantees, it is possible to modulus-switch to other compute-friendly modulus at the price of a slight increase of the error.

[45] shows that the variant of RLWE with discrete and short error (R-DLWE$_{q,\chi}$) is as hard as the original R-DLWE$_{q,\psi}$, by following the technique from [58]. These results can be adapted to our more general case as follows:

Claim 1. *The fraction of invertible elements in R_q is $\bigotimes_{i\in[l]} \mathcal{O}_{K_i}/\langle q\rangle$, for prime $q = 1 \bmod \phi(m_i)$ for all i is $(1-\frac{1}{q})^n$, with $n = \prod_i \phi(m_i)$. Thus, if $q \geq n$, this probability is constant.*

[1] $U(R_q^\vee)$ represents the uniform distribution over $R_q^\vee$.

[2] $\Gamma(2,1)$ refers to the Gamma distribution with shape 2 and rate 1.

Proof. Since R_q is in bijection with the ring $(\mathbb{Z}/q\mathbb{Z})^n$ via the tensor embedding mod q, an element is invertible iff its image does not contain any zero. Hence, there are $(q-1)^n$ invertible elements out of q^n. □

Pseudorandomness of m-RLWE: To show that the m-RLWE distribution is pseudorandom (that is, there exists a reduction from the search problem to the decision variant of the hardness problem) we rely on the results from [41], applied to the case of multivariate elements. The main needed properties are those related to the decomposition of $\langle q \rangle$ into n prime ideals ($q \equiv 1 \bmod \phi(m_i)$ for all i) and the use of the automorphisms that permute the prime ideals.

2.4. Proposed approach for secure multivariate rings

Despite the efficiency benefits of multivariate RLWE, its security can be much smaller than originally expected for those instances vulnerable to Bootland *et al.*'s attack [44]. This motivates us to redefine the set of instantiations that preserve the security in the tensor lattice dimension.

This section enumerates those secure instantiations of multivariate RLWE. With this in mind, we first briefly revise the choice of "coprime" order cyclotomics explicitly included in [45]. Afterwards, we discuss the possibility of using a more general set of number fields, enabling other multivariate rings that can be more convenient for practical applications.

2.4.1. Multivariate RLWE as a subset of RLWE

The m-th cyclotomic field $K = \mathbb{Q}(\zeta_m) \cong \mathbb{Q}[x]/(\Phi_m(x))$ (with ζ_m the m-th root of unity) is isomorphic to the multivariate field

$$K \cong \mathbb{Q}[x_1, \ldots, x_l]/(\Phi_{m_1}(x_1), \ldots, \Phi_{m_l}(x_l)), \tag{2.4}$$

where $m = \prod_i m_i$ is decomposed in its prime-power decomposition with $\gcd(m_j, m_k) = 1$ for all $j \neq k$.

This fact gives an alternative basis to the power basis $\{1, x, \ldots, x^{\phi(m)-1}\}$ for the ring of integers $R = \mathbb{Z}[x]/\Phi_m(x)$; this basis is the "powerful" basis of K composed of elements $\prod_i x_i^{j_i}$ with $0 \leq j_i < \phi(m_i)$.[3] This "powerful" basis has some very nice properties [45] which make it more appealing than the more "conventional" power basis. Additionally the authors of [45] provide a detailed analysis on how the performance of ring operations can be improved by means of this multivariate structure.

Besides [45], the use of the multivariate structure in Eq. (2.4) has been exploited to enhance polynomial operations in both the HElib [59, 60] and the MHEAAN [57] libraries. This gives us a first approach to deal with multivariate instantiations which do not suffer a decrease of the underlying lattice dimension. However, this structure is not flexible enough to convey the same benefits that general multivariate structures can achieve; in particular, it cannot preserve the interesting structure of power-of-two cyclotomics ($1 + x^n$).

[3] This basis does not coincide with the power basis under the mentioned automorphism and considering the map $x^{\frac{m}{m_i}} \rightarrow x_i$ for $i = 1, \ldots, l$ (see [45]).

2.4.2. More general RLWE instantiations

We look now beyond cyclotomics, into more general and flexible number fields and their parameterization. We first introduce the definitions of RLWE over any number field [53], and then give the intuition on the properties required to resist the Bootland *et al.*'s attack. A detailed discussion on the choice of good parameters and the security of RLWE on these number fields follows in Sections 2.5, 2.6 and 2.7.1.

RLWE over any number field

Peikert *et al.* [53] have recently generalized the RLWE problem to *any number field*. Let K be a number field with ring of integers $R = \mathcal{O}_K$; let $R^\vee$ be the fractional codifferent ideal of K, and let $\mathbb{T} = K_\mathbb{R}/R^\vee$. Let $q \geq 2$ be a (rational) integer modulus, and for any fractional ideal $\mathcal{I}$ of K, let $\mathcal{I}_q = \mathcal{I}/q\mathcal{I}$. We include now the relevant definitions of RLWE over any number field that we use in our formulation.

Definition 7 (Ring-LWE Distribution, Definition 2.14 in [53]). *For $s \in R_q^\vee$ and an error distribution ψ over $K_\mathbb{R}$, the R-LWE distribution $A_{s,\psi}$ over $R_q \times \mathbb{T}$ is sampled by independently choosing a uniformly random $a \leftarrow R_q$ and an error term $e \leftarrow \psi$, and outputting $(a, b = (a \cdot s)/q + e \bmod R^\vee)$.*

Definition 8 (Ring-LWE, Average-Case Decision, Definition 2.15 in [53]). *Let Υ be a distribution over a family of error distributions, each over $K_\mathbb{R}$. The average-case Ring-LWE decision problem, denoted R-LWE$_{q,\Upsilon}$, is to distinguish (with non-negligible advantage) between independent samples from $A_{s,\psi}$ for a* random *choice of $(s, \psi) \leftarrow U(R_q^\vee) \times \Upsilon$, and the same number of uniformly random and independent samples from $R_q \times \mathbb{T}$.*

Theorem 1 (Theorem 6.2 from [53]). *Let K be an arbitrary number field of degree n and $R = \mathcal{O}_K$. Let $\alpha = \alpha(n) \in (0,1)$, and let $q = q(n) \geq 2$ be an integer such that $\alpha q \geq 2 \cdot \omega(1)$. There is a polynomial-time quantum reduction from $K - DGS_\gamma$ to (average-case, decision) R-LWE$_{q,\Upsilon_\alpha}$, for any*

$$\gamma = \max\left\{\eta(\mathcal{I}) \cdot \sqrt{2}/\alpha \cdot \omega(1), \sqrt{2n}/\lambda_1(\mathcal{I}^\vee)\right\}.$$

Additionally, it is worth highlighting some observations regarding the choice of a particular number field in RLWE, as stated in [53]:

- The geometry of the dual ideal $R^\vee$ affects the error rate α (chosen to be smaller than the minimum distance $\lambda_1(R^\vee)$). As α decreases, worst-case hardness theorems give weaker guarantees (i.e., larger approximation factors), and known attacks on Ring-LWE become more efficient.

- A similar phenomenon arises for rings with large "expansion factors" (see [61]) which imposes smaller α for achieving correct decryption; hence, good rings for practical applications have small expansion factors.

- Besides the two previous relations, there is no practical evidence on which particular number field is better in terms of security.

Ad-hoc countermeasures to Bootland *et al.*'s attack

Bootland's attack [44] shows that a reduced RLWE sample is at least as hard as an m-RLWE sample. To prove the converse, we can use an oracle for m-RLWE. With access to such oracle and a set of RWLE samples with different keys, we can construct an m-RWLE sample (with a slight increase in the noise variance) by means of the reverse map of Bootland *et al.*'s attack (i.e., $\tilde{\Theta}^{-1}$). Once this oracle returns the secret key of the m-RLWE sample, the original keys of the RLWE sample can be recovered by means of the map $\tilde{\Theta}$.

We can therefore express the security of m-RLWE in terms of RLWE, but the decrease of the involved dimension considerably reduces the applicability of the problem with "non-coprime" modular functions. The security of $\prod_{j\neq k}\phi(\gcd(m_j,m_k))$ independent RLWE samples with dimension $\frac{\prod_{i\in[l]}\phi(m_i)}{\prod_{j\neq k}\phi(\gcd(m_j,m_k))}$ could be reduced to that of one m-RLWE sample (according to Definition 2) with dimensions $\{\phi(m_1),\ldots,\phi(m_l)\}$:

Proposition 1 ($\tilde{\Theta}^{-1}$ transform from [44] **).** *Let L independent univariate RLWE samples $(a_i,b_i)\in R_q\times\mathbb{T}$ for $i\in[L]$ and dimension n. We can transform (this transformation is invertible when q is prime) these L samples by means of the (inverse) of Bootland* et al.*'s attack into one m-RLWE sample with l dimensions $\{\phi(m_1),\ldots,\phi(m_l)\}$(see Definition 2) satisfying $L=\prod_{j\neq k}\phi\left(\gcd(m_j,m_k)\right)$ and having for the RLWE sample $n=\frac{\prod_{i\in[l]}\phi(m_i)}{L}$. This transformation slightly increases the variance of the error distribution by a factor L.*

The decrease in the lattice dimension by a factor $L=\prod_{j\neq k}\phi\left(\gcd(m_j,m_k)\right)$ brings about the question of whether we can *modify some of the multivariate RLWE constructions* where $L>1$ to *avoid Bootland et al.'s attack.*

Followed strategy

By considering instantiations satisfying $\gcd(m_j,m_k)=1$ for all $j\neq k$, we straightforwardly go back again to the RLWE problem. However, we would like to find other instantiations where the modular functions can have a similar form and degree. We will hence focus on modular functions as follows: $\{x^{n_x}+d_x, y^{n_y}+d_y, z^{n_z}+d_z,\ldots\}$, which can avoid Bootland's attack for certain parameters, while enabling NTT-like fast transforms and preserving the advantages of the originally introduced m-RLWE constructions.

However, the security of these instantiations is not based on any specific formulation of the RLWE problem, so we do not have a clear way of choosing the right parameters. In the next two sections, we focus on number fields satisfying Definition 7 and whose ring of integers (and their dual) has the aforementioned structure. In particular, we focus on multiquadratic number fields (Section 2.5) and more general multivariate rings (Section 2.6).

2.5. Multiquadratic Rings

Let $K=\mathbb{Q}(\sqrt{d_i})$ be a field with prime d_i (hence squarefree) satisfying $d_i=1 \bmod 4$; its ring of integers is $\mathcal{O}_K=\mathbb{Z}\left[\frac{1+\sqrt{d_i}}{2}\right]$ with basis $\{1,\frac{1+\sqrt{d_i}}{2}\}$ and discriminant $\Delta_K=d_i$, then we can also represent $\mathcal{O}_K$ as a polynomial ring $\mathbb{Z}[x]/x^2-x+\frac{1-d_i}{4}$ ($\mathcal{O}_K$ is free of rank 2), according to (see Proposition 2):

Proposition 2 (Proposition 2.24 from [62]). *Let $K = \mathbb{Q}(\sqrt{d})$ be a quadratic field with d a squarefree integer. If $d \equiv 2, 3(\mathrm{mod}4)$, then $\mathcal{O}_K = \mathbb{Z}\left[\sqrt{d}\right] \simeq \mathbb{Z}[x]/(x^2 - d)$ and $\mathcal{O}_K$ is free of rank 2 over $\mathbb{Z}$ with basis $\{1, \sqrt{d}\}$. If $d \equiv 1(\mod 4)$, then $\mathcal{O}_K = \mathbb{Z}\left[\frac{1+\sqrt{d}}{2}\right] \simeq \mathbb{Z}[x]/(x^2 - x + \frac{1-d}{4})$ and $\mathcal{O}_K$ is free of rank 2 over $\mathbb{Z}$ with basis $\{1, \frac{1+\sqrt{d}}{2}\}$.*

Let us extend the field to $\mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_l})$ (a multiquadratic field), with $\gcd(d_1, \ldots, d_l) = 1$, but still sticking to the case $d_i = 1 \bmod 4$. Taking into account that $\mathcal{O}_K\mathcal{O}_{K'} = \mathcal{O}_F$ when $\gcd(\Delta_K, \Delta_{K'}) = 1$, where F is the compositum over $\mathbb{Q}$ (see [63]) of two subfields $K = \mathbb{Q}(\sqrt{d_1})$ and $K' = \mathbb{Q}(\sqrt{d_2})$ (see [64]), we have that $\mathcal{O}_F = \mathbb{Z}\left[\frac{1+\sqrt{d_1}}{2}, \frac{1+\sqrt{d_2}}{2}\right]$. This can be generalized to the case of a field with l "coprime" squares, whose resulting ring of integers is

$$\mathcal{O}_K = \mathbb{Z}\left[\frac{1+\sqrt{d_1}}{2}\right] \cdot \ldots \cdot \mathbb{Z}\left[\frac{1+\sqrt{d_l}}{2}\right]. \tag{2.5}$$

Therefore, as all d_i are different primes, the discriminants of $\mathbb{Q}(\sqrt{d_i})$ are also coprime, which implies that the ring of integers can be expressed as the product of the respective univariate rings of integers.

However, the definition of RLWE (see Definition 8) works on the dual of the ring of integers, due to its geometric properties. The dual can be obtained through Theorem 2:

Theorem 2 (Theorem 3.7 from [65]). *Let $K = \mathbb{Q}(\alpha)$ and let $f(T)$ be the minimal polynomial of α in $\mathbb{Q}[T]$. Write*

$$f(T) = (T - \alpha)(c_0(\alpha) + c_1(\alpha)T + \ldots + c_{n-1}(\alpha)T^{n-1}),\ c_i(\alpha) \in K.$$

The dual basis to $\{1, \alpha, \ldots, \alpha^{n-1}\}$ relative to the trace product is

$$\left\{\frac{c_0(\alpha)}{f'(\alpha)}, \frac{c_1(\alpha)}{f'(\alpha)}, \ldots, \frac{c_{n-1}(\alpha)}{f'(\alpha)}\right\}.$$

In particular, if $K = \mathbb{Q}(\alpha)$ and $\alpha \in \mathcal{O}_K$ then

$$(\mathbb{Z} + \mathbb{Z}\alpha + \ldots + \mathbb{Z}\alpha^{n-1})^\vee = \frac{1}{f'(\alpha)}(\mathbb{Z} + \mathbb{Z}\alpha + \ldots + \mathbb{Z}\alpha^{n-1}).$$

Particularized to the quadratic case, Theorem 2 says that whenever the ring of integers has a power basis, the basis of the dual is

$$\left\{1, \frac{1+\sqrt{d_i}}{2}\right\}^\vee = \left\{\frac{1}{f'(\alpha)}, \frac{1}{f'(\alpha)}\frac{1+\sqrt{d_i}}{2}\right\}, \tag{2.6}$$

where $f(x) = x^2 - x + \frac{1-d}{4}$ and $\alpha = \frac{1+\sqrt{d}}{2}$, so $f'(x) = 2x - 1$; evaluated at $x = \alpha = \frac{1+\sqrt{d_i}}{2}$, it satisfies $f'(\alpha) = \sqrt{d_i}$.

As dual commutes with tensoring, this result can be straightforwardly extended to the compositum case with several d_i. Additionally, we see that we can go from the dual to $\mathcal{O}_K$ by just scaling with $\sqrt{d_i}$ (or multiplying with the polynomial $2x - 1$).

Following our requirements, we need a ring of the form $\mathbb{Z}[x_1, \ldots, x_l]/(x_1^2 - d_1, \ldots, x_l^2 - d_l)$, which is an *order* of the field $\mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_l})$, but not necessarily its ring of integers and a Dedekind domain.[4] However, we can only base its security on RLWE defined on a number field of the form $\mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_l})$ (see Definition 7) and its ring of integers satisfying $\mathbb{Z}[x_1, \ldots, x_l]/(x_1^2 - x_1 + \frac{1-d_1}{4}, \ldots, x_l^2 - x_l + \frac{1-d_l}{4})$. We will therefore show that we can define an invertible map modulo q from the ring $\mathcal{O}_K$ (and its dual $\mathcal{O}_K^\vee$) to the ring $\mathbb{Z}[x_1, \ldots, x_l]/(x_1^2 - d_1, \ldots, x_l^2 - d_l)$, while still basing its security on the original RLWE formulation from Definition 7. Additionally, this map does not significantly increase the noise; in fact, it also decorrelates it in the coefficient domain, enabling direct sampling of the noise in the coefficient representation with an independent error distribution.

The map, applied to each variable x_i, works as follows:

- We apply the change of variable $x \to \frac{x+1}{2}$.
- We multiply the sample by a factor 2.

This mapping can be applied whenever the inverse of 2 exists modulo q. The multiplication by 2 is applied afterwards to avoid the potentially high distortion introduced by the factor $\frac{1}{2}$ into the noise.

Canonical Embedding

Let $K = \mathbb{Q}(\sqrt{d})$, and note that $\frac{1}{2x-1}$ evaluated at $x = \frac{1+\sqrt{d}}{2}$ equals $\frac{1}{\sqrt{d}}$. We define the Embedding map $\mathcal{E}$ going from $\mathcal{O}_K^\vee \cong \frac{1}{\sqrt{d}}\mathbb{Z}[x]/x^2 - x + \frac{1-d}{4}$ to $\mathbb{C}^2$, as the substitutions $\{x \leftarrow \frac{1+\sqrt{d}}{2}, \sqrt{d} \leftarrow \sqrt{d}\}$ and $\{x \leftarrow \frac{1-\sqrt{d}}{2}, \sqrt{d} \leftarrow -\sqrt{d}\}$. This gives this transformation matrix for $\mathcal{E}$

$$\frac{1}{\sqrt{d}} \begin{pmatrix} 1 & \frac{1+\sqrt{d}}{2} \\ -1 & \frac{\sqrt{d}-1}{2} \end{pmatrix}. \tag{2.7}$$

The inverse map $\mathcal{E}^{-1}$ is defined as the product with matrix

$$\begin{pmatrix} \frac{\sqrt{d}-1}{2} & -\frac{1+\sqrt{d}}{2} \\ 1 & 1 \end{pmatrix}. \tag{2.8}$$

Sampling the error directly in the coefficient domain

Finally, if we define the noise in the embedding of the dual ring as two independent Gaussian variables $e_0, e_1 \in \chi$ with variance σ^2, we have in the ring $\frac{1}{x}\mathbb{Z}[x]/x^2 - d$ after following the whole "processing chain":

$$\frac{1}{x}\left(\underbrace{(e_0 + e_1)}_{2\sigma^2} x + \underbrace{\sqrt{d}(e_0 - e_1)}_{2d\sigma^2}\right) \bmod x^2 - d.$$

Hence, the noise is not correlated in the coefficient domain and we can easily sample the error distribution considering an appropriate variance per coefficient.

For simplicity, we have focused on a quadratic field, but the embedding can be extended to the multiquadratic case by means of the Kronecker product.

[4] A recent work [66] discusses the hardness of a generalization of Ring-LWE called Order-LWE.

Multiquadratic RLWE

Let us define the multiquadratic version of m-RLWE, where all the modular functions have the form $f_i(x_i) = d_i + x_i^2$, as

Definition 9 (multivariate polynomial RLWE with quadratic modular functions). *Given a multivariate polynomial ring $R_q^\vee[x_1, \ldots, x_l]$ with $f_j(x_j) = d_j + x_j^2$ for $j = 1, \ldots, l$ where $l = \log_2 n$ (with n a power of two) and an error distribution $\chi[x_1, \ldots, x_l] \in R_q^\vee[x_1, \ldots, x_l]$ that generates small-norm random multivariate polynomials in $R_q^\vee[x_1, \ldots, x_l]$, the multivariate polynomial RLWE relies upon the computational indistinguishability between samples $(a_i, b_i = a_i \cdot s + e_i)$ and (a_i, u_i), where $a_i \leftarrow R_q[x_1, \ldots, x_l]$, $u_i \leftarrow R_q^\vee[x_1, \ldots, x_l]$ are chosen uniformly at random from the rings $R_q[x_1, \ldots, x_l]$ and $R_q^\vee[x_1, \ldots, x_l]$; and $s, e_i \leftarrow \chi[x_1, \ldots, x_l]$ are drawn from the error distribution (see Section 2.5).*

The security reduction from Theorem 1 applies to this particular version of the m-RLWE problem whenever $-d_i = 1 \bmod 4$ and $\gcd(\Delta_K, \Delta_{K'}) = 1$. Section 2.7.1 gives further insights on the security and practicality of the chosen parameterization, and exemplifies it with a concrete instantiation. In particular, Proposition 6 gives a sufficient condition to consider the problem secure against known attacks.

Comparison with Gaussian integers

We now compare the multiquadratic RLWE with the particular case of power-of-two cyclotomics m-RLWE (see Definition 1) where all the used modular functions have the same form $f_i(x_i) = 1 + x_i^2$, as originally proposed in [5] (see Appendix A):

Definition 10 (multivariate polynomial RLWE with $\Phi_4(\cdot)$ as modular functions). *Given a multivariate polynomial ring $R_q[x_1, \ldots, x_l]$ with $f_j(x_j) = 1 + x_j^2$ for $j = 1, \ldots, l$ where $l = \log_2 n$ (with n a power of two) and an error distribution $\chi[x_1, \ldots, x_l] \in R_q[x_1, \ldots, x_l]$ that generates small-norm random multivariate polynomials in $R_q[x_1, \ldots, x_l]$, the multivariate polynomial RLWE relies upon the computational indistinguishability between samples $(a_i, b_i = a_i \cdot s + e_i)$ and (a_i, u_i), where $a_i, u_i \leftarrow R_q[x_1, \ldots, x_l]$ are chosen uniformly at random from the ring $R_q[x_1, \ldots, x_l]$; and $s, e_i \leftarrow \chi[x_1, \ldots, x_l]$ are drawn from the error distribution.*

The comparison of our secure multiquadratic RLWE samples with RLWE samples from Definition 10 is specially relevant, as the latter are severely affected by Bootland *et al.*'s attack. Samples from Definition 10 can be reduced to a dimension of 2, by applying the map $\tilde{\Theta}$ a total of $(\log_2 n - 1)$ times, yielding $n/2$ RLWE samples with $f(x) = 1 + x^2$ and error variance $n/2$ times higher than the original m-RLWE sample; this can be very easily solved. Consequently, despite of the efficiency of the polynomial operations on the rings instantiated according to Definition 10, they are not valid for cryptographic applications. Meanwhile, the samples from a secure instantiation of multiquadratic RLWE (Definition 9) preserve the lattice dimension n and withstand Bootland's attack.

Another advantage of the multiquadratic RLWE problem is that it also enables very efficient polynomial operations, without decreasing security. In particular, it is possible to apply a variant of the Fast Walsh-Hadamard transform (over finite rings instead of the usual real numbers), featuring a convolution property that relates the coefficient-wise representation with the transformed domain. This transform can be very efficiently computed with FFT-like algorithms (specifically, a variant of the Fast Walsh-Hadamard transform) whose computational cost is only $\mathcal{O}(n \log n)$ additions and $\mathcal{O}(n)$ products, hence considerably speeding up practical implementations. For more details, we refer the reader to the Section 3.2 from Chapter 3, where we show how the well-known asymptotic cost of $\mathcal{O}(n \log n)$ for cyclotomic rings with polynomials of n coefficients can be improved by a factor of $\log n$ in terms of elemental multiplications.

2.6. More general multivariate rings

Let us consider now general fields $\mathbb{Q}(a_1^{1/n}, \ldots, a_l^{1/n})$, for which the a_i are squarefree and coprime, but for simplicity we will assume that they are independent primes. The results shown in the previous section for multiquadratics cannot be straightforwardly generalized to these fields, as the individual univariate fields $\mathbb{Q}(a_i^{1/n})$ can easily have common factors in their discriminants (i.e., be non-coprime), in such a way that finding a basis for the multivariate ring of integers is not trivial.

We explain the followed path that leads to our definition of valid, secure and easily parameterizable multivariate rings. We start by choosing number fields whose ring of integers $\mathcal{O}_K$ can be represented as $\mathbb{Z}[x]/x^n + ax + b$, that is, as polynomial rings whose modular function has the form $x^n + ax + b$. For this to be a valid ring $\mathcal{O}_K$ for K, it has to be irreducible over $\mathbb{Q}$, for which we use Eisenstein's criterion:

Proposition 3 (Eisenstein's criterion [67]). *The polynomial $p(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$, where $a_i \in \mathbb{Z}$ for all $i = 0, \ldots, n$ and $a_n \neq 0$ (which means that the degree of $p(x)$ is n) is irreducible if some prime number p divides all coefficients $a_0, \ldots, a_{n-1}$, but not the leading coefficient a_n and, moreover, p^2 does not divide the constant term a_0.*

Therefore, we impose the following two conditions on $f(x) = x^n + ax + b$:

- Both a and b have to be divisible by a prime p and not by p^2 (Eisenstein's criterion).
- If we choose b as a prime, a has to be divisible by b.

Now, we can compute the discriminant for this number field by resorting to [68, Chapter 2.7]:

Proposition 4 (An example of the calculation of a discriminant [68]). *For the calculation of Δ_K in a number field $K = \mathbb{Q}(x)$ being a extension of finite degree n of $\mathbb{Q}$ and $f(x) = x^n + ax + b$ the minimal polynomial of x over $\mathbb{Q}$, we obtain*

$$\Delta_K = (-1)^{\frac{n(n-1)}{2}}(n^n b^{n-1} + (-1)^{n-1}(n-1)^{n-1}a^n). \tag{2.9}$$

For $n = 2$ (respectively, 3) we rediscover the well-known expressions $a^2 - 4b$ (respectively, $-27b^2 - 4a^3$).

Theorem 3 (Theorem 8.11 from [69]). *For $\mathbb{Z}$-lattices $\mathcal{L}' \subset \mathcal{L}$ inside K, $[\mathcal{L}' : \mathcal{L}]^2 < \infty$ and*

$$disc_{\mathbb{Z}}(\mathcal{L}') = [\mathcal{L}' : \mathcal{L}]^2 \cdot disc_{\mathbb{Z}}(\mathcal{L}).$$

In particular, if $\mathcal{L}' \subset \mathcal{O}_K$ and the integer $disc_{\mathbb{Z}}(\mathcal{L}') \in \mathbb{Z} - \{0\}$ is squarefree then $[\mathcal{O}_K : \mathcal{L}'] = 1$; i.e., $\mathcal{L}' = \mathcal{O}_K$.[5]

[5] In [69], $\mathrm{disc}_{\mathbb{Z}}(\mathcal{L})$ denotes the determinant of the basis of $\mathcal{L}$ for any $\mathbb{Z}$-basis of $\mathcal{L}$.

If we choose values for a and b such that Δ_K is squarefree, Theorem 3 guarantees that the ring of integers has a power basis of the form $\{1, \alpha, \alpha^2, \ldots\}$, with α a root of x^n+ax+b. Consequently, $\mathbb{Z}[x]/x^n + ax + b$ is a valid ring of integers.

By including more "univariate" subrings, $\mathbb{Z}[x_1, \ldots, x_l]/(x_1^n + a_1x + b_1, \ldots, x_l^n + a_lx + b_l)$ becomes a valid ring of integers when all the discriminants are coprime [64]. Therefore, this is a feasible strategy to define RLWE over a multivariate ring, as the product of univariate rings with modular functions $x^n + a_ix + b_i$.[6]

Finding valid parameters for $f(x) = x^n + ax + b$: Unfortunately, the two previous conditions (Eisenstein's criterion from Proposition 3 and Theorem 3) cannot be satisfied at the same time:

- To satisfy the Eisenstein's criterion, b and a have to be divisible by at least a prime p (i.e., $\gcd(a, b) = u \cdot p$ for some $u \in \mathbb{Z}$), this introduces a factor p^{n-1} in Δ_K (see Equation (2.9)), in such a way that Δ_K is not squarefree and not satisfying $[\mathcal{O}_K : \mathcal{L}'] = 1$ in Theorem 3.

  We could still work with these multivariate rings provided that their discriminants are coprime, but it seems that there is no straightforward way to determine the "powerful" basis of the ring of integers: starting from Proposition 4, it is known that $\mathbb{Z}[\alpha] \subseteq \mathcal{O}_K \subseteq \frac{1}{\Delta_K}\mathbb{Z}[\alpha]$ where $f(\alpha) = 0$.

- Additionally, Eisenstein's criterion is a sufficient but *not necessary* condition for irreducibility of the modular functions. Without the imposed restrictions, we could search for squarefree and coprime discriminants, but we would have to verify the irreducibility of the involved functions case-by-case. Nevertheless, this is not impossible to find, as it is known that monogenic fields are not scarce [70]; in fact, for random polynomials f, it has been conjectured that $\mathbb{Z}[x]/f(x)$ of degree ≥ 4 is a ring of integers with probability $\gtrsim 0.307$ [71].

Transformation based on Modulus Switching

Let us assume that we have found valid (monogenic) $x_i^n + a_ix_i + b_i$ functions defining the ring of integers $\mathbb{Z}[x_i]/x_i^n + a_ix_i + b_i$; they do not yet feature the desired $x^n + d$ form.

In order to achieve this, we consider a map from the original RLWE samples to RLWE samples modulo q, that removes the term ax if q divides a. It is worth noting that this transformation is nothing but a modulus switching to q, and if it were possible to break RLWE modulo q, the original secret key could be recovered or at least the indistinguishability assumption could be broken.

The trick relies on all the modular functions having the form $f_i(x_i) = x_i^n + \underbrace{a_i'q}_{a_i} x_i + b_i$. Hence, a reduction modulo q converts the modular functions into $f_i(x_i) = x_i^n + b_i$. We show the effect of this transformation on the ring $q\mathcal{O}_K^\vee$ for the univariate case (it extends to the multivariate case, as dual commutes tensoring):

- $\mathcal{O}_K^\vee$ is defined as $\frac{1}{f'(\alpha)}\mathcal{O}_K$; under the polynomial ring $\mathbb{Z}[x]/x^n + a_i'qx + b_i$, this implies that the dual is $\frac{1}{nx^{n-1}+a_i'q}\mathbb{Z}[x]/x^n + a_i'qx + b_i$.

[6] To define the dual $\mathcal{O}_K^\vee$ we can make use of Theorem 2 which states that whenever the ring of integers has a power basis, the basis of the dual is the same basis, scaled by $\frac{1}{f'(\alpha)} = \frac{1}{n\alpha^{n-1}+a}$, where α is a root of $f(x)$.

- After reducing modulo q, we obtain $\frac{1}{nx^{n-1}}\mathbb{Z}_q[x]/x^n + b_i$; considering that x has inverse modulo q, we can multiply numerator and denominator by x to obtain $\frac{x}{nx^n} = \frac{x}{-nb_i}$.
- The factor $\frac{1}{-nb_i}$ can be removed by just a scaling (moving to the ring of integers $\mathcal{O}_K$), so we can directly work on $\mathbb{Z}_q[x]/x^n + b_i$. This gives a "basis" $\{b_i, x, x^2, \ldots, x^{n-1}\}$ (or a basis $\{\frac{1}{n}, \frac{x}{nb_i}, \frac{x^2}{nb_i}, \ldots, \frac{x^{n-1}}{nb_i}\}$ without scaling).

Decodability of the transformed $x^n + ax + b$: Elias *et al.* [70] use an heuristic perturbation method to bound the spectral norm of the canonical embedding with $f(x) = x^n + ax + b$. As the condition number is stable for most of the random perturbations of the canonical embedding matrix associated to $x^n + 1$, they conjecture that many f functions have a bounded spectral norm in terms of a and b; therefore, we can consider that the spectral norm $s_1(N_f)$ (N_f represents the inverse of the canonical embedding matrix) is likely bounded by $\sqrt{\max(a, b)} \cdot \det(N_f)^{1/n}$ [72]. Consequently, the same arguments about noise behavior in [72, 55] still apply, and in order to guarantee the prevalence of the security reduction (see Proposition 6), the noise wraps around modulo q in some of the polynomial coefficients ($\max(a, b) \approx q$). This is due to the large q factor introduced in $f(x)$, which requires the use of a high error variance, rendering some of the polynomial coefficients modulo q useless. This makes these RLWE samples harder to use for cryptographic applications.

Valid and practical parameterizations for Multivariate Rings

The previous solutions to parameterize multivariate rings with modular functions x^n+d are not satisfactory, as (a) the search of valid univariate rings is not easy to handle (due to the impossibility of using Eisenstein's criterion) and (b) the obtained samples are not practical for cryptographic applications due to their high noise in some polynomial coefficients.

Here we follow a slightly different approach, releasing the condition on equal-degree modular functions; that is, we consider multivariate rings as $\mathbb{Z}[x_1, \ldots, x_l]/(x_1^{n_1} + d_1, \ldots, x_l^{n_l} + d_l)$. Again, to simplify the explanation we only consider an univariate ring with modular function $x^n + d$, but all the results can be analogously extended to the multivariate case (see Section 2.5) by requiring coprime discriminants.

First, for $f(x) = x^n + d$, Equation (2.9) simplifies to $\Delta_K = (-1)^{\frac{n(n-1)}{2}} n^n d^{n-1}$.

Let d be a prime number and $n = u^m$ a prime power. Then,

- $f(x)$ is an irreducible polynomial over $\mathbb{Q}$ by Eisenstein's criterion (Propostion 3).
- $f(x)$ is monogenic for d and n satisfying the following Proposition 5.

Proposition 5 (Adapted Proposition 3 from [70]). *Let n be a power of a prime u. If d is squarefree and u^2 does not divide $(-1)^n(d^{n-1} + 1)d$, then the polynomials $x^n + d$ are monogenic.*

Proposition 5 shows that $f(x)$ can be monogenic even when its discriminant is not squarefree. If $f(x)$ satisfies Proposition 5, we have $\mathcal{O}_K = \mathbb{Z}[x]/x^n + d$ and $\mathcal{O}_K^\vee = \frac{1}{nx^{n-1}}\mathbb{Z}[x]/x^n + d$.

In order to extend these results to multivariate rings $\mathbb{Z}[x_1, \ldots, x_l]/(x_1^{n_1} + d_1, \ldots, x_l^{n_l} + d_l)$, we only have to consider functions $\{x_1^{n_1} + d_1, \ldots, x_l^{n_l} + d_l\}$ satisfying Proposition 5 and having coprime discriminants. This basically means that all the d_i and n_i are respectively different primes and power primes.

Analogously to the *multiquadratic rings* in Section 2.5, we can directly map the error distribution in the coefficient domain. In particular, for the ring $\frac{1}{n_i x^{n_i-1}}\mathbb{Z}[x_i]/x_i^{n_i} + d_i$, the parameter for the error distribution in the $(j-1)$-th coefficient ($1 \leq j \leq n_i$) is given by $\sqrt{n_i} d_i^{\frac{n_i-j}{n_i}} r$, where r is the parameter of an independent spherical error distribution in the embedding domain [72]. This extends to multivariate rings by means of the Kronecker product. As the resulting embedding matrix is the Kronecker product of the embedding matrices associated to each univariate ring, the singular values are the result of the Kronecker product of the singular values for each univariate embedding matrix.

Finally, we introduce the definition of multivariate RLWE with the proposed modular functions $f_i(x_i) = d_i + x_i^{n_i}$:

Definition 11 (multivariate RLWE with modular functions as $x_i^{n_i} + d_i$)**.** *Given a multivariate polynomial ring $R_q[x_1, \dots, x_l]$ with $f_j(x_j) = d_j + x_j^{n_j}$ for $j = 1, \dots, l$ where $n = \prod_j n_j$ (where all n_j are prime powers) and an error distribution $\chi[x_1, \dots, x_l] \in R_q^\vee[x_1, \dots, x_l]$ that generates small-norm random multivariate polynomials in $R_q^\vee[x_1, \dots, x_l]$, the multivariate polynomial RLWE relies upon the computational indistinguishability between samples $(a_i, b_i = a_i \cdot s + e_i)$ and (a_i, u_i), where $a_i \leftarrow R_q[x_1, \dots, x_l]$, $u_i \leftarrow R_q^\vee[x_1, \dots, x_l]$ are chosen uniformly at random from the rings $R_q[x_1, \dots, x_l]$ and $R_q^\vee[x_1, \dots, x_l]$; $s, e_i \leftarrow \chi[x_1, \dots, x_l]$ are drawn from the error distribution.*

For the ring $R^\vee[x_1, \dots, x_l]$, we define $\chi[x_1, \dots, x_l]$ as the distribution generating polynomials belonging to $R^\vee[x_1, \dots, x_l]$ and whose parameter per coefficient satisfies $r \prod_{i \in [l]} \sqrt{n_i} d_i^{\frac{n_i - j_i}{n_i}}$, where $1 \leq j_i \leq n_i$ and $1 \leq i \leq l$, and hence represents the parameter for the coefficient associated to the monomial $x_1^{j_1-1} \cdot \dots \cdot x_l^{j_l-1}$.

Some examples of valid parameters: In order to show the feasibility of the proposed parameterization, we exemplify it with some practical use cases for bivariate RLWE; we will consider $n_1 = 2^{11} = 2048$ and $n_2 = 3^7 = 2187$, and $d_1 = 5$, $d_2 = 7$, for which we prove that they meet the conditions of Proposition 5

- $2^2 = 4$ does not divide $5^{2047} + 1$, or equivalently, $5^{2047} + 1 \neq 0 \bmod 4$. We have $5^{2047} + 1 \bmod 4 = 1^{2047} + 1 = 2 \neq 0$.
- $3^2 = 9$ does not divide $7^{3^7-1} + 1$, or equivalently, $7^{3^7-1} + 1 \neq 0 \bmod 9$. We have $7^{3^7-1} + 1 = 7^{-1}7^{3^7} + 1 = 7^{-1}7^{3^7 \bmod 6} + 1 = 7^2 + 1 = 50 = 5 \bmod 9 \neq 0$.

Consequently, with this choice of parameters we can work on the number field $K = \mathbb{Q}((-5)^{1/2048}, (-7)^{1/2187})$, with $\mathcal{O}_K = \mathbb{Z}[x,y]/(x^{2048} + 5, y^{2187} + 7)$ and $\mathcal{O}_K^\vee = \frac{1}{4478976 x^{2047} y^{2186}} \mathcal{O}_K$.

As for the example mentioned in the introduction, with functions $x^{64} + 1$ and $y^{27} + 5$, we can also verify that

- $x^{64} + 1$ is the $\Phi_{128}(x)$ power-of-two cyclomic, hence it is monogenic.
- $y^{27} + 5$ is monogenic by Proposition 5, as $3^2 = 9$ does not divide 5 or $5^{26} + 1$.

Additionally, as both discriminants are coprime, the product is directly the corresponding ring of integers.

2.7. Security of multivariate RLWE and example instantiations

This section includes a discussion on several aspects of the proposed solutions in this chapter, namely their security, the geometric interpretation of the problem, and the feasibility of the proposed parameterizations. With this purpose, we enumerate the known attacks in the literature and include an example instantiation of a simple bivariate RLWE scheme. Finally, we summarize some of the applications that our constructions enable. For a discussion on a more advanced set of efficiency improvements on cryptographic primitives we refer the reader to Chapter 3.

2.7.1. Resilience against known attacks

The formulation proposed in this chapter involves working with rings whose modular function is $x^n + d$ or, more generally, $x^n + ax + b$. Some particular instantiations of these rings have already been studied in the literature and we can find specific attacks to "variants" of the RLWE problem (e.g., PLWE together with non-dual and dual RLWE versions) defined over them.

In general, the known attacks can be divided in two main types [55]:

- Attacks using a reduction modulo an *ideal divisor* $\mathfrak{q}$ of the modulus qR [73, 74, 70, 75, 76, 77]. These attacks try to distinguish between the error distribution and the uniform distribution modulo an *ideal divisor*.
- A reduction to *errorless* LWE [72] which exploits the relation between RLWE and LWE. Expressing RLWE in its LWE form, the error term of some of the equations can be removed by means of a rounding operation, and linear algebra can be used to search for the secret key.

All these attacks have been generalized and studied in depth by Peikert in [55], where he concludes that *all the concrete insecure RLWE instantiations made use of error distributions which were insufficiently well spread relative to the rings*, meaning that none of the vulnerable instantiations satisfy the conditions from Theorem 1 to have worst-case hardness. In [55], Peikert also gives sufficient conditions to make RLWE secure against the previous attacks. We summarize the main relevant results for our constructions.

Proposition 6 (Invulnerability condition from [55]). *Let $\psi = D_r$ (see Definition 7) be a spherical Gaussian error distribution over $K_{\mathbb{R}}$ for some $r > 0$; a sufficient condition for invulnerability to the attacks from [73, 74, 70, 75, 72, 55, 76] is*

$$r \geq 2. \tag{2.10}$$

The validity of Proposition 6 to resist the previous attacks is shown in the following two theorems: Theorem 4 (for the attack based on reduction modulo an ideal divisor) and Theorem 5 (for the attack based on errorless LWE).

Theorem 4 (Theorem 5.2 from [55]). *Given a Ring-LWE sample $(a, b = s \cdot a + e) \in R_q \times K_{\mathbb{R}}/qR^{\vee}$ where $e \leftarrow D_r$ is transformed into n LWE samples $(\boldsymbol{A}_a, \boldsymbol{b} = \boldsymbol{s}^T \boldsymbol{A}_a + \boldsymbol{e}^T)$, where $\boldsymbol{b} \in (\mathbb{R}/q\mathbb{Z})^n$*

and $\mathbf{e} \in \mathbb{R}^n$ are respectively the coefficient vectors of $b \in K_{\mathbb{R}}/qR^\vee$ and $e \in K_{\mathbb{R}}$ (with respect to the chosen basis of $R^\vee$), and $\boldsymbol{A}_a \in \mathbb{Z}_q^{n \times n}$ is the matrix of multiplication by $a \in R_q$ with any element of $R_q^\vee$ (with respect to the chosen bases of $R, R^\vee$). Then, for any $\mathbb{Z}$-basis $B^\vee = (b_j^\vee)$ of $R^\vee$ used above, each entry of $\mathbf{e}$ is a continuous Gaussian of parameter at least $r\sqrt{n} \geq 2\sqrt{n}$ (which is the required lower bound to apply the worst-case hardness theorems for plain-LWE).

Theorem 5 (Theorem 5.1 from [55])**.** *Let $\mathfrak{q} \subseteq R$ be any ideal of norm $N(\mathfrak{q}) \leq 2^n$, and let the error parameter $r \geq 2$ satisfy condition* (2.10)*. Then the reduced error distribution $D_r \bmod \mathfrak{q}R^\vee$ is within statistical distance 2^{-2n} of uniform over $K_{\mathbb{R}}/\mathfrak{q}R^\vee$.*

2.7.2. Geometric interpretation and examples of multivariate RLWE

In this section, we give a high level overview of how to instantiate a secure multivariate RLWE sample from Definition 11, exemplifying it in the bivariate case (all rings are defined over variables x, y, omitted when unambiguous).

We also use this example as a means to showcase complex numbers packing into slots, obtaining a net improvement on the number of available slots per ciphertext when comparing to the recent results in [56]. For the sake of clarity, we introduce a simple SHE scheme which enables homomorphic additions and multiplications without taking into account some of the more advanced techniques typically considered in the literature (see Appendix 2.A for a brief explanation of the possible optimizations).

A multivariate RLWE sample

For simplicity, we consider a bivariate RLWE sample $(a, b = a \cdot s + e) \in R_q \times R_q^\vee$, where $a \in R_q[x, y]$, $s \in R_q^\vee[x, y]$ and $e \leftarrow \chi[x, y] \in R^\vee[x, y]$. We can use a uniformly random s or follow conventional approaches where s is a small key (see Section 2.3).

Geometry of R, its dual $R^\vee$ and an example for $\{x^2 + 3, y^2 - 5\}$: To easily illustrate the geometry of R and $R^\vee$, we use a simple example $R = \mathbb{Z}[x, y]/(x^2 + 3, y^2 - 5)$. By means of the canonical embedding, we know that the substitutions $\{x \leftarrow \pm\sqrt{-3}, y \leftarrow \pm\sqrt{5}\}$ yield the four different *slots* in the embedding domain.

This clearly shows that $\lambda_1(R) \leq \sqrt{n} = 2$ by the embedding of 1, and we can also obtain the embedding of the elements x, y and xy. xy can be used to obtain an upper-bound for $\lambda_4(R)$, such that $\lambda_4(R) \leq 2\sqrt{15}$.

This is easily generalizable to any multiquadratic with $l = \log_2 n$ variables, by considering the embedding of 1 and $\prod_{i \in [l]} x_i$, obtaining $\lambda_1(R) \leq \sqrt{n}$ and $\lambda_n(R) \leq \sqrt{n} \prod_{i \in [l]} \sqrt{d_i}$. As the l-th prime is asymptotically $p_l \sim l \log l$, a worst-case for $l = \log_2 n$ is $d_l^l \sim l^l (\log l)^l = (\log_2 n)^{\log_2 n} (\log_2 \log_2 n)^{\log_2 n}$. Combining the two previous expressions we have that $\lambda_n(R)$ (and hence also the ratio $\frac{\lambda_n R}{\lambda_1(R)}$) is polynomially upper-bounded by n.

These bounds are straightforwardly extended to the dual $R^\vee$ by taking into account the corresponding "tweak" factor. For the multiquadratic scenario, the dual only suffers a scaling by the square roots of the d_i terms (R is sparser than the dual $R^\vee$). However, considering higher degrees in the modular functions $x_i^{n_i} + d_i$, the tweak factor can turn the noise in the non-dual version of RLWE into highly non-spherical.

A very detailed analysis of these effects (including also some enlightening visual examples) can be found in [55].

Choice of parameters: We show now how to select correct parameters $\{n_x, n_y, d_x, d_y\}$ satisfying the conditions established in Sections 2.5 and 2.6 for valid number fields.

As a brief summary, and focusing on $n_x, n_y > 2$, this mainly implies that: (1) the discriminants of $K_x = \mathbb{Q}[x]/x^{n_x} + d_x$ and $K_y = \mathbb{Q}[y]/y^{n_y} + d_y$ are coprime, i.e., gcd $(\Delta_{K_x}, \Delta_{K_y}) = 1$, and (2) n_x, n_y are prime powers satisfying Proposition 5.

This enables the definition of $\mathcal{O}_K = R = \mathbb{Z}[x,y]/(x^{n_x} + d_x, y^{n_y} + d_y)$ as the ring of integers. Analogously, the dual is $\mathcal{O}_K^\vee = \frac{1}{n_x n_y x^{n_x - 1} y^{n_y - 1}} \mathbb{Z}[x,y]/(x^{n_x} + d_x, y^{n_y} + d_y)$ (see Section 2.6 for some particular choices).

In this bivariate case, the error distribution $\chi[x,y]$ samples polynomials in $\mathcal{O}_K^\vee$ whose coefficients are independently sampled from Gaussian distributions with different standard deviations. In particular, σ is equal to $r\sqrt{n} d_x^{\frac{n_x - j_x}{n_x}} d_y^{\frac{n_y - j_y}{n_y}}$ for the coefficient associated to the monomial $x^{j_x - 1} y^{j_y - 1}$ with $1 \leq j_x \leq n_x$ and $1 \leq j_y \leq n_y$.

Working on $q\mathcal{O}_K$: As it is usually done with power-of-two cyclotomics, we can directly transform the dual into the ring of integers by means of a scaling. If we have $\mathcal{O}_K^\vee = \frac{1}{n_x n_y x^{n_x - 1} y^{n_y - 1}} \mathbb{Z}[x,y]/(x^{n_x} + d_x, y^{n_y} + d_y)$, we can first multiply the dual by $\frac{xy}{xy}$, to see the simplified relation $\frac{xy}{xy} \mathcal{O}_K^\vee = \frac{xy}{n d_x d_y} \mathcal{O}_K$.

Finally, analogously to the $x^n + 1$ functions, we can scale the (a, b) sample by $n = n_x n_y$ and also $d_x d_y$. This gives us a sample $(a(x,y), b'(x,y) = n d_x d_y xy b(x,y)) \in R_q^2$. Consequently, we can directly work on the ring of integers with $(a, b = as + e) \in R_q^2$ where $a \leftarrow R_q$, $s \leftarrow R_q$ (or also $s \leftarrow \chi[x,y]$) and $e \leftarrow \chi[x,y]$. After the multiplication with the monomial xy, the error distribution $\chi[x,y]$ generates independent coefficients from a Gaussian distribution of $\sigma = r\sqrt{n} d_x^{\frac{n_x - j_x}{n_x}} d_y^{\frac{n_y - j_y}{n_y}}$ for $1 < j_x \leq n_x$ and $1 < j_y \leq n_y$, $\sigma = r\sqrt{n} d_x^{\frac{2n_x - j_x}{n_x}} d_y^{\frac{n_y - j_y}{n_y}}$ for $j_x = 1$ and $1 < j_y \leq n_y$, $\sigma = r\sqrt{n} d_x^{\frac{n_x - j_x}{n_x}} d_y^{\frac{2n_y - j_y}{n_y}}$ for $1 < j_x \leq n_x$ and $j_y = 1$ while $\sigma = r\sqrt{n} d_x^{\frac{2n_x - 1}{n_x}} d_y^{\frac{2n_y - 1}{n_y}}$ for $j_x = j_y = 1$.

SHE over Multivariate Rings: The basic example cryptosystem described in Table 2.6 follows the structure of the SHE version introduced in [78] and implemented in [79]. The main difference relies on the fact that our polynomial elements belong to the multivariate rings $R[x,y]$, $R_t[x,y]$ and $R_q[x,y]$ (see Definition 11), contrarily to the traditional univariate version $\mathbb{Z}[x]/1 + x^n$ and its analogous rings modulo t and q. In Table 2.6 the diagonal of $\boldsymbol{J}$ has the corresponding standard deviations of χ normalized by r (i.e., σ/r) for each coefficient of the bivariate polynomials.

In particular, our plaintext ring R_t is basically a bivariate polynomial $R_t[x,y] = \mathbb{Z}_t[x,y]/(x^{n_x} + d_x, y^{n_y} + d_y)$ which is encoded as a sub-module of $\mathbb{T} = K_{\mathbb{R}}/R^\vee$ (see Definition 7). Our example is based on the scheme introduced in [78], but other choices are possible, and we briefly discuss them in the Appendix 2.A. Regarding the achieved noise bounds, they are analogous to the computations from [78] by taking into account the expansion factor of the involved rings.

The additional variables of the multivariate structure bring about some significant advantages:

Table 2.6: Parameters and Primitives of a Somewhat Homomorphic Cryptosystem based on a secure multivariate version of RLWE from Definition 11 (see [4, 5]).

<table>
<tr><th colspan="3">Parameters</th></tr>
<tr><td colspan="3">Let $R_t[x, y]$ be the cleartext ring and $R_q[x, y]$ the ciphertext ring. The noise distribution $\chi[x, y]$ in $R_q[x, y]$ takes its coefficients from a spherically-symmetric truncated i.i.d Gaussian $\mathcal{N}(\mathbf{0}, r^2\boldsymbol{J}^2)$. q is an integer satisfying $t < q$ and is relatively prime to t. All the previous parameters are chosen in terms of the security parameter λ where $n = 2^{\lfloor \log \lambda \rceil - 1}$</td></tr>
<tr><th colspan="3">Example SHE Cryptographic Primitives</th></tr>
<tr><td>SH.KeyGen</td><td>Process</td><td>$s, e \leftarrow \chi[x, y], a_1 \leftarrow R_q[x, y]$; $sk = s$ and $pk = (a_0 = -(a_1 s + te), a_1)$</td></tr>
<tr><td rowspan="2">SH.Enc</td><td>Input</td><td>$pk = (a_0, a_1)$ and $m \in R_t[x, y]$</td></tr>
<tr><td>Process</td><td>$u, f, g \leftarrow \chi[x, y]$ and the fresh ciphertext is $\boldsymbol{c} = (c_0, c_1) = (a_0 u + tg + m, a_1 u + tf)$</td></tr>
<tr><td rowspan="2">SH.Dec</td><td>Input</td><td>sk and $\boldsymbol{c} = (c_0, c_1, \ldots, c_{\gamma-1})$</td></tr>
<tr><td>Process</td><td>$m = \left(\left(\sum_{i=0}^{\gamma-1} c_i s^i\right) \mod q\right) \mod t$</td></tr>
<tr><td rowspan="2">SH.Add</td><td>Input</td><td>$\boldsymbol{c} = (c_0, \ldots, c_{\beta-1})$ and $\boldsymbol{c}' = (c'_0, \ldots, c'_{\gamma-1})$</td></tr>
<tr><td>Process</td><td>$\boldsymbol{c}_{add} = (c_0 + c'_0, \ldots, c_{\max(\beta,\gamma)-1} + c'_{\max(\beta,\gamma)-1})$</td></tr>
<tr><td rowspan="2">SH.Mult</td><td>Input</td><td>$\boldsymbol{c} = (c_0, \ldots, c_{\beta-1})$ and $\boldsymbol{c}' = (c'_0, \ldots, c'_{\gamma-1})$</td></tr>
<tr><td>Process</td><td>Using a symbolic variable v their product $\boldsymbol{c}''$ can be obtained from the relation $\left(\sum_{i=0}^{\beta-1} c_i v^i\right) \cdot \left(\sum_{i=0}^{\gamma-1} c'_i v^i\right) = \sum_{i=0}^{\beta+\gamma-2} c''_i v^i$</td></tr>
</table>

more efficient polynomial operations (see Section 3.2 in Chapter 3), better space/efficiency tradeoffs when working with automorphisms (see Section 3.3 in Chapter 3), and can also be very useful when working with multidimensional structures (see Section 2.7.3, Chapters 7 and 8, and Appendix B; and also the works [4, 5, 56] for more details on practical applications). In particular, in [56, 57] the authors present a library called MHEAAN, based on multivariate RLWE, which is optimized to perform homomorphic matrix operations.

Correctness and Security: The condition for correct decryption is that the effective noise $||(\sum_{i=0}^{\gamma-1} c_i s^i) \bmod q)||_\infty$ remains smaller than $q/2$. Let us consider a simplified version of Theorem 2 from [78] where only the effect of noise is taken into account, and let $\max\{\sigma\}$ be the maximum standard deviation of the polynomials sampled from $\chi[x, y]$. Let M be the maximum coefficient of the evaluated degree-D polynomial; if $M(t \max\{\sigma\} d_x d_y n \sqrt{n})^D$ is smaller than $q/2$, the scheme of Table 2.6 can evaluate degree-D multivariate polynomials over elements which belong to $R_t[x, y]$. We could also consider a tighter empirical condition for q, as stated in [79].

Regarding the security of this SHE scheme, it relies on the indistiguishability assumption of the polynomial multivariate version of RLWE (with *adequately chosen secure parameters* $\chi[x, y]$, $\{n_x, d_x, n_y, d_y\}$ and q) featured in Definition 11; breaking this assumption implies, as stated in Theorem 1, the existence of a quantum algorithm which solves short vector problems over ideal lattices. For a practical estimation of the bit security, we can apply the LWE security estimator developed by Albrecht *et al.* [80, 81] to the cryptosystems built on multivariate RLWE and also the estimates included in the standards document [82] for a general random lattice with the same dimension ($n = \prod n_i$). This is plausible, analogously to what it is typically done with ideal lattices, a secure m-RLWE instantiation works with full-rank lattices, for which no substantially faster attacks are known than for general lattices.

Improving on the packing capacity of complex numbers

We address the packing of integer numbers in Chapter 3 (see Section 3.3), but complex numbers are more difficult to efficiently pack. Nevertheless, we can also leverage the multivariate structure to represent the complex arithmetic in a much more efficient way than previous recent approaches. Knowing that a total of $n/2$ complex slots can be packed over the ring $\mathbb{Z}[z]/1+z^n$, Cheon *et al.* [31, 56] expand these results to the bivariate case $\mathbb{Z}[x,y]/(1+x^{n_x}, 1+y^{n_y})$, packing a total of $\frac{n_x}{2}\frac{n_y}{2} = \frac{n}{4}$ complex slots. Generalizing this strategy to l dimensions, packing is restricted to $\frac{n}{2^l}$ complex slots (where $n = \prod_{i=1}^{l} n_i$) when working over multivariate rings as $\mathbb{Z}[x_1,\ldots,x_l]/(1+x_1^{n_1},\ldots 1+x_l^{n_l})$.[7] Consequently, this strategy leaves a huge gap of unused potential slots when transitioning to a multivariate ring.

Nevertheless, it is possible to achieve the same number of complex slots as the univariate counterpart (that is, $n/2$ complex slots), effectively substituting the multivariate complex embedding map (as used in [56]) by its univariate version. Let us consider the ring $\mathbb{Z}[x_1,\ldots,x_l]/(d_1 + x_1^{n_1},\ldots,d_l+x_l^{n_l})$, and choose one of the l independent variables to work with the canonical embedding map, x_1 without loss of generality. If we have a total of $n/2$ complex numbers to pack in one multivariate polynomial plaintext, we organize them as a set of $\frac{n}{n_1}$ complex vectors with length $n_1/2$. For each complex vector we use the encoding from [31], defined as the composition of the inverse of the complex embedding map and a discretization. This yields $\frac{n}{n_1}$ polynomials belonging to the ring $\mathbb{A} = \mathbb{Z}_t[x_1]/d_1 + x_1^{n_1}$.

Coming back to the multivariate ring representation, we can consider the new message as a polynomial in the ring $\mathbb{Z}_t[x_1,\ldots,x_l]/(d_1+x_1^{n_1},\ldots,d_l+x_l^{n_l})$. Hence, we gather all the polynomials in $\mathbb{A}$ as the different coefficients of the ring $\mathbb{A}[x_2,\ldots,x_l]/(d_2+x_2^{n_2},\ldots,d_l+x_l^{n_l})$, and we define encoding/decoding matrices working over $d_i + x_i^{n_i}$ modular functions (i.e., α-generalized INTTs/NTTs over t, see Section 3.2 in Chapter 3) for $i = 2,\ldots,l$, considering the identity matrix $\boldsymbol{I}_{n_1}$ of size $n_1 \times n_1$ for x_1 and the modular function $d_1+x_1^{n_1}$. Using the vector representation of the plaintext polynomial, the encoding/decoding is performed by means of one matrix multiplication which can be efficiently realized with FFT-like algorithms.

This method can pack a total of $n/2$ complex slots while preserving the properties for the automorphisms and also removing the gap of the method used in [56], where the fraction of used slots decreases exponentially with the number of dimensions.

Finally, it is worth looking at the case where the considered multivariate rings are those from Definition 9 in Section 2.5. In this case, the modular functions have the form $d_i + x_i^2$, so the variable x_1 can directly represent the imaginary unit, therefore perfectly mapping the complex arithmetic without the need of applying the canonical embedding map over the polynomials in $\mathbb{A}$.

2.7.3. Applications to Signal Processing

For the sake of completeness, this section focuses on some of the Secure Signal Processing (SSP) applications that benefit from m-RLWE to process encrypted signals in a more efficient and secure way than under RLWE, showcasing the applicability of m-RLWE (we refer the reader to Chapters 7 and 8, and Appendix B for a detailed exposition of the different signal processing applications). *While these results were originally presented on weak instances of m-RLWE vulnerable to the Bootland* et al.*'s attack, they can be adapted to deal with those rings from Definition 11.*

Image filtering (see Appendix B). In image processing, filtering is one of the most common building blocks, and it can be seamlessly implemented as a cyclic multidimensional convolution. While RLWE-based cryptosystems support univariate convolutions, they need to encrypt each row or column of the image or filter separately in order to implement a 2D or 3D convolution between

[7] While this strategy was introduced for a weak instance of multivariate RLWE (i.e., vulnerable to Bootland *et al.*'s attack), a similar approach works for rings following Definition 11.

two encrypted images (or an image and a filter). Conversely, m-RLWE introduces a natural way to work with multidimensional linear operations, and it achieves a more compact representation of the data, as it can effectively cipher one signal value per coefficient of the encryption polynomial. As shown in [4], the time needed for an encrypted convolution with an m-RLWE-based cryptosystem is between one and two orders of magnitude faster than with its RLWE counterpart for common image sizes,[8] while the security of the former can much higher (*whenever we work on a secure instantiation of multivariate RLWE*), due to the large degree of the multivariate polynomials.

Image denoising (see Chapters 7 and 8). Another ubiquitous image processing operation is image denoising. This operation involves a linear (Wavelet) transform, a thresholding non-linear operation applied to each sub-band, and an inverse transform. By resorting to 2-RLWE and a polynomial representation of the thresholding operation, it is possible to efficiently perform all these operations with a circuit of limited depth and without an intermediate decryption of the image [46]. This produces a denoised image of size 256×256 in a few minutes. *If the* 2*-RLWE scheme is not implemented in a weak instantiation*, the RLWE counterpart would require polynomials of large degree in each image dimension to achieve the same security level, which renders the computation several orders of magnitude slower than with a 2-RLWE cryptosystem.

Increased flexibility in image processing (see Appendix B). Finally, it is worth noting that the additional degrees of freedom that m-RLWE introduces give more flexibility to cope with signals with different structures, which is plainly impossible with the regular RLWE. In [5], mechanisms for converting across different signal structures and perform efficient block processing are shown. Hence, m-RLWE enables (a) better packing schemes by grouping image pixels in blocks (e.g., for encrypted JPEG de-/compression by using block Discrete Cosine Transforms), or video sequences in frames, (b) encrypted multi-dimensional transforms that can work on a block-by-block basis taking advantage of the large signal dimensionality to increase the cryptosystem security with respect to their RLWE counterpart, (c) the use of the extra variables to encode additional information which can be used to homomorphically evaluate encrypted divisions in the signal values, (d) flexible changes of the signal structure to update the packing and organization of the blocks, in order to seamlessly enable different operations on different dimensions.

2.8. Conclusions

This chapter addresses the main security flaw of the multivariate RLWE problem revealed by Bootland *et al.* For this purpose, we have defined and parameterized practical and secure instantiations of the multivariate Ring Learning With Errors problem, supported by the extended reduction of the original proof by Lyubashevsky *et al.* [40, 41]. The proposed instantiations are resilient against Bootland's attack to m-RLWE [44], while still preserving all the efficiency improvements that m-RLWE brings. We have shown how to find practical parameters for the proposed instantiations to make them both secure and usable.

[8] In [4], the authors implement an m-RLWE extension in C using GMP 6.0.0 and FLINT. For a filtering application with an image of size 1014×1014 and a filter of size 11×11, they achieve runtimes of 134 s with RLWE and 8 s with the extension.

2.A. Further Optimizations

The scheme we have chosen to exemplify the use of multivariate rings with RLWE in Section 2.7.2 can be further optimized. We based our choice on the scheme introduced in [78] for simplicity and clarity, but many other options could be taken into account. For example, in [83] the authors provide a detailed comparison among four of the main variants which are currently used in the literature: BGV [49, 50], NTRU [84] and their corresponding scale-invariant versions [85] which are, respectively, FV [86] and YASHE [87].

The use of a scale-invariant version simply involves additional division and rounding operations over the polynomial coefficients; these operations can be seamlessly addressed when working with multivariate polynomials.

The main optimizations which are considered for the comparison in [83] are modulus switching and key switching [88]. The first one has been used in RLWE to work with leveled SHE schemes [49, 50], and it requires a chain of decreasing moduli in such a way that, after each homomorphic multiplication, a switch to a smaller modulus is performed. The effect of this operation is a notable reduction in the noise increase after each multiplication. Similarly to scale-invariant schemes, the use of modulus switching requires division and rounding operations over the coefficients of the polynomials.

Regarding the key switching operation, its use removes the dependency between the number of polynomial elements in the ciphertexts and the depth of the evaluated circuits. It is also used when working with automorphisms, where it helps to recover the ciphertexts under the original secret key.

Both modulus and key switching can be extended to work with multivariate polynomials. Firstly, division and rounding can be directly applied over the coefficients of multivariate polynomials, and secondly, switching key matrices can be analogously generated with multivariate polynomials.

Finally, an additional "optimization" which we could incorporate is the use of bootstrapping to obtain a FHE scheme, hence removing the upper bound on the depth of the evaluated circuits. For this purpose, conventional procedures could be applied over the SHE scheme, mainly consisting of homomorphically evaluating the decryption circuit by having access to an encrypted version of the secret key.

After Gentry's seminal work [33, 34], different improvements on the use of bootstrapping have appeared in the literature, varying from the recryption of binary gates [89, 90, 35, 91, 51, 92] to the optimization of the depth of the decryption circuit for RLWE-based SHE schemes [60, 30, 32]. An interesting follow-up work would be to study the behavior of our multivariate scheme with these different approaches.

2.B. Concrete Security Estimates

In order to give an example of some concrete security estimates for different choices of parameters with the example cryptosystem, we can make use of the LWE security estimator developed by Albrecht *et al.* [80, 81]. [9] To this aim, we can call the function estimate_lwe(n, α, q, secret_distribution = "normal", reduction_cost_model = BKZ.sieve) considering the relation

[9] Available online in `https://bitbucket.org/malb/lwe-estimator`.

$\min\{\sigma\} = \alpha q/\sqrt{2\pi}$; where the minimum standard deviation of the χ distribution for the equivalent LWE samples is considered (see Section 2.7.2).

For the n parameter, analogously to what it is typically done with ideal and general lattices, we assume that the underlying lattices of the *secure* multivariate RLWE samples do not necessarily have substantially faster attacks than those known over a general random lattice with the same dimension (hence considering $n = \prod n_i$ as the dimensionality of the lattice).

Chapter 3

Applications of Multivariate RLWE on Lattice-based Cryptography

3.1. Introduction

Current hot problems in (fully) homomorphic encryption involve the optimization of elementary polynomial operations through fast transforms and, especially, the search for optimal strategies to execute homomorphic slot manipulations and trade off storage and computation needs for relinearization operations. These are fundamental blocks in homomorphic processing and in the implementation of the bootstrapping (see [60, 30, 32, 92]) primitives enabling fully homomorphic encryption.

To motivate the content of this chapter, we first present a survey on the state of the art on fully and somewhat homomorphic encryption, transformed-domain processing and the associated optimizations for rounding operations under RNS (Residue Number Systems) representation, SIMD techniques, and a brief discussion on the improvements that this chapter brings about with respect to the current state of the art.

Fully/Somewhat Homomorphic Encryption (FHE/SHE): Most of the efficiency improvements that RLWE has introduced are based on the algebraic structure of the used cyclotomic rings $R = \mathbb{Z}[z]/\Phi_m(z)$. With the adequate choice of modulus q for $R_q = \mathbb{Z}_q[z]/\Phi_m(z)$, the cyclotomic polynomial splits in $\phi(m)$ linear factors, and this enables the use of the CRT (Chinese Remainder Transform) to efficiently and independently add and multiply the corresponding elements belonging to R_q [45].

Additionally, this property has also been considered for the plaintext ring, as a tool to batch several integers in one encryption (as many as $n = \phi(m)$ values when the modular function fully splits in linear factors), which contributes to a reduction in the cipher expansion.

From a practical perspective, some of the most recent libraries dealing with lattice-based cryptography, such as the HElib [59, 60, 93], PALISADE,[1] SEAL[2] and NFLlib [94], take advantage of this fact to optimize the polynomial operations. The first one uses the double-CRT representation, that is, a first CRT (Chinese Remainder Theorem) splitting the cyclotomic polynomial in

[1] Available online in `https://git.njit.edu/palisade/PALISADE`.

[2] Available online in `http://sealcrypto.org`.

linear factors, and a second CRT factoring the coefficients of the polynomials depending on the prime-decomposition of the modulus q. The two latter libraries are specialized for power-of-two cyclotomic rings $\mathbb{Z}_q[z]/1+z^n$, so they consider a CRT representation for the coefficients together with an efficient NTT/INTT representation. For example, in the NFLlib the NTT is calculated with an efficient variant of a radix-2 FFT algorithm [1]. It is also worth mentioning that the HEAAN [31] library has been recently updated to work with this CRT and NTT representation. Conversely, the PALISADE library implements several cryptosystems and uses both approaches depending on the modular function of the involved rings.

In the case that all the involved operations are polynomial multiplications and additions, working in this transformed domain enables polynomial operations with a cost of $\mathcal{O}(n)$ elemental operations between coefficients. However, the current state-of-the-art homomorphic schemes, such as BGV [78, 50] and FV [86], apply a rounding operation over the polynomial coefficients which is not compatible with the double-CRT (or CRT and NTT) representation.

This means that this rounding has to be applied in the coefficient-wise representation, with the corresponding overhead for swapping between representations.

A rounding over the RNS (Residue Number System) representation: Rounding (quantization) is an essential operation for scale-invariant (e.g., FV) and leveled cryptosystems (e.g., BGV). Therefore, Bajard *et al.* [95] have studied in detail how to perform a rounding operation without leaving the CRT representation (also called RNS, Residue Number System).

They implement their method using the NFLlib library for the FV cryptosystem and show that for practical parameters, staying in the CRT domain outperforms the results of the usual approach of moving between domains. Additionally, it is also shown how the asymptotic complexity of decryption is improved by a factor of $\mathcal{O}(\log n)$ when staying in the CRT domain.

Unfortunately, this asymptotic improvement is not preserved when comparing the multiplication primitives, as the effect of the NTT/INTT computations is the same for both implementations.

In any case, even when there is no an asymptotic improvement for all the primitives, the use of the RNS representation proposed by Bajard *et al.* [95] enables to fit all the used values into the size of a machine word, which in practice helps in considerably improving the performance when comparing with an implementation requiring the use of multi-precision arithmetic.

In a recent work, Halevi *et al.* [96] propose further optimizations beyond the results of Bajard *et al.* [95], and implement them in the PALISADE library. They achieve a simplification in the procedure and also avoid the additional noise that their method introduces inside the ciphertexts. For a detailed comparison between the methods from [96] and [95] we refer the reader to [97], where CPU and GPU implementation runtimes are provided for both.

Ciphertext packing techniques and automorphisms: As we have already discussed, the special structure of the cyclotomic rings not only enables some optimizations on the involved operations. In fact, we can also batch several plaintext values into the same ciphertext by resorting to the CRT.

Smart and Vercauteren [98] showed how to exploit the factoring of the modular function over the plaintext space to encode a vector of “slots”. Therefore, a basic arithmetic operation over the encrypted plaintext is equivalent to applying the same operation component-wise over all the encoded slots. This property is one of the functionalities implemented in the HElib library[3] which enables a framework for encrypted SIMD operations.

When working over these "packed slots", being able to exchange contents among them is also convenient. This swapping operation among slots can be performed by means of the available automorphisms on the plaintext ring $R_t = \mathbb{Z}_t[z]/\Phi_m(z)$; these automorphisms can be seen as applying a change of variable $z \to z^i$ for $i \in \mathbb{Z}_m^*$ over the corresponding polynomial elements.[4]

The combination of the packed representation together with the use of the automorphisms has become one of the main blocks for several primitives working over the ring R (specifically, those which rely on the use of linear maps over encrypted vectors); one of the most representative examples is its use for the bootstrapping [59, 60].

Whereas the automorphism operations by themselves are very efficient (as they can be applied as linear operations over the ciphertext polynomials), the resulting ciphertexts are not encrypted over the original secret key. Hence, a relinearization or switching key operation has to be used to convert the ciphertext back to an encryption over the original secret key.

Consequently, it is important to reduce both the size of the evaluation keys (composed of the set of required relinearization matrices) and the runtime associated to the switching key process. A recent work [99] has introduced improvements on both the size of the evaluation key and also the corresponding runtimes when working with these automorphisms for linear maps over encrypted vectors, showcased in HElib.

However, in general, there exists a tradeoff between the number of required relinearization matrices and the increase on the computational cost of the operations, that we optimize in this chapter.

Motivation and contributions of this chapter: A careful examination of the previous results reveals that if we were able to either (a) efficiently compute the rounding operation without having to reverse the NTT/INTT (or, more generally, the CRT over the cyclotomic polynomial in the double-CRT representation) or (b) speed-up the runtimes involved on its calculation, then the efficiency of the current somewhat homomorphic encryption schemes could be considerably improved (as almost all the operations in these schemes need to call this basic block). Additionally, the effect of an efficiency improvement on these multiplications goes beyond somewhat homomorphic encryption schemes, enhancing also any cryptographic primitives involving polynomial multiplications, including the candidates of the NIST Post-Quantum challenge [81]. Our contributions in this matter are:

- We improve the cost of the underlying polynomial operations for cryptographic primitives based on RLWE (it could also be applied in the NTRU setting [100, 101]). We show how the well-known asymptotic cost of $\mathcal{O}(n \log n)$ for cyclotomic rings with polynomials of n coefficients can be improved by a factor of $\log n$ in terms of elemental multiplications. To this aim, we propose to work over a multivariate ring which possesses a convolution property relating the coefficient-wise representation with the transformed domain by means of

[3] It could also be considered in the libraries specialized for power-of-two modular functions by applying the INTT/NTT functionalities as a pre-/post-processing over the plaintexts before/after encryption/decryption (see Chapter 4). However, the HElib library provides more freedom in the definition of the slots.

[4] It might be useful to consider again the ring $\mathbb{Z}[z]/(1-z^n)$, which is typically used in Signal Processing applications (see Table 2.1 in Chapter 2). These automorphisms on the Z-transform would be analogous to a *circular shift in the frequency domain* (Shift theorem of the DFT), which also translates to a linear phase multiplication in the time domain.

an α-generalized variant of the Walsh-Hadamard transform (over finite rings instead of the usual real numbers). This transform can be very efficiently computed with FFT algorithms (specifically, with a variant of the Fast Walsh-Hadamard transform) whose computational cost is only $\mathcal{O}(n \log n)$ additions, hence being much more amenable for a practical implementation.

- We enable a considerable improvement on the homomorphic packing/unpacking with a single switching key operation (removing the dependency between the number of slots and the number of automorphisms/switching key operations), and we show that the available set of automorphisms in these multivariate rings presents a particular structure which enables to deal with different tradeoffs between the size of the involved evaluation keys and the efficiency of the switching key process. In general, we can show that with an increase of $\mathcal{O}(\log n)$ in the chain of the switching key process (it must be noted that in this multivariate rings the operations can be reduced by a factor of $\mathcal{O}(\log n)$ multiplicative operations, hence having a constant increase in terms of multiplicative cost), the number of required relinearization matrices is $\log_2 n$. We also discuss several tradeoffs between this size and computational cost; for example, when working with more general multivariate rings, we can have an increase by a factor of $\mathcal{O}(\sqrt{n})$ in the size of the evaluation key and 2 times the cost of a basic switching key process; in general, with a size of $\mathcal{O}(n^{1/k})$ we would have an increase in the cost by a factor of k. Taking a look to the improvements recently introduced in [99], our results can enhance their tradeoffs for those scenarios where the same "effective" slot encoding is used.

Additionally, it is also worth mentioning that in [31] the authors discuss how to pack complex numbers by means of the complex embedding. They extend this result to bivariate rings in [56], however their packing cannot work with as many complex slots as the usual univariate counterpart (they have a reduction by a factor of two per each new dimension). In this work we have also exemplified how to correctly embed complex slots into these multivariate rings so as to have as many complex slots as their univariate counterpart (see Chapter 2).

Structure: The rest of the chapter is organized as follows: Section 3.2 particularizes the problem to rings enabling an α-generalized Walsh-Hadamard Transform, and compares its performance with fast NTT algorithms currently used in state-of-the-art RLWE cryptosystems. Section 3.3 introduces the strategies for homomorphic packing/unpacking and the space/time tradeoffs improving on current RLWE relinearization and bootstrapping operations. Finally, Section 3.4 draws some conclusions and Appendix 3.A reviews the Full and Baby-step/giant-step strategies from [99, 93].

3.2. Multiquadratic Rings with Fast Walsh Hadamard Transforms

This section focuses on improving the cost of the underlying polynomial operations for cryptographic primitives based on RLWE, especially polynomial products (convolutions). We show how the well-known asymptotic cost of $\mathcal{O}(n \log n)$ for cyclotomic rings with polynomials of n coefficients can be improved by a factor of $\log n$ in terms of elemental multiplications when working on m-RLWE (or RLWE over a multivariate number field). To this aim, we particularize the multivariate version to degree-2 polynomials and introduce an (α-generalized) variant of the Walsh-Hadamard transform (over finite rings instead of the usual real numbers), featuring a convolution property that relates the coefficient-wise representation with the transformed domain. This transform can be very efficiently computed with FFT algorithms (specifically, with a variant of the Fast Walsh-Hadamard transform) whose computational cost is only $\mathcal{O}(n \log n)$ additions, hence being much more amenable for a practical implementation. It is worth noting that the effect of the efficiency improvement brought about by our approach goes beyond somewhat homomorphic encryption schemes (including also the NTRU setting [100, 101]), also enhancing any cryptographic primitives involving polynomial multiplications, e.g., the candidates of the NIST Post-Quantum challenge [81].

For this section, we deal with a specific version of m-RLWE (multiquadratic RLWE) where all the used modular functions have the same form $f_i(x_i) = d_i + x_i^2$ (see Definition 9).

The security reduction from Theorem 1 applies to this particular version of the m-RLWE problem. To this aim, parameters d_i have to be chosen as indicated in the beginning of Section 2.5. Additionally, Proposition 6 gives a sufficient condition to make the problem secure against the attacks described in Section 2.7.1.

After defining the specific version of the problem, we introduce the ($\boldsymbol{\alpha}$-generalized) Hadamard transform, that we apply to reach the aforementioned performance gains on polynomial convolutions.

3.2.1. Faster polynomial arithmetic over multivariate rings

The Hadamard transform over real numbers is usually applied by means of the recursion

$$\boldsymbol{H}_i = \frac{1}{\sqrt{2}} \begin{pmatrix} \boldsymbol{H}_{i-1} & \boldsymbol{H}_{i-1} \\ \boldsymbol{H}_{i-1} & -\boldsymbol{H}_{i-1} \end{pmatrix}, \tag{3.1}$$

where $i \in \mathbb{N}$ and $\boldsymbol{H}_0 = 1$.

It can be seen that matrix $\boldsymbol{H}_i$ with $i \geq 1$ is equivalent to the Kronecker product of i DFT (Discrete Fourier Transform) matrices of size 2 ($\boldsymbol{H}_1$ equals the DFT matrix of size 2); that is, it can be seen as a $\underbrace{2 \times 2 \times \cdots \times 2}_{i \text{ times}}$-DFT transform (defined over i dimensions of length 2 each).

Analogously to the DFT, the Walsh Hadamard Transform (WHT) of size n possesses a particular fast algorithm called FWHT (Fast Walsh Hadamard Transform) which can be very efficiently computed with no products and with a cost of $\mathcal{O}(n \log n)$ additions and subtractions (see [102, 103]). Hence, when working over rings satisfying a convolution property with the Hadamard transform, it is possible to efficiently compute the multiplication of elements from these rings with a cost of $\mathcal{O}(n)$ elemental multiplications.

Security reasons prevent us from directly working over rings satisfying this convolution property with the Walsh Hadamard transform (that is, multivariate rings whose modular functions are $f(x_i) = x_i^2 - 1$), as they can be easily factored into $(x_i - 1)(x_i + 1)$ over $\mathbb{Z}$. Therefore, we resort to the type of multivariate rings presented in Definition 9 and apply an ($\boldsymbol{\alpha}$-generalized) variant of the WHT.

α-generalized convolutions: An α-generalized convolution[5] corresponds to the ring operation defined over polynomials belonging to $\mathbb{Z}_q[z]/1 - \alpha z^n$. Figure 3.1 shows the realization of an α-

[5] For example, with $\alpha = -1$ we have a negacyclic convolution. In the literature, this convolution operation is also called negative wrapped convolution.

generalized convolution between vectors of length N (with $l = 0, \dots, N-1$), by means of a cyclic convolution combined with an element-wise pre/post-processing applied before/after [54, 29].

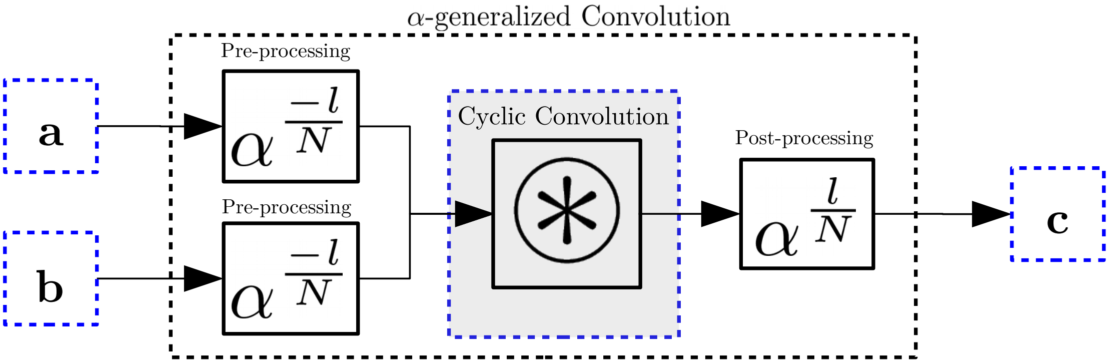


Figure 3.1: Block diagram for the implementation of an α-generalized convolution by means of a cyclic convolution.

As the cyclic convolution can be efficiently computed by means of standard fast algorithms, this means that an α-generalized convolution can be implemented with only a light overhead ($\mathcal{O}(n)$ scalar products). [6]

α-generalized Walsh-Hadamard transform: We are mainly interested in modular functions with the form $x_i^2 + d_i$. We can rewrite $1 - \alpha x^n$ as $-\alpha((-\alpha)^{-1} + x^n)$. Hence for $x_i^2 + d_i$ we have $d_i = (-\alpha_i)^{-1} = -\alpha_i^{-1}$. For this particular type of polynomial rings we can define the following direct and inverse transforms:

$$\boldsymbol{W}_1 = \boldsymbol{H}_1 \begin{pmatrix} 1 & 0 \\ 0 & (\alpha_1)^{-1/2} \end{pmatrix}, \quad \text{and} \quad \boldsymbol{W}_1^{-1} = 2^{-1} \begin{pmatrix} 1 & 0 \\ 0 & (\alpha_1)^{1/2} \end{pmatrix} \boldsymbol{H}_1,$$

where the square-roots $(\alpha_i)^{\frac{1}{2}}$ and $(\alpha_i)^{\frac{-1}{2}}$ have to exist in R_q for all i (see Definition 9). Equivalently, if q is an odd prime, we can make use of the Legendre symbol $\left(\frac{-d \bmod q}{q}\right)$ to check when the multivariate ring factors into linear terms. To this aim we need that $\left(\frac{-d_i \bmod q}{q}\right) = 1$ for a prime q and for all i. We also redefine $\boldsymbol{H}_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ without taking into account the normalizing factor $\frac{1}{2}$.

Therefore, now we can extend this definition to multivariate rings with modular functions of the form $x_i^2 + d_i$: we consider the Kronecker product of the matrices $\boldsymbol{W}_1$ and $\boldsymbol{W}_1^{-1}$ as $\boldsymbol{W}_i = \bigotimes_{j\in[i]} \boldsymbol{W}_1$ and $\boldsymbol{W}_i^{-1} = \bigotimes_{j\in[i]} \boldsymbol{W}_1^{-1}$, arriving at the following expression:

$$\boldsymbol{W}_i = \boldsymbol{H}_i \left(\bigotimes_{j\in[i]} \begin{pmatrix} 1 & 0 \\ 0 & (\alpha_j)^{-1/2} \end{pmatrix} \right), \quad \text{and} \quad \boldsymbol{W}_i^{-1} = 2^{-i} \left(\bigotimes_{j\in[i]} \begin{pmatrix} 1 & 0 \\ 0 & (\alpha_j)^{1/2} \end{pmatrix} \right) \boldsymbol{H}_i,$$

where the normalizing factors are again left outside $\boldsymbol{H}_i$.

Consequently, if we define the vector $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_l)^T$, when working over the multivariate ring $R_q[x_1, \dots, x_l]$ with $f_j(x_j) = d_j + x_j^2$ for $j = 1, \dots, l$ we can use the transforms $\boldsymbol{W}_l$ and

[6] It is common to include these additional scalar products inside the butterflies of the FFT algorithms to further enhance the efficiency.

$\boldsymbol{W}_l^{-1}$ analogously to the use of negacyclic NTTs in the univariate RLWE. Both $\boldsymbol{W}_l$ and $\boldsymbol{W}_l^{-1}$ transforms can be efficiently computed in $\mathcal{O}(n)$ (where $n = 2^l$) elemental multiplications thanks to the FWHT. This enables the computation of the $\boldsymbol{H}_l$ matrix multiplications with only $\mathcal{O}(n \log n)$ additions and subtractions and no products, which brings a net improvement with respect to the analogous and widespread radix implementation of the NTT.

Implementation of the Fast Walsh-Hadamard Transform (FWHT): Algorithm 1 shows a pseudocode implementation of the (cyclic) FWHT (Fast Walsh-Hadamard Transform) implementation (see [102, 103]), computing the Hadamard transform of a length-n vector $\boldsymbol{a}$. It can be easily seen that this algorithm requires a total of $n \log_2 n$ additions (specifically, $\frac{n \log_2 n}{2}$ additions and $\frac{n \log_2 n}{2}$ subtractions), instead of the n^2 additions/subtractions required when directly applying the matrix multiplication.

Algorithm 1 Fast Walsh-Hadamard Transform ($\boldsymbol{H}_i \boldsymbol{a}$ with $i \geq 1$).

```
procedure FastWalsh-HadamardTransform($\boldsymbol{a}$)
Input:
    $\boldsymbol{a}$ such that $\texttt{length}(\boldsymbol{a}) = n = 2^i$ and $i \geq 1$
Algorithm for $\texttt{FWHT}(\boldsymbol{a})$ (computing $\boldsymbol{H}_i \boldsymbol{a}$):
    $\texttt{depth} = 1$;
    for $j = 0$ until $\log_2 n - 1$ do
        $\texttt{scale} = 2 * \texttt{depth}$;
        for $k = 0$ until $\left\lfloor \frac{\texttt{length}(\boldsymbol{a})-1}{\texttt{scale}} \right\rfloor$ do
            for $l = \texttt{scale} * k$ until $\texttt{scale} * k + \texttt{depth} - 1$ do
                $\texttt{ac} = \boldsymbol{a}[l]$;
                $\boldsymbol{a}[l] = \boldsymbol{a}[l] + \boldsymbol{a}[l + \texttt{depth}]$;
                $\boldsymbol{a}[l + \texttt{depth}] = \texttt{ac} - \boldsymbol{a}[l + \texttt{depth}]$;
        $\texttt{depth} = 2 * \texttt{depth}$;
Output:
    $\boldsymbol{a} \leftarrow \boldsymbol{H}_i \boldsymbol{a}$
```

Finally, the $\boldsymbol{\alpha}$-generalized version of the direct (inverse) FWHT can be defined by adding a right (left) product by a diagonal matrix, so that the total cost for the negacyclic FWHT on a length-n vector is n elemental multiplications and $n \log_2 n$ additions.

Implementation and evaluation: Polynomial multiplications are the main bottleneck of lattice cryptography, as they are the most time-consuming basic blocks of any cryptographic algorithm, from encryption/decryption to relinearization and bootstrapping. The replacement of the traditional NTTs by FWHT by transitioning from cryptographic constructions built on univariate RLWE to the proposed multivariate version can bring a considerable improvement in terms of computational efficiency. To showcase the achieved gains, we have implemented Algorithm 1 in C++ and compared it with one of the currently most efficient radix-2 implementations of the NTT [1]; this is the algorithm featured in the NFLlib, one of the fastest lattice-based cryptographic libraries available for homomorphic encryption. NFL also off-loads the complexity of the bit-reversal operation to the INTT, in order to speed up the NTT, and makes use of SSE and AVX2 optimizations to further enhance the obtained performance. Figure 3.2 shows the comparison of the obtained run times for a wide range of practical values of n (vector size or polynomial degree), when using our FWHT implementations, including an SSE/AVX2 vectorized version. It can be seen that we obtain a reduction to about 45-50% of the time of the NTT (38-43% of the INTT) in the non-vectorized implementation of the FWHT with respect to the fast NTT of NFLlib, while the vectorized one further reduces this figure to 22-24% (19-22% of the INTT). Finally, it is worth noting that the memory consumption of the FWHT is much lower, as it does not need to store the tables of the twiddle factors.

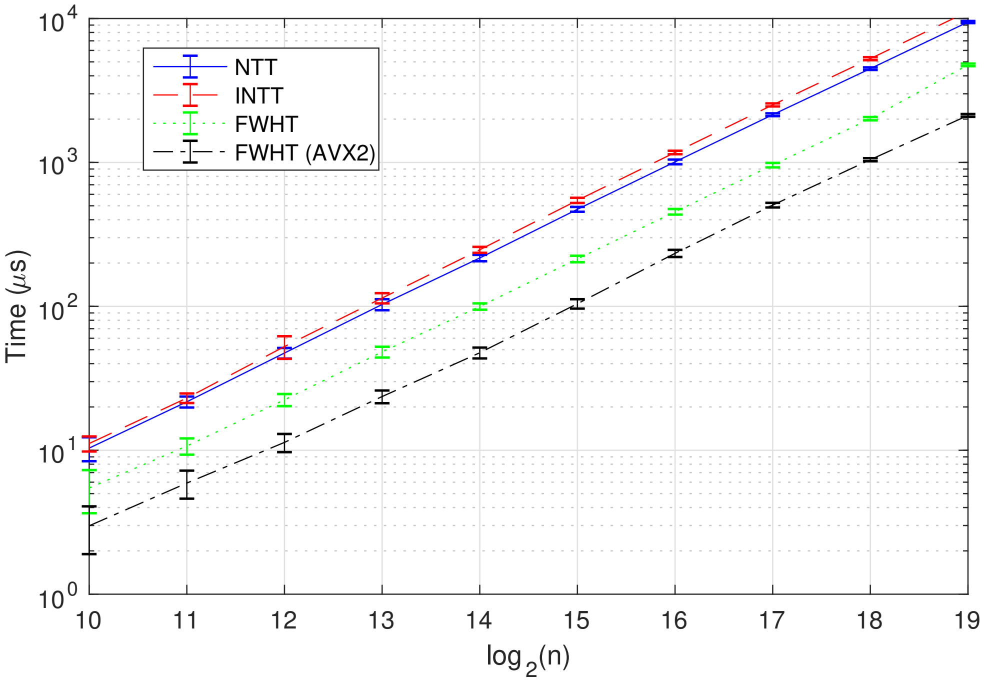


Figure 3.2: Runtimes of the proposed FWHT compared to the NTT/INTT from [1].

3.3. Slot manipulation in multivariate rings

In this section we introduce the main improvements that m-RLWE brings to slot manipulation when packing several plaintext inputs into a ciphertext, with applications in relinearization and bootstrapping operations. Packing into slots [98] helps to take advantage of the available space in the plaintext ring, therefore improving cipher expansion. The use of this packing strategy also enables working with homomorphic "slot"-wise additions and multiplications, i.e., SIMD (Single Instruction, Multiple Data) operations with encrypted data. This is usually combined with a mechanism to efficiently move and exchange the plaintext contents across slots, by taking advantage of the properties of the available automorphisms in the used ring. In general, in the ring $R_t = \mathbb{Z}_t[z]/\Phi_m(z)$, we can define a set of automorphisms $\phi(m)$ as different transformations $\rho_i : R_t \to R_t$ with $i \in \mathbb{Z}_m^*$, which apply a change of variable $z \to z^i$ over the elements in R_t.

Current lattice-based homomorphic cryptosystems leverage automorphisms to perform linear transformations across plaintext slots. Whereas applying an automorphism is a very efficient operation, it produces a ciphertext encrypted under a different secret key, and consequently, a switching key operation is needed to recover a ciphertext under the original secret key. This switching key process has two main drawbacks [99]: (a) a notable computational overhead and (b) an increase in the memory requirements due to the need of adding additional public information ("switching key/relinearization" matrices, a.k.a. evaluation keys).

In general, there is a tradeoff between these two dimensions: when the number of evaluation keys increases, the involved switching key runtime decreases, and conversely, when the number of keys is reduced, a chain of several switching key operations is needed, hence increasing the runtime. In a recent work [99], Halevi and Shoup explore several strategies to optimize this tradeoff, claiming improvements of even 75 times faster runtimes than those of their previous implementation, together with a reduction of up to a half in the required memory space to store the evaluation keys.

This section focuses on two different aspects: (1) We show how the introduced multivariate rings over the RLWE problem (see Sections 2.5 and 2.6 from Chapter 2) enable considerable improvements in the efficiency of the homomorphic packing/unpacking into slots, therefore greatly improving essential blocks for homomorphic encryption, such as bootstrapping, and (2) we analyze the structure of the available set of automorphisms on these rings, also showing that our solution can improve both the runtime and the memory requirements with respect to the state of the art in [99].

Remark: It is worth noting that, for simplicity in the exposition, all the exemplified solutions are sketched out with negacyclic rings. We plan to extend in the future these results to the more general multivariate rings showcased in Chapter 2. To this aim, we have to resort to the generalized pre-/post-processing presented in [29], together with the decomposition of the NTT/INTT transforms into a chain of automorphisms and convolution operations.[7]

3.3.1. Efficient Slot Packing/Unpacking

The homomorphic packing/unpacking of plaintext values into slots is one of the most important examples of the evaluation of linear transformations on the ciphertexts, bootstrapping being one of the most representative applications [60, 30, 32]. The way current cryptosystems implement this packing/unpacking is by means of a decomposition of the matrix multiplication into element-wise products between the different diagonals of the matrix and different rotated versions of the ciphertext (hence by adding the result of a set of multiplications between plaintexts and rotated ciphertexts).

The main bottleneck of this process is the number of switching key matrices required to rotate the ciphertexts. Working with n slots, a total of $n-1$ rotations, hence $n-1$ switching key matrices, is required in the worst case. Available strategies to reduce this number of matrices come at the cost of also increasing the runtimes per automorphism/switching key operation.

Thanks to the introduced n-RLWE, we break the need of a number of rotations (automorphisms/switching key operations) equal to the number of slots, and *we enable homomorphically packing/unpacking operations with a single switching key operation.* This is mainly due to the structure that the multivariate rings from Definition 9 present, which enables a much more efficient algorithm to compute the slot packing/unpacking, as we show next (again, we exemplify all the results with the negayclic variant from Definition 10).

Consider a plaintext ring $R_t[x_1, \ldots, x_i]$, then the required matrices for packing and unpacking

[7] For example, when discussing automorphisms with multiquadratics, the changes of variables $\{x_i \rightarrow -x_i\}$ used with functions $x_i^2 + 1$ are still valid.

are respectively:

$$\boldsymbol{V}_i = 2^{-i} \underbrace{\left(\bigotimes_{j\in[i]} \begin{pmatrix} 1 & 0 \\ 0 & (-1)^{1/2} \end{pmatrix} \right)}_{\boldsymbol{B}} \boldsymbol{H}_i \quad \text{and} \quad \boldsymbol{V}_i^{-1} = \boldsymbol{H}_i \underbrace{\left(\bigotimes_{j\in[i]} \begin{pmatrix} 1 & 0 \\ 0 & (-1)^{-1/2} \end{pmatrix} \right)}_{\boldsymbol{B}^{-1}},$$

where $\beta = (-1)^{1/2}$ is the 4-th root of unity over the plaintext modulo t. Instead of directly applying these linear transformations (following the conventional approach), we resort to the NTT pre-/post-processing presented in [29], where the authors show how a DFT/NTT transform can be expressed in terms of element-wise products (NTT and a one-stage pre-/post-processing) and a negacyclic convolution. We show this process step by step, by computing first $\boldsymbol{H}_i$ and then $\boldsymbol{B}$ (resp. $\boldsymbol{B}^{-1}$).

$\boldsymbol{H}_i$ evaluation: Adapting the results from [29] to the structure of our particular rings, it can be seen that the $\boldsymbol{H}_i$ matrix can be homomorphically evaluated by means of an automorphism and a negacyclic convolution with an all ones vector. That is, if we have encrypted a polynomial $a \in R_t[x_1, \ldots, x_i]$, let us define a polynomial $\mathbf{1}(x_1, \ldots, x_i) = \prod_{j\in[i]}(1+x_j)$, such that the result of the multiplication

$$\mathbf{1}(x_1, \ldots, x_i)a(-x_1, \ldots, -x_i) \in R_t[x_1, \ldots, x_i]$$

is a polynomial whose coefficients correspond to the cyclic Hadamard transform.

$\boldsymbol{B}^{-1}$ and $\boldsymbol{B}$ evaluation: The mentioned pre-/post-processing corresponds to the main diagonal of the matrices $\boldsymbol{B}^{-1}$ and $\boldsymbol{B}$, which comprise only four different values: $\{1, -1, \beta^{-1}, -\beta^{-1}\}$ for $\boldsymbol{B}^{-1}$ and $\{1, -1, \beta, -\beta\}$ for $\boldsymbol{B}$. This element-wise multiplication can be performed homomorphically over the encrypted polynomial coefficients through a change of variable in the ciphertext's polynomials: (1) $\{x_j \to \beta^{-1}x_j\}_{j\in[i]}$ to calculate the $\boldsymbol{B}^{-1}$ matrix multiplication, and (2) $\{x_j \to \beta x_j\}_{j\in[i]}$ for the $\boldsymbol{B}$ matrix multiplication.[8]

Finally, we only need a relinearization/key switching operation to recover the original secret key after the two changes of variables $\{x_j \to -x_j\}_{j\in[i]}$ and $\{x_j \to \beta x_j\}_{j\in[i]}$ for packing (respectively $\{x_j \to \beta^{-1}x_j\}_{j\in[i]}$ and $\{x_j \to -x_j\}_{j\in[i]}$ for unpacking).

3.3.2. Automorphisms and their Hypercube Structure

We show now how m-RLWE improves on the tradeoffs between space and computational cost when dealing with automorphisms, with respect to the univariate version.

Let $\mathbb{A}[z]/1+z^2$ be a polynomial ring as the one described by Definition 10, and α be an element $\alpha \in \mathbb{A}[z]/1 + z^2$; then, we denote by $\theta_i^{(z)}(\alpha) \in \mathbb{A}[z]/1 + z^2$ the transformation over α which applies the change of variable $z \to z^i$ with $i \in \mathbb{Z}_4^*$. For these particular rings, both transformations are, respectively, the identity $z \to z$ and the negation $z \to -z$. Reducing modulo t (the modulo of the plaintext ring), the effect of the latter transformation over the slots would be equivalent to a block shift where each block is composed by one half of the total slots. This shift is graphically

[8] Making use of the decomposition of the formulation of the Bluestein FFT algorithm from [29], we can implement this change of variable by means of a homomorphic negacyclic convolution with NTT/INTT(diag($\boldsymbol{B}$)) and NTT/INTT(diag($\boldsymbol{B}^{-1}$)).

illustrated in Figure 3.3, where ψ is the 4-th root of unity modulo t (i.e., $\psi^4 \equiv 1 \bmod t$), and the two blocks of slots encoded respectively in $\alpha(\psi)$ and $\alpha(\psi^3)$ get shifted by applying $z \to -z$.[9]

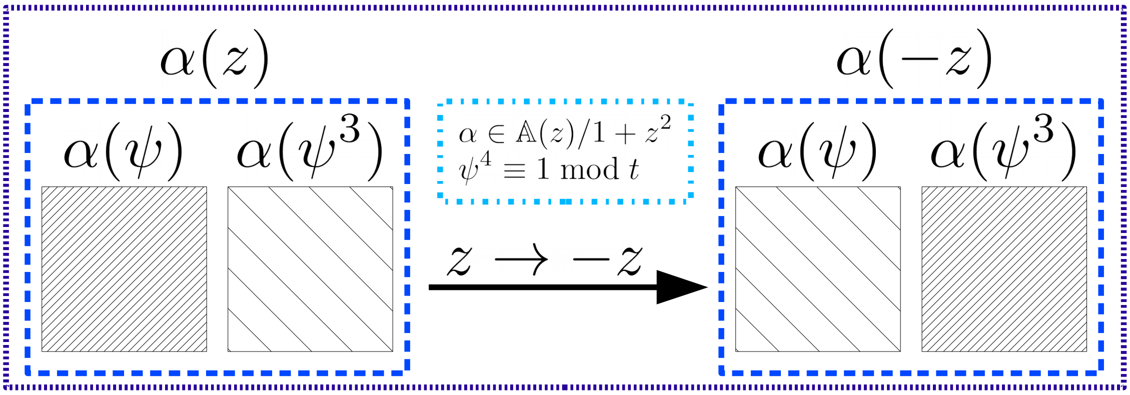


Figure 3.3: Representation of the rotation between two blocks of slots encoded in α.

Going back to the notation $R_t[x_1, \ldots, x_l]$ with $f_j(x_j) = 1 + x_j^2$ for our ring, we can then apply combinations of these two transformations with the different variables x_j for $j \in [l]$. Analogously to [99], this gives a multidimensional structure on the automorphisms group considering the composition of transformations

$$\theta_{i_1,\ldots,i_l}(\alpha) = \theta_{i_1}^{(x_1)}(\theta_{i_2}^{(x_2)}(\ldots \theta_{i_l}^{(x_l)}(\alpha)\ldots)) \in R_t[x_1, \ldots, x_l],$$

where $\alpha \in R_t[x_1, \ldots, x_l]$, $t \equiv 1 \bmod 4$ and $i_1, \ldots, i_l \in \mathbb{Z}_4^*$.

This multidimensional structure of the automorphisms group can be seen as an l-tuple with 2 different values per component (which gives a total of 2^l different automorphisms). Hence, similarly to the shift property of a multidimensional DFT [104], this group satisfies both the abelian and sharply transitive properties required to perform any type of permutation [105].

Logarithmic increase in space and computational cost (Strategy 1): The effect of each of the automorphisms over the slots can be visually represented as a hypercube with as many dimensions as independent variables the rings have, that is, with a total of $\log_2 n$ dimensions. As a graphical example, Figure 3.4 shows the slot structure corresponding to a multivariate ring with 7 independent variables; in this case, each different vertex of the hypercube represents one of the $n = 128$ available slots, where the allowed transitions between vertices depend on the chosen strategy, as we describe next.

In case of storing n switching key matrices (corresponding to all the automorphisms), any vertex transition will be allowed through one single switching key operation. However, it is possible to store less switching key matrices (which, combined, represent the whole set of automorphisms), hence increasing the number of subsequent automorphisms/switching key operations for transitioning from one vertex to another.

Due to the specific structure of our multivariate rings, we propose an optimal strategy with $\log_2 n$ switching key matrices, each one corresponding to a different transformation $x_i \to -x_i$; with the additional advantage that these transformations are their own inverses. Following this strategy, we can also see the different slots (vertices in Figure 3.4) as a binary vector of length $\log_2 n$, where the available operations are bit-wise XOR operations with vectors

[9] With rings $\mathbb{A}[z]/d + z^2$ we have similar automorphisms $\{z \to z\}$ and $\{z \to -z\}$.

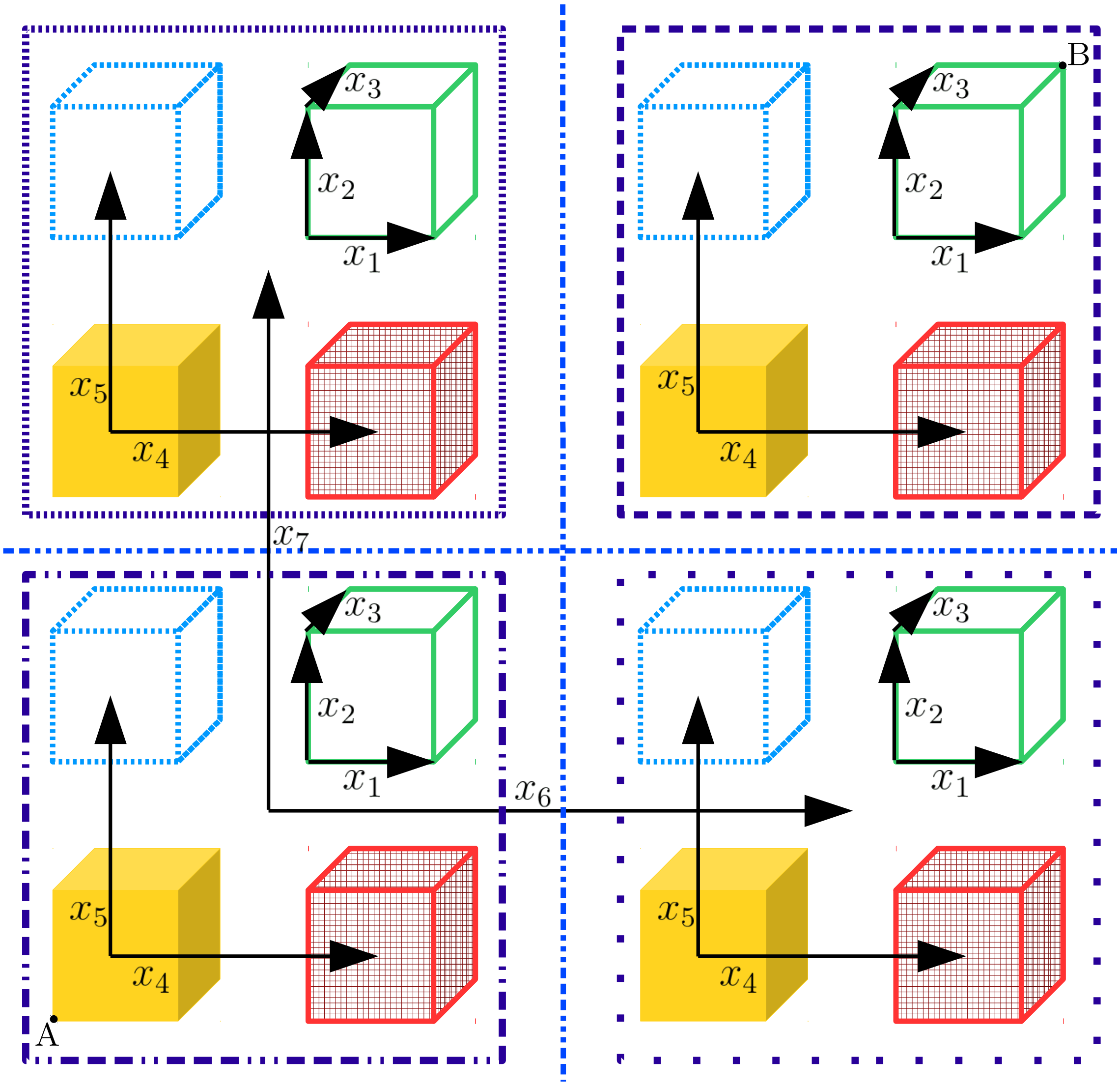


Figure 3.4: Representation of the hypercube structure of the group of automorphisms available in the multivariate polynomial RLWE with $\Phi_4(\cdot)$ as modular function and considering 7 independent variables $\{x_1, \ldots, x_7\}$.

$\{(1, 0, \ldots, 0), (0, 1, 0, \ldots, 0), \ldots, \ (0, \ldots, 0, 1)\}$ belonging to the standard basis of dimension $\log_2 n$. In the example of Figure 3.4 (with $\log_2 n = 7$), this method would be equivalent to working with 7 independent vectors (of the standard basis) enabling only movements between vertices in the dimension associated to the vector.

It can be seen that with this strategy the farthest slot to a given one is always the slot represented as its ones' complement, i.e., the opposite vertex. This implies a total of $\log_2 n$ automorphisms/switching key operations. Hence, in the worst case we have an increase in the computational cost by a factor of $\log_2 n$ when storing $\log_2 n$ switching key matrices and working with n slots. This is a considerable reduction in the memory requirements when compared to the approximately $\mathcal{O}(D)$ and $\mathcal{O}(\sqrt{D})$ factors considered by Halevi and Shoup [99] when working with D slots (in one dimension).

As a quick comparison, for the practical values reported in [99], i.e., $n = \phi(m) = 16384$, our strategy achieves an increase factor of 14 on the computational cost, which is not considerably higher than their results, but with huge savings in storage for our case: we store only 14 matrices, compared to the 51 matrices and 3 automorphisms/switching key operations achieved by [99] for a similar value of $\phi(m) = 15004$ and one dimension with $D = 682$ following a baby-step/giant-step strategy (see Appendix 3.A).

Finally, it must be noted that when applying a switching key, noise constraints force the need of decomposing the coefficients of the involved polynomials in some specific base.[10] As this decomposition does not straightforwardly commute with the NTT/INTT (or CRT over the polynomial modular function) representation, the inverse and direct transforms have to be applied over the polynomials. Our setting in multivariate rings with FWHT enables a reduction on complexity for these transforms by a factor of $\mathcal{O}(\log n)$ in terms of elemental products; i.e., *this yields a net gain factor of* $\log n$ *in storage while keeping the same order of (multiplicative) computational complexity.*

Efficiency/space tradeoffs: In practical scenarios, the tradeoff between used memory and computational cost might require a different balance with less space efficiency than the $\log_2 n$ achieved by the described strategy. Consequently, we also cover two additional strategies which lead to an improvement of the computational cost by a factor of 2.

Strategy 2: Our first approach adds to the previous $\log_2 n$ matrices those which are associated to "diagonal" vectors in our hypercube representation of the autormorphisms (see Figure 3.4); that is, we work with automorphisms $\{x_i \to x_i^{l_i}, x_j \to x_j^{l_j}\}$ where $l_i, l_j \in \mathbb{Z}_4^*$ and $i, j \in [\log_2 n]$, being $i \neq j$. Going back again to the binary representation of the slots, the additional automorphisms could be seen as the result of all pairwise XOR operations of different vectors of the standard basis of length $\log_2 n$.

The number of needed switching key matrices is therefore increased to

$$\binom{1+\log_2 n}{2} = \frac{(1+\log_2 n)\log_2 n}{2}.$$

In order to calculate the associated computational cost for this strategy, we resort to induction, working first with the odd natural numbers, and afterwards with the even natural numbers. Let the multivariate ring $R_t[x_1, \dots, x_l]$ with $f_i(x_i) = 1 + x_i^2$ where $i = 1, \dots, l$ and $l = \log_2 n$, if we consider only the odd values of l we have:

- For $l = 1$, any transition can be applied with only one automorphism/relinearization operation.
- Assuming that l variables require k automorphisms/relinearization operations, it can be shown that adding two variables (i.e., $l + 2$), $k + 1$ automorphisms/relinearization operations are needed. We can graphically see this by resorting to the binary representation: moving between any two slots implies, in the worst case (consider one vector and its ones' complement), one additional XOR operation.
- Therefore, by induction, odd values of l require $\lceil \frac{l}{2} \rceil$ automorphisms/relinearization operations.

[10] This is true unless we resort to the strategy of Bajard *et al.* [95] which takes advantage of the CRT decomposition over the polynomial coefficients. However, this strategy cannot be applied always, as it requires a highly composite modulo with primes of an adequate machine size (see [94]).

The argument is analogous for even l. First, we consider $l = 2$, where with only one automorphism/relinearization operation is enough to move between any of the slots. Next, the same reasoning as before could be applied between l and $l+2$ variables, resulting in a total of $\frac{l}{2}$ automorphisms/relinearization operations for l variables.

Taking into account both results, this strategy yields an increase in the number of automorphisms/switching key operations by a factor of $\lceil \frac{\log_2 n}{2} \rceil$. Hence, we can reduce by a half the computational cost compared to our previous strategy, with a quadratic increase in the memory requirements of $\frac{(1+\log_2 n)\log_2 n}{2}$ instead of $\log_2 n$. For instance, with $n = 16384$ this would give an increase in cost by a factor of 7 and a total of 105 stored matrices.

Strategy 3: The incurred increase in space requirements by Strategy 2 might not be acceptable for certain applications; therefore, our next approach preserves the cost improvement, but achieving a negligible increase in the number of required matrices: $1 + \log_2 n$ matrices instead of $\mathcal{O}((\log n)^2)$.

The idea behind this approach is adding to the switching key matrices for transformations of the form $\{x_i \rightarrow -x_i\}$ for $i = 1, \ldots, \log_2 n$ the following one

$$\{x_1 \rightarrow -x_1, \ldots, x_{\log_2 n} \rightarrow -x_{\log_2 n}\}.$$

As a graphical explanation, let us consider again the binary representation of the slots: in addition to working with those XOR operations with vectors belonging to the standard basis of length $\log_2 n$, now we can also apply the ones' complement of every "slot" in one operation (e.g., in Figure 3.4 we could directly move with one automorphism/switching key operation from point A to point B).

Therefore, the worst case automorphism requiring $l = \lceil \frac{\log_2 n}{2} \rceil$ matrices with our first strategy can now be computed with just one matrix. Moreover, as we know that $l - \lceil \frac{l}{2} \rceil \leq \lceil \frac{l}{2} \rceil$ for any $l \in \mathbb{N}$, then the farthest slot position can be achieved by only $\lceil \frac{l}{2} \rceil = \lceil \frac{\log_2 n}{2} \rceil$ automorphisms. Consequently, we can see that with $1 + \log_2 n$ matrices, we only need a maximum of $\lceil \frac{\log_2 n}{2} \rceil$ automorphism/switching key operations. For instance, with $n = 16384$ this would give an increase in cost by a factor of 7 and a total of 15 matrices in terms of use of memory.

3.3.3. Automorphisms in Multivariate Power-of-Two Cyclotomic Rings

It can be useful to expand Definition 9 to also cover more general multivariate rings, which can be leveraged by some applications (see Section 2.7.3). Most of these applications consider a general multivariate ring as R and R_q, where each of the modular functions can be defined as different power-of-two cyclotomic polynomials $f_i(x_i) = x_i^{n_i} + 1$.[11]

In this section the discussed efficiency/space tradeoffs achievable with automorphisms on the FWHT-enabled rings will be expanded to these rings (at the cost of lacking the faster FFT algorithms for the negacyclic Hadamard transform).

[11] Analogously to the procedure we followed with multiquadratics, we exemplify these results with power-of-two cyclotomics. These results can be similarly extended to more general rings with the form $x_i^{n_i} + d_i$ (see Definition 11).

Table 3.1: Practical space/efficiency tradeoffs of automorphisms for $n = 16384$.

l	2	3	4	5	6	7
# Matrices	256	80	52	36	34	28
# Calls to switching key (worst-case)	2	3	4	5	6	7

Tradeoffs in the size/efficiency of automorphisms

We consider the ring R introduced in Definition 1; particularly, we work with $R_t[x_1, \ldots, x_l]$ where $t \equiv 1 \bmod 2n_i$ for $i = 1, \ldots, l$. Analogously to our derivation in Section 3.3.2, when working with an element $\alpha \in R_t[x_1, \ldots, x_l]$, we have the transformations

$$\theta_{i_1,\ldots,i_l}(\alpha) = \theta_{i_1}^{(x_1)}(\theta_{i_2}^{(x_2)}(\ldots \theta_{i_l}^{(x_l)}(\alpha)\ldots)) \in R_t[x_1, \ldots, x_l],$$

now with $i_j \in \mathbb{Z}_{2n_j}^*$ for all j.

This multidimensional structure can be seen again as an l-tuple, where each component has n_i different values, hence giving a total of $n = \prod_{i=1}^{l} n_i$ different automorphisms.

Strategy 4: Our main strategy works with $n_i - 1$ matrices for each variable x_i, where each switching key matrix will correspond to an automorphism $\{x_i \to x_i^{l_i}\}$ for $l_i \in \mathbb{Z}_{2n_i}^*$ (except $\{x_i \to x_i\}$) and $i = 1, \ldots, l$. This strategy yields a total of $\sum_{i=1}^{l} n_i - l$ matrices with a computational cost of l automorphism/switching key operations. Let us assume that all the matrices for every "univariate" change of variable have to be stored. However, the number of required matrices per "univariate" change of variable could be further improved [99] (that is, we could work with subsets $\mathbb{A}_i \in \mathbb{Z}_{2n_i}^*$ in such a way that the corresponding automorphisms would be $\{x_i \to x_i^{l_i}\}$ for $l_i \in \mathbb{A}_i$ and $i = 1, \ldots, l$). [12]

To ease the analysis, we consider those $n_i = n^{\frac{1}{l}}$ for $i = 1, \ldots, l$ (hence being all n_i equal).[13] This gives us several tradeoffs depending on l and n where we have $l(n^{\frac{1}{l}} - 1)$ matrices and an increase in the computational cost by a factor of l. Table 3.1 shows the number of required matrices and the increase in computational cost for $n = 16384$ and several values of l. As $n^{\frac{1}{l}}$ is not always a valid value (that is, a power of two), the choice of n_i can be optimized to achieve the smallest possible number of automorphisms ($\sum n_i$) such that $n = \prod n_i$.

Conversely, Table 3.2 summarizes the different tradeoffs we have presented in this section. It is worth noting that we have focused on power-of-two cyclotomic modular functions, but this strategy could also be considered with any other cyclotomic modular functions and those used in Definition 11 for *secure* multivariate RLWE instantiations.

3.4. Conclusions

This chapter provides a set of possible applications with the *secure* multivariate RLWE instantiations discussed in Chapter 2. The applications of these secure instantiations are numerous,

[12] For a brief summary of Halevi and Shoup full and baby-step/giant-step strategies, see Appendix 3.A.

[13] It important to remark that the condition $n_i = n^{\frac{1}{l}}$ for all i includes non-secure multivariate RLWE instantiations (see Chapter 2). However, as in practice they could be chosen of relatively similar size, we assume this equality to make the analysis easier.

Table 3.2: Space/efficiency tradeoffs of automorphisms.

Strategy	# Matrices	# Calls to switching key (worst-case)
Strategy 1 from Section 3.3.2	$\log_2 n$	$\log_2 n$
Strategy 2 from Section 3.3.2	$\frac{(1+\log_2 n)\log_2 n}{2}$	$\lceil \frac{\log_2 n}{2} \rceil$
Strategy 3 from Section 3.3.2	$1+\log_2 n$	$\lceil \frac{\log_2 n}{2} \rceil$
Strategy 4 from Section 3.3.3	$\approx n^{\frac{1}{l}} l - l$	l
Strategy 4 (general) from Section 3.3.3	$\sum_{i=1}^{l} n_i - l$	l

achieving improved space-time tradeoffs in the most critical lattice operations, and therefore enabling more efficient homomorphic processing and closing the gap to the realization of practical fully homomorphic encryption. In particular:

- We introduced the $\boldsymbol{\alpha}$-generalized Walsh-Hadamard Transform as the basic block that can replace Number Theoretic Transforms in multivariate rings, achieving an improvement on the computational complexity of degree-n polynomial products by a factor $\log(n)$ in terms of elemental multiplications, with additional savings in memory usage (see Section 3.2).
- We enabled net improvements in cryptographic primitives built on top of m-RLWE, such as efficient time and space computation of automorphisms, relinearizations, packing, unpacking and homomorphic slot manipulation, and, consequently, bootstrapping, improving on current achievable trade-offs in RLWE (see Section 3.3).

It is worth highlighting that some of the exemplified solutions are sketched out with negacyclic rings (mainly those from Section 3.3). We plan to extend these results to the more general multivariate rings showcased in previous chapter. In any case, we have briefly explained how this extension can be realized.

3.A. Full and Baby-step/giant-step

In a recent paper [99, 93], Halevi and Shoup introduce several improvements on the operations with automorphisms and their associated switching key matrices, implemented in HElib. To this aim, they take advantage of the underlying algebraic structure that can be found on the group of automorphisms in RLWE. Specifically, they exploit the fact that these automorphisms can have a multidimensional structure [105] which depends on the group $\mathbb{Z}_m^*/\langle t \rangle$.

The HElib library considers a "basis" $g_1, \ldots, g_d \in \mathbb{Z}_m^*$ where each element has "order" $D_1, \ldots, D_d$, respectively (each D_i is a positive natural number). This basis induces the following representation for the elements belonging to $\mathbb{Z}_m^*/\langle t \rangle$:

$$\{g_1^{e_1} \ldots g_d^{e_d} : 0 \leq e_i < D_i, i = 1, \ldots, d\}.$$

Due to the existing bijection between the slots and vectors $(e_1, \ldots, e_d)$ now we can independently apply rotations[14] in each different "hypercolum" i (where $i = 1, \ldots, d$) by means of one

[14] A rotation by h in the dimension i is defined as a map from the slot associated with $(e_1, \ldots, e_i, \ldots, e_d)$ to the slot $(e_1, \ldots, e_i + h \bmod D_i, \ldots, e_d)$.

(if the i-th hypercolumn is a good dimension) or two (if the i-hypercolumn is a bad dimension) automorphisms.[15]

Without exploiting this multidimensional structure, we would have to work with a total of $\phi(m)$ different matrices to represent all the available automorphisms in the ring R; in a practical scenario, $\phi(m)$ can easily be above one or two thousand. However, by taking advantage of the different dimensions, we could represent the different automorphisms with as many as $\sum_{i=1}^{d} D_i$ matrices, and roughly increase the number of required switching key operations by a factor of d.

In [99], the authors describe two main strategies for working in each of these dimensions:

- Full strategy: D_i matrices are needed for a dimension i and produce a cost of one or two automorphisms/switching key operations depending on whether i is a good or bad dimension.
- Baby-step/giant-step: $g + \lceil D_i/g \rceil - 1$ (roughly $\mathcal{O}(\sqrt{D_i})$) matrices are needed for a dimension i where $g = \lceil \sqrt{D_i} \rceil$; this yields a cost of two or three automorphisms/switching key operations depending on whether i is a good or bad dimension.

The HElib library [99] works by default with the full strategy for those dimensions of length at most 50 and with the baby-step/giant-step for higher lengths.

As an example, in [99] the authors report runtimes for the parameters $m = 15709$ where $\phi(m) = 15004$, $r = 22$ and only one dimension with $D = 682$, hence working with 682 slots. With a full strategy and considering a good dimension we would have a total of 681 matrices; and 51 matrices with a baby-step/giant-step strategy (682 and 52 matrices considering a bad dimension).

[15] We say that the i-th hypercolumn is a good dimension if the order of g_i in $\mathbb{Z}_m^*$ is D_i; otherwise it is considered a bad dimension.

Part II

A Toolbox for Secure Signal Processing

Chapter 4

Number Theoretic Transforms

This chapter is adapted with permission from IEEE: Alberto Pedrouzo-Ulloa, Juan Ramón Troncoso-Pastoriza, and Fernando Pérez-González. Number Theoretic Transforms for Secure Signal Processing. IEEE Transactions on Information Forensics and Security, vol. 12, no. 5, pp. 1125-1140, May 2017.

4.1. Introduction

This chapter addresses the problem of *unattended secure signal processing* by providing a whole set of strategies and approaches to efficiently deal with composable unattended encrypted processing of sensitive signals, by relying on novel uses of Number Theoretic Transforms (NTTs) and appropriate pre- and post-processing techniques which enable efficient outsourced encrypted processing. Our proposal achieves a two-fold objective: replacing typical real or complex transforms for speeding up the underlying polynomial operations, and enabling an encrypted implementation of transformed processing in a flexible and efficient way. To the best of our knowledge, this was the first work that takes advantage of the polynomial structure of signals to represent them in a cryptosystem finite ring, where lattice cryptography can be very efficient, such that somewhat homomorphic cryptosystems can be leveraged to implement low-complexity and low-expansion ciphers and encrypted operations.

Main Contributions: Before delving into the description of our proposed techniques, we briefly enumerate our contributions here in order to clarify the targets and scope of this chapter:

- We propose the use of NTTs with Proth prime numbers as an efficient way for performing ciphertext multiplications.
- We present an efficient pre- and post-processing stage applied to the signals that allows us to perform: (a) very efficient generalized convolutions with only one ciphertext multiplication, including cyclic convolutions, (b) homomorphic NTTs with only one ciphertext multiplication, extensible to other typical fast transforms (like the Discrete Fourier Transform, DFT), and (c) any type of generalized linear convolution together with an NTT or INTT (respectively DFT or IDFT).
- We leverage the use of the relinearization primitive as a means to perform the pre- and post-processing homomorphically. Hence, we reduce the intervention of the secret key owner

in the middle of the process, allowing for a set of unattended encrypted signal processing applications. We also present several optimizations to further reduce key-owner intervention: (a) embed both the pre- and post-processing inside the homomorphic calculation, (b) enable component-wise multiplications together with encrypted linear convolutions without an intermediate decryption, (c) improve the efficiency and cipher expansion through batching/unbatching procedures.

- We introduce and discuss a set of exemplifying encrypted signal processing applications which can be performed thanks to our novel mechanisms, comprising, among others: elementary signal processing operations (shifts, changes in sampling rate, reflections, modulations), matrix multiplications, Cyclic Redundancy Check (CRC) codes, linear transforms and interleaving operations.

Structure: The rest of this chapter is structured as follows: Section 4.2 briefly reviews some preliminary notions and basic cryptographic concepts needed to develop the proposed approaches. Section 4.3 introduces the use of NTTs together with an optimal choice of parameters for enabling secure signal processing applications; Section 4.4 presents an approach to generalize convolutions and filtering in the encrypted domain; Section 4.5 proposes a series of optimizations to increase the efficiency of typical outsourced operations, and Section 4.6 exemplifies the use of the proposed techniques and primitives to produce a wide range of essential composable building blocks for unattended secure signal processing.

4.2. Preliminaries

The majority of the traditional SSP approaches make use of additive cryptosystems like Paillier [14], which enables the calculation of additions between encrypted values by multiplying their encryptions; however, additive homomorphisms lack flexibility for tackling more complex and non-linear operations. Hence, the use of lattice cryptosystems which present a ring homomorphism (addition and multiplication) is being progressively adopted by researchers in the field [25, 24, 106, 46]; an example is Lauter's cryptosystem [79], a Somewhat Homomorphic Encryption (SHE) based on the Ring Learning with Errors (RLWE) problem that can evaluate a bounded number of consecutive encrypted operations. Other recent representative RLWE-based examples are FV [86] and YASHE [87], cryptosystems that outperform Lauter's in terms of both efficiency and the upper bound on the number of encrypted operations. Moreover, novel lattice cryptosystems advance further in the direction of efficient processing and multi-key operation [84], and also in the fast execution of *bootstrapping* for achieving true Fully Homomorphic Encryption [107] (FHE).

We revisit here an RLWE lattice-based cryptosystem, which we use for our mechanisms, discussing the security properties of lattice cryptosystems and the choice of parameters; we also revise the basic form of Number Theoretic Transforms (NTTs), which are the building blocks that we use to produce efficient secure signal processing primitives.

4.2.1. RLWE-based cryptosystem

For the sake of the exposition, we have chosen Lauter [79] to showcase our proposed mechanisms, but they can be easily applied to any other RLWE-based cryptosystem (a brief comparison of some of the most recent homomorphic cryptosytems together with some additional reasons for choosing the Lauter cryptosystem can be found in Section 4.2.1). For completeness, a slightly adapted definition of the RLWE problem particularized to the case of Lauter cryptosystem is presented:

Definition 12 (RLWE problem [41], adapted from Definition 1 with $l = 1$)**.** *Given a polynomial ring $R_q[z] = \mathbb{Z}_q[z]/(1 + z^n)$ and an error distribution $\chi[z] \in R_q[z]$ that generates small-norm random polynomials in $R_q[z]$, RLWE relies upon the computational indistinguishability between samples $(a_i, b_i = a_i s + t \cdot e_i)$ and (a_i, u_i), where a_i, $u_i \leftarrow R_q[z]$ are chosen uniformly at random from the ring $R_q[z]$, while $s, e_i \leftarrow \chi[z]$ are drawn from the error distribution, and t is relatively prime to q.*

The fundamental primitives and parameters of Lauter's cryptosystem are described in Table 4.1. Lauter's ciphertexts are composed of at least 2 polynomial elements belonging to the ring $R_q[z]$; the cryptosystem allows for additions (the smallest ciphertext is previously zero-padded) and multiplications on these tuples of polynomials, whose size is increased after each multiplication (the original size can be brought back by resorting to the relinearization operation, explained in Section 4.5.1). The security of the cryptosystem is based on the hardness of reducing the n-dimensional lattices generated by the secret key and also on the semantic security provided by the RLWE problem (two encryptions of the same or different messages are indistinguishable). Further details about possible attacks to the cryptosystem are included in Section 4.2.2.

With this cryptosystem, messages encoded as univariate polynomials can be encrypted in only one ciphertext (instead of encrypting each coefficient in a different ciphertext). This has the main advantage of enabling to homomorphically perform encrypted linear convolution operations in a natural way with only one multiplication between ciphertexts; there is only a small overhead due to the larger cardinality of the involved encrypted polynomial coefficients, which belong to $\mathbb{Z}_q$ instead of the plaintext $\mathbb{Z}_t$, with $q > t$. In order to allow for D consecutive products and A sums over the same ciphertext, the needed q for correct decryption is lower-bounded by

$$q \geq 4(2t\sigma^2\sqrt{n})^{D+1}(2n)^{D/2}\sqrt{A}. \tag{4.1}$$

Remarkably, Lauter can be securely adapted to work efficiently with multidimensional signals (2D and 3D images or video), by extending the RLWE problem to a multi-variate case [4]; this extension enables working with complex-coefficient polynomials, by using bi-variate encryptions in which one of the modular polynomials is $f(w) = 1 + w^2$. We will revisit this idea for some of our constructions.

Choice of the RLWE-based cryptosystem

Although we have chosen the Lauter cryptosytem as the basis for our proposals, any RLWE-based cryptosytem can be used in order to apply the proposed methodologies and tools. The only requirement is the use of a modular function of the form $f(z) = 1 + z^n$ which, in fact, seems to be the most accepted and widely used by the cryptographic community due to its efficiency and well studied properties.

Besides its simplicity, there are some interesting motivations for our choice of the Lauter cryptosystem. Costache and Smart [83] recently presented a comparison in terms of efficiency, cipher expansion and security of the four main variants of RLWE-based cryptosytems: the NTRU and BGV schemes, which encode the messages in the lower bits of the decryption equation, and their corresponding scale-invariant versions YASHE and FV, encoding the messages in the upper

Table 4.1: RLWE-based Lauter Cryptosystem: Parameters and Primitives.

Parameters		
Let $R_t[z] = \mathbb{Z}_t[z]/(1+z^n)$ be the cleartext ring and $R_q[z] = \mathbb{Z}_q[z]/(1+z^n)$ the ciphertext's. The noise distribution $\chi[z]$ in $R_q[z]$ takes its coefficients from a spherically-symmetric truncated i.i.d Gaussian $\mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$; q is a prime $q \equiv 1 \bmod 2n$, and $t < q$ is relatively prime to q.		
Cryptographic Primitives		
SH.KeyGen	Process	$s, e \leftarrow \chi[z]$, $a_1 \leftarrow R_q[z]$ $sk = s$ and $pk = (a_0 = -(a_1 s + te), a_1)$
SH.Enc	Input	$pk = (a_0, a_1)$ and $m \in R_t[z]$
	Process	$u, f, g \leftarrow \chi[z]$ and the fresh ciphertext is $\boldsymbol{c} = (c_0, c_1) = (a_0 u + tg + m, a_1 u + tf)$
SH.Dec	Input	sk and $\boldsymbol{c} = (c_0, c_1, \ldots, c_{\gamma-1})$
	Process	$m = \left(\left(\sum_{i=0}^{\gamma-1} c_i s^i\right) \bmod q\right) \bmod t$
SH.Add	Input	$\boldsymbol{c}_0 = (c_0, \ldots, c_{\beta-1})$ and $\boldsymbol{c}_1 = (c'_0, \ldots, c'_{\gamma-1})$
	Process	$\boldsymbol{c}_{add} = (c_0 + c'_0, \ldots, c_{\max(\beta,\gamma)-1} + c'_{\max(\beta,\gamma)-1})$
SH.Mult	Input	$\boldsymbol{c}_0 = (c_0, \ldots, c_{\beta-1})$ and $\boldsymbol{c}_1 = (c'_0, \ldots, c'_{\gamma-1})$
	Process	Using a symbolic variable v their product is $\left(\sum_{i=0}^{\beta-1} c_i v^i\right) \cdot \left(\sum_{i=0}^{\gamma-1} c'_i v^i\right) = \sum_{i=0}^{\beta+\gamma-2} c''_i v^i$

bits of the decryption equation. They show that the most efficient schemes for the case of small and large moduli in the plaintext coefficients are respectively YASHE and BGV cryptosystems, where the former performs only slightly better than BGV for very small plaintext moduli ($t = 2$). Therefore, we chose Lauter cryptosystem as a representative of the BGV family, as a large number of signal processing applications work with reasonably large signal values and some of our contributions assume a relatively large value for t.

Finally, a recent attack [108] against NTRU cryptosytems also affects YASHE for some practical values that were considered secure until now. Considering λ as the security parameter, this attack allows to break these cryptosytems in sub-exponential time in λ for super-polynomial $q(\lambda)$; and even in polynomial time when $q(\lambda)$ increases. As this attack has no known effect on the BGV cryptosystem, our choice seems to be the most suitable thanks to both its efficiency and security.

4.2.2. Security of Lattice-based Cryptosystems

This section revisits some practical aspects related to security of lattice cryptosytems. The underlying assumption supporting the security of the used cryptosystems is the indistinguishability of the RLWE distribution w.r.t. a uniform distribution. There are mainly two types of attacks that can be considered: (a) distinguishing attacks [109], whose goal is to break the indistinguishability assumption through basis reduction algorithms, and (b) decoding attacks, which are aimed at obtaining the secret key s. We focus on the former.

Although we do not specifically account for decoding attacks, by using values for n similar to those used in [79], the cryptosystem achieves protection against them as described in [110]. Therefore, we adhere to these minimum values for n.

Security and runtime attack as a function of the root Hermite factor δ

The best attacks against lattice cryptosystems rely on basis reduction algorithms. Given an arbitrary basis of a lattice, these algorithms try to obtain a nearly orthogonal basis with small vectors. Among them, BKZ [111] is currently one of the most efficient ones. It uses blocks of size

Table 4.2: Performance of Lauter cryptosystem ($D = 1$, $A = 1$, $t = 65537$, $s = \sqrt{2\pi}$) and Paillier cryptosystem.

Lauter cryptosystem					
n	1024	2048	4096	8192	16384
$\lceil \log_2(q) \rceil$	53	55	56	58	59
δ	1.0090	1.0046	1.0024	1.0012	1.0006
Bit security (Eq.(4.3))	≈ 30	≈ 162	≈ 410	≈ 930	≈ 1270
Encrypt. time (μs)	114	224	444	860	1780
Decrypt. time (μs)	37	77	159	326	695
Poly. Mult. time (μs)	24	46	91	171	353
Poly. Add. time (μs)	7	12	24	47	107
Pre/Post time (μs)	4	8	17	32	64

Paillier cryptosystem				
Modulus size (bits)	1024	2048	3072	7680
Bit security	≤ 80	112	128	192
Encrypt. time (μs)	2947	19438	54122	521981
Decrypt. time (μs)	2806	19269	54006	521761
Scalar Mult. time (μs)	27	92	182	729
Scalar Add. time (μs)	3	10	21	87

ranging from 2 to the dimension of the lattice; increasing block sizes produce better bases at the cost of a higher computational load.

We take as a commonly adopted measure of security the root Hermite factor $\delta > 1$ for the underlying lattice, which is directly related to the running time needed for a basis-reduction algorithm to succeed. In fact, the runtime of an attack is approximately proportional to $e^{k/\log \delta}$ for a constant k; i.e., a lower δ implies a higher security. For the optimal distinguishing attack using BKZ, we obtain the following expression for δ [79]:

$$\log_2(\delta) = (\log_2(c \cdot q/s))^2/(4n\log_2(q)), \tag{4.2}$$

where n is the rank of the lattice, $c \approx \sqrt{\ln(\frac{1}{\epsilon})/\pi}$, ϵ is the attacker advantage ($\epsilon = 2^{-32}$), and s is a scale parameter of the error distribution (for the n-dimensional Gaussian $s = \sigma\sqrt{2\pi}$).

In order to calculate the corresponding bit security, we resort to the lower bound estimate $t_{BKZ}(\delta)$ of [110]:

$$t_{BKZ}(\delta) = \log_2\left(T_{BKZ}(\delta)\right) = \frac{1.8}{\log_2 \delta} - 110. \tag{4.3}$$

For the other cryptosystem parameters, we choose $\epsilon = 2^{-32}$, $\sigma = 1$ and for q, we choose the smallest prime that satisfies the bound (4.1), where we have $A = 1$, $D = 1$ and $t = 65537$. Table 4.2 shows different runtimes and the relevant security parameters (δ and bit security) of the used cryptosystem.

Paillier cryptosystem performance: Paillier cryptosystem [14] has been extensively used in recent years for secure signal processing. Therefore, we compare the efficiency of our proposed solutions exploiting Lauter with typical solutions resorting to Paillier. Due to the different hardness problems in which both cryptosystems are based on, we base our fair comparisons on the bit-security that can be achieved with both schemes. Table 4.2 reports the corresponding runtimes and bit security for different modulus size of the Paillier cryptosystem (with plaintext values upper-bounded by 256). For Paillier, we resort to the bit security estimate of RSA [112]. It is also important to note that Paillier can only deal with one scalar plaintext, while all the primitives using Lauter work in parallel with n plaintext values encrypted in one ciphertext, each one encoded in a different coefficient of the polynomials in $R_t[z]$, which is a clear advantage. Despite that, Table 4.2 shows that the encryption of one scalar with Paillier is much slower than the time needed to perform our proposed pre-/post-processing (see Section 4.4) and the encryption of n numbers with Lauter.

4.2.3. Number Theoretic Transforms (NTTs)

Signal processing heavily relies on transformed processing, for which the usually employed transforms are based on the DFT (Discrete Fourier Transform), due to the physical meaning of the frequency domain, the efficient algorithms for their computation, the good energy compaction properties, and the possibility of taking advantage of the cyclic convolution property. The latter implies a correspondence between the cyclic convolution of two signals and the element-wise product of their transforms, enabling very efficient computation of convolutions by working in the transformed domain.

When dealing with secure encrypted processing, DFTs cannot be directly translated to the encrypted signals, due to their reliance on complex arithmetic and non-integer numbers, which cannot fit in the finite rings of the cryptosystems without a quantization; this poses subsequent problems of accuracy loss and scale factor accumulation (cipher blow-up). When working in finite rings, we can find an alternative approach by resorting to integer transforms, more amenable to encrypted processing: NTTs (Number Theoretic Transforms) are transforms with the same structure as the DFT, with the peculiarity that they operate with elements belonging to a finite field or ring instead of the complex field.

More formally, in a finite ring $R_p = \mathbb{Z}_p[z]/f(z)$ with $p = \prod_{i=1}^{K} p_i^{m_i}$, an NTT of size N can be defined (with the cyclic convolution property) if the following properties hold [104]:

- There exists an N-th root of unity α in R_p, for which $\gcd(\alpha, p) = \gcd(N, p) = 1$.
- N is a divisor of $\gcd(p_1 - 1, \ldots, p_K - 1)$.

The expressions for the forward and inverse transforms are

$$\begin{aligned} X[k] &= \sum_{l=0}^{N-1} x[l]\alpha^{lk} \bmod p,\ k = 0, 1, \ldots, N-1 \\ x[l] &= N^{-1} \sum_{k=0}^{N-1} X[k]\alpha^{-lk} \bmod p,\ n = 0, 1, \ldots, N-1. \end{aligned} \tag{4.4}$$

Analogously to DFTs, NTTs possess a cyclic convolution property, and they also enable fast computation algorithms like radix-2 and radix-4. Remarkably, the NTTs lived a golden age in Signal Processing when the available hardware at the time (FPGAs and DSPs) could only work with finite precision arithmetic, but were later replaced due to the generalization of floating-point-capable hardware.

For our purposes, their important property is that they work in the same integer rings as lattice cryptosystems do and, therefore, they impose no rounding errors or cipher blow-up. Consequently, NTTs can be used to efficiently perform polynomial multiplications, and they have been recently proposed as a means to speed up finite-ring polynomial multiplications: there are some cryptosystem realizations that make use of NTTs for improving the efficiency of their polynomial operations [113, 114, 115, 116, 117]. Our proposed techniques go further, by focusing on an unexplored specific subset of the available NTTs and adjusting the cryptosystem parameters accordingly, to produce new primitives that enable highly efficient implementations, as we show in the next sections.

4.3. Number Theoretic Transforms in Secure Signal Processing

Once we have introduced the notions of lattice-based somewhat homomorphic cryptosystems and the basic formulation for NTTs, we discuss the proposed setup for the optimal combination of these two concepts with the goal of achieving extremely efficient, unattended outsourced signal processing.

We particularize the NTT for its application to the cryptosystems presented in [79] and [4]. In that case, the cryptosystems require the use of a ring $R_q = \mathbb{Z}_q[z]/(1+z^n)$, where q is prime, and n is a power of 2. Additionally, q must verify $q \equiv 1 \mod 2n$ and meet the lower bound defined by the number of operations allowed on the same ciphertext, cf. Eq. (4.1). Therefore, combining these restrictions with the existence of the size-n NTT in the ring R_q, we have

- n divides $\phi(q)$, where $\phi(q) = q - 1$.
- q verifies both Eq. (4.1) and $q \equiv 1 \mod 2n$.

Prior works simply assume that such a prime exists and do not address its generation or adaptation to efficient processing with a given cryptosystem. We can prove that these restrictions are verified by the set of Proth primes [118, 119, 120], which can be easily generated. A Proth number q is characterized by the form $q = k2^l + 1$ where l is an integer, k is an odd positive integer and $2^l > k$. The primality test for Proth numbers follows by virtue of Proth's theorem:

Theorem 6 ([121]). *For a Proth number q, if there is at least an integer a satisfying* $a^{(q-1)/2} \equiv -1 \mod q$*, then q is prime.*

Once q is fixed, an n-th root of unity α can be found by searching those numbers that have an order greater than or equal to n in the set of integers $[1, q-1]$. If the found α has an order d higher than n, then the n-th root of unity is obtained by considering $\alpha^{(\frac{d}{n})n} \equiv 1 \mod (q)$, where $\alpha^{\frac{d}{n}}$ is an n-th root of unity. Algorithm 2 details this procedure.

This choice of parameters enables efficient cyclic convolutions between the ciphertext elements with no rounding errors, as they allow for efficient algorithms for NTT and INTT (e.g., radix-2 or radix-4). The particularity here is that the cyclic convolutions allowed by this setting are actually nega-cyclic, and further processing has to be applied to enable "regular" cyclic convolutions, as we explain in the next section.

4.4. Generalizing Convolutions and Transforms in the Encrypted Domain

Equipped with the presented description of RLWE cryptosystems and the proposed optimal parameters for NTTs, we detail here the main contribution of this chapter, comprising a versatile

Algorithm 2 Cryptosystem's Parameters (q, n, α) Generation .

```
procedure SEARCH CRYPTOSYSTEM PARAMETERS(N)
Input:
    $N \leftarrow$ desired size for the NTT (where N is a power of 2)
Search prime $p$ for the NTT of size $N$:
    while $p$ not found do
        Find $p$ of the form $p = k2^l + 1$ (with $k$ and $l$ natural numbers, where $k$ is not a power of 2)
        if $p$ is prime and $N$ divides $p - 1$ then
            break;
Search $N$-th root of unity $\alpha$:
    for $i = 1$ until $p - 1$ do
        if $i$ is a $M$-th root of unity with $M \geq N$ then
            $\alpha \leftarrow i^{M/N}$
            break;
Output:
    For the ring $R_q$ we have $q = p$ and $n = N$
    An $N$-th root of unity $\alpha$
```

set of novel secure signal processing primitives:

- We show how to efficiently perform any encrypted cyclic, negacyclic or generalized convolution in an RLWE-based cryptosystem in a more efficient way and without wasting any coefficient. For that purpose, we propose an efficient pre- and post-processing for the input signals and the result respectively, enabling further operations in the encrypted domain.
- In order to allow for transformed operations under encryption, we propose a practical method for computing an encrypted NTT or DFT with an RLWE-based cryptosystem.
- Finally, both results are generalized in a framework that enables any kind of encrypted convolution and linear transforms with a convolution property.

4.4.1. Encrypted Cyclic Convolution

Along with the linear convolution, the circular or cyclic convolution is frequently used in signal processing. To implement a linear convolution with an RLWE cryptosystem [79, 78] and its extensions [4], we need a value of n large enough to store the result of the convolution. Moreover, the cyclic convolution poses additional problems as the modular function $f(z) = 1 + z^n$ in the cryptosystem only allows for negacyclic convolutions [122]. A straightforward approach for calculating the cyclic convolution would be the following:

- The cryptosystem modular function is of the form $f(z) = 1 + z^N$, with N power of two.
- The larger signal is assumed to have length $N/2$ (in other case, it is zero padded).
- By the homomorphic properties of the cryptosystem, the allowed polynomial multiplication enables computing $x(z)h(z)(1 + z^{N/2}) \bmod (1 + z^N)$.

It can be shown that the output of this product holds the result of the negacyclic convolution in the $N/2$ first coefficients, and the result of the cyclic convolution in the last $N/2$ coefficients. The drawbacks are that: (a) half of the used coefficients are wasted, unnecessarily increasing cipher expansion, and (b) the result is located in a portion of the ciphertext, so reusing it for further operations becomes harder.

We present our method for calculating the encrypted cyclic convolution, by using just one polynomial product and element-wise pre- and post-processing. This approach yields a more efficient cipher expansion, and it also enables to continue performing operations with the results of the convolutions.

Efficient Pre- and Post-processing

We rely on a generalization of the cyclic convolution between two signals $x[l], h[l]$ in terms of a complex value α, proposed by Murakami [54]. The α-generalized cyclic convolution is defined as

$$y(z) = x(z)h(z) \bmod (1 - \alpha z^n), \tag{4.5}$$

where $x(z)$, $h(z)$ and $y(z)$ are the Z-transforms of $x[l]$, $h[l]$ and $y[l]$ (we use the definition for the Z-transform as a power series in z instead of the more common z^{-1}).

This generalization lets us specify different types of convolutions depending on the chosen α: for $\alpha = -1$ we obtain a negacyclic convolution (we refer the reader to [122] for more details on negacyclic convolutions), which corresponds to the homomorphic operation offered by RLWE-based cryptosystems with $f(z) = 1 + z^n$. Conversely, $\alpha = 1$ conforms to the cyclic or circular convolution. We aim at a regular cyclic convolution ($\alpha = 1$), but we are bound to a negacyclic one by the cryptosystem homomorphism, as a modular function of the form $f(z) = 1 - z^n$ would not be irreducible in the integers (see [109, 123] for further details on the security reasons behind discarding $1 - z^n$ as the modular function).

Supported by Murakami's formulation, we can enable the calculation of a cyclic convolution between two N-length polynomials $x[l]$ and $h[l]$ by carrying out the following steps:

- Prior to encryption, the input signals are pre-processed with component-wise products:

  $$\begin{aligned} x'[l] =& x[l]\alpha^{-l/N}(-1)^{l/N}Q,\ l = 0, \ldots, N-1, \\ h'[l] =& h[l]\alpha^{-l/N}(-1)^{l/N}Q,\ l = 0, \ldots, N-1, \end{aligned}$$

  where Q is the quantization applied to signals $x'[l], h'[l]$.

- Then $y'(z)$ can be calculated under encryption with a homomorphic polynomial product

  $$y'(z) = x'(z)h'(z) \bmod (1 + z^N)$$

- The output decrypted signal $y'[n]$ is post-processed

  $$y[l] = y'[l]\frac{\alpha^{l/N}(-1)^{-l/N}}{Q^2}.$$

With the described procedure, the α-generalized cyclic convolution can be successfully implemented with a single product of encrypted polynomials; in particular, the cyclic convolution can be implemented with $\alpha = 1$.

It must be noted that the element-wise product between the pre- and post-processing vectors is equal to a constant signal of ones, so the polynomial product between two pre-processed polynomials "preserves" the pre-processing and can be subsequently operated. Consequently, we can implement several products between ciphertexts and the results of ciphertext products without the intervention of the key owner in the middle of the process. That is, for performing several cyclic convolutions we only need the key owner to apply the element-wise pre-processing to the original encrypted signals.

A final remark must be made regarding the complex arithmetic assumed by Murakami's formulation. The N-th roots of -1 needed for building the pre- and post-processing vectors can be tackled in two ways: (a) complex numbers can be embedded in the cryptosystem by incorporating a modular function $f(w) = 1 + w^2$ to the multivariate ring [4], and (b) Murakami's concepts can be applied to finite rings, so that $\alpha^{1/N}$ and $(-1)^{1/N}$ are elements of $\mathbb{Z}_t$, and the conditions for the existence of the N-point NTT are still satisfied. In that case, we could discard the quantization Q and perform the encrypted α-generalized convolution without rounding errors.

Consequently, we have solved the two main limitations that current approaches have for calculating a cyclic convolution under encryption: our approach does not introduce any rounding errors, it does not need to discard any coefficient (reducing the effective cipher expansion), and it can cope with successive operations without intermediate decryptions. We evaluate now its performance in terms of computational complexity.

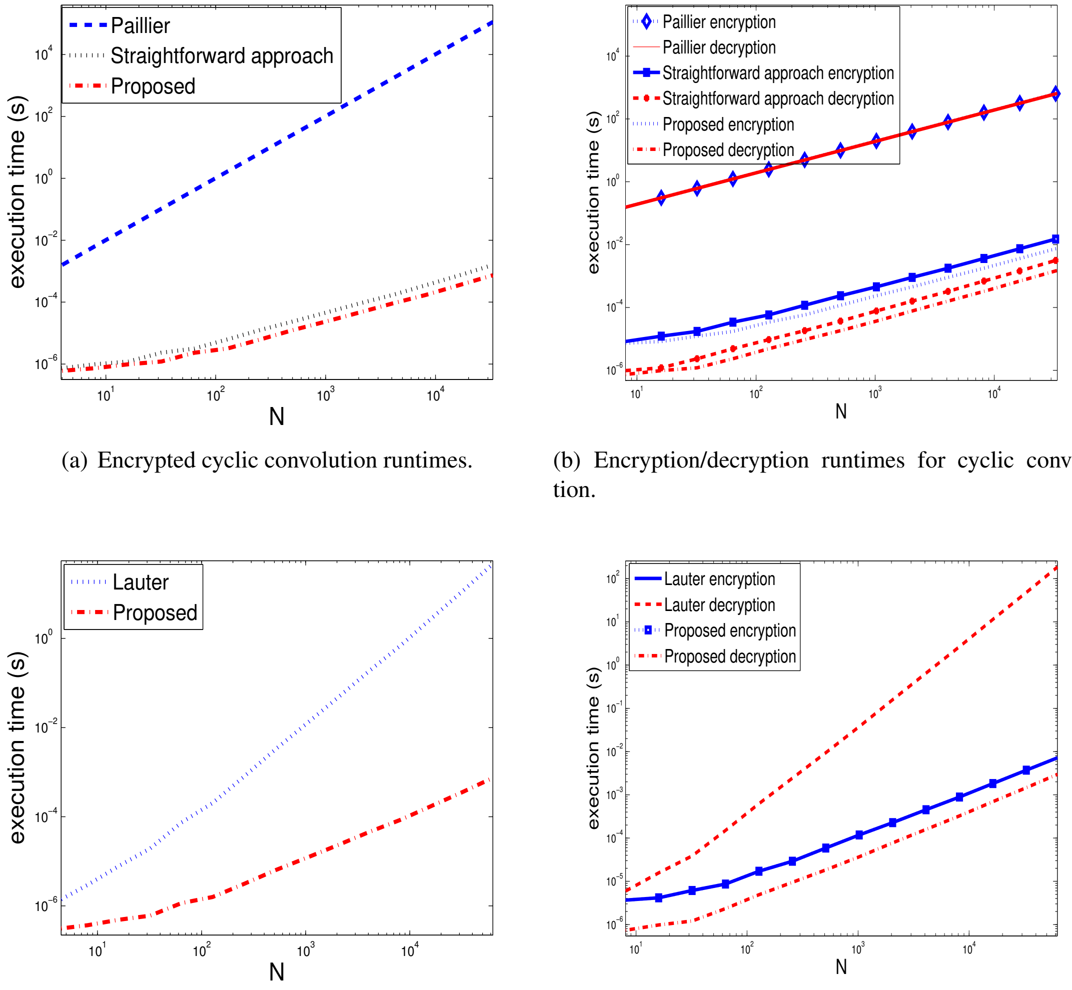
Performance evaluation of the encrypted cyclic convolution: We have implemented Lauter's RLWE-based cryptosytem in C++ using the GMP 6.0.0 [124] and NFLlib [94] libraries. Figures 4.1a and 4.1b compare the encrypted cyclic convolution performance with a 2048-bit modulus Paillier-based convolution (one of the convolved signals cannot be encrypted) versus the proposed method with Lauter's cryptosytem with $n = N$ on an Intel Xeon E5-2620 processor running Linux. We show the comparison of (a) the encrypted convolution, and (b) the encryption and decryption times with our pre- and post-processing. Additionally, the computational cost of performing a cyclic convolution of two encrypted signals with our scheme is lower than the straightforward method, due to the reduction in the needed coefficients (no coefficients are discarded).

We are not considering relinearization steps after each multiplication, but account for the decryption of the extended encryptions. We can see that RLWE-based cryptosystems are more efficient than Paillier, also having a much lower cipher expansion. Moreover, our method enables chaining several consecutive encrypted cyclic convolutions in a natural way.

Finally, we can see that in our method $n = N$; hence, when we increase the size of the encrypted signals we are also reducing the root Hermite factor δ, therefore increasing the runtime of a distinguishing attack (see Section 4.2.2) and achieving a much higher bit-security than Paillier (see Table 4.2).

4.4.2. Encrypted NTTs

With the relation of equivalence between the clear and encrypted convolutions, we can use the efficient radix-2/radix-4 algorithms of the NTTs for performing the negacyclic convolutions of encrypted signals as shown in Section 4.3. In this case, the NTT is applied directly on the encrypted signals as a means to speed up the calculation of the polynomial product (nega-cyclic convolution), which gets reduced to a component-wise product in the transformed domain.



(a) Encrypted cyclic convolution runtimes.

(b) Encryption/decryption runtimes for cyclic convolution.

(c) Encrypted NTT runtimes.

(d) Encryption/decryption NTT runtimes.

Figure 4.1: Performance of encrypted cyclic convolutions and NTT.

However, there are cases where the NTT must be applied to the clear-text signals, and we must replicate the computation of the NTT once the signals are already encrypted, in such a way that once we decrypt we get the transformed coefficients of the clear-text signal. Therefore, we aim here at the encrypted implementation of NTTs (homomorphically applied to the cleartext) independently of whether the underlying encrypted polynomial products are implemented with the aid of NTTs. Previous works have focused only on the implementation of the encrypted DCT or DFT [17], but not on NTTs. For this purpose, we propose a mechanism to obtain the NTT of a signal under encryption with only a cyclic convolution and a pre- and post-processing step. This procedure can also be applied to any other transform with a similar structure, with the difference of having to work with rounded real or complex numbers. In fact, we could separately operate with the real and imaginary parts of the signals or even embed complex numbers in the cryptosystem [4] by incorporating a modular function $f(w) = 1 + w^2$. Hence, we can use the same procedure to implement the encrypted versions of the corresponding real or complex transforms, but working over complex $2N$-th roots of unity; hence, by applying a pre- and post-processing step we get to homomorphically perform a DFT with only one cyclic convolution, or one DCT with two cyclic convolutions (for the DCT we would resort to Euler's formula to represent the cosines as a combination of complex roots of unity). It must be noted that unlike the NTT, both the DCT and the DFT would need quantization prior to encryption in order to be able to represent the real numbers as integers, with the corresponding increase in cipher expansion and quantization error.

We first introduce the proposed encrypted NTT algorithm, which we later extend for computing NTTs, INTTs and generalized cyclic convolutions.

Encrypted NTT with pre- and post-processing

By resorting to the formulation of Bluestein FFT algorithm (also called chirp Z-transform algorithm [125, 126]), we can compute the NTT of a signal as a single convolution and a pre- and post-processing. The expression for the NTT of a signal $x[l]$ is given in Eq. (4.4). We need that $\alpha^{\frac{1}{2}}$ be a $2N$-th root of unity in $\mathbb{Z}_t$ (and hence α is a N-th root of unity in $\mathbb{Z}_t$), so that we can write $kl = -\frac{(k-l)^2}{2} + \frac{l^2}{2} + \frac{k^2}{2}$. Hence,

$$X[k] = \alpha^{\frac{k^2}{2}} \sum_{l=0}^{N-1} \alpha^{\frac{l^2}{2}} x[l] \alpha^{\frac{-(k-l)^2}{2}}.$$

This shows the equivalence to a cyclic convolution followed by a component-wise product with $\alpha^{\frac{k^2}{2}}$

$$X[k] = \alpha^{\frac{k^2}{2}} \sum_{l=0}^{N-1} x'[l] \alpha^{\frac{-(k-l)^2}{2}} = \alpha^{\frac{k^2}{2}} (x'[k] \circledast \alpha^{\frac{-k^2}{2}}),$$

where $x'[l] = \alpha^{\frac{l^2}{2}} x[l]$ and $\circledast$ denotes the cyclic convolution (assuming N is even due to the cryptosytem requirements). Therefore, we can implement a generic NTT of N samples with a $2N$-th root of unity $\alpha^{\frac{1}{2}}$ by simply performing the pre-processing with $\alpha^{\frac{l^2}{2}}$, convolving the pre-processed signal with $\alpha^{\frac{-l^2}{2}}$ and post-processing the convolution result with $\alpha^{\frac{l^2}{2}}$.

This procedure allows to efficiently execute an encrypted NTT as shown in Figure 4.2. As negacyclic convolutions are the only homomorphically allowed convolutions, we resort to the pre- and post-processing shown in Section 4.4.1, which must be applied “inside” our convolution box (see Figure 4.2). Thus, $x'[l]$ is multiplied by the pre-processing vector before being encrypted. We apply the same pre-processing to $\alpha^{\frac{-l^2}{2}}$.

Finally, once the result is decrypted, we have to apply the component-wise post-processing for the cyclic convolution and, afterwards, the NTT post-processing.

The INTT (Inverse Number Theoretic Transform) implementation is analogous to the NTT, simply swapping the used signals and including a N^{-1} factor:

$$\begin{aligned} x[k] =& N^{-1} \alpha^{\frac{-k^2}{2}} \sum_{l=0}^{N-1} \alpha^{\frac{-l^2}{2}} X[l] \alpha^{\frac{(k-l)^2}{2}} \\ =& N^{-1} \alpha^{\frac{-k^2}{2}} ((\alpha^{\frac{-k^2}{2}} X[k]) \circledast \alpha^{\frac{k^2}{2}}). \end{aligned}$$

A new application enabled by encrypted NTT calculations is the element-wise signal multiplication. This is accomplished by simply leveraging the cyclic convolution property of the NTT to implement point-wise products as homomorphically allowed convolutions. Consequently, we obtain the desired product with an INTT of the decrypted result. While this could also be achieved with the DFT [17], the use of NTT avoids rounding and blow-up problems under encryption.

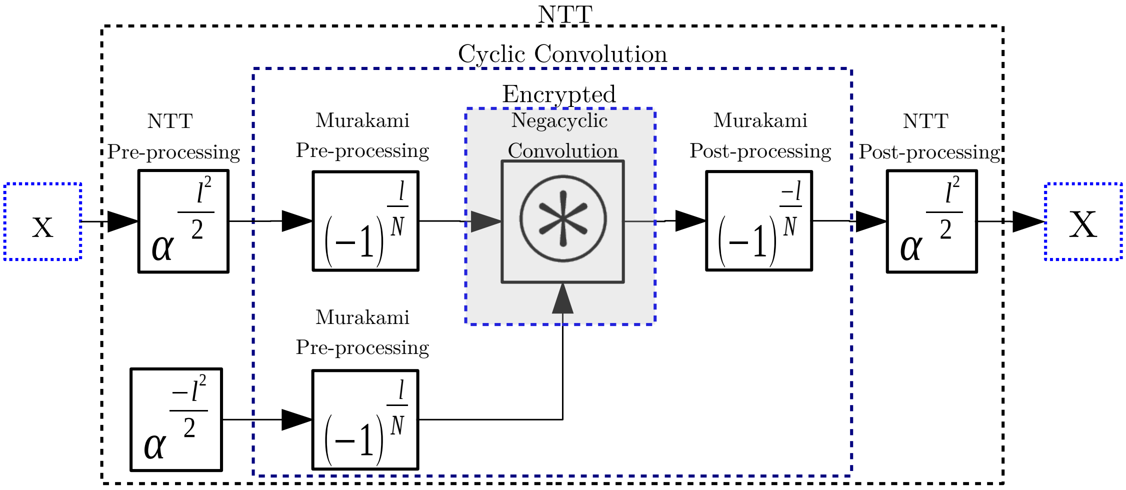


Figure 4.2: Block diagram for the implementation of the encrypted NTT.

Performance evaluation of the encrypted NTT: Prior works have proposed the use of the Paillier cryptosystem for performing the DFT [17]. Our method would require a multiplication step of the encrypted signal samples with powers of the corresponding N-th root of unity (see Section 4.2.3), which cannot be encrypted due to the limited homomorphism of Paillier. However, the security of Paillier relies on the hardness of computing $\phi(N)$ (Euler's totient function) without knowing the factorization of N. Of course, when the different powers of the N-th root of unity are known, $\phi(N)$ is disclosed.

As a consequence, Paillier cannot be used for calculating the NTT without resorting to a two-party protocol for secure multiplication [127] along with the corresponding increase of the execution overhead. Instead of Paillier, other schemes for which knowing the different powers of the N-th root of unity is not a security problem could be used (e.g., exponential El Gamal [128]), but they present additional drawbacks.

For this reason, we compare the efficiency of our proposed encrypted NTT with a straightforward encrypted realization of Eq. (4.4), in which one ciphertext multiplication is used for each output NTT coefficient. We use our aforementioned implementation of Lauter [124, 94] for comparing the runtimes of the different schemes. Figures 4.1c and 4.1d compare the encrypted NTT performance with the straightforward application of Eq. (4.4) and our proposed method, both with Lauter ($t = F_4 = 65537$ [1]). Our method enables the computation of the NTT with only one ciphertext multiplication instead of N, so the computational complexity is reduced in a factor of N.

Regarding the security, as we fix $n = N$, when we increase the length of the signals involved in the computation we also increase the achieved security (see Section 4.2.2).

Related works: In [129], Doröz *et al.* also homomorphically perform an NTT under encryption, exemplified under the LTV cryptosystem [84]. The main solution proposed in [129] takes advantage of a clever packing of the signals to encode each element of the original signal in different ciphertexts, and compute the corresponding fast algorithm for the NTT, eventually having the output of the NTT in N different ciphertexts (one per coefficient). In order to improve the

[1] F_4 is the fifth Fermat number, where Fermat numbers are defined as the set of numbers satisfying $F_l = 2^{2^l} + 1$ with $l \in \mathbb{N}$ including zero.

throughput they resort to batching, hence performing N parallel NTTs. The computational cost of their algorithm is equivalent to $\mathcal{O}(N^2 \log_2 N)$ elemental integer multiplications between ciphertext coefficients. Compared to our scheme,[2] it can be seen that we achieve the same computational cost; i.e., for one NTT we use one ciphertext product or $\mathcal{O}(N \log_2 N)$ elemental multiplications of coefficients. However, our solution presents two main advantages: (a) it is more flexible, as we do not have to pack several messages into one ciphertext and do not require packing/unpacking operations, and (b) our scheme only requires one homomorphic multiplication, while their solution requires $\log_2 N$ chained products over the same ciphertext, with the corresponding increase in ciphertext noise, in the required coefficient bitsize, and in complexity of the elemental operations, which our scheme does not incur.

If we compare the coefficient bitsize of our scheme $\log_2 q_1$ with respect to Doröz's $\log_2 q_2$, it can be shown that $(\log_2 q_2 - \log_2 q_1) \approx B \log_2 N - B$ bits, with $B > 0$ being a constant that depends on the cryptosystem parameters. Hence, our scheme is also more efficient in terms of coefficient size (cipher expansion), by a factor of $\log_2 N$. It must be noted that this also holds for a leveled cryptosystem, where their solution would require more levels and a deeper key chain.

The encrypted fast transform is always less space-efficient (due to the need of bigger q) than the direct implementation, but depending on the cost of the homomorphic products, the fast algorithm can be also less time-efficient than the naïve direct implementation due to the growth in plaintext size (accumulated quantization factors) that the former imposes, produced by its subsequent multiplications on the same ciphers ($\log_2 N$ levels), whereas the direct implementation only multiplies each cipher once. While this does not happen to Paillier [17], it is true for lattice-based cryptosystems. Therefore, in order to mitigate this effect in their work, Doröz *et al.* propose to implement the matrix multiplication associated to the NTT transform (direct transform) instead of the fast algorithm; hence, as their procedure only supports one multiplication with a cleartext constant, their cost to perform N parallel NTTs is $\mathcal{O}(N^3)$ elemental multiplications between coefficients. This is considerably outperformed by our solution.

4.4.3. Generalized Convolutions and Transforms

We can generalize the two prior primitives by adding the new pre- and post-processing and fixing one of the convolved signals, in such a way that we can achieve an INTT or NTT with any convolution type considered by Eq. (4.5). We can formulate this as a generalization of the Murakami scheme. The general matrix scheme is as follows:

$$\boldsymbol{X} = \boldsymbol{P}_{out} \boldsymbol{G}_{\beta} \left(\boldsymbol{P}_{in,y} \boldsymbol{y} \right) \boldsymbol{P}_{in,x} \boldsymbol{x},$$

with $\boldsymbol{P}_{out} = \boldsymbol{P}_2(\gamma^{\frac{-1}{2}}) \boldsymbol{P}_1(\beta^{\frac{-1}{N}})$, $\boldsymbol{P}_{in,x} = \boldsymbol{P}_1(\beta^{\frac{1}{N}}) \boldsymbol{P}_2(\gamma^{\frac{-1}{2}})$ and $\boldsymbol{P}_{in,y} = \boldsymbol{P}_1(\beta^{\frac{1}{N}}) \boldsymbol{P}_2(\gamma^{\frac{1}{2}})$; and where $\boldsymbol{x}$ and $\boldsymbol{y}$ are the two input vectors, and $\boldsymbol{X}$ represents the result vector. $\boldsymbol{P}_i(x)$ is the following matrix

$$\boldsymbol{P}_i(x) = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & x & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & x^{(N-1)^i} \end{pmatrix},$$

[2] Our algorithms and those in [129] are exemplified in different cryptosystems, but can be independently applied to LTV or Lauter, so we find it fairer to compare their theoretical computational costs in terms of elemental operations between ciphertext coefficients instead of implementation-dependent runtimes.

and $\boldsymbol{G}_\beta(\boldsymbol{v})$ is the β-generalized cyclic matrix of the vector $\boldsymbol{v}$

$$\boldsymbol{G}_\beta(\boldsymbol{v}) = \begin{pmatrix} v_0 & \beta v_{N-1} & \cdots & \beta v_1 \\ v_1 & v_0 & \cdots & \beta v_2 \\ \vdots & \vdots & \ddots & \vdots \\ v_{N-1} & v_{N-2} & \cdots & v_0 \end{pmatrix}.$$

The values for the different parameters depend on the choice of convolution or transform. In the case of cyclic convolutions with our cryptosystem (only negacyclic convolutions can be homomorphically performed), we consider $\gamma^{\frac{1}{2}} = 1$ and $\beta = -1$ (only Murakami pre-/post-processing is applied). The elements x_i and y_i, for an integer i such that $0 \leq i < N$, correspond to the samples of the signals we want to convolve.

Our NTT implementation uses $\gamma = \alpha^{-1}$, $\beta = -1$ and the elements y_i are equal to 1 for all i. On the other hand, an INTT would use $\gamma = \alpha$, $\beta = -1$, $y_i = 1$ for all i, and add a multiplication by N^{-1} as a post-processing step.

It must be noted that due to the requirements and structure of our cryptosystem, we implement the cyclic convolution with underlying negacyclic convolutions (i.e., we use $\beta = -1$). Conversely, it would be also possible to obtain a cyclic convolution, NTT or INTT by any other convolution type covered by Eq. (4.5) by simply using a different value of β.

4.5. Optimizations

This section presents a series of optimizations to the contributions of Section 4.4, targeted at: (a) efficiently performing the encrypted NTT operation by means of a relinearization primitive, (b) enabling component-wise encrypted products avoiding the pre- and post-processing needed for the encrypted NTT, therefore removing the need of interaction by the secret key owner for performing an encrypted NTT, and (c) enabling batch processing and maximizing the batched homomorphic capacity. For these purposes, we rely on a relinearization step and the CRT (Chinese Remainder Theorem), and we exploit the periodic structure of the input signals whenever they present it. We first revise the formulation of the relinearization primitive, and then explain the proposed optimizations.

4.5.1. Relinearization primitive

For the purpose of avoiding pre- and post-processing in the cleartexts, we can employ a *relinearization* primitive [79, 50, 88], commonly used in key switching algorithms to reduce the size of the encryptions after a multiplication: when multiplying two ciphertexts $\boldsymbol{c} = (c_0, c_1)$ and $\boldsymbol{c}' = (c_0', c_1')$ from the cryptosystems [79], [78] and [4], the number of elements of the resulting ciphertext is increased $\boldsymbol{c}'' = (c_0'', c_1'', c_2'')$. Hence, $\boldsymbol{c}''$ could be decrypted as $c_0'' + c_1''s + c_2''s^2$, which can be seen as a quadratic function of s.

This increase is undesired due to the induced overheads. Hence, a relinearization reduces $\boldsymbol{c}''$ to a new ciphertext formed by only two elements $\boldsymbol{c}''' = (c_0''', c_1''')$, satisfying $D(\boldsymbol{c}''', s) = D(\boldsymbol{c}'', s)$, where $D(\boldsymbol{c}, s)$ represents the decryption of $\boldsymbol{c}$ with key s (the decryption circuits for both cases are $c_0''' + c_1'''s$ and $c_0'' + c_1''s + c_2''s^2$, respectively). In order to perform this relinearization, the public key must comprise certain additional information about the successive powers of s, and circular security must hold for the cryptosystem to securely encrypt functions of the secret key. In case of applying the relinearization after each product, only information of s^2 is needed. As a drawback, the relinearization increases the ciphertext noise.

This is the conventional use of relinearization, but we use it additionally for performing other types of operations as upsampling, downsampling and reflections (see Section 4.6). We present now the formulation of a relinearization step; in our work we do not restrict the relinearization to only powers of the secret key s, and we consider a generic decryption for a ciphertext (c_0, c_1, c_2) as $c_0 + c_1 s_1 + c_2 s_2$, with $s_1, s_2 \leftarrow \chi$, where s_2 is not necessarily equal to s_1^2. It can be shown that the needed additional information would be $(h_1, \ldots, h_{\lceil \log_t q \rceil - 1})$, where the h_i are *key homomorphisms* defined as

$$h_i = (a_i, b_i = -(a_i s_1 + t e_i) + t^i s_2),\ i = 0, \ldots, \lceil \log_t q \rceil - 1,$$

where t is the module used for encoding the messages in R_t, $a_i \leftarrow R_q$ and $e_i \leftarrow \chi$. Expressing c_2 in base-t representation, we have $c_2 = \sum_{i=0}^{\lceil \log_t q \rceil - 1} c_{2,i} t^i$, and finally we obtain the ciphertext $(c_0^{relin}, c_1^{relin})$ under the key s_1 as

$$c_0^{relin} = c_0 + \sum_{i=0}^{\lceil \log_t q \rceil - 1} c_{2,i} b_i, \quad c_1^{relin} = c_1 + \sum_{i=0}^{\lceil \log_t q \rceil - 1} c_{2,i} a_i.$$

This step can be typically used either after each encrypted multiplication, in order to bring back the expanded ciphertext to a pair of polynomials, or after several consecutive multiplications, by using relinearizations h_i for each different key power. In general, given a ciphertext $(c_0, c_1, \ldots, c_{m-1})$, whose decryption function has the form $c_0 + \sum_{i=1}^{m-1} c_i s_i$, we can implement $m-2$ relinearizations to convert the ciphertext into a linear equation in terms of a unique key; e.g., if we want to express all polynomials as a function of key s_1, we use $m-2$ key homomorphisms $h_i^{(j)}$, where each homomorphism would have the key s_j "encrypted" in terms of s_1 for $j = 2, 3, \ldots, m-1$. By recursively applying these relinearizations, i.e. $h_i^{(2)}$ to the ciphertext composed by (c_0, c_1, c_2), then $h_i^{(3)}$ to the concatenation of the previous result and c_3, and so on, we arrive at the expression that encompasses all concatenated relinearizations in two equations:

$$\begin{aligned} c_0^{relin} =& c_0 + \sum_{i=0}^{\lceil \log_t q \rceil - 1} c_{2,i} b_i^{(2)} + \ldots + \sum_{i=0}^{\lceil \log_t q \rceil - 1} c_{m,i} b_i^{(m)}, \\ c_1^{relin} =& c_1 + \sum_{i=0}^{\lceil \log_t q \rceil - 1} c_{2,i} a_i^{(2)} + \ldots + \sum_{i=0}^{\lceil \log_t q \rceil - 1} c_{m,i} a_i^{(m)}. \end{aligned}$$

Now, considering the vectors

$$\begin{aligned} \boldsymbol{c}_{base-t} =& (\boldsymbol{c}_{2,base-t}^T, \boldsymbol{c}_{3,base-t}^T, \ldots, \boldsymbol{c}_{m,base-t}^T)^T, \\ \boldsymbol{a} =& \left(\left(\boldsymbol{a}^{(2)}\right)^T, \left(\boldsymbol{a}^{(3)}\right)^T, \ldots, \left(\boldsymbol{a}^{(m)}\right)^T \right)^T, \\ \boldsymbol{b} =& \left(\left(\boldsymbol{b}^{(2)}\right)^T, \left(\boldsymbol{b}^{(3)}\right)^T, \ldots, \left(\boldsymbol{b}^{(m)}\right)^T \right)^T, \end{aligned}$$

where

$$\begin{aligned} \boldsymbol{c}_{i,base-t} &= (c_{i,0}, c_{i,1}, \ldots, c_{i,\lceil \log_t q \rceil - 1})^T, \\ \boldsymbol{a}^{(i)} &= (a_0^{(i)}, a_1^{(i)}, \ldots, a_{\lceil \log_t q \rceil - 1}^{(i)})^T, \\ \boldsymbol{b}^{(i)} &= (b_0^{(i)}, b_1^{(i)}, \ldots, b_{\lceil \log_t q \rceil - 1}^{(i)})^T, \end{aligned}$$

we get the simplified vector expression of the relinearization

$$c_0^{relin} = c_0 + \boldsymbol{c}_{base-t} \cdot \boldsymbol{b}, \quad c_1^{relin} = c_1 + \boldsymbol{c}_{base-t} \cdot \boldsymbol{a},$$

where $\boldsymbol{a} \cdot \boldsymbol{b}$ is the scalar product between the vectors of polynomials $\boldsymbol{a}$ and $\boldsymbol{b}$. With this expression, if the key owner generates the appropriate relinearization matrices, we can flexibly convert encryptions between different keys (key switching) and even extract or linearly combine different individual components of an encrypted polynomial signal.

Increase of error after relinearization

The noise added to the ciphertext after the execution of one relinearization step is approximately equivalent to the noise added over the same ciphertext by as many homomomorphic additions of fresh ciphertexts as polynomials compose the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. Therefore, if both vectors have $m\lceil \log_t q \rceil$ polynomial elements, it is equivalent to $m\lceil \log_t q \rceil$ homomorphic additions. Hence, even when some of the proposed methods in this chapter resort to a relinearization step, they still allow for $\mathcal{O}(D)$ homomorphic products between ciphertexts, being D the number of products allowed by the choice of q (see Eq. (4.1)).

4.5.2. Proposed Optimizations

This section introduces several strategies based on relinearization, aimed at optimizing the realization of the encrypted NTT proposed in Section 4.4.2, avoiding the interaction with the secret-key owner for the pre- and post-processing steps; for this purpose, we take advantage of the specific structure of the transform matrices. We first present a polyphase-decomposition-based approach which reduces the key size, and then enhance it by preserving the key security. The polyphase decomposition is a common tool used in signal processing [130], which has also been applied in a cryptographic setting as a means to achieve different tradeoffs in the General Learning with Errors (GLWE) problem [50].

Our target is to calculate the NTT of an already encrypted version of $x[l]$, which has not been pre-processed (Section 4.4.2). We first note that our encrypted NTT algorithm allows to perform one of the processings under encryption, by expressing it as a convolution. For the NTT we have

$$\mathrm{NTT}(x[l]) = N^{-1}\left(\left(x[k] \circledast \alpha^{\frac{-k^2}{2}}\right)\alpha^{\frac{k^2}{2}}\right) \circledast \mathrm{NTT}\left(\alpha^{\frac{l^2}{2}}\right).$$

Analogously, for the INTT we have

$$\mathrm{INTT}(x[l]) = N^{-1}\left(\left(x[k] \circledast \alpha^{\frac{k^2}{2}}\right)\alpha^{\frac{-k^2}{2}}\right) \circledast \mathrm{INTT}\left(\alpha^{\frac{-l^2}{2}}\right).$$

With this structure, we only have to implement *one of the component-wise products* with the (known) pre- or post-processing vector under encryption to get an unattended implementation of the encrypted NTT; we use the polyphase decomposition of the inputs and exploit the use of the relinearization to homomorphically calculate the pre- or post-processing.

Encrypted NTT with polyphase decompositions

In order to calculate a component-wise product of the signals $x[l]$ and $h[l]$, we can decompose them in as many polyphase components as their length. A relinearization can be used to extract each of these components into separate encryptions, and subsequently, element-wise multiplication can be straightforwardly performed. Then, an inverse relinearization step would enable regrouping the signals in a sole encryption which can be decrypted using the initial secret key. This approach suffers from an excessive computational cost to perform N relinearizations; moreover, the corresponding relinearizations to one polyphase component reduce the problem to a lattice with $n = 1$, which would imply no security.

We can exploit the use of the polyphase decomposition in a smarter way, balancing the size of the used relinearization matrices and the reduction in the security: We divide the signal in a number of components equal to a constant M (the previous solution corresponds to $M = N$). For our choice of cryptosystem parameters (see Section 4.2), we need M be a power of two dividing N. Hence, we can express the element-wise multiplication in terms of M smaller and independent homomorphic element-wise multiplications, where each signal has a size N/M. Therefore, we are able to divide the sought encrypted operations as a set of simpler and easier element-wise operations (with the corresponding reduction in the considered lattice). This process could be recursively performed, at the cost of eventually reducing again the ciphertexts to a lattice with $n = 1$, which is unacceptable in terms of security.

We still need a method to homomorphically perform the element-wise operations without resorting to a reduction in the dimension of the lattice, which is presented in detail in Section 4.5.2. By combining this method with the partial polyphase decomposition we can produce several possible solutions for an encrypted element-wise multiplication. Depending on the chosen M, we can trade-off efficiency (lower size for the relinearization matrices) for security (lower n).

Our proposed process for an encrypted component-wise product $x[l]\alpha^{\frac{l^2}{2}}$ is the following:

- Decimate $x[l + m]$ with $m \in [0, M)$ by a factor M.

- For each polyphase component, apply a relinearization encrypting the corresponding component in a polynomial ring isomorphic to a lattice of dimension $\frac{N}{M}$ (if $M > 1$).

- Perform the element-wise multiplication of each component by the corresponding component of the signal $\alpha^{\frac{l^2}{2}}$ by resorting to the method proposed in Section 4.5.2 (multiplication between a ciphertext and a cleartext). If $M = N$, the multiplication can be directly performed.

- Finally, a reverse relinearization process is applied to each component so that they are regrouped into a new ciphertext under the same key (if $M > 1$).

This method produces an element-wise product by the pre- or post-processing vector without the intervention of the secret key owner, enabling a fully non-interactive computation of the encrypted NTT. Moreover, the used relinearization decreases the dimension by a factor M (each polyphase component has a length of $\frac{N}{M}$ samples), thereby achieving a net improvement in both computational cost and security with respect to a direct application of the polyphase decomposition ($M = N$).

Encrypted NTT without lattice dimension reduction

Decreasing the size of the used key as done by the previous method implies a reduction in security. However, it is possible to perform the sought element-wise multiplication between a ciphertext and a known cleartext vector with no such reduction. Hence, we enable additional secure and efficient operations like modulation or demodulation with an unencrypted carrier, or the implementation of the encrypted NTT without the intervention of the key owner, which is our purpose.

First, we show how to perform the element-wise multiplication of a ciphertext and cleartext without a reduction in the lattice dimension. Finally, we explain how to use smaller relinearizations and achieve a net improvement in the efficiency of the operations when the cleartext is periodic.

Element-wise product between ciphertext and cleartext: We consider the ciphertext $\boldsymbol{c} = (c_0, c_1)$, whose decryption circuit would be $c_0 + c_1 s$, and the polynomial represented as a column vector $\boldsymbol{g} = (g_0, g_1, \ldots, g_{n-1})^T$. If we denote by the polynomial c_0' the result of the element-wise multiplication between c_0 and g, the decryption circuit in matrix form will be $\boldsymbol{c}_0' + \text{diag}(\boldsymbol{g})\boldsymbol{C_1 s}$, where $\text{diag}(\boldsymbol{g})$ is a diagonal matrix composed of the elements of the vector $\boldsymbol{g}$, and $\boldsymbol{C_1}$ is the skew circulant matrix [122] of the polynomial c_1.

Now, we apply the relinearization algorithm and express the decryption circuit in terms of polynomial products or, in matrix form, products between vectors and skew circulant matrices. Considering the key homomorphism $h_i = (a_i, b_i = -(sa_i + te_i) + t^i s)$, with $i = 0, \ldots, \lceil \log_t q \rceil - 1$, we have

$$\boldsymbol{c}_0^{relin} = \boldsymbol{c}_0' + \sum_{i=0}^{\lceil \log_t q \rceil - 1} \tilde{\boldsymbol{C}}_i \boldsymbol{b}_i, \quad \boldsymbol{c}_1^{relin} = \sum_{i=0}^{\lceil \log_t q \rceil - 1} \tilde{\boldsymbol{C}}_i \boldsymbol{a}_i,$$

where $\tilde{\boldsymbol{C}}_i$, $i = 0, \ldots, \lceil \log_t q \rceil - 1$, is the base-$t$ decomposition of the matrix product $\text{diag}(\boldsymbol{g})\boldsymbol{C}_1$. For the decryption circuit $c_0^{relin} + c_1^{relin} s$ to be correct, $\boldsymbol{S}\tilde{\boldsymbol{C}}_i$ has to be equal to $\tilde{\boldsymbol{C}}_i \boldsymbol{S}$ for all i, with $\boldsymbol{S}$ being the negacyclic or skew circulant matrix corresponding to the polynomial s. The previous equality is true when all the g_i are equal (multiplication by a polynomial of maximum degree 0), but in our general case all the g_i are different, and equality is not achieved. Therefore, the ciphertext must be modified to perform the sought relinearization. $\text{diag}(\boldsymbol{g})\boldsymbol{C}_1\boldsymbol{s}$ can be expressed equivalently in polynomial form as $\sum_{j=0}^{n-1} c^{(j)}(z)s_j$, being $c^{(j)}(z)$ the polynomial whose coefficients are the j-th column of the matrix product $\text{diag}(\boldsymbol{g})\boldsymbol{C}_1$. Finally, with these requirements the new decryption circuit has the form $c_0'(z) + \sum_{j=0}^{n-1} c^{(j)}(z)s_j$.

Now, considering n key homomorphisms $h_i^{(j)}$ with $i = 0, \ldots, \lceil \log_t q \rceil - 1$ and $j = 0, \ldots, n-1$, in which $h_i^{(j)}$ has the coefficient s_j "encrypted" under the key s, we can perform a unique relinearization by concatenating all the $h_i^{(j)}$ and the corresponding polynomials $c^{(j)}(z)$ as discussed in Section 4.5.1. As we targeted, the proposed relinearization does not convey a reduction in the size n of the lattice. Regarding the computational cost of the approach, the key owner needs to generate the vectors $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}_{base-t}$ of size $n\lceil \log_t q \rceil$, which are composed of polynomials of n coefficients. The relinearization comprises one polynomial addition and two scalar products $\boldsymbol{c}_{base-t} \cdot \boldsymbol{b}$ and $\boldsymbol{c}_{base-t} \cdot \boldsymbol{a}$, i.e, $2n\lceil \log_t q \rceil$ polynomial products and $2n\lceil \log_t q \rceil - 1$ polynomial additions.

Element-wise multiplication between ciphertext and periodic cleartext: When the cleartext used in the element-wise multiplication is periodic, the length of the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ required for the relinearization process can be reduced, therefore decreasing the number of addends in the decryption circuit. If g is a periodic signal with m samples per period, and m divides n, we can use the following decryption circuit:

$$c_0' + \sum_{i=0}^{m-1} \underbrace{z^i \left(\sum_{j=0}^{m-1} g_{i+j \bmod m} z^j c^{(j)} \left(z^m\right) \right) s^{(i)} \left(z^m\right)}_{c'_{i+1}},$$

where $s^{(i)}(z) = \sum_{d=0}^{\frac{n}{m}-1} s[md+i]z^d$ and $c^{(j)}(z) = \sum_{d=0}^{\frac{n}{m}-1} c_1[md+j]z^d$ are the i-th and j-th components of the polyphase decomposition in m components of s and c_1, respectively, and $s^{(i)}(z^m)$, $i = 0, \ldots, m-1$, are the corresponding secret keys.

The obtained ciphertext consists of $m+1$ polynomials; applying the relinearization as in the previous section, we reduce it to only two components. Hence, when g is periodic with a period of m samples, it is possible to reduce the size of the vectors $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}_{base-t}$ to $m\lceil \log_t q \rceil$ components each. Regarding the complexity, the proposed relinearization requires $2m\lceil \log_t q \rceil$ polynomial products and $2m\lceil \log_t q \rceil - 1$ polynomial additions.

It is worth noting that the pre and post-processing vectors needed to implement the encrypted NTT and INTT present some additional structure and periodicities which could be exploited in order to increase the efficiency of the computation of the encrypted NTT. Additionally, the solutions presented in this section could also be useful for other typical signal processing applications involving periodic signals, like modulations and demodulations (see 4.6.1).

4.5.3. Element-wise multiplication of two encrypted messages

The previous sections describe a fully non-interactive encrypted NTT with efficient relinearization operations which enable component-wise products between an encrypted vector and a known clear-text vector. We can now leverage the encrypted NTT to perform component-wise products between two fully encrypted vectors of length N without reducing the security of the scheme. Fixing the parameter n and working with $\lceil \frac{N}{n} \rceil$ pairs of ciphertexts, the computational cost in terms of elemental products $\bmod q$ would be $\mathcal{O}(2^{\lceil \log_2 N \rceil} n \log_2 n) \approx \mathcal{O}(Nn \log_2 n)$, step-wise linear in terms of N.

For the sake of comparison, we define cost_1 as the computational cost of the techniques presented in Section 4.5.2 for computing of the encrypted NTT, cost_2 for the polyphase-decomposition-based method which with M components, and cost_3 for the straightforward decomposition in N components (see Section 4.5.2). Hence, we obtain the ratios $\frac{\text{cost}_2}{\text{cost}_1} \approx \left(\frac{M}{n} + \frac{1}{M}\right)\left(1 - \frac{\log_2 M}{\log_2 n}\right) + \frac{1}{n}$ and $\frac{\text{cost}_3}{\text{cost}_1} \approx \frac{1}{\log_2 n} + \frac{1}{n}$. The computational cost for the element-wise multiplication between two encrypted messages is approximately bounded by three times the computational cost for an encrypted and a clear-text message. This is due to the need of homomorphically computing two NTT and one INTT, which amounts to three executions of the algorithms (or only some parts of the algorithms) previously presented in 4.5.2. Table 4.3 summarizes the computational cost in terms of elemental products modulo q, the total size of the relinearization matrices in terms of coefficients modulo q and the minimum required dimension for the lattices in the three methods.

By increasing the parameter M we achieve a net improvement in both the size of the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, and the efficiency of the encrypted element-wise multiplication, at the cost of a reduction in

Table 4.3: Comparison of the proposed encrypted element-wise multiplication methods.

Computational Cost
$\text{cost}_1 \approx \mathcal{O}\left(\lceil \frac{N}{n} \rceil n^2 \log_2 n\right)$
$\text{cost}_2 \approx \mathcal{O}\left(\lceil \frac{N}{n} \rceil \left(\left(Mn + \frac{n^2}{M}\right) \log_2\left(\frac{n}{M}\right) + n \log_2 n\right)\right)$
$\text{cost}_3 \approx \mathcal{O}\left(\lceil \frac{N}{n} \rceil \left(n^2 + n \log_2 n\right)\right)$

Total size of the relin. matrices
$\text{bits}_1 = 2(n^2 + n)\lceil \log_t q \rceil \lceil \log_2 q \rceil$
$\text{bits}_2 = \left(4n + \frac{2n^2}{M^2} + \frac{2n}{M}\right) \lceil \log_t q \rceil \lceil \log_2 q \rceil$
$\text{bits}_3 = (4n + 2)\lceil \log_t q \rceil \lceil \log_2 q \rceil$

Minimum lattice dimension
$n_{min_1} = n$
$n_{min_2} = \frac{n}{M}$
$n_{min_3} = 1$

the underlying dimension of the lattice. This trade-off between the security and the implementation runtimes is quantified in the next section.

Performance evaluation of the element-wise multiplication

The proposed constructions enable component-wise processing in RLWE cryptosystems, which are suited and very efficient for polynomial processing. Therefore, we compare our proposed methods with the use of a cryptosystem (Paillier) which is apparently more amenable to element-wise multiplication than RLWE-based cryptosystems. We use Lauter's cryptosystem to implement our proposed methods for element-wise products (Sections 4.5.2, 4.5.2 and 4.5.3).

In general, the computational cost for performing N element-wise multiplications with Paillier (with one of the messages in clear) is N modular exponentiations of Paillier ciphertexts. With Lauter, we would need only N element-wise multiplications (see Section 4.5.2), but the computational cost for the relinearization is relatively high. In order to have a fair comparison, we fix a value of n (polynomial degree) independent of N (message size) for Lauter, as a function of the needed level of security. Then, the computational cost for the element-wise multiplication of two pairs of N encrypted integers would be approximately $\mathcal{O}\left(\lceil \frac{N}{n} \rceil n^2 \log_2 n\right) = \mathcal{O}\left(2^{\lceil \log_2 N \rceil} n \log_2 n\right) \approx \mathcal{O}\left(N n \log_2 n\right)$ for the techniques from Section 4.5.2 or $\mathcal{O}(2^{\lceil \log_2 N \rceil}\left(M + \frac{n}{M}\right) \log_2\left(\frac{n}{M}\right) + \log_2 n) \approx \mathcal{O}\left(\left(NM + \frac{Nn}{M}\right) \log_2\left(\frac{n}{M}\right) + N \log_2 n\right)$ when resorting to polyphase decompositions (Section 4.5.2), using for both a radix-2 algorithm (as we have already described in the previous section 4.5.3).

Figure 4.3 compares the different runtimes for (a) the element-wise method in Section 4.5.2, (b) the polyphase-based method from Section 4.5.3, (c) the partial polyphase method from Section 4.5.3 with $M = 8$, (d) Paillier-based element-wise multiplication between an encrypted message and a message in the clear, and (e) the combination of Paillier-encrypted messages and a secure interactive multiplication protocol (SMP) [127] for computing the product of two encrypted messages. In all the experiments, we have chosen practical parameters for the Paillier cryptosystem (2048-bit, 3072-bit and 7680-bit moduli) and we vary the n used for the lattice cryptosystems. We are considering $N = 131072$ (the runtimes increase linearly with N). Additionally, Figure 4.3 also compares the bit-size of the relinearization matrices used for the first and second methods from Section 4.5.3 in terms of n.

As mentioned in previous sections, increasing n produces a smaller δ, and therefore a higher security (see Section 4.2.2). Hence, depending on the required security for the applications, we can choose an adequate value for n.

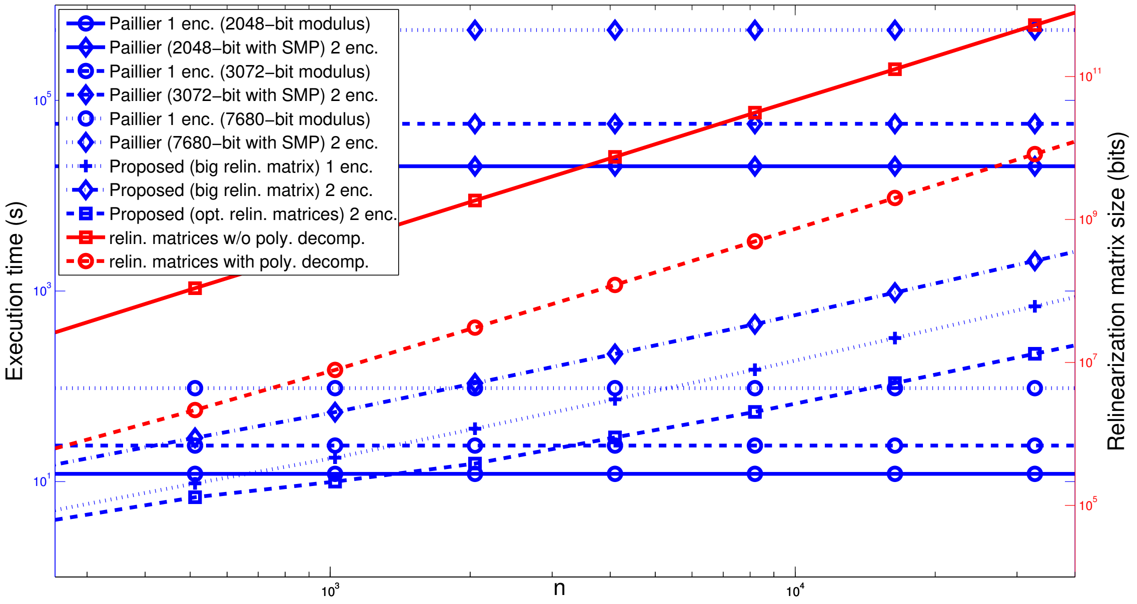


Figure 4.3: Comparison of element-wise runtimes for different schemes.

We can see that using practical values for both RLWE cryptosystems (e.g., $n = 2048$) and Paillier, the proposed optimized methods for element-wise multiplications of two encrypted vectors outperform the other approaches in terms of efficiency. In fact, Paillier can only achieve better performance when one of the vectors is unencrypted, but even in such case, RLWE-based cryptosystems are still much more efficient for polynomial operations and, when combined with our methods, they provide greater flexibility and a full toolset of unattended efficient encrypted operations along with these element-wise operations, which Paillier could not provide.

4.5.4. CRT for cleartext batching

The second optimization we propose deals with cleartext batching and enhancing the homomorphic capacity when SIMD (Single-Instruction-Multiple-Data) operations are implemented. For this purpose, we resort to the Chinese Remainder Theorem (CRT). The CRT has been used in numerous different applications, ranging from the conversion of a one-dimensional convolution into a convolution with smaller multidimensional signals, to the development of error correcting codes, secret-sharing and many more [131].

We first revisit the CRT with a notation slightly adapted to our particular scheme. We begin with the rings $R_{t_i^{k_i}}[z] = \mathbb{Z}_{t_i^{k_i}}[z]/f(z)$ and polynomials $a_i \leftarrow R_{t_i^{k_i}}[z]$, with $i = 1, \ldots, m$. If $a_i \equiv a_j \bmod (\gcd(t_i^{k_i}, t_j^{k_j}))$ holds for $1 \leq i, j \leq m$, then there exists a polynomial $a \in R_t[z]$ with $R_t[z] = \mathbb{Z}_t[z]/f(z)$ and $t = t_1^{k_1} t_2^{k_2} \ldots t_m^{k_m}$ which verifies:

$$a \equiv a_i \bmod (t_i^{k_i}), \text{ for } i = 1, \ldots, m. \tag{4.6}$$

For the existence of a, we can impose a less demanding requirement: it suffices that the t_i be pairwise coprime, i.e., $\gcd(t_i, t_j) = 1$ with $i \neq j$. In order to find the polynomial $a \in R_t[z]$ that satisfies the above congruences (4.6), we write

$$a = \sum_{i=1}^{m} T_i a_i d_i,$$

where $T_i = t/t_i^{k_i}$ and d_i fulfill $d_i T_i \equiv 1 \bmod (t_i^{k_i})$.

Therefore, the existing isomorphism between $R_t[z]$ and $R_{t_1^{k_1}}[z] \oplus R_{t_2^{k_2}}[z] \oplus \cdots \oplus R_{t_m^{k_m}}[z]$ enables several cleartext operations through a single encrypted homomorphic operation. The possibility of exploiting this isomorphism to parallelize cleartext operations has been previously suggested by several authors [50, 98]. Smart and Vercauteren [98] propose and exemplify SIMD operations using FHE (Fully Homomorphic Encryption) cryptosystems, for performing AES encryption homomorphically and for searching in an encrypted database. Brakerski *et al.* [50] propose batching the bootstrapping operation for improving the efficiency of the cryptosystem.

Our contribution comprises choosing appropriate values for t, such that the CRT can be applied to parallelize any of the encrypted operations introduced in the previous sections.

Throughput optimizations for signal processing applications

In general, prior works dealing with batching operations (see [132] for a comparison) are mainly focused in maximizing the throughput of operations, but generally overlook the type and usefulness of the parallelized encrypted operations, which might be severely affected by the decomposition of the cryptosystem ring in unequal prime ideals. Contrarily, we want to present the use of the NTT as a tool for batching operations which can be more suitable for typical signal processing applications. Then, following the steps presented in Section 4.4, we briefly explain how to achieve the maximum number of parallel operations between either integers or discrete signals, keeping the meaning and usefulness of the batched operations.

We assume that the used modular function is $f(z) = 1 + z^n$ and we also assume, without loss of generality, that all the t_i from Eq. (4.6) are different prime numbers. Then, for the previously introduced ring R_t, an addition or multiplication between two ciphertexts actually conveys the element-wise addition or multiplication between the vectors whose i-th element belongs to $R_{t_i^{k_i}}[z]$. In this case, if we want to perform the maximum number of parallel multiplications between integers, we have to restrict the input signals to zero-degree polynomials, therefore wasting much of the plaintext space.

By resorting to the proposed pre-processing techniques and the use of the NTT, we can fully utilize all the available plaintext space. That is, combining the CRT and the NTT we can perform multiplications among integers belonging to the finite field $\mathbb{Z}_{t_i^{k_i}}$. In any case, we can maximize the number of encoded integers if we choose the right values for the different t_i and k_i, and we show how in the following discussion.

There exists an isomorphism between the finite field $\mathbb{Z}_{t_i^{k_i}}$ and $\mathbb{Z}_{t_i}[x]/g(x)$ where $g(x)$ is an irreducible function over $\mathbb{Z}_{t_i}$. Then, in order to have $g(x) = 1 + x^{k_i}$ ($1 - x^{k_i}$ is not irreducible), we must use a cyclotomic polynomial $\Phi_{2k_i}(x) = 1 + x^{k_i}$ (with k_i a power of two) and, finally, we can assert that $\Phi_{2k_i}(x)$ is irreducible over $\mathbb{Z}_{t_i}$ when it satisfies

$$t_i^{\phi(2k_i)} = t_i^{k_i} \equiv 1 \bmod 2k_i, \tag{4.7}$$

where $\phi(\cdot)$ is Euler's totient function and k_i is the smallest integer satisfying the above condition, that is, t_i is a generator of the multiplicative group $\mathbb{Z}_{2k_i}^*$. Further details about working with finite fields can be found in [133].

As an example, if we consider $k_i = 2$ for all i, we know that if $t_i = 3 \bmod 4$, then $\mathbb{Z}_{t_i^2}$ is equivalent to $\mathbb{Z}_{t_i}[x]/(1 + x^2)$. Therefore, reusing again the proposed pre-processing and the

NTT over each $\mathbb{Z}_{t_i}[x]/(1+x^{k_i})$ we can implement the batched multiplication of more integers. Unfortunately [134], the only cyclotomic polynomial that allows to encode more integers while satisfying our requirements is $\Phi_4(x)$; e.g., if we have two messages $x_1, x_2 \in R_t[z]$ and each one encodes $2mn$ integers ($2n$ integers over $\mathbb{Z}_{t_i}$ for $i = 1, \ldots, m$) thanks to the CRT, we can apply the proposed pre-processing for performing a cyclic convolution and, afterwards, use the NTT for performing the element-wise product as a cyclic convolution (first, for distinguishing the different rings $R_{t_i^2}$, and afterwards, for distinguishing the two different $\mathbb{Z}_{t_i}$ which we can find on the finite field $\mathbb{Z}_{t_i^2}$). In this way, with only one multiplication between two ciphertexts we can perfom the element-wise multiplication between $2mn$ pairs of integers. Conversely, the element-wise addition is easily obtained without the need of the NTT or pre-processing. In case we want to perform linear filterings instead of multiplications of integers, we can use the techniques presented in [20, 21] for packing a different signal in each of the involved integers.

4.6. Applications: Encrypted Signal Processing Toolset

This section exemplifies the use of the primitives and algorithms presented in previous sections by proposing a set of practical tools and applications, which comprise matrix operations, Cyclic Redundancy Check (CRC) codes, changes in sampling rate and linear transforms; we also show how they can be seamlessly adopted within any RLWE-based cryptosystem [79, 78], by taking advantage of its polynomial structure. For simplicity, we assume that all signals have only one independent variable (univariate) but the methods can be easily extended to the multivariate case [4].

4.6.1. Typical Operations in Signal Processing

We present efficient methods to implement elementary signal processing operations in the encrypted domain when using lattice-based cryptosystems. Besides the different types of convolutions tackled above, shifts and scaling of the independent variable of the signals are also very common operations in signal processing. In general, shift operations do not involve any change in the cryptosystem parameters, but this is not true for operations that cause a change in the sampling rate of the encrypted signal. In that case, it is necessary to "reset" the secret key to the new sampling rate of the signal. Below, we address shift operations and changes in sampling rate together with modulation and demodulation operations which are enabled by using different types of relinearizations.

Shift

A shift $x[l-l_0]$ represents the signal $x[l]$ delayed by l_0 samples. This operation can be implemented on encrypted signals, represented as z-transform polynomials, by simply multiplying them by the monomial z^{l_0}. Therefore, the cost of the operation is the product of a single polynomial. Also, if the polynomial z^{l_0} is available in the clear, the cost is much lower, since it only involves a product by z^{l_0} in the clear with modular function $f(z)$. In case the shift makes the signal wrap-around, the same effects explained for the α-generalized convolution would apply, and pre- and post-procesing can be used to preserve the desired sign for the wrapped components.

Changes in the sampling rate

The changes in the sampling rate of encrypted signals can imply a change in the entropy and dimension (therefore, in security) of the used key. Considering again the use of modular functions of the form $f(z) = 1 + z^n$ with n power of 2, changes in the sampling rate which are powers of 2 can be implemented in the encrypted domain following the procedure we explain in the next two paragraphs.

Upsampling: For the upsampling, we only need to perform a change of variable in the polynomial ring. For an upsampling $x[l/G]$ with $G > 1$ in the ring $R_q[z]$, we just replace $z' = z^G$, ending in $R_q[z^G]$. Hence, for upsampling we apply $x(z^G)$ and consider $f(z^G)$ and the secret key $s(z^G)$. Since the variables of our polynomial rings can only involve natural degrees, G has to be a natural number. In our case, using $f(z) = 1 + z^n$ with n power of 2, G must also be a power of 2. After increasing the number of samples, the encrypted signal can then be low-pass filtered through a homomorphic convolution, obtaining the encryption of the interpolated signal. Regarding security, the change of variable implies an increase in the lattice dimension; however, the entropy of the key remains unchanged. From the point of view of the key owner, the key is the same, simply considering different degrees for the coefficients.

Downsampling: Considering the ring $R_q[z]$, in order to perform a downsampling, we apply a change of variable $z^{1/G}$ with $1/G < 1$. If the corresponding coefficients of the polynomials from the ring $R_q[z^{1/G}]$ with no integer degree are discarded (plain decimation), we end up with the ciphertext $\left(c_0(z^{1/G}), c_1(z^{1/G})\right)$ and secret key $s(z^{1/G})$. As in the case of upsampling, it is considered that G is a power of 2. Back in the variable z, decrypting $\boldsymbol{c} = (c_0, c_1)$ implies calculating $c_0 + c_1 s$, where s is the secret key. Hence, a downsampling of the encrypted message involves performing a decimation of both c_0 and the result of the multiplication of c_1 with s. After an upsampling with a factor G, we can directly perform the corresponding downsampling by G without any impact on the number of ring elements that form the ciphertext. In contrast, for downsampling without relying on a previous upsampling, we need to use the polyphase decomposition of the decryption circuit, with the particularity that we are working in a ring with negacyclic convolutions instead of the typical cyclic convolutions. Therefore, the downsampling of the ciphertext $\boldsymbol{c} = (c_0, c_1)$ by a factor G is equivalent to the first component of the polyphase decomposition in G components of $\boldsymbol{c}$. If decryption computes $c_0 + c_1 s$, the decryption of the decimated ciphertext would compute

$$c_0'(z) + c^{(0)}(z)\, s^{(0)}(z) + z \sum_{i=1}^{G-1} c^{(i)}(z)\, s^{(G-i)}(z) \bmod 1 + z^{\frac{n}{G}},$$

where $c_0'(z)$ is the downsampling of $c_0(z)$ and both $c^{(i)}(z)$ and $s^{(i)}(z)$ are the i-th polyphase components of c_1 and s, respectively. Now, we can reduce the ciphertext to a function of a single key. For this purpose, we can use $G - 1$ concatenated relinearizations with the corresponding key homomorphisms $h_i^{(j)}$ with $i = 0, \ldots, \lceil \log_t q \rceil - 1$ and $j = 1, \ldots, G - 1$, which can be performed in just one step (see Section 4.5.1). Regarding security, the entropy and size of the key are reduced in proportion to the downsampling factor.

Reflection

We denote the reflection of the signal a by a^{ref}. As the ciphertext $\boldsymbol{c}^{ref} = (c_0^{ref}, c_1^{ref})$ contains the reflection of the encrypted signal under the key s^{ref}, to implement the reflection of the ciphertext $\boldsymbol{c} = (c_0, c_1)$ we have to use a key change of s^{ref} instead of s. Finally, the key change circuit can be represented as a relinearization of the decryption circuit $c_0^{ref} + 0s + c_1^{ref} s^{ref}$, therefore considering $c_1 = 0$ in the decryption circuit introduced in Section 4.5.1.

Modulation and demodulation

Typical modulations involve the multiplication by a periodic carrier. This element-wise multiplication can be addressed by the general method presented in Section 4.5.2. However, the element-wise multiplication between a ciphertext and a known periodic carrier can be efficiently implemented through the method proposed in Section 4.5.2 which takes advantage of the periodic structure of the carrier signal, and achieves better efficiency by reducing the size of the needed relinearization matrices.

4.6.2. Encrypted Matrix Multiplication

We rely on Yagle's [135] method to write a matrix multiplication as a single polynomial product to implement matrix multiplications on RLWE-based encrypted signals.

For calculating a matrix multiplication of size $N \times N$ as $\boldsymbol{C} = \boldsymbol{AB}$, Yagle proposes to compute $c(z) = a(z)b(z)$, where we will denote the elements of matrices or polynomial coefficients with two or one subscripts respectively, such that

$$c(z) = \sum_{i=0}^{N^3+N^2-N-1} c_i z^i, \; a(z) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} a_{i+jN} z^{i+jN},$$
$$b(z) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} b_{N-1-i+jN} z^{N(N-1-i+jN)},$$

with $a_{i,j} = a_{i+jN}$, $b_{i,j} = b_{N-1-i+jN}$, $c_{i,j} = c_{N^2-N+i+jN^2}$ and integers i and j such that $0 \leq i,j < N$.

This imposes a lower bound $n \geq N^3 + N^2 - N - 1$ on the needed maximum degree n of the modular function $f(z) = 1 + z^n$, in order to store the result of a matrix multiplication of size $N \times N$ within the cryptosystem. Possible applications of this encrypted matrix multiplication algorithms for signal processing comprise, among others, linear codes and computing N encrypted linear transforms of N different signals of size N through a single polynomial product.

Operations with the scaling variable d: The necessary lattice size for encrypting Yagle's matrix product in an encrypted algorithm can be very large, so the method can become computationally too expensive even for matrices of moderate sizes. Yagle's approach uses a scaling variable (denoted s in [135]) to lower the number of coefficients of the proposed polynomials, reducing also the dimension of the considered lattice. This comes at the cost of imposing certain conditions on the magnitude of the elements of the result.

Assuming a plaintext from a ring $R_t[z] = Z_t[z]/f(z)$ with $f(z) = 1 + z^n$ and n power of 2, we can reduce the degree of the polynomial of the modular function and still get the desired result. For this, we need to do a change of variable $z = dw$ and a change of the modular function by $f(w) = 1 + w^l$, $1 \leq l < n$, being l the desired new degree (due to the cryptosystem requirements, l must be a power of two). Therefore, if all the elements of the resulting matrix $\boldsymbol{C}$ are less than d^l, they can be recovered from the result with a base-d decomposition of all the coefficients. For more details on using the scaling variable d we refer the reader to [135].

This approach can be adapted to the operations described in this chapter in order to achieve a size reduction of the polynomials. Nevertheless, its use makes it difficult to perform subsequent encrypted homomorphic operations.

4.6.3. Encrypted CRC (Cyclic Redundancy Check)

Given a generator polynomial $g(z)$ of maximum degree $n - k$, and a message $m(z)$ of maximum degree $k - 1$, CCCs (Cyclic Convolution Codes) encode the signal $m(z)$ as the polynomial product $m(z)g(z)$. After encrypting signals $g(z)$ and $m(z)$, their polynomial product can be homomorphically performed if the result fits in the length allowed by the ring $R_t[z]$, i.e., it does not *wrap around*.

Consequently, these types of codes seem to perfectly adapt to the structure of RLWE cryptosystems, enabling the application of new encrypted operations, such as encrypted CRC checks of the encrypted message. Some specific types of cyclic convolution codes, such as BCH or Reed-Solomon, require the use of the NTT and INTT for encoding the messages. The calculation of the encrypted NTT and INTT has been addressed in Section 4.4. Below, we include an example of the use of cyclic codes for the reduction of the cipher expansion in RLWE or m-RLWE based cryptosystems.

Cyclic codes for better cipher expansion: In [114], the authors show that in practical situations, if the least significant bits of the encrypted coefficients are discarded, the decryption error rate does not increase significantly. This line of thought can also be found in other recent works [85] showing how discarding the least significant bits does not increase too much the ciphertext noise in a scale-invariant cryptosystem.

Therefore, a possible improvement would be the homomorphic application of a cyclic code to the encrypted values, in such a way that we could discard more bits and protect against the quantization errors without decoding first. Of course, there exists a trade-off between the increase of the polynomial size (due to the introduced redundancy in the message) and the number of discarded bits in the polynomial coefficients.

Fortunately, when the messages have a size k smaller than n we can apply the cyclic encoding without increasing the polynomial size of the ciphertexts, therefore achieving a reduction in the cipher expansion. Regarding the increase of computational cost at decryption, the key owner only has to apply the corresponding cyclic decoding after the decryption of the encoded message.

4.6.4. Generic Linear Transforms for encrypted vectors

We can implement any kind of linear transform by relying on the method presented in Section 4.5.2 to perform the element-wise multiplication between a ciphertext and a cleartext. In matrix form, the element-wise multiplication can be seen as a multiplication between the ciphertext components and a diagonal matrix whose diagonal is composed of the cleartext coefficients. We briefly introduce a generalization of the previously considered diagonal matrix to a general square matrix. With this approach, we can perform any linear transform of an encrypted signal, provided that the matrix considered for the linear transform is available in cleartext.

For completeness, we show below the process for calculating the product between the public matrix and the encrypted vector. Additionally, we also exemplify its use for a typical signal processing application: interleaving.

Implementation of the Linear Transform for an encrypted vector

We follow an analogous process to Section 4.5.2. First, we consider the ciphertext $\boldsymbol{c} = (c_0, c_1)$, whose decryption circuit is $c_0 + c_1 s$, and the linear transform represented by the square matrix $\boldsymbol{L}$ of size $n \times n$. If we denote the polynomial c_0' as the result of the multiplication $\boldsymbol{L}\boldsymbol{c_0}$, the decryption circuit in matrix form will be $\boldsymbol{c}_0' + \boldsymbol{L}\boldsymbol{C_1}\boldsymbol{s}$.

Additionally, the product $\boldsymbol{L}\boldsymbol{C}_1\boldsymbol{s}$ can be expressed equivalently in polynomial form as $\sum_{j=0}^{n-1} c^{(j)}(z)s_j$, being $c^{(j)}(z)$ the polynomial whose coefficients are the j-th column of the matrix product $\boldsymbol{L}\boldsymbol{C}_1$. Consequently, the new decryption circuit has the form $c_0'(z) + \sum_{j=0}^{n-1} c^{(j)}(z)s_j$.

Finally, considering n key homomorphisms $h_i^{(j)}$ with $i = 0, \ldots, \lceil \log_t q \rceil - 1$ and $j = 0, \ldots, n-1$, in which $h_i^{(j)}$ has the coefficient s_j "encrypted" under the key s, we can do a unique relinearization by concatenating all the $h_i^{(j)}$ and the corresponding polynomials $c^{(j)}(z)$.

Regarding the computational cost, the only difference with respect to the element-wise multiplication between a ciphertext and a cleartext shown in Section 4.5.2 is the following: instead of multiplying the coefficients of the ciphertext with a diagonal matrix, here we use a general square matrix; that is, if we have N/n ciphertexts (where $N \geq n$), then we perform: (a) $\mathcal{O}(N)$ products between coefficients for the element-wise case, and (b) $\mathcal{O}(Nn)$ products with the naïve matrix multiplication algorithm between a square matrix and a vector (we are computing N/n matrix multiplications among vectors and matrices with size n and $n \times n$ respectively).

In any case, the part of the algorithm that determines the total computation time is the execution of N/n relinearization steps with $2n\lceil \log_t q \rceil$ polynomial products each; i.e., approximately $\mathcal{O}(Nn(\log_2 n)^2)$ products between coefficients. As the relinearization process is the same for both cases, the computational cost for the general (known) linear transform of an encrypted vector is approximately the same as the suggested method for element-wise product between a ciphertext and a cleartext shown in Section 4.5.2.

Interleaving

The interleaving process can be represented as a matrix product with a concatenation of permutation matrices, which conform one "interleaving matrix"; therefore, we can implement the interleaving of the encrypted signal as a linear transform. As an example, this can be useful when performing an encrypted matrix multiplication (see Section 4.6.2), because by changing some rows of the interleaving matrices for other rows which contain all zeros, we can relocate the coefficients of the result and zero those coefficients which are not needed.

Hence, the computational cost for the relinearization processes involved in our interleaving is the same as for the case of linear transforms discussed in Section 4.6.4. Additionally, the remaining cost of encrypted interleaving is smaller than the cost of an encrypted linear transform, as the interleaving prior to relinearization is much faster than a matrix product.

4.7. Conclusions

This chapter presents a novel way of using Number Theoretic Transforms paired with lattice-based cryptosystems to take advantage of the polynomial structure of typical signal processing applications and enable a wide range of unattended secure signal processing primitives for noninteractive privacy-preserving processing of sensitive signals.

On the one hand, we show a parameterization of RLWE-based cryptosystems to fully optimize the use of underlying NTTs to speed up polynomial products; additionally we show how to perform cyclic, negacyclic and generalized convolutions in the encrypted domain, encrypted component-wise products, and efficient encrypted NTT, either by applying pre- and post-processing operations, or in a fully unattended manner through the use of relinearization primitives.

We illustrate the use of our proposed approaches in several composable signal processing blocks, ranging from generalized convolutions to error correcting codes and matrix-based operations. Therefore, this chapter opens up a wide variety of novel secure signal processing primitives over fully encrypted signals in a non-interactive way, working either with polynomial or component-wise operations, and efficiently batching SIMD processes.

Chapter 5

Revisiting Multivariate Lattices

This chapter is adapted with permission from ACM: Alberto Pedrouzo-Ulloa, Juan Ramón Troncoso-Pastoriza, and Fernando Pérez-González. Revisiting Multivariate Lattices for Encrypted Signal Processing. 7th ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec), July 2019.

5.1. Introduction

We have seen in Chapter 4 that cryptosystems based on RLWE (Ring Learning with Errors) present a clear advantage when dealing with signals, as its underlying polynomial structure allows for very efficient filtering and convolution operations [29]; hence, most of the applications involving correlations and filtering can benefit from recent RLWE-based schemes, which keep constantly evolving [30, 31, 32].

Nevertheless, applications working with images or higher dimensional signals are much more demanding. One example is multimedia forensics, which deals with high volumes of signals with an inherent multidimensional structure [136]. For this scenario, several solutions have been proposed to adapt the structure of RLWE cryptosystems for efficiently dealing with this multi-dimensionality [4, 46, 5]. These works propose a generalization of RLWE called multivariate RLWE (m-RLWE), and their results show improved efficiency/space tradeoffs. Actually, the authors of [5] show the flexibility of these structures and their advantages in several conventional signal processing operations, such as block-processing and multidimensional convolutions/transforms. These schemes have been used in even more complex applications inside the field of multimedia forensics, namely camera attribution in the encrypted domain [137].

In Chapters 2 and 3 (see also Appendix A) we introduced the multivariate RLWE problem and discussed in depth its pecularariaties for cryptographic primitives. With this in mind, our objective in this chapter is a little bit different and we focus more on its consequences for secure signal processing.

Recently, Bootland *et al.* [44] introduced an attack that reduces the security of schemes based on m-RLWE (see Chapter 2). This attack has important consequences on the validity of the results presented in [4, 46, 5] and a careful analysis is needed to correctly reevaluate the security of these schemes.

This chapter carries out this analysis and recalculates the correct security estimates for m-

RLWE applications in light of this new attack. Additionally, we introduce a novel pre-/post-coding paradigm for RLWE cryptosystems, which we denote "packed"-RLWE, that preserves all the security properties of works based on m-RLWE, but now basing their security directly on RLWE, which is not affected by Bootland's attack.

We also provide an extensive comparison between a conventional use of an RLWE cryptosystem (baseline RLWE), an m-RLWE cryptosystem and an RLWE cryptosystem equipped with our proposed pre-/post-coding. For the sake of clarity and space, we focus on applications based on multidimensional filtering, but all the solutions previously presented for m-RLWE can be adapted to our new packed-RLWE. Finally, we analyze the optimal combination of the three approaches, baseline RLWE, m-RLWE and packed-RLWE, depending on the efficiency/space trade-offs required by the target application.

Main Contributions: This chapter features the following contributions:

- We revisit the security analysis of previous m-RLWE cryptosystems in light of the recent attack introduced in [44].
- We survey the best existing algorithms to homomorphically evaluate multidimensional convolutions with an RLWE cryptosystem (denoted as baseline RLWE), noting that some of the best solutions in one dimension (as the use of FFT algorithms) result in a much worse performance in a multidimensional setting, due to the increase of the circuit depth.
- We propose a new pre-/post-coding paradigm over RLWE cryptosystems (we denote it packed-RLWE), that directly "emulates" multidimensional convolutions over the encrypted signals, and comprises very efficient element-wise products and FFT operations on the plaintext ring.
- We show how previous solutions based on m-RLWE can be adapted to our packed-RLWE version, hence getting all the advantages of these structures while still preserving the high security of a lattice with dimension equal to the full length of the involved signals.
- We provide an extensive comparison between a baseline RLWE, an m-RLWE based solution (with non-"coprime" modular functions, which is a "worst-case" for security) and our packed-RLWE proposal. Our results show that m-RLWE and packed-RLWE still outperform those results of baseline RLWE.
- We briefly discuss how the three approaches can be combined to fit the specific requirements of a real application, optimizing the space/efficiency trade-offs. Additionally, we describe several practical applications which can greatly benefit from the use of these tools.

Structure: The rest of the chapter is organized as follows: in Section 5.2, we briefly revisit the used RLWE-based cryptosystems, their security and the use of NTT/INTT transforms. Section 5.3 includes a description of the different approaches for both baseline and multivariate RLWE solutions. We introduce the main contribution of this chapter in Section 5.4, comprising our new pre-/post-coding blocks for packed-RLWE. Section 5.5 includes an extensive comparison between the different proposed approaches in terms of security, efficiency and cipher expansion. Finally, we discuss a set of example encrypted applications that greatly benefit from our solutions in Section 5.6.

5.2. Preliminaries

In this section, we revisit the RLWE problem and RLWE-based cryptosystems, together with their multivariate RLWE counterparts. We also summarize the recent attack [44] to multivariate RLWE and detail its effects on the choice of security parameters. Finally, we briefly revisit the use of Number Theoretic Transforms (NTTs).

5.2.1. Multivariate RLWE problem

Firstly, for completeness, we include an informal definition of the multivariate RLWE problem as is stated in Chapter 2. We focus on the most widespread case where the modular functions are cyclotomic polynomials of power-of-two order, i.e., $f_i(z_i) = \Phi_{2n_i}(z_i) = z_i^{n_i} + 1$ with n_i a power-of-two. Additionally, this general definition allows us to also cover the RLWE problem as a particular case when the number of dimensions is one (i.e. $l = 1$).

Definition (multivariate RLWE problem [4, 5, 22], adapted Definition 1 from Chapter 2)**.** *Given a polynomial ring $R_q[z_1, \dots, z_l] = \mathbb{Z}_q[z_1, \dots, z_l]/(z_1^{n_1} + 1, \dots, z_l^{n_l} + 1)$ and an error distribution $\chi[z_1, \dots, z_l] \in R_q[z_1, \dots, z_l]$ that generates small-norm random polynomials in $R_q[z_1, \dots, z_l]$, m-RLWE relies upon the computational indistinguishability between samples $(a_i, b_i = a_i s + e_i)$ and (a_i, u_i), where a_i, $u_i \leftarrow R_q[z_1, \dots, z_l]$ are chosen uniformly at random, whereas $s, e_i \leftarrow \chi[z_1, \dots, z_l]$ are drawn from the error distribution.*

A remark on RLWE: For cyclotomic modular functions $\{\Phi_{m_1}(z_1), \dots, \Phi_{m_l}(z_l)\}$ where $\gcd(m_1, \dots, m_l) = 1$, m-RLWE is isomorphic to RLWE with modular function $\Phi_{\prod_i m_i}(z)$ [45]. Unfortunately, this is not the case for the version stated in Definition 1, and the security of m-RLWE is highly dependent on the form of the different modular functions (see Section 5.2.3).

5.2.2. An $(m-)$RLWE based Cryptosystem

We instantiate univariate and multivariate versions of the FV cryptosystem [86] as examples for our proposed schemes (see Sections 5.3 and 5.4) and our performance comparisons (see Section 5.5), but the results are generalizable to other cryptosystems such as BGV and CKKS [50, 31]. Due to space constraints, we do not include here a description of all the cryptosystem primitives (we refer to [86] for a detailed description). Instead, we summarize the cryptosystems' properties relevant to our analysis.

The plaintext elements belong to the ring $R_t[z_1, \dots, z_l]$, and ciphertexts are composed of (at least) two polynomial elements belonging to $R_q[z_1, \dots, z_l]$. The security of the scheme relies on the indistinguishability assumption of the m-RLWE problem (see Definition 1), which reduces to RLWE when $l = 1$.

Cipher expansion

In FV, we can use the following noise bound (Theorem 1 in [86]) when evaluating an arithmetic circuit of multiplicative depth L

$$4\delta_R^L(\delta_R + 1.25)^{L+1} \cdot t^{L-1} < \left\lfloor \frac{q}{B} \right\rfloor, \tag{5.1}$$

where $\delta_R = \prod_i n_i$ is the ring expansion ratio, q is the modulo of the ciphertext ring R_q, t is the modulo of the plaintext ring R_t, and $||\chi|| < B$, that is, χ is a B-bounded distribution of variance σ^2.

RLWE in secure signal processing

The use of an RLWE-based cryptosystem brings about two main advantages in secure signal processing: (a) its security is highly dependent on the length of the involved polynomials, which directly impacts the cipher expansion if the input data cannot be fully packed; practical signals are usually long sequences, such that they can be encrypted in only one encryption; this helps in increasing the security of the underlying RLWE-based cryptosystem without significantly increasing its expansion; (b) homomorphic properties of the cryptosystem translate into addition and multiplication of plaintext polynomials, which represent signal addition and convolution (filtering), the basic blocks required in any signal processing application.

5.2.3. Security of multivariate RLWE

The original formulation of multivariate RLWE [4, 5] assumes that the m-RLWE problem (Definition 1) in dimension $n = \prod_{i=1}^{l} n_i$ is as hard as the RLWE problem in dimension n. However, in [44] Bootland *et al.* introduce an attack on m-RLWE; this attack exploits the fact that some of the modular functions enable repeated "low-norm" roots in the multivariate ring. As a result, when common roots exist, this attack is able to factor the m-RLWE samples into RLWE samples of smaller dimension, hence reducing the security of these m-RLWE samples to that of solving a set of independent RLWE samples of the maximum individual degree $\max_i\{n_i\}$.

This attack is specially relevant for m-RLWE samples $(a_i, b_i = a_i s + te_i)$ chosen as in Definition 1, where all the modular functions introduce common roots.[1] For a detailed explanation of the Bootland *et al.*'s attack we refer the reader to Section 2.2 from Chapter 2.

5.2.4. Number Theoretic Transforms

Consider a ring $\mathbb{Z}_p$ where $p = \prod_{i=1}^{k} p_i^{l_i}$, an NTT of size N can be defined if the introduced properties in Chapter 4.2.3 hold. Instead of dealing with the Equations (4.4) for the forward and inverse transform, we work with an alternative representation which is more convenient to showcase the results of this chapter.

We can see NTT/INTT transforms as matrix multiplications

$$\tilde{\boldsymbol{x}} = \boldsymbol{W}\boldsymbol{x}, \quad \text{and} \quad \boldsymbol{x} = \boldsymbol{W}^{-1}\tilde{\boldsymbol{x}}, \tag{5.2}$$

where

$$\tilde{\boldsymbol{x}} = (\tilde{x}[0], \ldots, \tilde{x}[N-1])^T, \quad \boldsymbol{x} = (x[0], \ldots, x[N-1])^T,$$

[1] As an example, consider the functions $f(x) = x^n + 1$ and $g(y) = y^{2n} + 1$. It is easy to verify that the square of the roots of $g(y)$ are also roots of $f(x)$.

and

$$\boldsymbol{W} = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \alpha & \alpha^2 & \dots & \alpha^{N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{N-1} & \alpha^{2(N-1)} & \dots & \alpha^{(N-1)(N-1)} \end{pmatrix}.$$

5.3. An analysis of Previous Schemes

In this section, we survey the available algorithms to homomorphically evaluate a multidimensional convolution operation with both an RLWE and an m-RLWE based cryptosystem. We give approximations for computational cost, cipher expansion and security with relative expressions between the different algorithms (we refer the reader to Appendices 5.A and 5.B for more details on the derivation of these expressions).

5.3.1. Our Setup

We set the following parameters to enable a fair comparison:

- The used FV cryptosystem (see Section 5.2) is based on either RLWE or m-RLWE (see Definition 5.2.1) with power-of-two modular functions ($f_i(z_i) = z_i^{n_i} + 1$). The noise distribution of a fresh ciphertext has variance σ^2 and its noise coefficients are upper-bounded by B.
- We use RLWE with $n = n_l$ and m-RLWE with $n = \prod_{i=1}^{l} n_i$. Hence, the ring expansion ratio $\delta_R = n_l$ for RLWE, and $\delta_R = \prod_{i=1}^{l} n_i$ for m-RLWE.
- The computational cost is measured in terms of polynomial coefficient multiplications, without explicitly taking the cost of each coefficient multiplication into account. In Section 5.5 we introduce this additional factor to have a fair comparison between the analyzed schemes.
- The elemental ring operations are polynomial multiplications and additions in $R[z]$ (RLWE) or $R[z_1, \dots, z_l]$ (m-RLWE). By means of FFT algorithms, the computational cost of polynomial products is $n_l \log\, n_l$ for RLWE and $n_1 \dots n_l \log{(n_1 \dots n_l)}$ for m-RLWE.
- Bit security is measured relative to $\texttt{BitSecurity}(\sigma^2, n)$, which represents the bit security of an RLWE instance with error distribution of variance σ^2 and polynomial degree n. In Section 5.5 we give concrete bit security estimations for the different solutions.

We work with l-dimensional signals and filters whose length per dimension is, respectively, N_i and $F_i \leq N_i$ for $i = 1, \dots, l$, and consider two main scenarios: (a) a **linear (non-cyclic) convolution** where we reserve enough space inside the ciphertexts to store the result (i.e. $n_i = N_i + F_i - 1$), and (b) a **cyclic convolution**, enabled by means of the pre-/post-processing from [29] on top of the homomorphic negacyclic ring operation (i.e. $n_i = N_i$).

In the next sections, we introduce two RLWE-based approaches for performing a multivariate convolution, and the natural m-RLWE approach, and compare them in terms of computation cost, ciphertext noise and relative bit security, before presenting our proposed scheme.

5.3.2. Multidimensional convolutions in baseline RLWE

Convolution, correlation and filtering can all be expressed as a linear convolution between two l-dimensional signals $y[u_1, \ldots, u_l] = x[u_1, \ldots, u_l] * h[u_1, \ldots, u_l]$ (where $u_i \in \mathbb{N}$). With a polynomial representation, this reduces to a polynomial product $y(z_1, \ldots, z_l) = x(z_1, \ldots, z_l) \cdot h(z_1, \ldots, z_l)$.

As discussed in [4], implementing a multidimensional convolution with an RLWE-based cryptosystem can be achieved by internally encoding only one of the dimensions (u_l or z_l in this case), and externally evaluating the whole convolution on the remaining $l-1$ dimensions. This means that the two l-dimensional signals are represented with $(l-1)$-dimensional elements $x'[u_1, \ldots, u_{l-1}]$ and $h'[u_1, \ldots, u_{l-1}]$, where each element belongs to $R_q[z_l]$. The resulting operation is $y'[u_1, \ldots, u_{l-1}] = x'[u_1, \ldots, u_{l-1}] * h'[u_1, \ldots, u_{l-1}]$ with $x = 0$ (resp. $h = 0$) for elements outside of the interval $0 \leq u_i < N_i$ (resp. $0 \leq u_i < F_i$).

This external convolution operation can be realized by leveraging the circular convolution property of DFT transforms and using FFT algorithms. However, the implementation of the FFT introduces a multiplicative depth equal to $\log\left(\prod_{i=1}^{l-1} N_i\right)$, where N_i is the number of samples in dimension i; the complexity of RLWE-based cryptosystems strongly depends on the number of levels, due to the increase in the size of the ciphertext coefficients (q depends exponentially on the number of levels in Eq. (5.1)); hence, as noted in [129], it turns out that more basic approaches with a multiplicative depth of one perform better, even if they feature a higher (quadratic) computational cost in terms of coefficient multiplications. Hence, we rule out the "fast" algorithms and we detail the two main direct approaches in the following, to enable a fair comparison of computational complexity with fixed q.

NTT matrix Convolution

Let P be the total number of elements in the convolution signal ($P = \prod_{i=1}^{l-1} N_i$ for the cyclic convolution scenario, and $P = \prod_{i=1}^{l-1}(N_i + F_i - 1)$ for the linear one). The NTT can be implemented by using its matrix formulation, Eq. (5.2). This results in a total of P^2 multiplications between ciphertexts and clear-text scalar values (and roughly P^2 ciphertext additions). As these operations can be much faster than the P ciphertext multiplications corresponding to the Hadamard product in the NTT domain, we do not take into account the runtime corresponding to the NTT/INTTs matrix computations, but we do consider its effects in the noise of the ciphertext. Table 5.1(a) shows the computational cost, ciphertext's noise and bit security for this method, particularized for the two scenarios presented in Section 5.3.1 (linear and cyclic convolution).

Direct Convolution

The second approach is to directly realize the convolution equation in polynomial form, which has a computational cost of roughly the product of the lengths of the involved signals in the convolution. While this solution has a higher computational cost than the previous one, it can be seen that the NTT matrix product incurs in a higher noise than the polynomial version; furthermore, for the case where the length of the filter signal is much smaller than that of the signal (i.e. $\prod_i F_i \ll \prod_i N_i$), the direct convolution approach can be much more efficient than the NTT matrix convolution, due to a smaller cipher expansion (caused by a much more reduced noise increase). Table 5.1(b) summarizes the computational cost, ciphertext's noise and bit security for this method.

5.3.3. Multivariate RLWE

RLWE-based cryptosystems lack support for seamlessly encrypting a multidimensional signal in one ciphertext, whereas m-RLWE enables a more compact representation achieving one encryption per signal. By considering the polynomial representation of the signals $y(z_1, \dots, z_l) = x(z_1, \dots, z_l) \cdot h(z_1, \dots, z_l)$, m-RLWE can homomorphically evaluate the multidimensional convolution operation with only one ciphertext multiplication [4, 5], which can be realized leveraging efficient FFT algorithms with no penalty on the required ciphertext size, which is a clear advantage with respect to baseline RLWE. Nevertheless, due to the recent attack presented in [44], the security of m-RLWE cannot be based on the product dimension of the multidimensional polynomial ($n = \prod_{i=1}^{l} n_i$), but instead on the highest degree of the univariate rings (that is, $\max_i \{n_i\}$). Table 5.1(c) summarizes the computational cost, ciphertext's noise and bit security for the m-RLWE multidimensional convolution.

5.3.4. Comparison between RLWE and m-RLWE

In light of the results shown in Tables 5.1(a), 5.1(b), and 5.1(c), it is clear that m-RLWE is much more efficient than RLWE when implementing multidimensional convolutions, but the increase in ciphertext size is not paired with an analogous increase in the bit security of m-RLWE, in general. Actually, depending on the chosen modular functions, m-RLWE can be isomorphic to RLWE when the modular functions $\{\Phi_{m_1}(z_1), \dots, \Phi_{m_l}(z_l)\}$ satisfy $\gcd(m_1, \dots, m_l) = 1$ (see Section 5.2.1). Hence, it is possible to preserve some of the advantages of m-RLWE while still keeping the security reduction to a lattice of dimension equal to the product of the degrees of each univariate ring, by resorting to "uneven" non-power-of-two (coprime) univariate modular functions (we discuss this possibility in detail in Chapter 2).

As an example, Cheon and Kim [56] initially proposed using m-RLWE with modulo power-of-two cyclotomic polynomials, and updated their application to use “coprime” cyclotomic polynomials [57] after the publication of Bootland *et al.* attack [44].

In the next section we focus on the “worst-case” scenario, where the security of m-RLWE reduces to only the highest of the univariate degrees. Even after this reduction on security, we show that m-RLWE can outperform the use of a simpler RLWE instance, due to two key advantages: (1) working with power-of-two univariate modular functions $1 + z^n$ which enable faster algorithms for product and reduction computations, and (2) more flexibility on the choice of the encrypted “lengths”.

However, we want to remark that the results presented here can be analogously applied to more general RLWE instances with other cyclotomic polynomial modular functions.

5.4. Proposed Scheme

This section describes the main contribution of this chapter. We introduce a new pre-/post-coding block which, when applied before/after RLWE-based encryption/decryption, transforms the polynomial multiplication (1D negacyclic convolution) of RLWE samples with power-of-two modular function ($l = 1$ in Definition 1) into an l-dimensional cyclic convolution operation. This enables the efficient realization of multivariate convolutions under the RLWE problem without a loss in security; i.e., the bit security is that of the whole lattice dimension $n = \prod_{i=1}^{l} n_i$. Therefore, we can encrypt the whole multidimensional signal in just one RLWE encryption with a security based on RLWE and not affected by Bootland's attack, while preserving all the properties of m-RLWE claimed in [4, 5].

We start by defining multivariate NTT/INTTs, as one of the main building blocks of our proposed scheme, and then we present our proposed framework for pre-/post-coding.

5.4.1. Multivariate Number Theoretic Transforms

Consider a length-N NTT transform over $\mathbb{Z}_p$, as defined in equations (5.2) by a matrix multiplication with $\boldsymbol{W}$ (and $\boldsymbol{W}^{-1}$ for the INTT).

If $\boldsymbol{x}$ represents a "flattened" vector[2] with the samples of an l-dimensional signal x, we can define an l-dimensional NTT/INTT as the Kronecker product of the NTT matrices for the l dimensions as follows:

$$\tilde{\boldsymbol{x}} = \underbrace{\left(\bigotimes_{i=1}^{l} \boldsymbol{W}^{(z_i)}\right)}_{\boldsymbol{V}^{(l)}} \boldsymbol{x}, \quad \boldsymbol{x} = \underbrace{\left(\bigotimes_{i=1}^{l} \left(\boldsymbol{W}^{(z_i)}\right)^{-1}\right)}_{\left(\boldsymbol{V}^{(l)}\right)^{-1}} \tilde{\boldsymbol{x}}, \tag{5.3}$$

where each $\boldsymbol{W}^{(z_i)}$ (resp. $\left(\boldsymbol{W}^{(z_i)}\right)^{-1}$) is the NTT (resp. INTT) of length N_i for the i-th dimension (z_i) of x. Equivalently in signal representation, the i-th NTT matrix is applied to $x[u_1, \ldots, u_i, \ldots, u_l]$ as a vector of N_i $(l-1)$-dimensional samples indexed by $u_i = 0, \ldots, N_i - 1$, for each $i = 1, \ldots, l$. Hence, the matrices $\boldsymbol{V}^{(l)}$ (resp. $\left(\boldsymbol{V}^{(l)}\right)^{-1}$) represent the l-dimensional NTT (resp. l-dimensional INTT). Additionally, the conditions in Section 5.2.4 must be satisfied, so for each matrix $\boldsymbol{W}^{(z_i)}$ there must exist an N_i-th root of unity in $\mathbb{Z}_p$.

The l-dimensional NTT/INTT satisfies a multivariate circular convolution property that we exploit in our proposed scheme

$$\boldsymbol{V}^{(l)}\boldsymbol{y} = (\boldsymbol{V}^{(l)}\boldsymbol{x}) \circ (\boldsymbol{V}^{(l)}\boldsymbol{h}), \tag{5.4}$$

where $\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{h}$ are the "flattened" vectors corresponding to the signals $y[u_1, \ldots, u_l]$, $x[u_1, \ldots, u_l]$, $h[u_1, \ldots, u_l]$, and $y[u_1, \ldots, u_l]$ is the l-dimensional circular convolution between $x[u_1, \ldots, u_l]$ and $h[u_1, \ldots, u_l]$.

Analogously to their univariate counterparts, multidimensional NTT/INTTs can be efficiently implemented with FFT algorithms.

5.4.2. "Packed"-RLWE and its underlying Multivariate Structure

Once we have introduced the formulation for multivariate NTTs/INTTs applied to flattened vectors, we can present the pre-/post-processing adapted from [29, 54] which allows to transform the negacyclic convolutions of the rings from Definition 1 into cyclic convolutions.

Consider two length-N signals $x[j]$ and $h[j]$, with polynomial representations $x(z), h(z)$

$$x(z) = \sum_{i=0}^{N-1} x[i]z^i \quad \text{and} \quad h(z) = \sum_{i=0}^{N-1} h[i]z^i.$$

[2] A "flattened" $\boldsymbol{x}$ vector is a reshape of the multdimensional signal x into a column vector.

We want to calculate their circular convolution $y(z) = x(z)h(z) \bmod 1 - z^N$, but the ring operation enabled as a homomorphic product is a polynomial product modulo $1 + z^N$ (negacyclic convolutions).

Assume that there exists a $2N$-th root of unity β in $\mathbb{Z}_p$ (that is, $\beta = (-1)^{\frac{1}{N}} \bmod p$), the pre-/post-processing [29, 54] consists of the following steps (we term it Murakami pre-/post-processing, see Chapter 4):

- The input signals are pre-processed with component-wise products

$$\begin{aligned} x'[j] &= x[j](1)^{\frac{-j}{N}}(-1)^{\frac{j}{N}}, && j = 0, \ldots, N-1, \\ h'[j] &= h[j](1)^{\frac{-j}{N}}(-1)^{\frac{j}{N}}, && j = 0, \ldots, N-1. \end{aligned}$$

- Then, $y'(z)$ can be calculated with a negacyclic convolution as $y'(z) = x'(z)h'(z) \bmod 1 + z^N$.

- The output signal is post-processed with component-wise products

$$y[j] = y'[j]1^{\frac{j}{N}}(-1)^{\frac{-j}{N}}.$$

Equipped with the Murakami pre-/post-processing, we can emulate the operation from a ring with a circular convolution property. The last step is to find a way of transforming the unidimensional circular convolution into a multidimensional one. To this aim, we combine both a unidimensional NTT/INTT (see Section 5.2.4) and a multidimensional NTT/INTT.

Let $\boldsymbol{y}$ be the flattened l-dimensional circular convolution of $\boldsymbol{x}$ and $\boldsymbol{h}$. By the convolution property of the NTTs, we have

$$\left(\boldsymbol{W}^{-1}\boldsymbol{x}'\right) \circledast \left(\boldsymbol{W}^{-1}\boldsymbol{h}'\right) = \boldsymbol{W}^{-1}\left(\boldsymbol{x}' \circ \boldsymbol{h}'\right),$$

where $\boldsymbol{x}' = \boldsymbol{V}^{(l)}\boldsymbol{x}$ and $\boldsymbol{h}' = \boldsymbol{V}^{(l)}\boldsymbol{h}$. If we make use of the convolution property of the l-dimensional NTT, Eq (5.4) with $N = \prod_{i=1}^{l} N_i$, we have

$$\begin{aligned} (\boldsymbol{W}^{-1}\boldsymbol{V}^{(l)}\boldsymbol{x}) \circledast (\boldsymbol{W}^{-1}\boldsymbol{V}^{(l)}\boldsymbol{h}) = && \boldsymbol{W}^{-1}\left((\boldsymbol{V}^{(l)}\boldsymbol{x}) \circ (\boldsymbol{V}^{(l)}\boldsymbol{h})\right) \\ = && \boldsymbol{W}^{-1}\boldsymbol{V}^{(l)}(\boldsymbol{y}). \end{aligned}$$

This represents a chain of matrix transformations that relates the unidimensional circular and l-dimensional circular convolutions.[3]

Hence, the resulting structure of our proposed pre-/post-coding, detailed in Figure 5.1 is as follows:

- A pre-coding is applied to the input signals

$$\boldsymbol{x}'' = \boldsymbol{\Upsilon}\boldsymbol{W}^{-1}\boldsymbol{V}^{(l)}\boldsymbol{x} \quad \text{and} \quad \boldsymbol{h}'' = \boldsymbol{\Upsilon}\boldsymbol{W}^{-1}\boldsymbol{V}^{(l)}\boldsymbol{h}.$$

- $y''(z)$ is calculated as $x''(z)h''(z) \bmod 1 + z^N$.

- A post-coding is applied to $y''(z)$

$$\boldsymbol{y} = (\boldsymbol{V}^{(l)})^{-1}\boldsymbol{W}\boldsymbol{\Upsilon}^{-1}\boldsymbol{y}''.$$

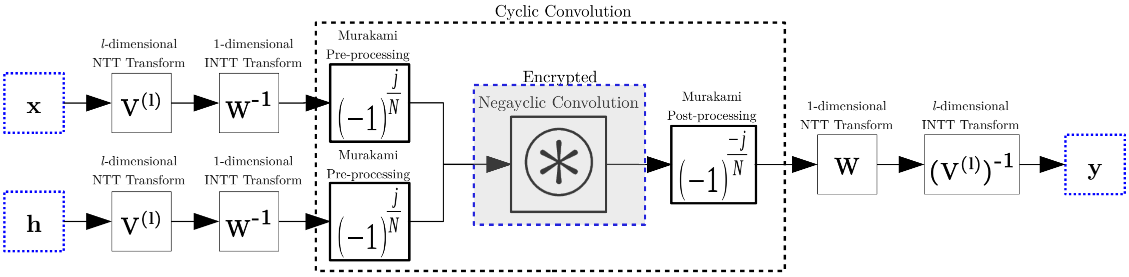


Figure 5.1: Block diagram of the proposed scheme for "packed"-RLWE.

The matrices $\boldsymbol{\Upsilon}$ and $\boldsymbol{\Upsilon}^{-1}$ are diagonal matrices containing the elements of the Murakami pre-/post-processing $(1)^{\frac{-j}{N}}(-1)^{\frac{j}{N}}$ and $(1)^{\frac{j}{N}}(-1)^{\frac{-j}{N}}$ for $j = 0, \ldots, N-1$.

Table 5.1(d) includes a summary with the computational cost, ciphertext's noise and bit security for the execution of a multidimensional convolution with the proposed method. This table includes the cost of the actual convolution without the pre-/post-coding, which would be executed at the client-side in a homomorphic processing scenario, and is evaluated as part of the encryption/decryption in Section 5.5. In any case, this pre-/post-coding only comprises element-wise multiplications and a chain of two FFT computations on the plaintext ring, so the computational cost of both encryption and decryption with the FV cryptosystem is higher than this processing chain.

5.5. Security and Performance Evaluation

This section includes a comparison of RLWE, m-RLWE and the proposed packed-RLWE in terms of security, computational cost and cipher expansion. We start by describing the procedure followed to analyze the security of the different schemes. Afterwards, we analyze and compare the expressions reported in Tables 5.1(a), 5.1(b), 5.1(c) and 5.1(d), and we highlight the tradeoffs for each scenario. Finally, we also include execution runtimes for the case of image and 3D-signal filtering.

5.5.1. Evaluation for Encrypted Processing of Multidimensional Signals

In [4, 5] we compared several encrypted multidimensional operations implemented with an RLWE or an m-RLWE based scheme. We concluded that the m-RLWE implementation enabled a much higher security with faster runtimes.

However, after the attack presented in [44], we know that the security estimations with m-RLWE are no longer valid. Nevertheless, the packed-RLWE solution we have introduced in this chapter can preserve all the claimed security and efficiency results in [4, 5].

In this chapter, instead of using a slack variable as in [4, 5], we follow a different approach, and we compare baseline RLWE, m-RLWE (see Section 5.3) and our packed-RLWE solution (see Section 5.4) by fixing the minimum level of security in terms of the lattice dimension for baseline

[3] While we focus on NTT transforms, similar results could be considered with chains of CRT matrices (see [45]). This would enable the encoding of different multidimensional signals in any instance of RLWE with a general cyclotomic modular function.

RLWE. This dimension is kept constant for each univariate ring of m-RLWE and packed-RLWE, even though the security obtained with the two latter will be higher, and hence, the comparison on efficiency represents a worst-case scenario for m-RLWE and an unfavorable case for our packed-RLWE. We show that even in this pessimistic scenario, both m-RLWE and packed-RLWE can outperform baseline RLWE both in terms of efficiency and security.

For simplicity, we make the following set of assumptions (we refer the reader to Appendices 5.A and 5.B for further details):

A1 We work with "hyper-cubic" l-dimensional signals with the same length N in each dimension (length-F in case of filters) and we assume n_i to be equal to the value required to store the result of the linear or cyclic convolution.

A2 We define $F = C \cdot N$ where C is a constant satisfying $0 < C \leq 1$, so that we can express the results in terms of N to compare the behavior of both linear and cyclic convolutions under the same formulation.

A3 For estimating the cost of each coefficient multiplication in $\mathbb{Z}_q$, we assume the use of Schönhage-Strassen algorithm with a cost of $\mathcal{O}(w(\log w)(\log\log w))$, with $w = \mathcal{O}(\log_2 q)$. For the asymptotic analysis, we simplify the cost to $\mathcal{O}(\log_2 q)$.

Comparison of Computational Cost and Cipher Expansion

A summary with the computational cost and ciphertexts' noise for each of the analyzed approaches, particularized for assumptions A1-A3, is included in Tables 5.2(a), 5.2(b) and 5.2(c). We refer the reader to Appendices 5.A and 5.B for the detailed derivation of the approximate costs and noise bounds. We first compare the asymptotic cost ratios for increasing N between the four approaches, and then move on to a more precise analysis of the effect of each parameter for a given N.

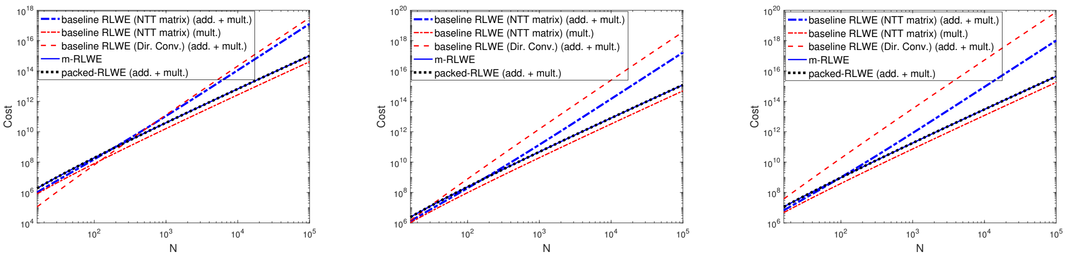
Asymptotic computational cost ratios: By neglecting the effect of some logarithmic factors in the computational cost, we can provide some approximate asymptotic comparisons between the different schemes, in order to highlight the most significant effects. In particular, we consider $N \gg F$, so we approximate $N + F - 1 \approx (1 + C)N$ and neglect the effect of $(1 + C)$ and its powers with respect to powers of N. This allows us to cover both linear and cyclic convolutions with the same computational cost expressions (we refer the reader to Appendices 5.A and 5.B for more details on the simplifications).

If we neglect the effect of additions and consider only products as the operation driving the complexity, we obtain the following ratios

$$\frac{\text{Cost}_{rd}}{\text{Cost}_{rn}} \approx N^l, \quad \frac{\text{Cost}_{mr}}{\text{Cost}_{rn}} \approx l, \quad \frac{\text{Cost}_{pr}}{\text{Cost}_{rn}} \approx l, \quad \frac{\text{Cost}_{pr}}{\text{Cost}_{mr}} \approx \frac{3}{2},$$

where the costs $\{\text{Cost}_{rn}, \text{Cost}_{rd}, \text{Cost}_{mr}, \text{Cost}_{pr}\}$ correspond, respectively, to $\{$Baseline RLWE (NTT matrix comp.), Baseline RLWE (Dir. Conv.), m-RLWE, and packed-RLWE$\}$.

We can see that Cost_{rn} is approximately l times lower than Cost_{mr} and Cost_{pr}, but it has also a lower bit security, which grows with l for packed RLWE.



(a) $C = 0.01$ (b) $C = 0.1$ (c) $C = 1$

Figure 5.2: Computational cost of encrypted image linear filtering for different relative filter sizes $C = \{0.01, 0.1, 1\}$.

If we factor in additions by assuming a cost of $\mathcal{O}(\log_2 q)$ for each coefficient addition (linear in the size of the coefficients), the asymptotic ratios become

$$\frac{\text{Cost}_{rd}}{\text{Cost}^*_{rn}} \approx \log_2 N, \quad \frac{\text{Cost}_{mr}}{\text{Cost}^*_{rn}} \approx \frac{l \log_2 N}{N^{l-1}}, \quad \frac{\text{Cost}_{pr}}{\text{Cost}^*_{rn}} \approx \frac{l \log_2 N}{N^{l-1}},$$

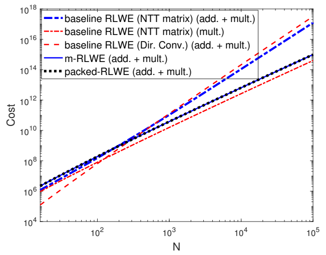
where Cost^*_{rn} represents the cost of the NTT/INTT matrix computation in baseline RLWE. Consequently, we see that m-RLWE and packed-RLWE are not only more secure, but also asymptotically more efficient than baseline RLWE for a wide set of scenarios.

Precise computational cost: While the previous asymptotic analysis is useful to extract the relative behavior of the schemes for very large N, it neglects the effects of some parameters. Now, we calculate the exact costs of the different methods by using the Schönhage-Strassen algorithm for coefficient multiplication, considering $\log_2 q$ for the cost of coefficient additions and without removing any non-significant factors.

We choose two filtering scenarios with 2- and 3-dimensional signals. In all figures we represent the cost (in terms of N) of a convolution between a "hyper-cubic" 2D or 3D signal with length N per dimension and a filter with length $F = \{0.01N, 0.1N, N\}$ per dimension.[4] Figure 5.2 (resp. Figure 5.3) represents the cost for a linear (resp. cyclic) convolution of 2D images, while Figure 5.4 (resp. Figure 5.5) represents the cost for a linear (resp. cyclic) convolution of 3D signals. All of them plot the relative cost of RLWE with NTT matrix and direct convolution, m-RLWE, and packed-RLWE, as a function of the per-bit elementary operation cost for growing signal size; the ciphertext size q is taken as the minimum value that enables the operation with no decryption errors for a constant noise power; therefore, security is also increased together with N (see Section 5.5.2). Hence, we are accounting for the raw growth in complexity produced by a change in the signal dimensions.

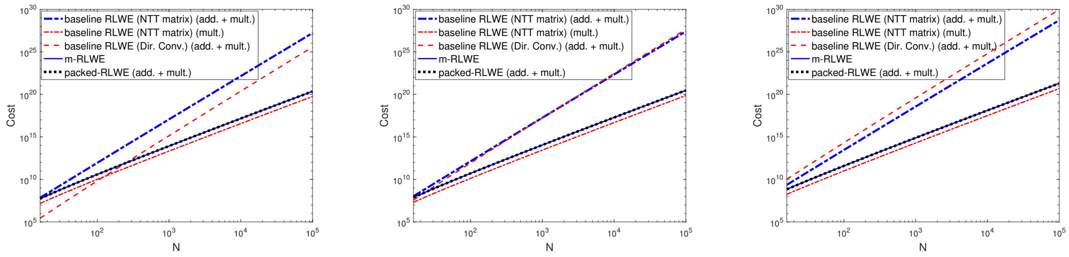
We can see that changes in the relative filter size C have a higher impact when the dimensionality of the signals increases, and in particular, the expansion in baseline RLWE with direct convolution is strongly influenced by small C values, which explains why it can be better when working with very small filters. In this case, if baseline RLWE gives enough security, it can be the best option, because both m-RLWE/packed-RLWE would require to further increase each of the n_i to store the results. In general, there is a minimum value of C for which packed-RLWE and m-RLWE start outperforming baseline RLWE, and this value decreases when increasing the

[4] The cost plotted in Figures 5.2, 5.3, 5.4 and 5.5 considers $n_i \approx N$ or $n_i \approx N + F - 1$, but it is worth noting that in practice each n_i will be rounded up to a power of two (see Definition 5.2.1), so performance will show a step-wise behavior for growing N instead of the smooth figures we show.



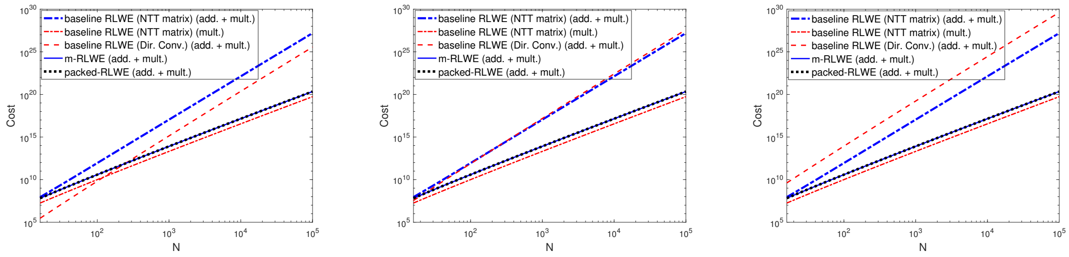
(a) $C = 0.01$ (b) $C = 0.1$ (c) $C = 1$

Figure 5.3: Computational cost of encrypted image cyclic filtering for different relative filter sizes $C = \{0.01, 0.1, 1\}$.



(a) $C = 0.01$ (b) $C = 0.1$ (c) $C = 1$

Figure 5.4: Computational cost of encrypted 3D-signal linear filtering for different relative filter sizes $C = \{0.01, 0.1, 1\}$.



(a) $C = 0.01$ (b) $C = 0.1$ (c) $C = 1$

Figure 5.5: Computational cost of encrypted 3D-signal cyclic filtering for different relative filter sizes $C = \{0.01, 0.1, 1\}$.

dimensionality, showing that packed-RLWE and m-RLWE perform better with high-dimensional signals and/or with filters of moderate or big size.

It is worth noting that none of the approaches is universally better than the others, and a combination of all of them may produce the best efficiency/security trade-offs. As an example, if the used filter has one particularly small dimension, it could be worth to encode this dimension as external to the encryption scheme. Conversely, if the security of the largest dimension is enough, the structure of m-RLWE could be preferable, as it can be more easily parallelizable than packed-RLWE and also avoids the pre-/post-coding stage at the client. Nevertheless, packed-RLWE is shown to outperform baseline RLWE and m-RLWE both in efficiency and security in a wide range of parameterizations.

5.5.2. Security evaluation

Tables 5.1(a), 5.1(b), 5.1(c) and 5.1(d) express the security of the schemes relative to $\texttt{BitSecurity}(\sigma^2, n)$ (see Sections 5.3 and 5.4). This function grows when increasing σ^2 or n (it is much more sensitive to n).

In order to give concrete values for $\texttt{BitSecurity}(\sigma^2, n)$, we make use of the LWE security estimator developed by Albrecht *et al.* [80, 81],[5] by calling the function $\texttt{estimate\_lwe}(n, \alpha, q,$ secret_ distribution $=$ "normal", reduction_cost_model $=$ BKZ.sieve), where $\sigma = \frac{\alpha q}{\sqrt{2\pi}}$. The results for the analyzed cases are shown in Tables 5.3 and 5.4, which are discussed in the next subsection in the context of the achieved security-efficiency tradeoffs.

5.5.3. Implementation and execution times

We have implemented the methods from Sections 5.3 and 5.4 making use of the RNS variant of the FV cryptosystem [95], in order to have concrete runtimes, instantiating the complexity measures introduced in the previous section. Execution runtimes were measured on an Intel Xeon E5-2667v3 at 3.2 GHz using one core (no parallelization).

We remark that we have not included results using the Paillier cryptosystem [14] in our performance comparison, but its runtimes and bit security can be easily extrapolated from [4, 5] and [112] respectively. In any case, Paillier cannot address the operations with encrypted signals and filters, and even with clear-text filters it is much slower than any RLWE-based scheme for this type of operations.

Tables 5.3 and 5.4 report runtimes for, respectively, encrypted $2D$-image linear filtering and encrypted $3D$-signal cyclic filtering for the same signal length per dimension. We have used $n_i = N_i + F_i - 1$ and $n_i = N_i$ (lattice dimensions equal to the signal dimensions) to show the maximum achievable efficiency for each scheme. In both scenarios, packed-RLWE provides similar runtimes to multivariate RLWE and faster runtimes than both baseline RLWE solutions, while also having a much higher bit security. Actually, with previous approaches we can only guarantee a very reduced security for the chosen polynomial degree, which is clearly below the current recommended bit security estimations (≥ 128 and ≥ 256 for quantum-resistance), and means that their computational complexity for the same acceptable security level as packed-RLWE would be substantially worse.

[5] Available online at `https://bitbucket.org/malb/lwe-estimator`.

5.6. A Discussion: Multidimensional Structures and their Applications

This chapter introduces a new pre-/post-coding block which enables significant efficiency advantages with respect to regular RLWE when processing multidimensional signals, bringing the benefits of m-RLWE while avoiding the recent attack by Bootland *et al.* [44] by basing the security only on that of RLWE.

While we focus on multidimensional filtering and correlation scenarios with encrypted signals, the proposed multivariate structures can be leveraged in a much wider set of applications. These range from block-processing (where we could apply homomorphic transforms between different block structures), better encrypted packing, multi-scale approaches such as pyramids and wavelet transforms, and even block-DCTs (see Appendix B). These solutions could also be combined with conventional signal processing approaches such as overlap-save and overlap-add algorithms (see [137]) and used to enhance encrypted matrix operations [57]. Hence, multivariate structures can produce notable efficiency improvements in many applications, when combining the solutions proposed in this chapter to optimize the security-efficiency trade-offs.

The use of packed-RLWE could also provide clear improvements in more complex applications such as forensic analysis and, in particular, camera attribution in the encrypted domain (see Chapters 7 and 8), where we can already find some works such as [137, 136, 138]. The last two make use of the BGN (Boneh-Goh-Nissim) cryptosystem to implement an homomorphic correlation operation between images. By the use of our proposed method, their runtimes could be greatly improved with no impact (or with an increase) on security.

5.7. Conclusions

This chapter proposes a novel framework for secure outsourced processing of encrypted multidimensional signals. As a fundamental block in our framework, we present a new pre-/post-coding block which enables multivariate structures directly on RLWE-based cryptosystems without compromising the security of the RLWE problem. We have also reevaluated the security of previous solutions based on multivariate RLWE by taking into account a recent attack which exploits the use of modular functions by introducing repeated roots in the ring. We have included an extensive comparison in terms of security and performance between the different approaches, showing the advantages of our scheme with respect to the previous solutions in terms of both faster runtimes and higher security; and also analyzing the possibility of adapting a combination of different methods to the needs of the specific scenario. Consequently, this chapter opens up a broad set of encrypted processing applications which deal with multidimensional signals and shows the viability of somewhat homomorphic encryption for the privacy-preserving processing of this type of signals.

5.A. Cipher Expansion Analysis

In order to calculate the bounds on q (see Section 5.2) depending on the chosen scheme, we rely on Lemma 3 from [86], which relates noise growth in the FV cryptosystem after each addition and multiplication. We include here a slightly modified version of the lemma:

Lemma 1 (Lemma 3 from [86]). *Let ct_i for $i = 1, 2$ be two ciphertexts with $[ct_i(s)]_q = \Delta \cdot m_i + v_i$ where $\Delta = \lfloor \frac{q}{t} \rfloor$, and $||v_i|| < E < \frac{\Delta}{2}$. Set $ct_{add} = FV.SH.Add(ct_1, ct_2)$ and $ct_{mult} = FV.SH.Mul(ct_1, ct_2, rlk)$ then*

$$[ct_{add}(s)]_q = \Delta \cdot [m_1 + m_2]_t + v_{add},$$
$$[ct_{mul}(s)]_q = \Delta \cdot [m_1 \cdot m_2]_t + v_{mul},$$

with $||v_{add}|| < 2E + t$ and $||v_{mul}|| < Et\delta_R(\delta_R + 1.25) + E_{Relin}$.

Taking into account Lemma 1 and the approximation for the noise in a fresh ciphertext, $E = 2\delta_R B$ (see [86]), the noise after L levels of multiplication is approximately $2B\delta_R^{2L+1}t^L$. This expression can be directly used to estimate the size of q (hence the cipher expansion) for both the multivariate and "packed" RLWE schemes.

However, when working with baseline RLWE for a multidimensional convolution, the effect of additions cannot be neglected, as their number is of the order of (or even higher than) δ_R, so we explicitly take them into account in the size of q. After one addition, $||v_{add}|| < 2E+t = 2\delta_R B+t$, where we neglect t because in our scheme $\delta_R B$ dominates the right hand term. Murakami pre-/post-processing (see [29, 54]) needs a t higher than the lattice dimension, so we choose a slightly higher t, that is $t \approx \max_i \{N_i + F_i - 1\}$ for all schemes but for packed-RLWE, for which $t \approx \prod(N_i + F_i - 1)$.

The effect of these additions into the size of the noise is equivalent to a multiplicative factor A_{add}, yielding a noise of $A_{add}^L \cdot 2B\delta_R^{2L+1}t^L$ after L multiplication levels.

The expressions for A_{add} for baseline RLWE are

- NTT matrix Convolution:

$$A_{add}^{(linear)} = \prod_{i=1}^{l-1} N_i(N_i + F_i - 1), \quad A_{add}^{(cyclic)} = \prod_{i=1}^{l-1} N_i^2.$$

- Direct Convolution:

$$A_{add}^{(linear)} = \prod_{i=1}^{l-1} F_i, \quad A_{add}^{(cyclic)} = \prod_{i=1}^{l-1} F_i.$$

If we now assume $N_i = N$ and $F_i = CN_i$ with $0 < C \leq 1$, we have the following noise size approximations after a linear convolution in each scheme

- baseline RLWE (NTT matrix Convolution):

$$\begin{aligned}\frac{\Delta}{2} \approx & 2B(1+C)^{L(l-1)}N^{2L(l-1)}\delta_R^{2L+1}t^L \\ \approx & 2B(1+C)^{Ll+L+1}N^{2Ll+1}t^L.\end{aligned}$$

- baseline RLWE (Direct Convolution):

$$\begin{aligned}\frac{\Delta}{2} \approx & 2BC^{L(l-1)}N^{L(l-1)}\delta_R^{2L+1}t^L \\ \approx & 2BC^{L(l-1)}(1+C)^{2L+1}N^{L(l+1)+1}t^L.\end{aligned}$$

- multivariate and "packed" RLWE:

$$\frac{\Delta}{2} \approx 2B\delta_R^{2L+1}t^L \approx 2B(1+C)^{2Ll+l}N^{2Ll+l}t^L.$$

We know that $0 < 1 + C \leq 2 \ll N$ and its exponent is not higher than the exponent of N, so in the following we will ignore powers of $(1+C)$. This allows us to use the same expression for both linear and cyclic convolutions (see Table 5.5) in the asymptotic cost ratio analysis in Section 5.5.1.

5.B. Computational Cost analysis

An integer multiplication in $\mathbb{Z}_q$ using a Schönhage-Strassen algorithm has a cost of $\mathcal{O}(\log_2 q \cdot (\log_2 \log_2 q) \cdot (\log_2 \log_2 \log_2 q))$. We can compare the computational cost of all the schemes by considering the number of coefficient multiplications and the cost of each coefficient multiplication. For simplicity, we only keep the $\log_2 q$ term in the cost of the Schönhage-Strassen algorithm

- baseline RLWE (NTT matrix Convolution, $t \approx N$):

$$\text{Cost}_{rn} \approx \underbrace{LN^l \log_2 N}_{\text{Num. Coeff. Mult}} \cdot \underbrace{(2Ll + L + 1) \log_2 N}_{\approx \log_2 q}.$$

- baseline RLWE (Direct Convolution, $t \approx N$):

$$\text{Cost}_{rd} \approx \overbrace{LN^{2l-1} \log_2 N}^{\text{Num. Coeff. Mult}} \cdot \overbrace{((L(l+1) + L + 1) \log_2 N}^{\approx \log_2 q} + \underbrace{(L(l-1)) \log_2 C}_{\leq 0}).$$

- multivariate RLWE ($t \approx N$):

$$\text{Cost}_{mr} \approx \underbrace{LlN^l \log_2 N}_{\text{Num. Coeff. Mult}} \cdot \underbrace{(2Ll + l + L) \log_2 N}_{\approx \log_2 q}.$$

- "packed" RLWE ($t \approx N^l$):

$$\text{Cost}_{pr} \approx \underbrace{LlN^l \log_2 N}_{\text{Num. Coeff. Mult}} \cdot \underbrace{(3Ll + l) \log_2 N}_{\approx \log_2 q}.$$

By ignoring the effect of the logarithmic terms and considering that F is not a very small filter, this gives the following approximate ratios:

$$\frac{\text{Cost}_{rd}}{\text{Cost}_{rn}} \approx N^l, \quad \frac{\text{Cost}_{mr}}{\text{Cost}_{rn}} \approx l, \quad \frac{\text{Cost}_{pr}}{\text{Cost}_{rn}} \approx l, \quad \frac{\text{Cost}_{pr}}{\text{Cost}_{mr}} \approx \frac{3}{2}$$

Hence, we can see that the baseline RLWE algorithm still gives a reduction factor in cost linear in the number of dimensions with respect to m-RLWE and packed-RLWE. However, it must be noted that the bit security of both m-RLWE and, especially, packed-RLWE is higher than baseline RLWE; in fact, this security also increases with l (see Tables 5.1(a), 5.1(b), 5.1(c) and 5.1(d)).

5.B.1. Some additional considerations

There are some considerations on the effect of the performed approximations in the computed costs Cost_{rn} to Cost_{pr}. Cost_{rd} can be much smaller than the obtained approximation when the filter is very small (i.e., C very close to zero). As we discuss in Section 5.5, for a small enough filter, the factor $\log_2 q$ can become so small that it compensates the higher number of coefficient multiplications of baseline RLWE compared to the other methods.

The main difference between Cost_{mr} and Cost_{pr} relies on the need of a higher t with packed-RLWE. This imposes more costly coefficient multiplications due to the higher ciphertext noise. Additionally, the obtained cost measures do not take into account the pre-/post-coding stage introduced by packed-RLWE before/after encryption/decryption, which is not needed in m-RLWE, but this step is negligible when compared to the encryption/decryption complexity.

The cost of the baseline RLWE scheme (Cost_{rn}) can be much higher when coefficient addition is not fast enough, as the previous expressions do not take into account the cost of ciphertext additions required for the NTT/INTT matrix computations. Hence, we introduce now this factor. The number of ciphertext additions required in baseline RLWE (with NTT matrix computation) is roughly 3 times $N^{2(l-1)}$ per level (2 NTTs of $l-1$ dimensions and 1 INTT of $l-1$ dimensions). It has an order higher than the maximum exponent in Cost_{mr} and Cost_{pr}; depending on the cost of ciphertext addition, this dependency can make the baseline algorithm slower than m-RLWE and packed-RLWE. In fact, assuming that the cost of addition per coefficient is roughly $\mathcal{O}(\log_2 q)$, we can see that the asymptotic cost of multivariate and "packed" RLWE is smaller, even for a higher security level than that of baseline RLWE. We have

$$\begin{aligned}\text{Cost}^*_{rn} &\approx \underbrace{L(1+C)^{2l-1}N^{2l-1}}_{\text{Num. Coeff. Adds.}} \cdot \underbrace{(Ll+L+1)\log_2 N}_{\approx \log_2 q} \\ &\approx LN^{2l-1}\cdot(Ll+L+1)\log_2 N,\end{aligned}$$

where Cost^*_{rn} represents the cost that the ciphertext additions in the NTT/INTT transforms incur on for baseline RLWE with NTT matrix computation, that adds up to the previous Cost_{rn} to obtain the total cost.

Again, taking the most significant factors into account, and considering that F is not a very small filter, we obtain the following approximate ratios

$$\frac{\text{Cost}_{rd}}{\text{Cost}^*_{rn}} \approx \log_2 N, \quad \frac{\text{Cost}_{mr}}{\text{Cost}^*_{rn}} \approx \frac{l\log_2 N}{N^{l-1}}, \quad \frac{\text{Cost}_{pr}}{\text{Cost}^*_{rn}} \approx \frac{l\log_2 N}{N^{l-1}}.$$

Table 5.1: Figures for (a) baseline RLWE with NTT matrix Convolution ($t \approx n_l$), (b) baseline RLWE with Direct Convolution ($t \approx n_l$), (c) multivariate RLWE ($t \approx \max\{n_1, \ldots, n_l\}$) and (d) our packed RLWE ($t \approx \prod_{i=1}^{l} n_i$).

(a) baseline RLWE with NTT matrix Convolution

Computational Cost
$\text{Cost}_{\text{linear}} = L \cdot \mathcal{O}(n_l \prod_{i=1}^{l-1}(N_i + F_i - 1) \log n_l)$ coeff. mult. $+L \cdot \mathcal{O}(n_l \left(\prod_{i=1}^{l-1}(N_i + F_i - 1)\right)^2)$ coeff. add. $\text{Cost}_{\text{cyclic}} = L \cdot \mathcal{O}(n_l \prod_{i=1}^{l-1} N_i \log n_l)$ coeff. mult. $+L \cdot \mathcal{O}(n_l \left(\prod_{i=1}^{l-1} N_i\right)^2)$ coeff. add.
Ciphertext's noise (upper bound on $\frac{\Delta}{2B}$)
(linear) $\frac{\Delta}{2B} \approx 2\left(\prod_{i=1}^{l-1} N_i(N_i + F_i - 1)\right)^L t^L (N_l + F_l - 1)^{2L+1}$ (cyclic) $\frac{\Delta}{2B} \approx 2\left(\prod_{i=1}^{l-1} N_i^2\right)^L t^L (N_l)^{2L+1}$
Bit Security
(linear) $\texttt{BitSecurity}(\sigma^2, N_l + F_l - 1)$ (cyclic) $\texttt{BitSecurity}(\sigma^2, N_l)$

(b) baseline RLWE with Direct Convolution

Computational Cost
$\text{Cost}_{\text{linear}} = L \cdot \mathcal{O}(n_l \prod_{i=1}^{l-1}(N_i F_i) \log n_l)$ coeff. mult. $\text{Cost}_{\text{cyclic}} = L \cdot \mathcal{O}(n_l \prod_{i=1}^{l-1}(N_i F_i) \log n_l)$ coeff. mult.
Ciphertext's noise (upper bound on $\frac{\Delta}{2B}$)
(linear) $\frac{\Delta}{2B} \approx 2\left(\prod_{i=1}^{l-1} F_i\right)^L t^L (N_l + F_l - 1)^{2L+1}$ (cyclic) $\frac{\Delta}{2B} \approx 2\left(\prod_{i=1}^{l-1} F_i\right)^L t^L (N_l)^{2L+1}$
Bit Security
(linear) $\texttt{BitSecurity}(\sigma^2, N_l + F_l - 1)$ (cyclic) $\texttt{BitSecurity}(\sigma^2, N_l)$

(c) multivariate RLWE

Computational Cost
$\text{Cost}_{\text{linear}} = L \cdot \mathcal{O}(\prod_{i=1}^{l}(N_i + F_i - 1) \log(\prod_{i=1}^{l} N_i + F_i - 1))$ coeff. mult. $\text{Cost}_{\text{cyclic}} = L \cdot \mathcal{O}((\prod_{i=1}^{l} N_i) \log(\prod_{i=1}^{l} N_i))$ coeff. mult.
Ciphertext's noise (upper bound on $\frac{\Delta}{2B}$)
(linear) $\frac{\Delta}{2B} \approx 2t^L \left(\prod_{i=1}^{l}(N_i + F_i - 1)\right)^{2L+1}$ (cylic) $\frac{\Delta}{2B} \approx 2t^L \left(\prod_{i=1}^{l} N_i\right)^{2L+1}$
Bit Security
(linear) $\texttt{BitSecurity}(\sigma^2 \prod_{i=1}^{l-1}(N_i + F_i - 1), N_l + F_l - 1)$ (cylic) $\texttt{BitSecurity}(\sigma^2 \prod_{i=1}^{l-1} N_i, N_l)$

(d) packed RLWE

Computational Cost
$\text{Cost}_{\text{linear}} = L \cdot \mathcal{O}(\prod_{i=1}^{l}(N_i + F_i - 1) \log(\prod_{i=1}^{l} N_i + F_i - 1))$ coeff. mult. $\text{Cost}_{\text{cyclic}} = L \cdot \mathcal{O}((\prod_{i=1}^{l} N_i) \log(\prod_{i=1}^{l} N_i))$ coeff. mult.
Ciphertext's noise (upper bound on $\frac{\Delta}{2B}$)
(linear) $\frac{\Delta}{2B} \approx 2t^L \left(\prod_{i=1}^{l}(N_i + F_i - 1)\right)^{2L+1}$ (cylic) $\frac{\Delta}{2B} \approx 2t^L \left(\prod_{i=1}^{l} N_i\right)^{2L+1}$
Bit Security
(linear) $\texttt{BitSecurity}(\sigma^2, \prod_{i=1}^{l}(N_i + F_i - 1)$ (cylic) $\texttt{BitSecurity}(\sigma^2, \prod_{i=1}^{l} N_i)$

Table 5.2: Cost and noise bounds for (a) baseline RLWE with NTT matrix Convolution ($t \approx n_l$, $N_i = N, F_i = C \cdot N$), (b) baseline RLWE with Direct Convolution ($t \approx n_l$, $N_i = N, F_i = C \cdot N$), (c) m-RLWE ($t \approx \max\{n_1, \ldots, n_l\}$) and packed-RLWE ($t \approx \prod_{i=1}^{l} n_i$, $N_i = N, F_i = C \cdot N$).

(a) baseline RLWE with NTT matrix Convolution

Computational Cost
$\text{Cost}_{\text{linear}} = L \cdot \mathcal{O}((1+C)^l N^l \log((1+C)N))$ coeff. mult. $+L \cdot \mathcal{O}(((1+C)N)^{2l-1})$ coeff. add. $\text{Cost}_{\text{cyclic}} = L \cdot \mathcal{O}(N^l \log N)$ coeff. mult. $+L \cdot \mathcal{O}(N^{2l-1})$ coeff. add.
Ciphertext's noise (upper bound on $\frac{\Delta}{2B}$)
(linear) $\frac{\Delta}{2B} \approx 2(1+C)^{L(l+2)+1} t^L N^{2L(l+1)+1}$ (cyclic) $\frac{\Delta}{2B} \approx 2t^L N^{2L(l+1)+1}$

(b) baseline RLWE with Direct Convolution

Computational Cost
$\text{Cost}_{\text{linear}} = L \cdot \mathcal{O}((1+C)C^{l-1}N^{2l-1} \log((1+C)N))$ $\text{Cost}_{\text{cyclic}} = L \cdot \mathcal{O}(C^{l-1}N^{2l-1} \log N)$
Ciphertext's noise (upper bound on $\frac{\Delta}{2B}$)
(linear) $\frac{\Delta}{2B} \approx 2C^{Ll}(1+C)^{2L+1} t^L N^{L(l+2)+1}$ (cyclic) $\frac{\Delta}{2B} \approx 2C^{Ll} t^L N^{L(l+2)+1}$

(c) m-RLWE and packed-RLWE

Computational Cost
$\text{Cost}_{\text{linear}} = L \cdot \mathcal{O}(l(1+C)^l N^l \log((1+C)N))$ $\text{Cost}_{\text{cyclic}} = L \cdot \mathcal{O}(lN^l \log N)$
Ciphertext's noise (upper bound on $\frac{\Delta}{2B}$)
(linear) $\frac{\Delta}{2B} \approx 2(1+C)^{2Ll+l} t^L N^{2Ll+l}$ (cylic) $\frac{\Delta}{2B} \approx 2t^L N^{2Ll+l}$

Table 5.3: Runtimes and security for encrypted 2D Linear Filtering ($L = 1$, $\sigma = 8$, $B = 6\sigma$, 2 limbs for q, $F = 11$).

$N \times N$	118×118	246×246
baseline RLWE (NTT matrix Convolution)		
n	128	256
Enc. (image + filter) size (bits)	$4.09 \cdot 10^6$	$16.32 \cdot 10^6$
Bit security	≈ 31	≈ 33
Encryption time (*ms*)	2.4	5.8
Decryption time (*ms*)	1.4	3.7
Convolution time (*ms*)	43.3	142.4
Baseline RLWE (Direct Convolution		
n	128	256
Enc. (image + filter) size (bits)	$4.09 \cdot 10^6$	$16.32 \cdot 10^6$
Bit security	≈ 31	≈ 33
Encryption time (*ms*)	2.4	5.8
Decryption time (*ms*)	1.4	3.7
Convolution time (*ms*)	272.5	812.6
Multivariate RLWE		
n (effective n)	16384 (128)	65536 (256)
Enc. (image + filter) size (bits)	$8.13 \cdot 10^6$	$32.51 \cdot 10^6$
Bit security	≈ 32	≈ 33
Encryption time (*ms*)	1.6	8.6
Decryption time (*ms*)	1.3	7.8
Convolution time (*ms*)	28.2	127.5
Packed RLWE		
n	16384	65536
Enc. (image + filter) size (bits)	$8.13 \cdot 10^6$	$32.51 \cdot 10^6$
Bit security	> 128	> 128
Encryption time (*ms*)	3.1	12.6
Decryption time (*ms*)	2.8	11.8
Convolution time (*ms*)	28.2	127.5

Table 5.4: Runtimes and security for encrypted 3D Cyclic Filtering ($L = 1$, $\sigma = 8$, $B = 6\sigma$, 2 limbs for q, $F = 5$).

$N \times N \times N$	$16 \times 16 \times 16$	$32 \times 32 \times 32$
baseline RLWE (NTT matrix Convolution)		
n	16	32
Enc. (image + filter) size (bits)	$1.12 \cdot 10^6$	$8.32 \cdot 10^6$
Bit security	< 30	< 30
Encryption time (*ms*)	2.9	5.6
Decryption time (*ms*)	0.3	2.6
Convolution time (*ms*)	6.0	58.1
Baseline RLWE (Direct Convolution		
n	16	32
Enc. (image + filter) size (bits)	$1.12 \cdot 10^6$	$8.32 \cdot 10^6$
Bit security	< 30	< 30
Encryption time (*ms*)	2.9	5.6
Decryption time (*ms*)	0.3	2.6
Convolution time (*ms*)	150.1	1452.8
Multivariate RLWE		
n (effective n)	4096 (16)	32768 (32)
Enc. (image + filter) size (bits)	$2.03 \cdot 10^6$	$16.25 \cdot 10^6$
Bit security	< 30	< 30
Encryption time (*ms*)	0.6	3.7
Decryption time (*ms*)	0.4	3.0
Convolution time (*ms*)	6.4	53.3
Packed RLWE		
n	4096	32768
Enc. (image + filter) size (*bits*)	$2.03 \cdot 10^6$	$16.25 \cdot 10^6$
Bit security	> 128	> 128
Encryption time (*ms*)	0.8	6.0
Decryption time (*ms*)	0.7	5.4
Convolution time (*ms*)	6.4	53.3

Table 5.5: Ciphertext noise bounds for all schemes ($N_i = N, F_i = C \cdot N$, ignoring $(1 + C)$ factor).

Ciphertext noise (upper bound on $\frac{\Delta}{2}$)
(baseline RLWE, NTT matrix), (linear, cyclic) $\frac{\Delta}{2} \approx 2BN^{2Ll+1}t^L$
(baseline RLWE, Dir. Conv.), (linear, cyclic) $\frac{\Delta}{2} \approx 2BC^{L(l-1)}N^{L(l+1)+1}t^L$
(m-/packed RLWE),(linear, cyclic) $\frac{\Delta}{2} \approx 2BN^{2Ll+l}t^L$

Part III

Signal Processing Applications

Chapter 6

Genomic Susceptibility Testing

This chapter is adapted with permission from IEEE: Juan Ramón Troncoso-Pastoriza, Alberto Pedrouzo-Ulloa, and Fernando Pérez-González. Secure Genomic Susceptibility Testing based on Lattice Encryption. The 42nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP17), March 2017.

6.1. Introduction

Genomic research has experienced a considerable growth in the last years due to the advances in Next Generation Sequencing (NGS), which enable potentially better analyses, tests, diagnostics and treatments based on genomic data. The growing volume of genomic data available to be processed, cannot be managed by current facilities at hospitals and laboratories. The need for outsourced genomic processing is urgent, but it entails severe privacy risks [139] comprising, among others, re-identification threats (it is not possible to entirely anonymize genomic data), phenotype inference (sharing aggregate genomic data, even pseudonymized, enables kin privacy breaches), and other threats (anonymous paternity breaches, legal and forensic inferences), affecting not only the individual but also his/her ancestors and descendants.

Several proposals of privacy-preserving mechanisms have arisen to cope with these threats in two main fields: research studies like Genome-Wide Association Studies (GWAS), and personalized health-care. While the former has been recently tackled through differentially-private mechanisms [140, 141], dealing with person-level genome sequence records prevents the use of generalization techniques or differentially-private mechanisms, and the solution must involve cryptographic primitives, which are generally costlier than other approaches.

One of the most recent privacy-preserving mechanisms for disease susceptibility outsourced processing was proposed by Ayday *et al.* [142], which introduce an untrustworthy Storage and Processing Unit (SPU) to deal with the outsourced encrypted processing, and devise a protocol based on additive homomorphic encryption and proxy decryption to enable the calculation of simple susceptibility tests on a set of Single Nucleotide Polymorphisms (SNPs) of one patient; this encrypted test is eventually handled by the medical center due to the limitations of the used homomorphism. Subsequently, Namazi *et al.* [143] proposed the use of lattice-based somewhat homomorphic encryption (SHE) to move the computation complexity to the SPU, but they did not evaluate it nor addressed the shortcomings introduced when dealing with SHE, namely increased cipher expansion, higher bandwidth requirements and much higher storage needs for the encrypted sequences.

Contributions: In this chapter, we propose an efficient protocol to deal with encrypted genomic susceptibility tests based on Ring Learning with Errors (RLWE) cryptosystems, and introduce optimizations which lead to a considerable improvement in terms of computation, bandwidth and storage with respect to both the original protocol by Ayday *et al.* and Namazi *et al.*.

Notation and structure: Uppercase letters denote matrices and lowercase letters denote elements from a vector space. a_{E_P} denotes the result of the encryption of a with the key belonging to P. The rest of the chapter is organized as follows: Section 6.2 briefly introduces the used cryptosystem and its primitives. Section 6.3 revisits the scheme by Namazi *et al.* [143]. Section 6.4 describes our proposed protocol and the introduced optimizations. Section 6.5 evaluates the secure protocol in terms of ciphertext size, run times and communication, and compares it to the prior works.

6.2. RLWE-based SHE

We choose Lauter *et al.*'s [79] as our cryptosystem, due to its simplicity, efficiency and security, but any other RLWE cryptosystem (as FV [86] or BGV [50]) can be used as well. Table 4.1 (see Chapter 4.2) summarizes its parameters and primitives.

Furthermore, by means of a relinearization matrix $\boldsymbol{B}$ it is possible to transform three-component encryptions after a homomorphic product back into two-component fresh-like encryptions. This matrix can also be used as a proxy reencryption in order to perform the key change needed at the end of the protocol (see Section 6.3). In this case, the relinearization process of a multiplied ciphertext (vector of 3 components in R_q) $\boldsymbol{c}_{P_1} = \{c_1, c_2, c_3\}$ under P_1's key into P_2's key can be expressed as a matrix product $\boldsymbol{c}_{P_2} = \{c_1, c_2\} + \boldsymbol{c_{3,base-t}} \cdot \boldsymbol{B}$, where $\boldsymbol{c_{3,base-t}}$ is a $\lceil \log_t q \rceil$-length row vector with the base-t decomposition of the polynomial c_3, and matrix $\boldsymbol{B}$ has size $\lceil \log_t q \rceil \times 2$ (see Section 4.5.1 of Chapter 4 for further details).

The equivalent bit-security of this cryptosystem can be lower-bounded [110] by $t_{BKZ}(\delta)$ (see Eq. (4.3)), where δ is the Root Hermite Factor of the used polynomial lattice.

6.3. Encrypted Susceptibility Tests

The genomic sequence of each individual presents variations with respect to the reference sequence which fully identify the individual. The most common and relevant variants are called SNPs (Single Nucleotide Polymorphisms), which are particularly suitable for running susceptibility tests of certain diseases. Weighted averaging [144] is the simplest way to measure the susceptibility of a patient P to a disease x:

$$S^{P,x} = \sum_{i \in \Omega_x} \overline{c}^{x,i} \{pr_0^{x,i}[1 - \text{SNP}^{P,i}] + pr_1^{x,i}[\text{SNP}^{P,i}]\}. \tag{6.1}$$

The symbols used in Eq. (6.1) are defined in Table 6.1. As this test involves a bounded number of additions and products, an SHE scheme allows to execute it with all the inputs encrypted. We

Table 6.1: Used Notation.

Γ^P	Set of positions of real SNPs of patient $\mathcal{P}$
γ^P	Set of positions of potential SNPs of patient $\mathcal{P}$
$\text{SNP}^{P,i}$	i-th SNP for patient $\mathcal{P}$. $\text{SNP}^{P,i}$ equals 0 when it belongs to γ^P, and 1 when the patient presents a variant (it belongs to Γ^P)
Ω_x	Set of *relevant* positions of SNPs which are related to disease x.
$pr_b^{x,i}$	$\Pr(x\|SNP^{P,i} = b)$, with $b \in 0,1$. Probability of developing disease x conditioned on the value of the i-th SNP
$\overline{c}^{x,i}$	Normalized contribution of $SNP^{P,i}$ to the susceptibility to x.
$S^{P,x}$	Predicted susceptibility of patient $\mathcal{P}$ to disease x

briefly revisit the protocol by Namazi et al. [143] to calculate Eq. (6.1) homomorphically, with the following parties: a patient P owns a biological sample; a medical center MC has the knowledge of the parameters (pr, c) for calculating the susceptibility to disease x; the certified institution CI is a trusted party that sequences the patient's DNA and generates all the used cryptographic keys; the Storage and Processing Unit SPU is an untrustworthy party with computational power to execute the encrypted test. The patient does not trust the MC to share all his/her genomic data, and both MC and P distrust SPU with respect to the analysis parameters and the patient's data. All parties are considered to be semi-honest.

The protocol works as follows (see Figure 6.1):

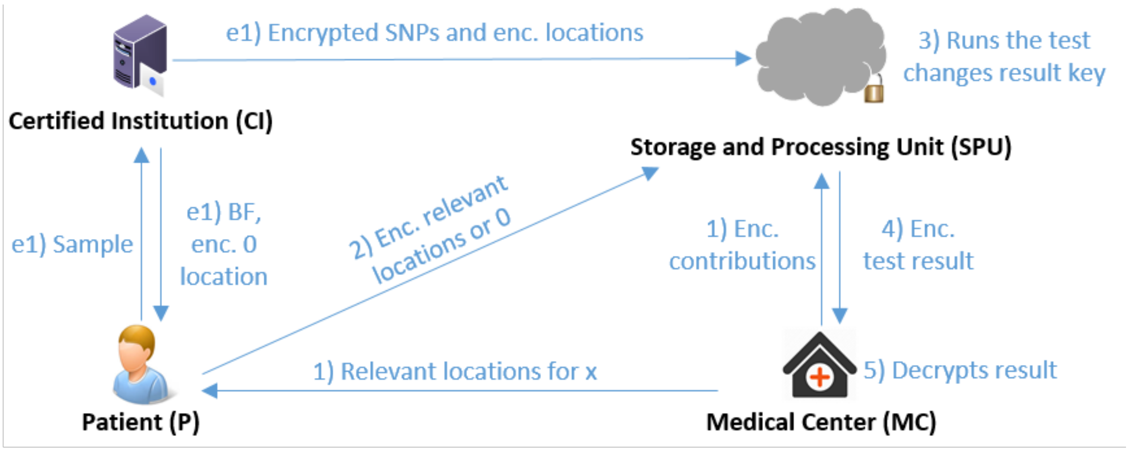

e1) Encrypted SNPs and enc. locations
3) Runs the test changes result key
Certified Institution (CI)
Storage and Processing Unit (SPU)
e1) Sample
e1) BF, enc. 0 location
2) Enc. relevant locations or 0
1) Enc. contributions
4) Enc. test result
1) Relevant locations for x
5) Decrypts result
Patient (P)
Medical Center (MC)

Figure 6.1: Encrypted susceptibility testing protocol.

Step s1: The CI generates and distributes the needed keys: P and MC have one SHE keypair each, while P and CI share a symmetric key $sk_{P,CI}$; the CI also produces a relinearization matrix $\boldsymbol{B}$ to change encryptions from P's key into MC key, and sends it to the SPU.

Sequencing and generation of input encryptions

Step e1: After P sends the biological sample to CI, the latter sequences it, builds a Bloom Filter representing the positions for which the patient presents SNPs, and sends it to P; CI encrypts these positions $\{l_{i,E_{P,CI}}\}$ and a "zero position" $l_{0,E_{P,CI}}$ with $sk_{P,CI}$, and the values of all SNPs $\text{SNP}^{P,i}$ with P's SHE key, and sends all these encryptions to the SPU.

Encrypted susceptibility test

Step 1: The MC marks the location of SNPs in Ω_x and sends them to P. Additionally, it sends the contributions of these SNPs to the disease x encrypted under P's SHE key to SPU: $\{[pr_b^{x,i} \cdot \overline{c}^{x,i}]_{E_P}\}_{b\in\{0,1\},i\in\Omega_x}$.

Step 2: P runs the Bloom filter for these positions; for those in the filter (present variants), P encrypts the corresponding location $l_{i,E_{P,CI}}$ and sends it to SPU; otherwise, P sends the encryption $l_{0,E_{P,CI}}$.

Step 3: The SPU computes the susceptibility Eq. (6.1) on patient's encrypted SNPs and MC's encrypted susceptibility parameters for x by using the homomorphic properties of the SHE scheme, obtaining the encryption of $S_{E_P}^{P,x}$ under P's key.

Step 4: The SPU uses the relinearization matrix to switch the result into MC's key, and sends it to MC.

Step 5: The MC decrypts the clear-text test result $S^{P,x}$ of patient $\mathcal{P}$ for the disease x using its own SHE secret key.

This protocol succeeds in moving all homomorphic computation to the SPU and keeping the locations and values of P's SNPs concealed from the SPU and the MC, and the test parameters concealed from the SPU. Conversely, its high cipher expansion makes it much more demanding in terms of storage and bandwidth compared to the Paillier based scheme by Ayday *et al.*, as we show in Section 6.5.

6.4. Proposed Approach

As can be seen from the protocol description in Section 6.3, the only elements which have to be encrypted with a homomorphic encryption are the patient SNPs, and the susceptibility parameters; Ayday *et al.* [142] encrypted only the patient SNPs, as the computation was done at the MC, who already knows the clear-text susceptibility contributions. Blindly applying lattice encryptions to the protocol produces a huge growth in the cipher expansion: SNPs are binary values (either present 1 or absent 0), which get encrypted into several thousand bits in Paillier, and several hundred thousand bits with an RLWE cryptosystem. Hence, even when the lattice-based operations are more efficient than their Paillier-based counterparts, the large cipher expansion becomes a serious drawback when coping with 4 million SNPs per patient. Figure 6.2 presents a high-level view of our proposed approach for dealing with the encrypted calculation of the susceptibility. We present four main contributions described in the following paragraphs: a clever choice of the cryptosystem parameters to optimize the performance and maximize the security of the protocol; an input packing strategy to minimize storage and bandwidth; a pre-processing mechanism based on transformed coefficients to enable the homomorphic calculation of component-wise products between vectors of susceptibility coefficients and SNPs, and a homomorphic blinding strategy to enable the seamless calculation of the addition of all the components in one vector while avoiding costly unpacking/repacking operations at the SPU.

6.4.1. Parameter choice

RLWE cryptosystems work with polynomials in R_q; i.e., the ring product is a polynomial product (convolution). In order to speed up products, it is more convenient to work in a trans-

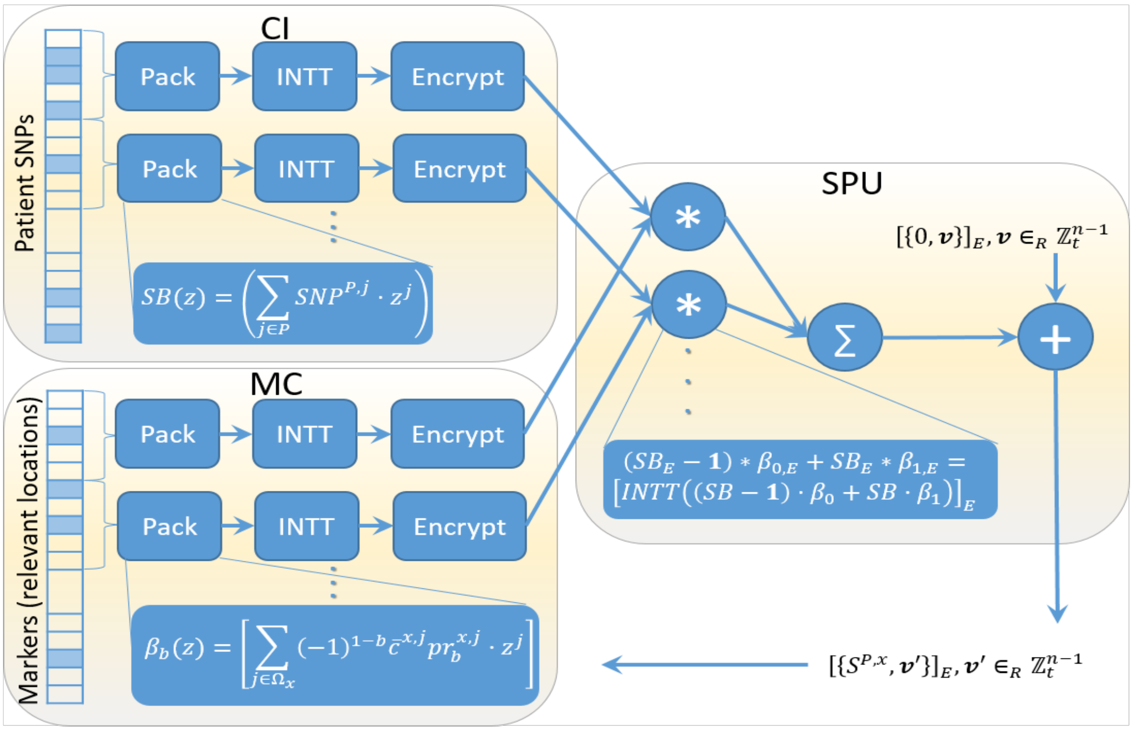

CI
Patient SNPs
Pack
INTT
Encrypt
Pack
INTT
Encrypt
$SB(z) = \left(\sum_{j\in P} SNP^{P,j} \cdot z^j\right)$
MC
Markers (relevant locations)
Pack
INTT
Encrypt
Pack
INTT
Encrypt
$\beta_b(z) = \left[\sum_{j\in\Omega_x} (-1)^{1-b}\bar{c}^{x,j} pr_b^{x,j} \cdot z^j\right]$
SPU
$*$
$*$
$\sum$
$+$
$[\{0, \boldsymbol{v}\}]_E, \boldsymbol{v} \in_R \mathbb{Z}_t^{n-1}$
$(SB_E - \mathbf{1}) * \beta_{0,E} + SB_E * \beta_{1,E} =$
$[INTT((SB - \mathbf{1}) \cdot \beta_0 + SB \cdot \beta_1)]_E$
$[\{S^{P,x}, \boldsymbol{v}'\}]_E, \boldsymbol{v}' \in_R \mathbb{Z}_t^{n-1}$

Figure 6.2: Diagram of the encrypted susceptibility computation.

formed domain with the convolution property, where convolutions become much more efficient component-wise products. As these cryptosystems work in finite rings, we stick to Number Theoretic Transforms (NTTs) instead of Discrete Fourier Transforms (DFTs), which would introduce undesirable rounding errors [29]. For an n-th root of unity α in the ring, the NTT has a similar form to the DFT (see Chapter 4):

$$NTT\{x\} = \sum_{i=0}^{n-1} x[i] \cdot \alpha^{ik}, \; INTT\{X\} = n^{-1} \cdot \sum_{k=0}^{n-1} X[k] \cdot \alpha^{-ik}.$$

Therefore, we parameterize the cryptosystem to enable component-wise operations in the NTT domain. We choose $n = 2^k$ (polynomial degree in R_q) as a power of 2, and q and t as Proth primes $(c \cdot 2^k + 1)$ [29]; this choice guarantees that an n-th root of unity exists in $\mathbb{Z}_q$ (ciphertext coefficients) and in $\mathbb{Z}_t$ (plaintext coefficients), in such a way that NTTs of size n exist both in $\mathbb{Z}_q$ and $\mathbb{Z}_t$. All the used polynomials (random polynomials, input plaintexts and keys) undergo an NTT prior to encryption, all ciphertexts are always expressed in the NTT domain, and decryptions are followed by an INTT of the resulting polynomial.[1] Hence, all the intermediate operations are considerably faster (component-wise), and encryption and decryption suffer from a slight overhead for calculating the NTT/INTT with fast algorithms ($\mathcal{O}(n \log(n))$).

6.4.2. Input Packing

Due to the polynomial structure of RLWE cryptosystems, the cipher expansion can be reduced by packing the inputs in vectors of n elements (as many as the degree of the polynomials in R_q,

[1] In order to perform cyclic convolutions inside a negacyclic ring (mod$x^n + 1$), signals must be pre- and post-processed with a component-wise product with a vector of powers of a root of -1 in $\mathbb{Z}_q$ [29] (see Chapter 4). This operation is already accounted for in all the measured run times.

Table 6.2: Evaluation runtimes, bandwidth and storage for 4M SNPs and a 10-marker test ($|\Omega_x| = 10$).

Run time [ms] / transferred size		CI	SPU	
Ayday *et al.*[142]		Encrypt per SNP	Recrypt	Proxy recrypt
2048 bit modulus, 112 bit sec.		33,2 ms / 4,1 GB	304,3 ms / 10,2 kB	30,3 ms / 1,02 kB
		Encrypt per SNP	Homomorphic calc.	Relinearization
RLWE $n = 4096$	Unpacked	0,45 ms / 262,1 GB	2,17 ms / —	2,32 ms / 65,5 kB
364 bit sec. ($\delta = 1.002$)	Packed	0,00011 ms / 64 MB	0,1 – 2,17 ms / —	2,32 ms / 65,5 kB
RLWE $n = 2048$	Unpacked	0,22 ms / 131,1 GB	1,08 ms / —	1,1 ms / 32,8 kB
127 bit sec. ($\delta = 1.005$)	Packed	0,00011 ms / 64 MB	0,05 – 1,08 ms / —	1,1 ms / 32,8 kB

Run time [ms] / transferred size		MC	
Ayday *et al.*[142]		Homomorphic calc.	Paillier decrypt
2048 bit modulus, 112 bit sec.		39,3 ms / 1,02 kB	30,3 ms / —
		Encrypt params	RLWE Decrypt
RLWE $n = 4096$	Unpacked	9,1 ms / 1,31 MB	0,96 ms / —
364 bit sec. ($\delta = 1.002$)	Packed	0,45 – 9,1 ms / 0,131 – 1,31 MB	0,96 ms / —
RLWE $n = 2048$	Unpacked	4,5 ms / 655 kB	0,46 ms / —
127 bit sec. ($\delta = 1.005$)	Packed	0,22 – 4,5 ms / 65,5 – 655 kB	0,46 ms / —

see Table 4.1) instead of encrypting one scalar value per ciphertext. For the devised susceptibility test protocol, the CI can encrypt the SNPs of the patient in blocks of n SNPs per ciphertext, which divides the storage overhead by a factor of n. This creates a two-level indexing of the SNPs (i, j), where i indexes the block where the SNP was encrypted, and j indexes the polynomial coefficient ($j \in \{0, n-1\}$) where the SNP was packed inside the block. The mapping between the SNP location and the indices (i, j) can be freely chosen by the CI, and must be known by the MC. This alters steps 1 and 2 of the protocol: In step 1, the MC encrypts the contributions of a SNP indexed by (i, j) in the j-th coefficient of the polynomial, and zeros in the other coefficients. If several relevant SNPs belong to the same block, their contributions are packed together in the same encryption. In step 2, after running the Bloom Filter, P sends to the SPU the encrypted location $l_{i,E_{P,CI}}$ indexing the chunks of SNPs where the relevant positions belong, and sends no information about j.

6.4.3. Packed operations: pre-processing

Once the inputs are packed, the calculation of the Eq. (6.1) requires the homomorphic execution of component-wise products of SNP contributions and SNP values. This is not possible if we encrypt the input blocks of SNPs directly, as the cryptosystem only allows for homomorphic convolutions. Hence, the CI (resp. MC) first applies an INTT to the polynomial of SNP values (resp. contributions), and then encrypts the transformed values. Then, due to the convolution property of the NTT, the homomorphic operations become:

$$\begin{aligned} INTT(\{SNP^{P,(i,j)}\}_j) \circledast INTT(\{pr_b^{x,(i,j)}\overline{c}^{x,(i,j)}\}_j) = \\ INTT(\{SNP^{P,(i,j)} \cdot pr_b^{x,(i,j)}\overline{c}^{x,(i,j)}\}_j). \end{aligned}$$

These transforms are enabled by our choice of t and n, that guarantees that the n-size INTTs exist for coefficients in $\mathbb{Z}_t$. Therefore, the SPU can seamlessly obtain the encrypted component-wise products contributing to the susceptibility.

6.4.4. Obtaining the test result

After the previous process, the SPU ends up with an encrypted vector holding the INTT coefficients of the component-wise products, but the cryptosystem homomorphism does not allow to add them together without decrypting and unpacking them first. To overcome this limitation, we leverage the structure of the NTT, by realizing that the first coefficient of the INTT is just the sum of all the signal coefficients in the time domain, multiplied by the modular inverse of n in $\mathbb{Z}_t$. Hence, the SPU generates a random vector $\boldsymbol{v} \in \mathbb{Z}_t^{n-1}$ to blind the remaining INTT coefficients, and homomorphically adds it to the packed susceptibility encryption (at the end of step 3). Then, after performing the relinearization and sending back the resulting encryption to MC (step 4), the latter can decrypt the result and obtain a vector which holds the susceptibility result $S^{P,x}$ in the first coefficient (multiplied by $n^{-1} \mod t$) and random values in the remaining coefficients. Hence, we also avoid that the MC has to execute an NTT to revert the INTT that was applied to the inputs.

6.5. Implementation and Evaluation

We implemented the full protocol in C++ with and without packing, using the NFLlib [94] library, and Ayday's Paillier-based version with GMP [124]. According to Section 6.4.1, we choose $t = 65537$, as it is enough to deal with all the input values with a precision of 10^{-3} for a test of up to 65 markers; due to efficiency reasons, we fix q to 62 bits, such that it fits in a limb (8 bytes) and all operations on polynomial coefficients are performed in just one machine cycle; additionally, this choice of q and t allows for the correct computation of one encrypted polynomial product between two fresh encryptions, which is enough to homomorphically calculate Eq. (6.1).

We choose medium-term security for Paillier, with 2048-bit modulus (112 bits of security), and two levels of security for our lattice-based protocol: $n = 2048$, which produces an equivalent security of 127 bits ($\delta = 1.005$, see Section 6.2), and $n = 4096$, with 364 bits of security ($\delta = 1.002$). Table 6.2 shows the run times for each party on an Intel Core i5-2500 processor at 3.3 GHz running Linux, and the sizes of the transferred encryptions at each step for 4 million SNPs per patient and a test with 10 relevant SNPs (markers) in Ω_x.

The RLWE-based protocols greatly outperform the Paillier-based Ayday *et al.* protocol in terms of efficiency (two orders of magnitude for SPU and CI, and one order of magnitude for the MC), while keeping all the homomorphic computation at the SPU instead of the MC. As for the bandwidth, the unpacked solution suffers from the big cipher expansion of the RLWE encryptions, producing a huge set of encrypted SNPs at the CI. The proposed strategies greatly reduce this overhead, limiting the stream of the 4 million encrypted SNPs to just 64 MB, notably lower than the 4 GB needed for the Paillier encryptions, improving on storage needs. The improvement achieved on homomorphic computation depends on the number of blocks spanned by the positions of the relevant SNPs, analogously to the bandwidth needed between SPU and CI. Both can be optimized by configuring the (public) ordering of the SNPs (mapping of the indices (i, j)) so that most of the SNPs relevant for the same diseases be together in the same block.

It must be noted that the performed packing, the used SNP indexing and the blinding of the resulting vector leak no information either to the SPU or to the MC, in such a way that the same security properties and privacy guarantees of the unpacked Paillier-based protocol are preserved here.

6.6. Conclusions

This chapter proposes a privacy-preserving genomic susceptibility protocol based on a Ring Learning with Errors SHE cryptosystem which outperforms previous protocols in terms of efficiency, bandwidth and storage needs. We introduce a choice of cryptosystem parameters to optimize the performance and the security of the protocol, and propose a transformed input packing strategy to minimize storage and bandwidth, and enable the homomorphic calculation of the susceptibility function while avoiding costly unpacking/repacking operations.

Chapter 7

Image Denoising

This chapter is adapted with permission from IEEE: Alberto Pedrouzo-Ulloa, Juan Ramón Troncoso-Pastoriza, and Fernando Pérez-González. Image Denoising in the Encrypted Domain. The 8th IEEE International Workshop on Information Forensics and Security (WIFS16), December 2016.

7.1. Introduction

The problem of image (or signal) denoising is ubiquitous in signal processing and has a broad set of applications. It appears in any possible scenario looking for the best possible estimate of a signal from a noisy version. Nowadays, outsourced services are increasingly used, so it is not hard to imagine a situation where someone wants to obtain an enhanced version of a noisy signal by relying on a third party to perform the task, therefore incurring in a threat for the privacy of the involved sensitive information. The approaches presented in [4] to deal with images are not enough to tackle the problem non-interactively, requiring interactive secure protocols to obtain a feasible solution. Some current proposals for encrypted domain processing target unattended processing, without resorting to interactive secure protocols [29], but they are limited to polynomial operations.

We can find some recent works dealing with privacy-preserving denoising: Hu *et al.* [145] propose an scheme for performing nonlocal means (NLM) denoising of encrypted images, and Saghaian *et al.* [146] propose a scheme for wavelet denoising resorting to secret sharing. However, the former does not deal with wavelet denoising algorithms (it performs a filtering operation and leaks pixel distances) and the latter is based on interactive protocols (secret sharing).

This chapter proposes a new solution to the problem of denoising of an image (or a more general multidimensional signal) in the encrypted domain in a fully unattended way. For this purpose, we solve the problem of homomorphically computing both filtering and threshold operations in a sole round without resorting to the intervention of the secret key owner.

Main Contributions: We briefly summarize the main ideas and contributions of this chapter:

- We introduce a practical scheme for homomorphically denoising images in the encrypted domain. The results can be easily adapted to work with either uni- or multi-dimensional signals.

- The main advantage of our scheme is that it avoids interactive protocols. Therefore, the secret key owner does not need to participate in the middle of the encrypted computation to complete the denoising process.
- We show how to adapt the structure of modern lattice-based cryptosystems to efficiently compute a wavelet transform.
- In the same round, we show how to homomorphically perform the threshold of encrypted values without the need of intermediate decryption or interaction with the secret key owner.

Structure: The rest of the chapter is organized as follows: Section 7.2 revisits some relevant concepts related to the used 2-RLWE (Ring Learning with Errors) based cryptosytem and a brief overview of the image denoising problem. Section 7.3 introduces the main contributions of this chapter, including the description of the proposed scheme for encrypted image denoising. Section 7.4 discusses some practical aspects aimed towards an efficient implementation of the proposed scheme, and evaluates its security and efficiency.

7.2. Preliminaries

This section revises the lattice-based cryptosystem chosen to exemplify our schemes, together with its main parameters and primitives. It also includes a brief explanation of the image denoising problem.

7.2.1. 2-RLWE based Cryptosystem

Firstly, we revisit a slightly adapted definition of the m-RLWE problem [22, 4] particularized to our bivariate case. For a general discussion of the m-RLWE problem with power-of-two modular functions, we refer the reader to Chapters 2 and 5. Here we particularize the Definition 1 of m-RLWE to 2-RLWE.

Definition 13 (2-RLWE problem [22, 4], Definition 1 particularized to bivariate rings)**.** *Given a polynomial ring $R_q[x,y] = (\mathbb{Z}_q[x,y]/(x^{n_x}+1))\,/(y^{n_y}+1)$ and an error distribution $\chi[x,y] \in R_q[x,y]$ that generates small-norm random polynomials in $R_q[x,y]$, 2-RLWE relies upon the computational indistinguishability between samples $(a_i, b_i = a_i s + t \cdot e_i)$ and (a_i, u_i), where a_i, u_i $\leftarrow$ $R_q[x,y]$ are chosen uniformly at random from the ring $R_q[x,y]$, while $s, e_i \leftarrow \chi[x,y]$ are drawn from the error distribution, and t is relatively prime to q.*

The primitives and parameters of the 2-RLWE cryptosystem are described in Table B.1 from Appendix B; being the only difference that in this chapter we consider bivariate rings. Its ciphertexts are composed of at least 2 polynomial elements from the ring $R_q[x,y]$; the cryptosystem allows for additions (the smallest ciphertext is previously zero-padded) and multiplications on these tuples of polynomials, whose size increases after each multiplication. They can be brought back to the original size by resorting to a relinearization operation.

The security of the cryptosystem is based on the hardness of the 2-RLWE problem with power-of-two modular functions. In [46] it was assumed that its hardness was equivalent to that of reducing n-dimensional lattices ($n = n_x n_y$) generated by the secret key. However, we know from Chapter 5 that due to the Bootland *et al.*'s attack [44] its security is in fact based on a

$\max\{n_x, n_y\}$-dimensional lattice. In Section 7.4.1, we update the security estimates initially presented in [46] and discuss some possibilities to increase the security. Further details about possible attacks are discussed in Section 7.4.

We choose this cryptosystem as it enables us to encrypt 2-dimensional messages in only one ciphertext, instead of encrypting each coefficient in a different ciphertext. It also enables efficient bivariate negacyclic linear convolutions with only one ciphertext multiplication at the cost of a small overhead (we refer the reader to [4] for a more detailed comparison between homomorphic cryptosystems when dealing with images). This overhead is caused by the use of the ring $\mathbb{Z}_q$ for the polynomial coefficients of the ciphertexts instead of $\mathbb{Z}_t$, where $q > t$. In order to correctly compute D consecutive products and A sums over the same ciphertext, the needed q for correct decryption is lower-bounded by

$$q \geq 4(2t\sigma^2\sqrt{n_x n_y})^{D+1}(2n_x n_y)^{D/2}\sqrt{A}, \tag{7.1}$$

where we adapt the bound from Equation B.1 to our bivariate case with $n = n_x n_y$.

Our proposed approach involves a multiplication tree with a determined number of levels to achieve a logarithmic complexity. Therefore, we work with a scale-invariant version [86] of the 2-RLWE cryptosystem, where D in eq. (7.1) represents the number of levels of the multiplication tree.

7.2.2. Basic Structure of an Image Denoising Scenario

This section briefly introduces the general scheme of the nonlinear image denoising method which we later perform in the encrypted domain (see Section 7.3).

There are several methods to perform the denoising of one image [147]; we resort here to the use of a wavelet transform to compact the energy of the image in a few values [148]. As the wavelet transform is an orthonormal transformation, the noise distribution is invariant after computing it, and therefore, we have two main components in the transformed domain: (a) the signal component, with most of its energy compacted in a few values, and (b) the noise distribution component, typically considered Gaussian noise, which is invariant after the transformation.

Hence, in order to separate the two components, a thresholding operation in the transformed domain can preserve the signal information while discarding most of the noise. Afterwards, we can compute the inverse wavelet transform to recover the estimated image. Figure 7.1 depicts a basic scheme of the clear-text image denoising process.

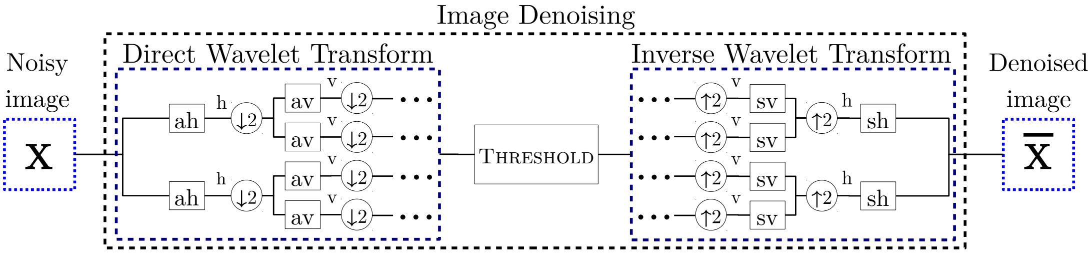


Figure 7.1: Basic structure of the image denoising method.

Figure 7.1 shows the different components of an image denoising method based on wavelet transform. Both direct and inverse wavelet transforms are typically implemented by means of filter banks where a/s and v/h stand respectively for analysis/synthesis and vertical/horizontal filters;

and ↓2 (↑2) represents downsampling (upsampling) by a factor of two. The threshold operation performs the element-wise threshold of the different transformed coefficients.

7.3. Proposed Scheme

This section introduces the proposed scheme for encrypted image denoising and details its main blocks. First, we show the general structure of the scheme and the purpose of each component. Afterwards, we focus on the two main parts of the scheme: (a) the encrypted wavelet transform (both direct and inverse transforms), and (b) the encrypted thresholding in the wavelet domain. We reiterate that we exemplify the scheme with images, but the results can be seamlessly adapted to work with higher dimensional signals [4] (see Appendix B).

7.3.1. General Overview

We exemplify the denoising operation with a typical nonlinear scheme that leverages the properties of the wavelet transform to compact the energy of the signal in a few values while keeping the energy of the noise spread through all the coefficients. This allows for separating noise and signal through a thresholding operation in the wavelet transformed domain. Currently, this problem can only be tackled efficiently in a privacy-preserving manner by resorting to interactive protocols. Our main focus is on an unattended solution which completely avoids interaction, therefore overcoming the need of intervention of the secret key owner during the process.

This paradigm introduces many challenges on the different parts of the process, the hardest one comprising the combination of both polynomial and thresholding operations in the encrypted domain without the help of the secrey key owner at each step.

Figure 7.2 depicts the general structure of our proposed solution for encryped image denoising. First, we rely on the cryptosystem presented in [4] to work with encrypted images, and we apply a light-weight pre-/post-processing [29] to enable a homomorphism with the cyclic convolution when multiplying two ciphertexts (see Section 7.3.2). The remaining blocks correspond to the homomorphic computation of the bivariate (direct and inverse) wavelet transform and the homomorphic threshold of each coefficient in the transformed domain. The following sections explain the details of these blocks.

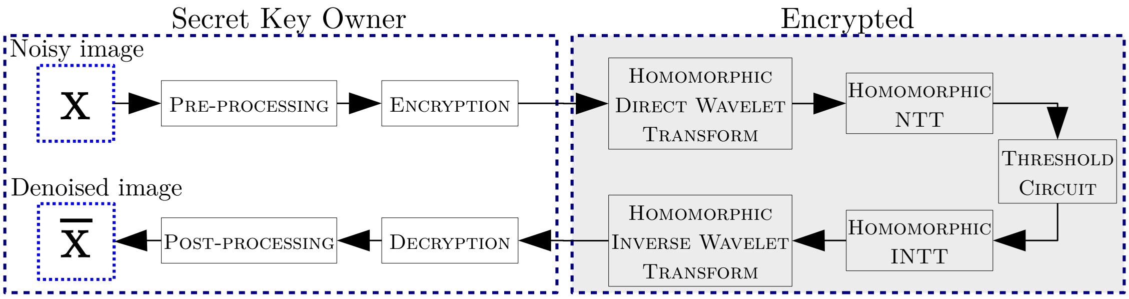


Figure 7.2: Structure of the proposed encrypted image denoising method.

7.3.2. Homomorphic Wavelet Transform by means of filter banks

This section describes the homomorphic execution of the first and last blocks from Figure 7.2 (direct and inverse wavelet transforms). For the sake of efficiency, we resort to the filter bank implementation of the wavelet transform, which uses a matrix transformation for the i-th stage of the bivariate case as follows

$$\boldsymbol{W}_i = \boldsymbol{W}_i^{(x)} \otimes \boldsymbol{W}_i^{(y)} = \begin{bmatrix} \boldsymbol{D}_i^{(x)} \boldsymbol{A}_{l_i}^{(x)} \\ \boldsymbol{D}_i^{(x)} \boldsymbol{A}_{h_i}^{(x)} \end{bmatrix} \otimes \begin{bmatrix} \boldsymbol{D}_i^{(y)} \boldsymbol{A}_{l_i}^{(y)} \\ \boldsymbol{D}_i^{(y)} \boldsymbol{A}_{h_i}^{(y)} \end{bmatrix},$$

where matrix $\boldsymbol{D}_i^{(z)}$ downsamples the input vectors of the i-th stage by a factor of two in the dimension z, and $\boldsymbol{A}_{l_i}^{(z)}, \boldsymbol{A}_{h_i}^{(z)}$ represent the circulant matrices which correspond, respectively, to the low-pass and high-pass analysis filters of the first stage in dimension z.

Analogously, we can define the inverse transform as $\boldsymbol{W}_i^{-1} = \left(\boldsymbol{W}_i^{(x)}\right)^{-1} \otimes \left(\boldsymbol{W}_i^{(y)}\right)^{-1}$ where $\left(\boldsymbol{W}_i^{(z)}\right)^{-1} = \begin{bmatrix} \boldsymbol{S}_{l_i}^{(z)} \boldsymbol{U}_i^{(z)} & \boldsymbol{S}_{h_i}^{(z)} \boldsymbol{U}_i^{(z)} \end{bmatrix}$ with $\boldsymbol{U}_i^{(z)} = \left(\boldsymbol{D}_i^{(z)}\right)^T$ and the circulant matrices $\boldsymbol{S}_{l_i}^{(z)}, \boldsymbol{S}_{h_i}^{(z)}$ are, respectively, the synthesis low-pass and high-pass filters of the i-th stage for perfect reconstruction (i.e., $\boldsymbol{W}_i^{-1}\boldsymbol{W}_i = \boldsymbol{I}_{N^{(i)}}$ with $N^{(i)} = \frac{N_x N_y}{4^{i-1}}$).

Finally, this process is recursively applied for the four outputs at each stage of the filter bank.

In light of this structure, the main needed homomorphic operations under encryption are (a) block-circulant matrix operations (multivariate cyclic convolutions), and (b) changes on the sampling rate. The following sections detail the process to achieve these operations by preserving the multivariate structure of the images.

Homomorphic Bivariate Cyclic Convolutions

The filter bank implementation of the (direct or inverse) wavelet transform for images involves a total of 4^i filtering operations in the i-th stage. In general, when working with m-dimensional signals, the i-th stage will need a total of 2^{im} filtering operations. In order to securely and efficiently compute these operations we combine two contributions:

- We resort to the multivariate cryptosystem in [4] to encrypt each image in only one ciphertext and to enable encrypted multidimensional linear and negacylic convolutions (see Section 7.2).
- We adapt the techniques from [29] for our multivariate case, in such a way that with a lightweight pre-/post-processing (negligible with respect to the encryption and decryption primitives) of the images before (after) encryption (decryption), we can homomorphically perform multivariate cyclic convolutions (see Chapters 4 and 5, and Appendix B).

Pre-/Post-processing: In [29], the authors enable homomorphic cyclic convolutions between two one-dimensional signals of length N by performing an element-wise multiplication of both signals with $(-1)^{l/N}$ for $l = \{0, \ldots, N-1\}$ before encryption. The clear-text output of the cyclic convolution can be recovered by multiplying the pre-processed encryptions, decrypting the result and applying an element-wise multiplication with $(-1)^{-l/N}$ for $l = \{0, \ldots, N-1\}$. It is worth noting that, in order for this scheme to be valid, $(-1)^{1/N}$ has to be an element of $\mathbb{Z}_t$, that is, we must be able to find a $2N$-th root of unity in $\mathbb{Z}_t$.

We present a modified version of this pre-/post-processing that transforms the homomorphism on bivariate negacylic convolutions into bivariate cyclic convolutions. Therefore, if we consider two 2-dimensional signals $w[l_x, l_y]$ and $h[l_x, l_y]$ of length N_x and N_y in each dimension (both powers of two), our method works as follows:

- First, we assume the existence of $2N_x$-th and $2N_y$-th roots of unity in $\mathbb{Z}_t$, denoted α_x and α_y (they can be efficiently found).
- We pre-process the signals before encrypting them:

$$w'[l_x, l_y] = w[l_x, l_y]\left(\alpha_x^{l_x} \otimes \alpha_y^{l_y}\right),$$
$$h'[l_x, l_y] = h[l_x, l_y]\left(\alpha_x^{l_x} \otimes \alpha_y^{l_y}\right),$$

  where $l_x = 0, \dots, N_x - 1$ and $l_y = 0, \dots, N_y - 1$.

- Analogously, as described in Chapter 4, we can compute $v'(x, y)$ under encryption with only one ciphertext product modulo the two functions $x^{N_x} + 1$ and $y^{N_y} + 1$:

$$v'(x, y) = \left(w'(x, y)h'(x, y) \bmod x^{N_x} + 1\right) \bmod y^{N_y} + 1.$$

- Finally, the decrypted signal $v'[l_x, l_y]$ is post-processed:

$$v[l_x, l_y] = v'[l_x, l_y]\left(\alpha_x^{-l_x} \otimes \alpha_y^{-l_y}\right).$$

This approach can be easily extended to the multivariate case. Therefore, considering m-dimensional signals (i.e., $h[l_1, \dots, l_m]$ where $l_i = 0, \dots, N_i - 1$) with a length of N_i (all of them powers of two) in each dimension, let α_i be $2N_i$-roots of unity for $i = 1, \dots, m$; the pre-processing and post-processing vectors are $\left(\bigotimes_{i=1}^{m} \alpha_i^{l_i}\right)$ and $\left(\bigotimes_{i=1}^{m} \alpha_i^{-l_i}\right)$ respectively.

Homomorphic Downsampling and Upsampling

This section addresses the implementation of downsampling/upsampling steps in the filter bank. For simplicity, we employ here univariate polynomials of n coefficients; we could extend this change of rate to the bivariate case by resorting to the Kronecker product, as done in previous sections. The structure of the filter bank (see Figure 7.1) requires a change in the sampling rate at each filter: (a) one downsampling by a factor of two after each analysis filter ($\boldsymbol{D}_i^{(z)}$), and (b) one upsampling by a factor of two before each synthesis filter ($\boldsymbol{U}_i^{(z)}$).

The required upsampling operation of a signal $x(z) \bmod z^n + 1$ represented as a polynomial can be seen as a scaling of the independent variable, $x(z^2) \bmod z^{2n} + 1$; conversely, the downsampling operation yields $x(z^{\frac{1}{2}}) \bmod z^{\frac{n}{2}} + 1$ by discarding the coefficients of the non integer exponents of z.

Hence, for a ciphertext $c = (c_0, c_1)$ with the corresponding $((c_0 + c_1 s) \bmod q) \bmod t$ decryption primitive, where s denotes the secret key, the new decryption circuit for the downsampling of

c is:

$$((c_0(z^{\frac{1}{2}}) + c_1^{(even)}(z)s^{(even)}(z) \\ +zc_1^{(odd)}(z)s^{(odd)}(z)) \bmod q) \bmod t,$$

where $c_0(z^{\frac{1}{2}})$ denotes the downsampling by a factor of two, and the upperscript denotes the phase (even or odd) of the polynomials.

Therefore, downsampling reduces the number of coefficients of the involved polynomials, but it also increases the number of polynomials of the ciphertexts. We reduce this expansion on the number of polynomial elements after each downsampling by resorting to a relinearization primitive (see Chapter 4 and Appendix B).

Interestingly, if our target were to reduce the cipher expansion of the ciphertexts (compressing the signal instead of denoising it), we could skip the relinearization primitive and leverage the encrypted wavelet transform to just discard the detail coefficients, approximating the signal with the (encrypted) approximation coefficients: we would have $\frac{3n}{2}$ coefficients modulo q instead of the $2n$ coefficients of a fresh ciphertext, hence reducing the expansion by a factor of $\frac{4}{3}$.

7.3.3. Homomorphic Threshold

After homomorphically computing the wavelet transform, the denoising scheme involves thresholding the encrypted transformed output. Previous approaches [6] to encrypted thresholding resort to the use of Paillier encryptions [14] and an interactive protocol between the secret key owner and the third party, as there is no efficient method proposed so far to deal with homomorphic thresholding and additions/multiplications at the same time. Conversely, our main objective is to reach an unattended solution without intervention of the secret key owner during the process.

Paillier cryptosystem cannot support additions and multiplications between two encrypted messages at the same time. This drawback is severe for our scenario, as our approach to the homomorphic computation of the threshold requires to homomorphically compute both encrypted additions and multiplications. Therefore, an m-RLWE based cryptosystem [4] also allows us to tackle this challenge, at the cost of additional issues derived from its peculiar polynomial structure, which we address in Section 7.3.3.

Our approach to a homomorphic thresholding block is the following: let $f(x)$ be a function, and consider that we have a set of different points $\{x_0, \ldots, x_l\}$ and their corresponding outputs $\{f(x_0), \ldots, f(x_l)\}$. Now, let us compute the smallest-degree polynomial $p(x) = \sum_{i=0}^{l} a_i x^i$ which satisfies $p(x_i) = f(x_i)$ for $i = 0, \ldots, l$, that is, we find the interpolating polynomial of $f(x)$ for a given set of $l + 1$ different points (we refer the reader to [149] for more details on polynomial interpolation).

The solution for polynomial coefficients a_i can be expressed in matrix form as:

$$\underbrace{\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^l \\ 1 & x_1 & x_1^2 & \ldots & x_1^l \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_l & x_l^2 & \cdots & x_l^l \end{bmatrix}}_{\boldsymbol{X}} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_l \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_l) \end{bmatrix},$$

where all the operations are carried out modulo-t (the plaintext domain); it can be easily seen that considering a prime t, $\boldsymbol{X}$ is a nonsingular Vandermonde matrix, whose determinant det(V) =

$\prod_{1\leq j<i\leq l}(x_i - x_j) \bmod t$ where

$$\boldsymbol{V} = \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^l \\ 1 & x_1 & x_1^2 & \ldots & x_1^l \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_l & x_l^2 & \cdots & x_l^l \end{bmatrix},$$

and the matrix $\boldsymbol{V}$ is clearly nonsingular because as all the x_i are different and t is prime (this implies that there are no zero divisors in $\mathbb{Z}_t$), the determinant det($\boldsymbol{X}$) is not zero. Therefore, the linear system has a unique solution for the coefficients a_i. This interpolating polynomial is typically computed resorting to its Lagrange form:

$$p(x) = \sum_{i=0}^{l} f(x_i)L_i(x) \bmod t, \tag{7.2}$$

where $L_j(x) = \prod_{i\neq j}(x - x_i)(x_j - x_i)^{-1} \bmod t$.

We leverage this interpolating polynomial $p(x)$ for a threshold computation as follows: (a) we consider a function $f(x)$ which encodes the desired threshold function for $x \in \mathbb{Z}_t$, and (b) we obtain the interpolating polynomial $p(x)$ for the required inputs.

The polynomial $p(x)$ describes one arithmetic circuit with several layers of additions and products over the same input x; thus, we can homomorphically compute the threshold if the chosen cryptosystem can perform both the addition and multiplication of two encrypted messages up to the depth of such circuit.

It is important to note that the proposed procedure is not limited to threshold functions; in fact, it can be analogously applied to general functions described by any $f(x)$. Additionally, the particular shape of $f(x)$ or the value of the corresponding threshold do not affect our contribution. Therefore, we assume that either the threshold or $f(x)$ are pre-defined in the clear, and we focus on how to homomorphically apply the threshold function as a circuit in the encrypted values.

Element-wise threshold

We resort to the use of the m-RLWE based cryptosystem [4] which, as explained in previous sections, allows us to efficiently perform the wavelet transform and to encrypt multidimensional signals. Its main advantage is enabling encrypted cyclic convolutions with only one ciphertext multiplication.

However, the threshold circuit has to be independently computed for each coefficient, so we need element-wise operations, which are not supported by the homomorphism. Consequently, the advantage of having the signal encoded with a polynomial structure becomes a problem for applying the threshold. To address this problem, we introduce an unattended homomorphic NTT (Number Theoretic Transform) [29] of the encrypted signal. The NTT has a convolution property (similar to that of the Fourier Transform), such that the convolutions in the transformed domain get translated into component-wise products in the original domain. We proceed as follows:

- Compute the homomorphic NTT of the encrypted signal.
- The encryped NTT of the signal is the input to the arithmetic threshold circuit.
- After the threshold circuit, we perform a homomorphic INTT.

As each ciphertext addition performs the addition of two NTTs and the ciphertext multiplication is equivalent to the cyclic convolution between two NTTs, we are homomorphically performing the element-wise multiplication between the values of the encrypted signal. Hence, when we consider the NTT of the encrypted signal as the input of the threshold circuit (see Eq. (7.2)), we are actually homomorphically computing the threshold for all the signal values.

Optimization for square images: In our proposed scheme, we perform a bidimensional NTT of the image. As the NTT is a separable transform, this can be easily realized by concatenating two homomorphic univariate NTTs (horizontal and vertical). For this purpose, a direct application of the methods proposed in [29] is not the optimal procedure, as they would be considering more relinearization matrices than needed. Therefore, we propose an optimization on the additional information required to perform the bivariate NTT for a square image (or in general, the multivariate NTT of any multidimensional signal with the same length in each dimension).

The general algorithm presented in [29] for performing our two NTTs (one for each dimension) would need one relinearization matrix for each NTT. However, when working with square images, it can be seen that one of the matrices can be replaced by a basic relinearization (see Chapter 4 and Appendix B), hence reducing in half the additional information with respect to the direct application of the original method in [29].

Our optimization reuses the relinearization matrix of one of the NTTs by performing two changes of variables $x \to y$ and $y \to x$. This procedure allows to apply the homomorphic NTT to the second variable, but it also introduces a change on the considered secret key, which now has its variables reversed. This problem can be solved with a basic relinearization for performing the switching key (see Chapter 4 and Appendix B) which has a size negligible compared to the original relinearization matrix.

Efficient computation of the threshold circuit

This section evaluates the computational cost of the threshold circuit and proposes methods for efficiently computing it in the encrypted domain.

In the worst case scenario, the maximum number of different points that our threshold circuit can have as input is t, which is the modulo considered for the plaintext (see Section 7.2). Therefore, we can find an interpolating polynomial whose maximum possible degree is $t - 1$.

It is also known that there exist algorithms for computing general polynomials of degree $t - 1$ with as many multiplications as the degree of the polynomial [150], for example, resorting to Horner's rule [151] we can easily compute a polynomial of degree $t - 1$ with $t - 1$ multiplications. However, dealing with a homomorphic cryptosystem brings about two important points:

- Horner's rule considers that all the multiplications have the same cost; hence, it does not take into account our special case dealing with a homomorphic cryptosystem, where multiplications between a ciphertext and a known scalar value are negligible with respect to the product between two encrypted values.

- Horner's rule does not take into account that a somewhat homomorphic cryptosystem bounds the number of allowed multiplications over the same encryped value "x" (in our case it is bounded by D; see Section 7.2).

Hence, in order to deal with these constraints, we resort to the algorithms for polynomial evaluation proposed by Paterson and Stockmeyer [152], which only count non-scalar multiplications, i.e., those multiplications involving the variable of the polynomial on both sides. Therefore, if we adapt their algorithms for bounding the number of multiplications over the same encrypted value, we can compute an arithmetic circuit of an l-degree polynomial with an order of $\mathcal{O}(\sqrt{l})$ non-scalar multiplications (ciphertext multiplications).

The smallest number of multiplications can be achieved with the algorithm C from [152], which has a computional cost equivalent to $\sqrt{2l} + \log_2 l + \mathcal{O}(1)$ ciphertext multiplications:

- It assumes $l = k2^{m-1}$. If this is not the case, we decompose l in smaller pieces of length $k2^{i-1}$, evaluate them separately and subsequently join them using the powers $\{x^{2k}, \ldots, x^{2k\lceil \log_2 \frac{l}{k} \rceil}\}$. This implies an additional cost of $\log_2 l/k$ multiplications.
- Compute the powers $\{x^2, x^3, \ldots, x^k\}$.
- Compute the powers $\{x^{2k}, x^{4k}, \ldots, x^{2^{m-1}k}\}$.
- After computing these powers, we can evaluate the polynomial with a total of $\sqrt{2l} + \log_2 l + \mathcal{O}(1)$ nonscalar multiplications if we consider $k \approx \sqrt{\frac{l}{2}}$.

7.4. Security and Performance Evaluation

This section evaluates both the performance and security of our proposed scheme. First, we briefly revisit and discuss some important concepts regarding the security of lattice-based cryptosystems. Afterwards, we show which are the changes that we can apply to the scheme so as to improve its efficiency when working in practical applications. Finally, we present the achieved runtimes together with the corresponding security parameters.

7.4.1. Security of Lattice Cryptosystems

All the proposed methods are noninteractive, and their security is entirely based on the semantic security of the used cryptosystem and the hardness assumptions on which it is grounded (m-RLWE problem). We can analyze the security of lattice-based cryptosystems by following the same procedures of prior works [4, 29, 79]. Hence, we focus on distinguishing attacks [109], which aim at breaking the indistinguishability assumption resorting to basis reduction algorithms.

In [46] we do not specifically deal with decoding attacks, which are aimed at obtaining the secret key, but we considered minimum values for $n = n_x n_y$ similar to those used in [79], this apparently gave us protection against the decoding attacks described in [110]. However, as we discussed in Chapter 5, when working with 2-RLWE with power-of-two cyclotomic modular functions the effective n is equal to $\max\{n_x, n_y\}$. This implies a considerable reduction on the bit security estimates initially claimed in [46]. With this in mind, in this section we update those bit security estimates and briefly discuss some possible alternatives to avoid this decrease on security.

Distinguishing attacks: The best attacks against lattice-based cryptosystems rely on basis reduction algorithms, being BKZ [111] one of the most efficient ones. The parameter which establishes the complexity of reduction atacks on the lattice is the root Hermite factor $\delta > 1$, such that for a constant k the runtime of an attack is approximately proportional to $e^{k/\log \delta}$ (see Chapter 4 for more details on how δ is obtained). So as to calculate the corresponding bit security (and be able to compare our chosen cryptosystem with other "traditional" cryptosystems), we resort to the accepted pessimistic lower bound estimate $t_{BKZ}(\delta)$ of [110] (see Eq. (4.3)).

7.4.2. Performance Evaluation

This section discusses some additional implementation challenges that can appear when realizing our proposed scheme in a practical scenario. We also bring about some approaches which can help to considerably improve the efficiency and cipher expansion of the proposed solutions for these practical situations. Additionally, we also include different runtimes together with the corresponding bit security (Eq. (4.3)) for several image sizes when performing our image denoising in the encrypted domain.

Practical considerations

Carefully looking at all the stages of our proposed encrypted image denoising process, it can be seen that the most costly operation is the element-wise threshold circuit, whose worst-case degree is highly dependent on the input cardinality.

For practical input images, their pixel values vary in range, therefore determining the degree of the threshold circuit, together with the corresponding computational cost for its execution.

In order to alleviate the computational cost of the threshold circuit, we can reduce the maximum value that the image coefficients can achieve as a result of the homomorphic wavelet transform.

Hence, for a practical implementation of our encrypted image denoising, we resort to the use of the Haar wavelet. Its use allows to easily analyze how the encrypted image coefficients increase after each stage, yielding a factor of 4^k after k stages. So, for a practical range for images like $[0, 255]$, by mapping $[0, 255] \rightarrow [-127, 128]$ before encryption, we have that the output of the k-stage belongs to the possible interval $4^k[-127, 128] = [-2^{7+2k} + 2^{2k}, 2^{7+2k}]$ for the coefficients. Now, we can take the number of values of this interval minus one as the considered maximum degree for the threshold circuit (in practical cases the degree of the interpolating polynomial would be much lower), therefore obtaining a clear improvement comparing with the case of using $t - 1$ as the maximum degree. Additionally, the structure of the Haar wavelet allows us to express the computational cost of the wavelet transform as very efficient additions among shifted polynomials.

Implementation and execution times

We have implemented the 2-RLWE cryptosystem in C++ using the GMP,[1] MPFR[2] and NFLlib [94] libraries. Table 7.1 compares the performance for encrypted image denoising for a range of four different sizes of images and for the two lower bounds on the bit security originally claimed in [46] (above 128 and above 256 bits of security), when running on an Intel Core-i5 2500 at 3.3 GHz using only one core (but the code is very amenable to parallelization). For all the cases, we consider a Haar wavelet and two stages for the filter-bank implementation. Additionally, the

[1] `www.gmplib.org`

[2] `www.mpfr.org`

range of values for the pixels is $[0, 255]$, mapped to $[-127, 128]$ before pre-processing the input images. The possible interval for the values of the (clear-text) coefficients at the input of the threshold circuit is $[-2032, 2048]$; hence, for preserving correctness in decryption, we consider $D = \lceil \log_2 4081 \rceil = 12$ for obtaining the bound on q (see Eq. (7.1)). This value for D yields a conservative pessimistic q, as the optimizations of [29] allow to consider ciphertext multiplications with polynomials of less than n coefficients. In any case, we take into account this fact for the estimation of δ and the calculation of the equivalent bit security. We report here the achieved performance when denoising is used as a standalone block, but it is possible to perform further homomorphic operations supported by the cryptosystem before or after the denoising, being the only requirement to increase D to account for the rest of the processes in the chain.

We include the corresponding runtimes for each of the operations in the pipeline: the pre-/post-processing together with encryption/decryption, and the homomorphic image denoising. Additionally, we have included the root Hermite factor δ, the bit security (see eq. (4.3)) for each scenario and the ratio in bits between the size of the encrypted image and the size of the image in clear (cipher expansion). For the given δ and bit-security we have included the updated estimates working with $\max\{n_x, n_y\}$ (see Chapter 5), which in this scenario means n equal to 128 and 256 for the, respectively, "mid-term" and "long-term" security. We also include the original estimates from [46] where we assume that there is no decrease in the lattice dimension, hence considering $n = n_x n_y$ in Equations (4.2) and (4.3).

Table 7.1: Performance of Image Denoising ($D = 12$, $A = 1$, $t = 65537$, $\sigma = 1$, 2 stages).

2-RLWE cryptosystem (claimed bit security $>$ 128 in [46])				
Image size	128×128	256×256	512×512	1024×1024
Cipher Exp. (ratio)	101.25	107.5	113.75	120
δ (assuming $n = n_x n_y$)	1.0043	1.0045	1.0048	1.0051
Bit security (Eq.(4.3), assuming $n = n_x n_y$)	≈ 182	≈ 165	≈ 150	≈ 136
δ ($n = \max\{n_x, n_y\}$)	1.7283	1.7878	1.8494	1.9131
Bit security (Eq.(4.3), $n = \max\{n_x, n_y\}$)	≈ -108	≈ -108	≈ -108	≈ -109
Encrypt. + Pre-proc. (ms)	9	41	199	939
Decrypt. + Post-proc. (ms)	10	42	211	1428
Enc. Denoising (*min*)	1.46	6.06	25.74	106.77

2-RLWE cryptosystem (claimed bit security $>$ 256 in [46])				
Image size	128×128	256×256	512×512	1024×1024
Cipher Exp. (ratio)	104.25	110.5	116.75	123
δ (assuming $n = n_x n_y$)	1.0022	1.0023	1.0025	1.0026
Bit security (Eq.(4.3), assuming $n = n_x n_y$)	≈ 456	≈ 424	≈ 396	≈ 370
δ ($n = \max\{n_x, n_y\}$)	1.3258	1.3485	1.3715	1.3949
Bit security (Eq.(4.3), $n = \max\{n_x, n_y\}$)	≈ -106	≈ -106	≈ -107	≈ -107
Encrypt. + Pre-proc. (ms)	19	97	417	1973
Decrypt. + Post-proc. (ms)	20	101	441	2998
Enc. Denoising (*min*)	4.26	17.85	76.49	316.69

Improvements on Security: It is clear in Table 7.1 that due to the Bootland *et al.*'s attack [44] the parameterization of 2-RLWE with power-of-two cyclotomic modular functions is not secure for the presented scenario. However, we can consider a set of modifications to increase the bit-security:

- We can add a slack variable h to increase the dimension of the underlying RLWE instatiation. To hold the bit-security estimates from [46], this slack variable must be at least $h = 128$. As a result, the basic polynomial operations increase roughly by a multiplicative factor of $h(1 + \log_2 h) = 1024$.

- We can adapt the 2-RLWE problem to those instantiations which do not suffer a decrease in its effective dimension. We explain in detail these solutions in Chapters 2 and 3. By using these secure m-RLWE instantiations, we do not need to incorporate a slack variable to guarantee a minimum level of security, and hence we can preserve the performance shown in Table 7.1.

The performance of the proposed methods shown in Table 7.1 proves the practicality of the scheme, requiring a few minutes (using just one core) to process an entire image of moderate size with a bit-security over 128 bits (mid-term security) if we consider a secure instantiation of 2-RLWE (see Chapter 2), and few milliseconds for encryption/decryption. The denoising runtime shows a quasi-linear behavior in terms of the image size, which is basically caused by the computational cost of the polynomial operations. This is much more efficient than using a comparison protocol with Paillier; e.g., [4] shows that for a basic filtering operation of a 1024×1024 image size, an RLWE-based solution provides runtimes 3 orders of magnitude faster than Paillier. Even a fully interactive secret sharing solution like [146] (which claims to be more efficient than a garbled-circuit based solution) needs over 16 minutes for a two-level denoising of a 128×128 image; considering a very favourable case with a communication cost of a LAN. For this case, our solution, besides not requiring any interaction, performs one order of magnitude faster.

7.5. Conclusions

This chapter proposes non-interactive methods based on 2-RLWE (Ring Learning with Errors) that overcome the limitations of previous Signal Processing in the Encrypted Domain solutions to efficiently perform encrypted image denoising. We have shown how to combine homomorphic polynomial operations and thresholding without involving decryption or interaction, therefore enabling fully unattended encrypted image denoising.

The performance of our proposed methods proves their practicality, improving on the usage of interactive comparison protocols with Paillier, and also comparing favorably with respect to fully interactive secret sharing solutions, even when we do not require any interaction. However, the proposed method is severely affected by the Bootland *et al.*'s attack, which has important consequences on its effective bit security. In this chapter we have briefly enumerated some possible solutions which enable to hold the originally claimed bit-security estimates (mid-term with 128 and long-term with more than 256 bits) without an impact of the efficiency. In Chapters 2 and 3 we propose and explain in detail how to deal with secure instantiations.

Chapter 8

Camera Attribution Forensic Analyzer

This chapter is adapted with permission from IEEE: Alberto Pedrouzo-Ulloa, Miguel Masciopinto, Juan Ramón Troncoso-Pastoriza, and Fernando Pérez-González. Camera Attribution Forensic Analyzer in the Encrypted Domain. The 10th IEEE International Workshop on Information Forensics and Security (WIFS18), December 2018.

8.1. Introduction

Digital media forensics is rapidly evolving as an answer to societal demands. Besides lively research topic, several commercial applications already exist that are able to (semi-) automatically detect forgeries and tampering, or identify and/or cluster acquisition devices. Although most of these tools have relatively low computational complexity, they must be run on very large and ever increasing databases, with efficiency thus becoming a major concern. On the other hand, the still growing popularity of content-sharing websites such as YouTube, Instagram or Facebook, and the Dark Web [153], leads to rapidly obsolescent forensic analysis platforms, especially in times of budget shortfalls, and quite conspicuously so in the case of law enforcement. An increasingly appealing solution is to buy computing power and database storage as needed, by running software and keeping data on outsourced platforms such as Amazon Web Services, Microsoft Azure or Google Cloud. This approach cuts down maintenance costs and dynamically scales with computing needs. However, outsourcing faces the problem of guaranteeing confidentiality and privacy at the server end, much more so considering that forensic data is highly sensitive.

One salient instance of extremely sensitive data is related to child pornography. Some of the existing tools for camera attribution or device clustering [3, 154, 155, 156] find an immediate application in fighting against crimes involving depictions of minors [157, 158, 159]. To get an estimate of the sheer size of this problem, researchers looked during a one-year period (2010-2011) at two of the then most common peer-to-peer networks, to find more than 2,500,000 peers worldwide sharing child pornography [160]. Obviously, processing this type of files outside of law enforcement's own infrastructure is currently out of the question; encryption alone is not a solution either, because contents must be opened at the server end in order to analyze them. This, of course, extends to storage: even camera fingerprints should be encrypted at all times as they may leak information that compromise an investigation.

Opportunely, recent advances in the field of Secure Signal Processing (SSP) [6] hint at a potential solution to cloudify forensic analysis software and forensic data storage in such a privacy-conscious way with zero information leakage. This means that the server does not even learn the outcome of a binary forensic test. Recently, some works have introduced new solutions based on lattice cryptography which are especially adapted to efficiently work with images, covering encrypted operations that range from image filtering [4] and image denoising [46] to more general image processing operations [5] (we have covered them in detail in Chapters 5 and 7, and also in Appendix B).

Most camera-attribution methods rely on the so-called Photoresponse Non-Uniformity (PRNU). The PRNU is a specific noise pattern inherent to digital imaging sensors which represents the difference in response of the sensor array to a uniform light source [161]. It is caused by random imperfections in the manufacturing process and it, due to its random nature, can be used as a fingerprint of the camera device, serving to determine whether a given test image was taken by a certain camera, by matching a residual obtained from the test image with the fingerprint. Due to its great potential for image forensics, many works have studied the use and properties of the PRNU, from the peculiarities of its mathematical modeling [162, 163] to a wide range of possible applications, including source attribution [164], source-based clustering [155], and tampering detection [165].

This chapter proposes a new framework for the secure outsourcing of PRNU-based source attribution (including secure PRNU extraction, detection and storage) in a fully unattended way, that is, without the intervention of the secret key owner during the process. To this end, since denoising is one of the main building blocks, we improve on the efficiency of the state-of-the-art in secure, unattended solutions for image denoising (namely, the solution introduced in Chapter 7), and we show how filtering, polynomial, denoising and pixel-wise operations (e.g. element-wise division) can be homomorphically performed in a single round without the need of an interactive protocol.

Main Contributions: To the best of our knowledge,[1] this is the first work in the literature that proposes a secure implementation of a forensic analyzer. The framework is here epitomized by a PRNU-based extractor/detector, but it embraces many other existing forensic tools. Other main contributions of this chapter are:

- Rooting in the secure wavelet-based denoising primitive presented in [46], we improve the results therein by means of a new threshold function. Our new procedure enables a considerable reduction in both the depth of the evaluated circuit and the number of effective ciphertext multiplications.
- We discuss the application of our novel homomorphic wavelet-based denoising primitive within a complex use case: PRNU extraction/detection for camera attribution.
- As such application requires many calls to the homomorphic wavelet denoising primitive, we show how to optimize its implementation. The resulting method is able to evaluate the full extraction/detection processes while avoiding execution-time interactions between client and server.

Structure: The rest of the chapter is organized as follows: Section 8.2 briefly revises the used lattice-based cryptosystems and the PRNU matching scenario. Section 8.3 introduces the main

[1] Thanks to the anonymous reviewers of WIFS2018, we were made aware of a related work by Mohanty et al. [136, 138]. It requires the use of a trusted environment (ARM TrustZone), while our approach can be fully implemented on a general purpose architecture.

scheme for secure PRNU extraction and detection, and Section 8.4 evaluates it in terms of security, efficiency and performance.

8.2. Preliminaries

This section summarizes the main operations performed in a PRNU-based extractor/detector and revisits the lattice-based cryptosystem used in our proposed scheme, highlighting its convenience for this scenario.

8.2.1. Basic structure of PRNU extraction/detection

The sensor output model can be approximated by the first two terms of its Taylor series [163], as

$$\boldsymbol{Y} = (\mathbf{1} + \boldsymbol{K}) \circ \boldsymbol{X} + \boldsymbol{N}, \tag{8.1}$$

where $\boldsymbol{Y}$ is the output matrix of the imaging sensor, $\boldsymbol{K}$ is the PRNU signal, $\mathbf{1}$ is a matrix filled with ones, $\boldsymbol{X}$ is the incident light intensity and $\boldsymbol{N}$ represents other noise sources.

It is worth noting that $\boldsymbol{X}$ is unknown in practice, but an estimate $\hat{\boldsymbol{X}}$ can be obtained with a denoising operation over $\boldsymbol{Y}$.

PRNU fingerprint extraction: Let $\{\boldsymbol{Y}^{(l)}\}_{l=1}^{M}$ be a set of M images taken with the same camera device of N_k pixels at native resolution. The PRNU can be estimated by using the maximum likelihood estimator (MLE) derived in [158]:

$$\hat{\boldsymbol{K}} = \left(\sum_{l=1}^{M} \boldsymbol{W}^{(l)} \circ \hat{\boldsymbol{X}}^{(l)}\right) \circ \left(\sum_{l=1}^{M} (\hat{\boldsymbol{X}}^{(l)})^{\circ 2}\right)^{\circ -1}, \tag{8.2}$$

where $\boldsymbol{W}^{(l)} = \boldsymbol{Y}^{(l)} - \hat{\boldsymbol{X}}^{(l)}$ is the denoising residue of the image $\boldsymbol{Y}^{(l)}$, and $\boldsymbol{A}^{\circ -1}$ (resp. $\boldsymbol{A}^{\circ 2}$) stands for the Hadamard inverse (resp. square) of matrix $\boldsymbol{A}$.

PRNU detection: Given a test image $\boldsymbol{Y}_t$ with residue $\boldsymbol{W}_t = \boldsymbol{Y}_t - \hat{\boldsymbol{X}}_t$ and a PRNU estimate $\hat{\boldsymbol{K}}$, the following hypothesis testing problem can be formulated:

H_0: $\boldsymbol{W}_t$ and $\hat{\boldsymbol{K}}$ correspond to different PRNUs.

H_1: $\boldsymbol{W}_t$ and $\hat{\boldsymbol{K}}$ correspond to the same PRNU.

As a computationally simpler alternative to the use of the Peak to Correlation Energy (PCE) statistic [3], here we consider

$$u = \boldsymbol{W}_t \cdot \hat{\boldsymbol{K}}, \tag{8.3}$$

for which an estimate of the variance is

$$\sigma_u^2 = \frac{1}{N_k}(\hat{\boldsymbol{K}} \cdot \hat{\boldsymbol{K}})(\boldsymbol{W}_t \cdot \boldsymbol{W}_t); \tag{8.4}$$

then, for a given probability of false alarm, the test becomes [158]

$$\frac{u}{\sigma_u} \underset{H_0}{\overset{H_1}{\gtrless}} \lambda, \tag{8.5}$$

where λ is a fixed threshold that changes depending on the desired false positive probability. In (8.3) we assume that the signals $\boldsymbol{W}_t$ and $\hat{\boldsymbol{K}}$ are aligned; otherwise, the maximum of the cross-correlation for every possible lag must be chosen as u in (8.5) [154].

8.2.2. A 2-RLWE based Cryptosystem

We use univariate and bivariate versions of the FV cryptosystem [86] as the underlying block for our secure forensic analyzer. Due to space constraints, we do not include here a description of all the cryptosystem primitives (we refer the reader to [86] for a detailed description). The plaintext elements belong to the ring $R_t[x, y]$ and ciphertexts are composed of two elements belonging to $R_q[x, y]$. When we work with bivariate polynomials instead of the usual univariate ones, security relies on the indistinguishability assumption of the 2-RLWE problem (see Definition 13).

The bivariate cryptosystem can encrypt images in only one ciphertext, instead of encrypting each pixel in a different ciphertext. It also enables efficient pixel-wise additions with one ciphertext addition and bivariate linear/cyclic convolutions with only one ciphertext multiplication at the cost of a small overhead (operations are performed over $\mathbb{Z}_q$ instead of $\mathbb{Z}_t$ with $q > t$). We refer the reader to [86, 4, 46] for further details on these homomorphic operations.

To evaluate an arithmetic circuit of multiplicative depth L, we can consider the following condition to have correct decryption (Theorem 1 in [86])

$$4n^L(n + 1.25)^{L+1}t^{L-1} < \left\lfloor \frac{q}{B} \right\rfloor, \tag{8.6}$$

where $n = n_x n_y$ and $||\chi|| < B$, that is, χ is a B-bounded distribution.

Some comments on the security of bivariate polynomials: Analogously to the case addressed in Chapter 7, when using rings with power-of-two cyclotomic modular functions, we have a reduction on the effective lattice dimension. If the security provided by the maximum univariate polynomial degree is not enough for the application, several modifications can be considered which enable to preserve the security. We elaborate more on the possible fixes in Section 8.4.

8.3. Proposed Scheme

This section describes the proposed scheme for securely evaluating the PRNU extractor/detector. First, we give a general overview of its structure with a brief description of each block. Afterwards, we focus on the secure image denoising block due to its importance for the PRNU extractor/detector. Finally, the two main tasks (PRNU extraction/detection) which form part of the scheme are discussed in more depth.

8.3.1. General Overview

We establish the following two working hypotheses for the proposed secure solution:

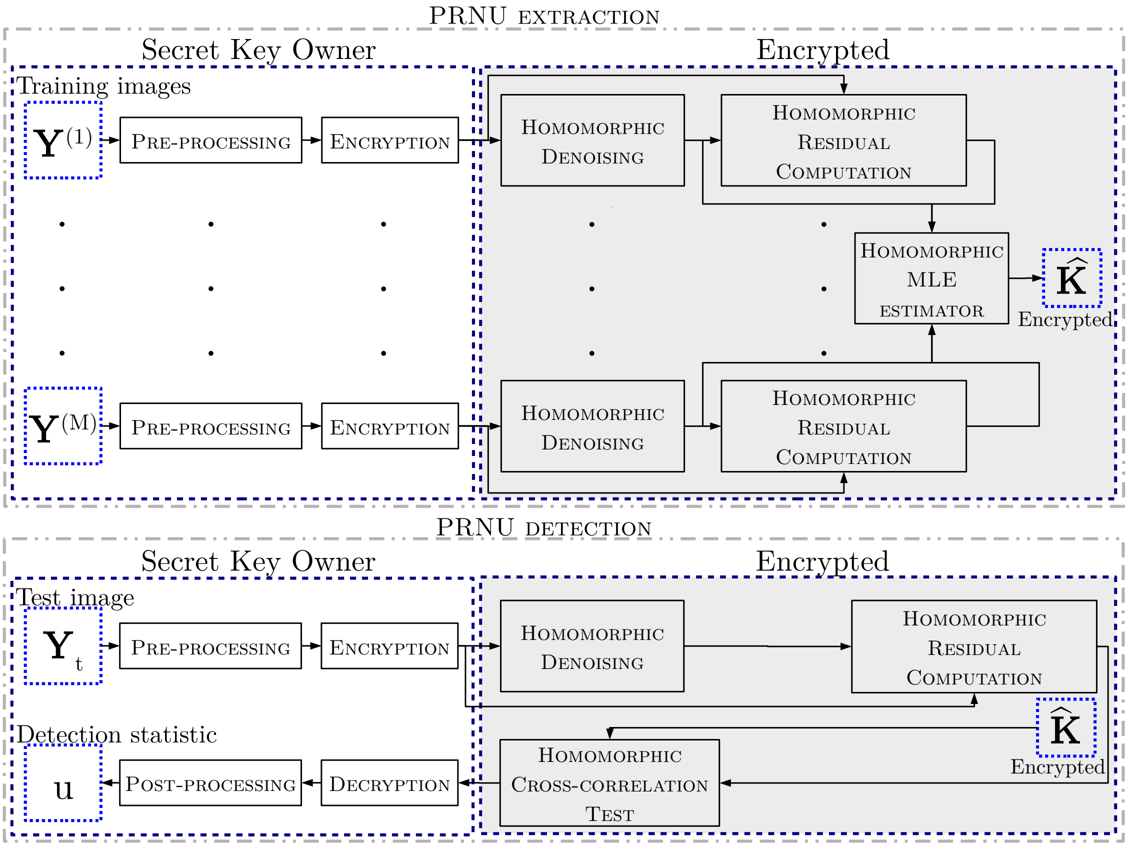


Figure 8.1: Secure scheme for the PRNU extractor/detector.

- The adversary model is based on a semi-honest setting, where the party who evaluates the encrypted PRNU extractor/detector tries to gather as much information of the content of the input images as possible, but does not deviate from the protocol.
- We require an unattended solution where the secret key owner does not have to participate in the middle of the process.

Taking into account these constraints, Figure 8.1 sketches the proposed scheme, which involves the two main attribution stages: (1) The extraction of the PRNU fingerprint given a training set of images from the same camera, and (2) the detection of the PRNU in an input image taking the previously extracted PRNU fingerprint as a template to be matched (see Section 8.2.1).

Our solution uses the RLWE and 2-RLWE versions [4] of the FV cryptosystem [86] as a means to perform encrypted arithmetic operations. We also make use of some of the techniques described in [46], such as (a) a lightweight pre-/post-processing (for homomorphic cyclic convolutions when multiplying two ciphertexts) and (b) the use of homomorphic NTT/INTTs (Direct/Inverse Number Theoretic Transforms) from [29] (for element-wise additions and multiplications between encrypted vectors).

Whereas the two main stages securely implement two different processes (represented by, respectively, (8.2) and (8.3)), both make use of an encrypted image denoising block. In fact, due to the high number of denoising operations, optimizing this common block is especially important for the efficiency of the whole pipeline.

In the following sections we explain in more detail the two main stages in Figure 8.1, including our optimizations over the state-of-the-art encrypted denoising block proposed in [46] (see

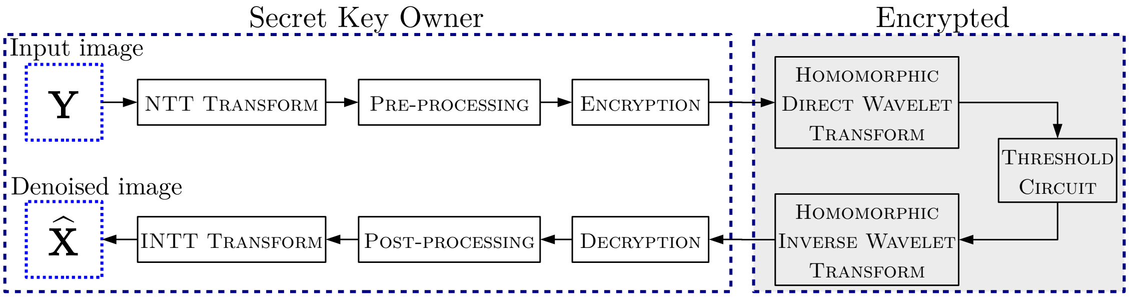


Figure 8.2: Encrypted Wavelet-based Denosing.

Chapter 7).

8.3.2. Encrypted Image Denoising

We consider as baseline the method for image denoising introduced in [46],[2] which comprises three elements: (1) homomorphic direct/inverse wavelet transform, (2) homomorphic NTT/INTT, and 3) threshold circuit. We considerably improve on the performance of this method by modifying the second and third elements.

Firstly, our solution moves the homomorphic NTT/INTT to the pre-/post-processing stage, avoiding its costly homomorphic computation and performing most of the operations in this batched setting. Figure 8.2 details the new structure of this primitive after substituting the homomorphic NTT/INTT block.

Regarding the last element, instead of directly applying a threshold function, we consider a quantization function which, in practice, works similarly to the hard threshold function from [46]. The advantage of this quantization is that it can be implemented by means of the "lowest digit removal" polynomials defined in [166, 30]. Their use allows for a smaller depth on the threshold circuit, hence considerably reducing the runtime of the primitive.

Homomorphic Wavelet Transform

We consider a filter-bank implementation for computing both the homomorphic direct and inverse wavelet transforms of the denoising algorithm. In [46] the authors introduce a light pre-/post-processing which enables the efficient application of low-/high-pass wavelets with cyclic convolutions by means of only one multiplication between a ciphertext and a plaintext encoding the corresponding wavelets.

After each homomorphic filtering operation, a downsampling or upsampling by a factor of 2 has to be applied depending on whether we work with the direct or the inverse transform. This downsampling/upsampling operation is very efficient, but it has to be followed by a costly relinearization.

In this chapter, we avoid these downsampling/upsampling steps (together with the relinearization) by previously dividing and separately encrypting the original image into as many polyphase components as required in the last level of the homomorphic wavelet transform (see [29]). Restricting the wavelet transform to Haar wavelets, their particular structure enables to express the

[2] This choice is mainly motivated by the widespread use of Wavelet denoising and its good tradeoff between cost and performance.

transform as very efficient additions among the polyphase components.[3]

Homomorphic Threshold

The approach considered in [46] for the homomorphic threshold (see Figure 8.2) directly interpolates the desired function (together with a normalization factor corresponding to the wavelet transform) over the plaintext. However, as the plaintext cardinality increases after each stage of the filter bank,[4] the complexity of the threshold circuit also increases. Hence, the results from [46] do not scale well when working with a high number of stages (in [46] the authors evaluate a denoising algorithm with only 2 stages).

This section introduces our quantization method to homomorphically evaluate both the normalization and the threshold. By choosing the plaintext modulo t as a prime power p^2 (where p is roughly equal to the number of possible input values for the images, e.g., $p = 257$), we can evaluate the quantization step with a polynomial whose maximum degree is equal to the cardinality of the plaintext before applying the wavelet transform, which considerably improves its performance with respect to the solution of Chapter 7.

This technique is based on the use of the "lowest digit removal" polynomials recently introduced in [166, 30] as a means to enhance the performance of bootstrapping for FHE (Fully Homomorphic Encryption) schemes. Here we leverage their properties for a different purpose: the homomorphic quantization of the plaintext.

We first present these polynomials (Lemma 3 from [30]) and how to construct them for our particular scenario:

Lemma 2 ([30])**.** *Let p be a prime and $e \geq 1$. Then there exists a polynomial f of degree at most $(e-1)(p-1)+1$ such that for every integer $0 \leq x < p^e$,*

$$f(x) \equiv (x - (x \bmod p)) \bmod p^e, \tag{8.7}$$

where $|x \bmod p| \leq \frac{p-1}{2}$ when p is odd.

For $e = 2$, $f(x) = -x(x-1)\dots(x-p+1)$ (Example 4 in [30]).

In our case, the quantization function which we want to evaluate is $\lfloor \frac{x}{Q} \rfloor$ for positive x and $\lceil \frac{x}{Q} \rceil$ for negative x. To have this functionality, and considering $e = 2$, we can define $f(x) = -(x + \frac{p-1}{2})\dots(x+1)x(x-1)\dots(x-\frac{p-1}{2})$, which implements the desired function for a quantization step $Q = p$. Once we have $f(x) \bmod p^e$ we can directly divide by p to have $\frac{f(x)}{p} \bmod p^{e-1}$. When working with the FV cryptosystem (see Section 8.2.2), after homomorphically evaluating $f(x)$, this division can be done for free, only introducing a slight increase in the ciphertext's noise (see [30]).

8.3.3. Homomorphic Cross-correlation Test

To securely perform the detection test, we have to homomorphically evaluate (8.3) (the general flow is depicted in Figure 8.1). After the encrypted denoising block (see Section 8.3.2), computing

[3] A total of $i4^i$ ciphertext additions for i levels, where each ciphertext encrypts a polynomial of size $\frac{n}{4^i}$ and n is the size of the original image.

[4] For example, considering a Haar wavelet the range of plaintext values is increased by a factor of 4 after each stage.

the residuals is straightforward by means of a homomorphic subtraction. Afterwards, as the test image may have been cropped, depending on whether it is aligned or not with the PRNU estimate (see Section 8.2.1), an encrypted scalar product or an encrypted cross-correlation operation is required.

Aligned case: To calculate the scalar product, we take advantage of the fact that the first coefficient of the NTT is the addition of all the coefficients in the time domain. Therefore, the server divides the encrypted PRNU into blocks and obtains the homomorphic NTT transform of each block, multiplies each PRNU block with the corresponding encrypted polyphase component of the residual, and finally adds all the encrypted polyphase components. This method encodes the scalar product in the first coefficient of the encrypted result. [5]

Non-aligned case: Here we want to calculate the full cross-correlation between the encrypted residuals and the reference PRNU. To do this, the client applies a pre-/post-processing over the plaintexts before/after encryption/decryption, and works with a cryptosystem based on the 2-RLWE problem. Then, the server can exploit the cyclic convolution property of the bivariate homomorphic INTT from [46] with the purpose of obtaining the time domain representation of the encrypted polyphase components (we refer the reader to [46] and Chapter 7 for details on this operation).

Once this is done, as the test image is encrypted in different blocks with a cyclic convolution property, the server can resort to the traditional "overlap-save" method [167] for calculating the linear convolution between the PRNU template and the encrypted polyphase components of the test image.

It must be noted that overlap-save discards part of the computed values, so the server has to generate enough space inside the ciphertexts. To achieve it, the server breaks the content of each encrypted polyphase component into four new ciphertexts before applying the homomorphic INTT, where each one has a quarter of the original polyphase component (for simplicity we consider that we are working with square images) and the rest are zero values. This increases the number of ciphertexts by a factor of 4, yielding a total of 4^{i+1} when working with an i-level wavelet denoising. The computational cost of the mentioned operation is equivalent to applying 4^{i+1} times an overlap-save algorithm over a filter encoded in the ciphertext and a PRNU 4^{i} times smaller.

Variance normalization: The statistic presented in (8.3) is normalized by $\sigma_w \sigma_k \sqrt{N_t}$ where $N_t \sigma_w^2 = \boldsymbol{W}_t \cdot \boldsymbol{W}_t$ and $N_k \sigma_k^2 = \hat{\boldsymbol{K}} \cdot \hat{\boldsymbol{K}}$ (N_t and N_k are the number of elements in $\boldsymbol{W}_t$ and $\hat{\boldsymbol{K}}$ respectively). For efficiency reasons, the server calculates the desired λ and returns the encrypted result of the scalar product or cross-correlation together with an encryption of λ scaled by this normalizing factor. The server could also homomorphically evaluate the division as we describe next for the PRNU extraction.

To compute these normalizing factors, the server can homomorphically evaluate the square of the residuals and PRNU, and add for both the polyphase components of their results. The desired values are stored in the first coefficient of the NTTs (see Chapter 6).

[5] For this scalar product we do not take advantage of the bivariate structure of the image, so we could consider an RLWE based-cryptosystem.

8.3.4. Secure PRNU extraction

The secure PRNU extraction involves the computation of (8.2) in an encrypted way. The encrypted denoising block for the input images and the pixel-wise operations on the encrypted image and residuals are analogous to those in (8.2), which are explained in Section 8.3.2, so we do not repeat them again here.

Finally, several strategies can be considered to implement the encrypted division needed to fully realize (8.2) under encryption; we briefly describe them here.

Approximate division: We can consider the methods for encrypted division used in [168, 31, 56], with which we can approximate the result of the division with a predefined bit precision.

For example, the server can approximate the inverse of a number b with 2^r bits of precision with the expression:

$$\rho^{-2^r} \prod_{j=0}^{r-1} \left(\rho^{2^j} + (\rho - b)^{2^j} \right) \text{ where } \frac{\rho}{2} \leq |b| \leq \frac{3\rho}{2}. \tag{8.8}$$

This approximation can be applied by adding an adequate value to the denoised images in (8.2) (for both numerator and denominator), such that all the pixels lie in the right range for convergence (for example, if $p = 257$ and pixel values are in the interval $[-128, 128]$, the server could add 256).

The server can also use a gradient descent algorithm (previously shifting the negative values to the positive side) as the Newton's root finding algorithm proposed in [168], where the inverse of a number b can be calculated through an iteration of the form

$$a_{i+1} = a_i(2\rho^{2^i} - ba_i), \tag{8.9}$$

with $b \in [0, 2^k]$ scaled by $\rho^{2^{\mu-1}}$ (that is, $(\rho^{2^{\mu-1}}/b)$), and being μ the number of iterations, $a_0 = 1$ and $\rho = 2^{k-1}$.

8.4. Security and Performance Evaluation

This section provides a complete evaluation of the proposed scheme, in terms of security, efficiency and performance.

For such evaluation, and due to space constraints, we assume that the client has control over the content of the images. This scenario could arise when the police have seized the camera of a suspect and wants to verify whether a certain image was taken from that camera, but due to legal constraints it cannot be directly outsourced in the clear. In this setting, we can safely assume that the client can gather a set of non-sensitive training images from the same camera (e.g., flatfield images); we can then perform the extraction in the clear. Once the PRNU has been extracted, we do consider that the test images may have a very high sensitive content, which requires the client to encrypt them before outsourcing.

Alternatives for PRNU extraction: As can be seen in (8.2), the extraction is more costly than detection due to its high number of denoising operations. However, we can consider other approaches more amenable to the allowed encrypted operations. For example, instead of separately denoising each image from the training set, we could previously add them and apply (8.2) to the resulting image. This computation is very similar to the PRNU detection, and the homomorphic addition of all the images can be done very efficiently with the used cryptosystem (see Section 8.2.2).

8.4.1. Security of the Proposed Scheme

The security of the proposed scheme is based on the semantic security of the used cryptosystem, which relies on the indistinguishability of the RLWE and 2-RLWE distributions (see Definition 13). In this chapter, we consider again distinguishing attacks [109] (although the considered values of n also resist the decoding attacks described in [110]), aimed at breaking the indistinguishability assumption through basis reduction algorithms (such as BKZ [111]). The runtime of basis reduction attacks is parameterized by the root Hermite factor $\delta > 1$ (for details on how to calculate δ see [79, 29]) as approximately $e^{k/\log \delta}$ with a constant k; hence, a lower δ gives higher attack runtimes. To estimate the bit security, we use the lower bound estimate[6] for BKZ $t_{BKZ}(\delta)$ given in [110] (see Equations (4.2) and (4.3)).

Bit security estimates (together with execution times) for our proposed scheme are included in Tables 8.1 and 8.2. As the encrypted image denoising algorithms from Table 8.1 are implemented by means of an RLWE-based cryptosystem, there is no decrease on the effective lattice dimension. Hence, this chapter shows how the denoising from Chapter 7 can be improved in terms of efficiency and security by only moving the NTT/INTT blocks and removing the need of a bivariate convolution. The same applies for the aligned detection in Table 8.2. However, for the non-aligned detection we make use of a 2-RLWE based cryptosystem for the homomorphic bivariate cross-correlation operation. We can see how the originally claimed security in [4] is much higher than the real obtained due to the Bootland *et al.*'s attack [44]. To solve this issue, we can consider several approaches:

- The proposed "packed"-RLWE in Chapter 5 allows to work with a bivariate convolution on RLWE.
- A slack variable h could be considered to fix a minimum level of security (see Chapter 7).
- We can adapt the results to secure multivariate RLWE instantiations as those discussed in Chapters 2 and 3. This solution allows to preserve roughly the same performance while not having a reduction on security.

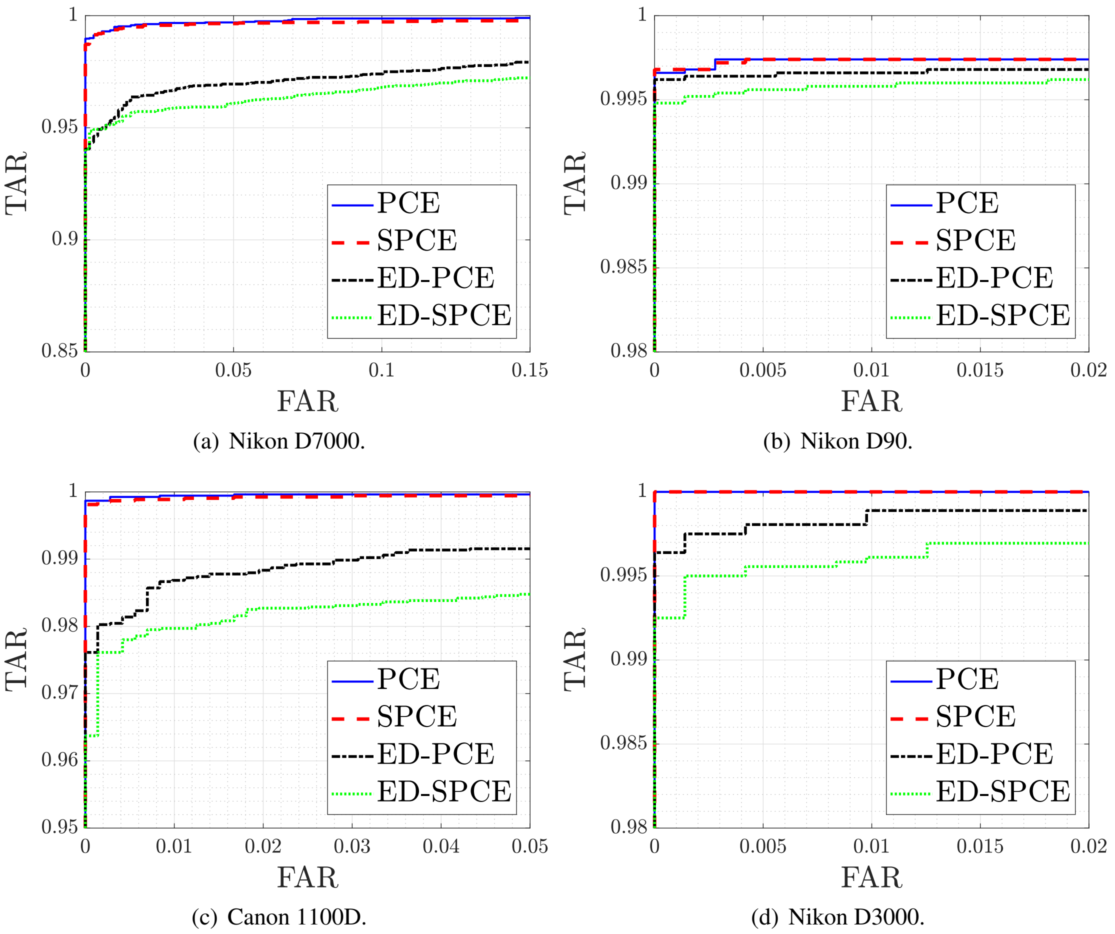
8.4.2. Implementation and execution times

We have implemented our scheme making use of the RNS variant of the FV cryptosystem [95].[7] Table 8.1 compares the runtimes of our proposed encrypted denoising (with the new threshold circuit) and the original algorithm from [46], which we already optimized by applying the NTT/INTT before/after the pre-/post-processing, to fairly compare the raw performance of the denoising primitive.

The runtimes substantially improve those from [46] (the improvement would be even more significant if we did not include our optimized NTT/INTT in our implementation of [46]). First,

[6] This estimate is more pessimistic than the security estimator recently developed by Albrecht et al. [80].

[7] Execution times were measured on an Intel Xeon E5-2667v3 at 3.2 GHz using one core for the non-parallelized option.



(a) Nikon D7000.

(b) Nikon D90.

(c) Canon 1100D.

(d) Nikon D3000.

Figure 8.3: True Acceptance Rate (TAR) vs. False Alarm Rate (FAR) for 4 different target camera devices. PCE represents the result obtained with the denoising in [2] and the PCE statistic [3], SPCE is the simplified detector in (8.5) applying the denoising in [2], ED-PCE is the PCE statistic using the encrypted image denoising described in Section 8.3.2, and ED-SPCE stands for the simplified detector discussed in Section 8.3.

we avoid the heavy homomorphic INTT/NTT computation. Secondly, the use of a new threshold function considerably reduces the ciphertext size and the depth of the evaluated circuit, resulting in a much faster computation.

Table 8.2 reports the runtimes for the detection scenario of our proposed scheme for PRNU extraction/detection. For efficiency reasons we separately compute the detection statistic u and the normalizing factor in two different ciphertexts (avoiding the costly encrypted division, which can be computed by the client as post-processing). The additional process of division does not add an important overhead to the secret key owner (see Table 8.2). Moreover, due to the highly parallelizable structure of the operations (they can be seamlessly parallelized even with a factor of 256), we include the runtimes considering different levels of parallelization.

Table 8.1: Performance of Encrypted Image Denoising ($\sigma = 8$).

Encrypted Denoising with RLWE cryptosystem (bit security > 110)				
Denoising with 2 stages	Optimized from [46]		Our denoising	
N (size image $N \times N$)	1024	2048	1024	2048
t	65537	65537	257^2	257^2
Cipher Exp. (ratio)	200.6250	210.0000	134.6250	134.6250
δ	1.00561	1.00294	1.00374	1.00374
Bit security (Eq.(4.3))	≈ 112	≈ 315	≈ 223	≈ 223
L (multiplicative depth)	12	12	8	8
Encrypt. + Pre-proc. (ms)	308.5	1333.4	211.2	844.9
Decrypt. + Post-proc. (ms)	591.4	2518.2	392.0	1568.2
Enc. Denoising (*min*)	17.42	74.21	2.79	11.19
Denoising with 3 stages	Optimized from [46]		Our denoising	
N (size image $N \times N$)	1024	2048	1024	2048
t	65537	65537	257^2	257^2
Cipher Exp. (ratio)	652.0000	326.0000	179.5000	179.5000
δ	1.00342	1.00342	1.00374	1.00374
Bit security (Eq.(4.3))	≈ 255	≈ 255	≈ 223	≈ 223
L (multipplicative depth)	14	14	8	8
Encrypt. + Pre-proc. (ms)	797.2	1594.4	211.2	844.9
Decrypt. + Post-proc. (ms)	2311.7	4623.5	588.1	2352.3
Enc. Denoising (*min*)	98.12	196.25	2.80	11.20

Table 8.2: Performance of Encrypted PRNU detection (2048 × 2048 image, PRNU of 16 Mpixels, $L = 11$, $t = 257^5$, $\sigma = 8$, denoising with 2 stages).

Aligned detection with RLWE cryptosystem (bit security > 128)				
Parallelization	1	8	16	20
Cipher Exp. (ratio)	379.95			
δ	1.00396			
Bit security (Eq.(4.3))	≈ 205			
Encrypt. + Pre-proc. (ms)	3642.62			
Decrypt. + Post-proc. (ms)	26.87			
Enc. Detection (*min*)	128.33	16.05	8.03	6.53
Non-aligned detection with 2-RLWE cryptosystem (bit security > 128)				
Parallelization	1	8	16	20
Cipher Exp. (ratio)	113.13			
δ (assuming $n = n_x n_y$)	1.00396			
Bit security (Eq.(4.3), assuming an effective $n = n_x n_y$)	≈ 205			
δ (assuming $n = \max\{n_x, n_y\}$)	1.65960			
Bit security (Eq.(4.3), effective $n = \max\{n_x, n_y\}$)	≈ -108			
Encrypt. + Pre-proc. (ms)	3642.62			
Decrypt. + Post-proc. (ms)	6878.50			
Enc. Detection (*min*)	1140.10	142.50	71.30	57.90

8.4.3. Performance of the PRNU extraction/detection

In order to evaluate the feasibility of the proposed scheme in terms of detection probabilities, we securely perform the PRNU detection test proposed in Sect. 8.2.1 as described in Sect. 8.3.3. To do so, we employed images from a database containing 2639 TIFF images from 16 digital single lens reflex camera devices [169, 170].

For a given target camera device, the fingerprint is extracted from $M = 50$ randomly chosen TIFF images, while crops of the JPEG-compressed version of the TIFF images with size 1536 × 1536 and quality factor 95 are considered for detection purposes. To test the H_1 hypothesis, after discarding the M images used for extraction, 20 different crops per image with random origins are considered on the images from the target camera, while H_0 hypothesis is tested by considering one crop per image for all images from each remaining camera device.

Figure 8.3 compares the performance of the detector in (8.3) with the Peak to Correlation Energy (PCE) state of the art detector [3], both when the widely used image denoising in [2] and when the proposed encrypted denoising filter with 2 stages (see Section 8.3) are used to obtain the residue of the test images. Notice that the denoising procedure from [2] is used for extraction ($\hat{\mathbf{K}}$ estimation) in all experiments, since the fingerprint is estimated in the clear.

The performance loss in the encrypted domain is mainly due to: (1) The simpler encrypted denoising algorithm, and (2) the simpler variance estimation on the detector in (8.3).

In spite of this slight loss in performance, the source attribution problem based on PRNU detection is feasible in the encrypted domain, achieving high true detection rates with low false alarm rates on JPEG test images, as shown in Fig. 8.3.

8.5. Conclusions

This chapter proposes a novel framework for secure outsourced camera attribution in a fully unattended way. As a fundamental block for both PRNU extraction and detection, we also present a new image denoising algorithm which improves the efficiency of the state of the art. Our solutions focus on unattended processing, where no interaction with the client is needed during the outsourced computation. We show that suboptimal choices can be more adequate for homomorphic operations. Finally, we also evaluate our proposed scheme in terms of security, efficiency and performance, showing the feasibility of secure camera attribution in the encrypted domain.

Chapter 9

A Discussion: Conclusions and Future Work

In this chapter we summarize the main contributions proposed in the thesis. To introduce our results we have followed a *bottom-up strategy*, starting from both the main definitions and the descriptions of the basic blocks, and ending up with a full list of examples of practical signal processing applications.

With this in mind, here we briefly discuss the implications of our solutions for privacy-preserving processing, paying special attention to the existing interdependences between the initially established high-level objectives and the most technical contributions of the thesis. Finally, we sketch out some of the possible future research lines which can be followed from now on.

From a high-level point of view, the *main contribution is twofold*: (1) we propose methods to securely *process multidimensional signals*, and (2) we present a general set of secure signal processing primitives which allow for *unattended secure processing*. We also show how to combine these two contributions so as to securely implement more complex multimedia applications in an unattended way.

In order to fulfill these two general points, we have studied in detail the particularities of secure signal processing primitives, mainly working on three different layers:

- **multivariate RLWE problem:** We have studied the grounds of lattice-based homomorphic cryptography with the purpose of adapting this building blocks to the specific requirements of signal processing applications. As a result, we proposed a *hard problem called multivariate RLWE* which fits to multidimensional signals better than its univariate counterpart. Interestingly, this hard problem is not only useful for signal processing, but also to *improve the efficiency of fundamental primitives in homomorphic cryptography* (e.g., polynomial operations, automorphisms, homomorphic slot manipulation, etc.).

  During the development of this thesis, an attack was presented which severely affects the security of multivariate RLWE. Consequently, we have *reevaluated the security of this problem*, and studied in detail how to parameterize it to have secure instantiations. These *secure instantiations maintain the efficiency properties originally claimed for multivariate RLWE*, while also avoiding a reduction on the effective lattice dimension.

- **A secure toolbox for unattended signal processing:** We have taken advantage of the polynomial structure of RLWE-based cryptosystems to implement *very efficient basic signal pro-*

cessing operations (e.g., generalized convolutions, linear transforms and matrix operations, among others). Our focus is on *unattended processing*, where we avoid the intervention of the secret key owner in the middle of the computation.

To do this, we have proposed *novel uses of NTTs* (Number Theoretic Transforms) and *relinearization techniques* combined with *very efficient pre-/post-coding operations* before/after encryption/decryption. It is worth noting that although these methods were originally proposed for our secure toolbox, *they can also be applied in the low-level cryptographic blocks to enhance the efficiency and define new operations*. This fact clearly showcases the interdependence between the different technical contributions and layers.

We have also proposed *an alternative solution to our secure parameterization of multivariate RLWE*. This solution mainly consists of a new pre-/post-coding operation which enables to transform homomorphic univariate into multivariate polynomial operations. Hence, *allowing us to work with multidimensional signals while working with univariate RLWE-based cryptosystems*.

- **Signal processing applications:** We *exemplify the proposed secure tools in several practical signal processing applications*. The set of possible applications is very wide, ranging from genomics to multimedia forensics. It is worth noting that during the development of this thesis, we have worked with more scenarios (e.g., some financial scenarios) than those exemplified here, but due to space restrictions we have decided to omit them. In any case, they are briefly mentioned in Chapter 1.

  Additionally, the described scenarios introduce *additional difficulties due to the nature of the involved signals*. Consequently, instead of directly applying our previous general solutions, we *take advantage of the signals' particular structure in each application to improve the results with respect to the standalone solution*. We briefly enumerate the exemplified applications: (1) we propose a secure genomic susceptibility testing protocol to a particular disease, (2) we propose block-processing operations and several transformations which allow for converting signals with different structures, (3) we propose a framework for secure and unattended image denoising, and finally, (4) we propose a camera forensic analyzer in the encrypted domain.

  Some of the solutions working with multidimensional signals were originally proposed in *insecure* multivariate RLWE instantiations (by insecure we mean that the effective lattice dimension is equal to the maximum of the univariate polynomial degrees). We also discuss how to readapt those results to preserve both the efficiency and security originally claimed.

9.1. Future Lines of Research

Even though the field of secure signal processing has rapidly evolved in the last years, it is still a very young discipline and there are many open problems. Many of them relate to the improvement of the underlying MPC techniques. In this thesis, where we focus on unattended secure processing between two parties, it is of fundamental importance the efficiency improvement of the underlying cryptograhic primitives like e.g. bootstrapping, which is a key block to have truly practical FHE. Additionally, from the point of view of secure signal processing, little work has been done on considering more realistic threat models in addition to passive security.

Next, we discuss these ones and more possible follow-up works which can be derived from the results obtained in this thesis.

Signal Processing Applications: Two of the main issues restricting the implementation of practical secure signal processing are those related to (1) the plaintext blow-up problem, (2) conversions of plaintext representation, and (3) the growth on cipher expansion (the ratio between ciphertext and cleartext size).

Many improvements have appeared regarding the two first problems, but they are still two of the most important obstacles to implement practical secure signal processing in an unattended way.

Blow-up problem: For the former, one easy solution is to resort to interactive protocols as a means to normalize the plaintext (we decrypt, normalize and reencrypt). However, this process introduces an additional communication overhead which was not our initial objective.

Plaintext representation: Efficient conversions between plaintext representation are especially important: we can choose either very efficient arithmetic operations or operations (efficient comparisons, maximum, minimum, etc.) more amenable to a a binary plaintext representation. We know how to very efficiently perform both but no with the same representation. Advances in this matter would greatly benefit encrypted computation.

Cipher expansion: In spite of the previous disadvantages, lattice-based cryptosytems (particularly those based on RLWE) already provide very efficient homomorphic operations, and considering most of the proposed optimizations in the literature they can be used for many practical scenarios. However, they can present a huge growth in cipher expansion. This introduces a high communication and storage overhead when working with them. As we have seen in some of the exemplified applications, with an adequate packing strategy this shortcoming can be considerably improved, but in general it is still very high. If the evaluated circuits do not have a high depth, the growth is not that important, but when increasing the depth of the evaluated circuits this defect starts to be a serious limitation.

A concrete example - Camera Attribution: Chapter 8 showcases how to implement a secure forensic analyzer based on exploiting the properties of the PRNU fingerprint. In this thesis, we have assumed that for training, the client can gather a set of non-sensitive images. We plan to extend these results, providing a complete evaluation of the PRNU extraction phase.

Another concrete example - Genomic Susceptibility Testing: For the genomic susceptibility testing protocol we assume a passive security threat model, where the SPU does not deviate from the protocol. A possible follow-up work could consider to extend these results to deal with malicious adversaries.

Cryptographic Building blocks: Some follow-up paths are also related to the different low-level cryptographic primitives of this thesis. We briefly summarize some options next.

Extending our results to secure multivariate RLWE instantiations: It is worth highlighting that some of the exemplified solutions are sketched out with negacyclic rings (mainly those from Section 3.3 in Chapter 3 and Chapters 7 and 8, and also Appendix B). Although we have briefly explained how this extension can be realized, a much more detailed analysis is required and we plan to extend these results to the more general multivariate rings showcased in Chapter 2.

Signal conversions and pre-/post-coding with any number field: In this thesis we have only exemplified the use of signal conversions and pre-/post-coding with power-of-two cyclotomics, but more general instances of multivariate RLWE could be greatly benefited of its use.

Joining the pieces: Although we have studied many of our contributions as independent pieces, most of them are perfectly compatible. The next step must be to study the combination of all of them.

Updates on the Threat Model: Finally, some possible lines of future work can focus on dealing with more realistic threat models:

Verifiable computation: In our two-party scenario composed of client and server, we can have passive security and even input privacy for a malicious server. However, there is no guarantee on the output correctness and hence we do not have active security. If both parties know the function to evaluate, a possible follow-up work is to use verifiable computation of encrypted data [171, 172, 173].

Circuit privacy: Our solutions do not provide circuit privacy; that is, when applying a function to an input, the evaluated ciphertext has to hide every information relative to the function except the output itself. This is an interesting property for those situations where the service provider may want to hide its propietary algorithm.

Conventional lattice-based homomorphic cryptosystems do not provide this property by default, as the homomorphic circuit is also applied to the additive ciphertexts' noise. Hence, at decryption, the noise presents some traces of the evaluated circuit.

There are several solutions in the literature: (1) Using bootstrapping (either by assuming circular security with the conventional approach which homomorphically evaluates the decryption circuit or by means of Garbled Circuits). (2) Adding superpolynomial noise (noise flooding) which hides the circuit information of the ciphertexts' noise. Finally, (3) some solutions are based on the GSW cryptosystem [174, 175]. An interesting follow-up work is to study the use of these techniques in several practical scenarios.

Bibliography

[1] D. Harvey, "Faster arithmetic for number-theoretic transforms," *J. Symb. Comput.*, vol. 60, pp. 113–119, 2014.

[2] M. K. Mihcak, I. Kozintsev, and K. Ramchandran, "Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising," in *IEEE ICASSP*, vol. 6, pp. 3253–3256, March 1999.

[3] M. Goljan, J. Fridrich, and T. Filler, "Large scale test of sensor fingerprint camera identification," in *Proc. SPIE, Electronic Imaging, Media Forensics and Security XI*, vol. 7254, pp. 0I 1-0I 12, Feb. 2009.

[4] A. Pedrouzo-Ulloa, J. Troncoso-Pastoriza, and F. Pérez-González, "Multivariate Lattices for Encrypted Image Processing," in *IEEE ICASSP*, pp. 1707–1711, 2015.

[5] A. Pedrouzo-Ulloa, J. R. Troncoso-Pastoriza, and F. Pérez-González, "Multivariate Cryptosystems for Secure Processing of Multidimensional Signals," *CoRR*, vol. abs/1712.00848, 2017.

[6] R. L. Lagendijk, Z. Erkin, and M. Barni, "Encrypted Signal Processing for Privacy Protection," *IEEE SP Mag.*, vol. 30, no. 1, pp. 82–105, 2013.

[7] A. C. Yao, "Protocols for secure computations (extended abstract)," in *IEEE FOCS*, pp. 160–164, 1982.

[8] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game or A completeness theorem for protocols with honest majority," in *ACM STOC*, pp. 218–229, 1987.

[9] J. Kilian, "Founding Cryptography on Oblivious Transfer," in *ACM STOC*, pp. 20–31, 1988.

[10] A. Shamir, "How to Share a Secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[11] D. W. Archer, D. Bogdanov, Y. Lindell, L. Kamm, K. Nielsen, J. I. Pagter, N. P. Smart, and R. N. Wright, "From Keys to Databases - Real-World Applications of Secure Multi-Party Computation," *Comput. J.*, vol. 61, no. 12, pp. 1749–1771, 2018.

[12] I. Damgård, M. Keller, E. Larraia, V. Pastro, P. Scholl, and N. P. Smart, "Practical Covertly Secure MPC for Dishonest Majority - Or: Breaking the SPDZ Limits," in *ESORICS*, pp. 1–18, 2013.

[13] D. Beaver, "Efficient Multiparty Protocols Using Circuit Randomization," in *CRYPTO*, vol. 576 of *LNCS*, pp. 420–432, 1991.

[14] P. Paillier, "Public-key Cryptosystems Based on Composite Degree Residuosity Classes," in *EUROCRYPT*, vol. 1716 of *LNCS*, pp. 223–238, Springer, 1999.

[15] T. E. Gamal, "A Public Key Cryptosystem and a Signature Scheme based on Discrete Logarithms," *IEEE Trans. Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[16] Y. Lindell, "How to Simulate It - A Tutorial on the Simulation Proof Technique," in *Tutorials on the Foundations of Cryptography.*, pp. 277–346, 2017.

[17] T. Bianchi, A. Piva, and M. Barni, "On the Implementation of the Discrete Fourier Transform in the Encrypted Domain," *IEEE Trans. on Information Forensics and Security*, vol. 4, pp. 86–97, March 2009.

[18] P. Zheng and J. Huang, "Efficient Encrypted Images Filtering and Transform Coding with Walsh-Hadamard Transform and Parallelization," *IEEE Trans. Image Processing*, vol. 27, no. 5, pp. 2541–2556, 2018.

[19] J. R. Troncoso-Pastoriza and F. Pérez-González, "Secure Adaptive Filtering," *IEEE Trans. on Information Forensics and Security*, vol. 6, pp. 469–485, June 2011.

[20] J. Troncoso-Pastoriza, S. Katzenbeisser, M. Celik, and A. Lemma, "A Secure Multidimensional Point Inclusion Protocol," in *ACM MM&Sec*, pp. 109–120, 2007.

[21] T. Bianchi, A. Piva, and M. Barni, "Composite Signal Representation for Fast and Storage-Efficient Processing of Encrypted Signals," *IEEE Trans. on Information Forensics and Security*, vol. 5, pp. 180–187, March 2010.

[22] A. Pedrouzo-Ulloa, J. R. Troncoso-Pastoriza, and F. Pérez-González, "On Ring Learning with Errors over the Tensor Product of Number Fields," *CoRR*, vol. abs/1607.05244, 2016.

[23] A. Pedrouzo-Ulloa, J. R. Troncoso-Pastoriza, and F. Pérez-González, "Revisiting Multivariate Lattices for Encrypted Signal Processing," in *ACM IH&MMSec*, 2019.

[24] C. Aguilar-Melchor, J. Barrier, L. Fousse, and M.-O. Killijian, "XPIR: Private Information Retrieval for Everyone," *PoPETs*, vol. 2016, no. 2, pp. 155–174, 2016.

[25] J. Troncoso-Pastoriza, D. Gonzalez-Jimenez, and F. Perez-Gonzalez, "Fully Private Non-interactive Face Verification," *IEEE Trans. on Information Forensics and Security*, vol. 8, pp. 1101–1114, July 2013.

[26] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. E. Lauter, M. Naehrig, and J. Wernsing, "CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy," in *ICML*, pp. 201–210, 2016.

[27] C. Aguilar-Melchor, M. Killijian, C. Lefebvre, and T. Ricosset, "A Comparison of the Homomorphic Encryption Libraries HElib, SEAL and FV-NFLlib," in *SecITC*, pp. 425–442, 2018.

[28] J. H. Cheon, D. Kim, Y. Kim, and Y. Song, "Ensemble Method for Privacy-Preserving Logistic Regression Based on Homomorphic Encryption," *IEEE Access*, vol. 6, pp. 46938–46948, 2018.

[29] A. Pedrouzo-Ulloa, J. R. Troncoso-Pastoriza, and F. Pérez-González, "Number theoretic transforms for secure signal processing," *IEEE Trans. on Information Forensics and Security*, vol. 12, pp. 1125–1140, May 2017.

[30] H. Chen and K. Han, "Homomorphic lower digits removal and improved FHE bootstrapping," in *EUROCRYPT*, vol. 10820 of *LNCS*, pp. 315–337, 2018.

[31] J. H. Cheon, A. Kim, M. Kim, and Y. S. Song, “Homomorphic Encryption for Arithmetic of Approximate Numbers,” in *ASIACRYPT*, vol. 10624 of *LNCS*, pp. 409–437, 2017.

[32] J. H. Cheon, K. Han, A. Kim, M. Kim, and Y. Song, “Bootstrapping for approximate homomorphic encryption,” in *EUROCRYPT*, vol. 10820 of *LNCS*, pp. 360–384, 2018.

[33] C. Gentry, “Fully homomorphic encryption using ideal lattices,” in *ACM STOC*, pp. 169–178, 2009.

[34] C. Gentry, *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. `crypto.stanford.edu/craig`.

[35] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, “Faster Fully Homomorphic Encryption: Bootstrapping in Less Than 0.1 Seconds,” in *ASIACRYPT*, vol. 10031 of *LNCS*, pp. 3–33, 2016.

[36] D. J. Bernstein, J. Buchmann, and E. Dahmen, *Post Quantum Cryptography*. Springer Publishing Company, 1st ed., 2008.

[37] P. W. Shor, “Algorithms for quantum computation: Discrete logarithms and factoring,” in *IEEE FOCS*, pp. 124–134, 1994.

[38] O. Regev, “On lattices, learning with errors, random linear codes, and cryptography,” in *ACM STOC*, pp. 84–93, 2005.

[39] O. Regev, “On Lattices, Learning with Errors, Random Linear Codes, and Cryptography,” *J. ACM*, vol. 56, pp. 34:1–34:40, Sept. 2009.

[40] V. Lyubashevsky, C. Peikert, and O. Regev, “On Ideal Lattices and Learning with Errors over Rings,” in *EUROCRYPT*, vol. 6110 of *LNCS*, pp. 1–23, Springer, 2010.

[41] V. Lyubashevsky, C. Peikert, and O. Regev, “On Ideal Lattices and Learning with Errors over Rings,” *J. ACM*, vol. 60, pp. 43:1–43:35, Nov. 2013.

[42] C. Peikert, “Lattice Cryptography for the Internet.” Crypt. ePrint Archive, Report 2014/070, 2014.

[43] L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky, “Lattice Signatures and Bimodal Gaussians.” Crypt. ePrint Archive, Report 2013/383, 2013.

[44] C. Bootland, W. Castryck, and F. Vercauteren, “On the security of the multivariate ring learning with errors problem,” *Crypt. ePrint Archive*, vol. 2018, p. 966, 2018.

[45] V. Lyubashevsky, C. Peikert, and O. Regev, “A Toolkit for Ring-LWE Cryptography,” in *EUROCRYPT*, vol. 7881 of *LNCS*, pp. 35–54, Springer, 2013.

[46] A. Pedrouzo-Ulloa, J. R. Troncoso-Pastoriza, and F. Pérez-González, “Image denoising in the encrypted domain,” in *IEEE WIFS*, pp. 1–6, Dec 2016.

[47] I. Haviv and O. Regev, “Tensor-based Hardness of the Shortest Vector Problem to within Almost Polynomial Factors,” *Theory of Computing*, vol. 8, no. 1, pp. 513–531, 2012.

[48] A. Langlois and D. Stehlé, “Worst-case to average-case reductions for module lattices,” *Des. Codes Cryptography*, vol. 75, no. 3, pp. 565–599, 2015.

[49] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, “(leveled) fully homomorphic encryption without bootstrapping,” in *ITCS*, pp. 309–325, 2012.

[50] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, “(Leveled) Fully Homomorphic Encryption without Bootstrapping,” *ACM Trans. Comput. Theory*, vol. 6, pp. 13:1–13:36, July 2014.

[51] G. Bonnoron, L. Ducas, and M. Fillinger, “Large FHE gates from tensored homomorphic accumulator,” in *AFRICACRYPT*, vol. 10831 of *LNCS*, pp. 217–251, 2018.

[52] D. Micciancio and J. Sorrell, “Ring Packing and Amortized FHEW Bootstrapping,” in *ICALP*, pp. 100:1–100:14, 2018.

[53] C. Peikert, O. Regev, and N. Stephens-Davidowitz, “Pseudorandomness of ring-lwe for any ring and modulus,” in *ACM STOC*, pp. 461–473, 2017.

[54] H. Murakami, “Generalization of the cyclic convolution system and its applications,” in *IEEE ICASSP*, vol. 6, pp. 3351–3353, 2000.

[55] C. Peikert, “How (Not) to Instantiate Ring-LWE,” in *SCN*, pp. 411–430, 2016.

[56] J. H. Cheon and A. Kim, “Homomorphic Encryption for Approximate Matrix Arithmetic.” Crypt. ePrint Archive, Report 2018/565, 2018.

[57] J. H. Cheon, A. Kim, and D. Yhee, “Multi-dimensional packing for HEAAN for approximate matrix arithmetics,” *IACR Cryptology ePrint Archive, Report 2018/1245*, 2018.

[58] B. Applebaum, D. Cash, C. Peikert, and A. Sahai, “Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems,” in *CRYPTO*, vol. 5677 of *LNCS*, pp. 595–618, Springer, 2009.

[59] S. Halevi and V. Shoup, “Algorithms in helib,” in *CRYPTO*, vol. 8616 of *LNCS*, pp. 554–571, 2014.

[60] S. Halevi and V. Shoup, “Bootstrapping for helib,” in *EUROCRYPT*, vol. 9056 of *LNCS*, pp. 641–670, 2015.

[61] V. Lyubashevsky and D. Micciancio, “Generalized compact knapsacks are collision resistant,” in *ICALP*, pp. 144–155, 2006.

[62] T. Weston, “Algebraic Number Theory.” `https://www.math.wisc.edu/~mmwood/748Fall2016/weston.pdf`. Accessed: 11 March 2019.

[63] E. of Mathematics, “Compositum.” `https://www.encyclopediaofmath.org/index.php/Compositum`. Accessed: 11 March 2019.

[64] B. Conrad, “Math 154: Discrimant of Composite Fields.” `http://math.stanford.edu/~conrad/154Page/handouts/disccomposite.pdf`. Accessed: 11 March 2019.

[65] K. Conrad, “The Different Ideal.” `https://kconrad.math.uconn.edu/blurbs/gradnumthy/different.pdf`. Accessed: 11 March 2019.

[66] Z. Brakerski and R. Perlman, “Order-lwe and the hardness of ring-lwe with entropic secrets,” *Crypt. ePrint Archive, Report 2018/494*, 2018.

[67] M. Barile, "Eisenstein's Irreducibility Criterion." From MathWorld, A Wolfram Web Resource, created by Eric. W. Weisstein, `http://mathworld.wolfram.com/EisensteinsIrreducibilityCriterion.html`. Accessed: 11 March 2019.

[68] P. Samuel, *Algebraic Theory of Numbers*. Dover Publications, 2008.

[69] B. Conrad and A. Landesman, "Math 154: Algebraic Number Theory." `http://math.stanford.edu/~conrad/154Page/handouts/undergraduate-number-theory.pdf`. Accessed: 11 March 2019.

[70] Y. Elias, K. E. Lauter, E. Ozman, and K. E. Stange, "Provably weak instances of ring-lwe," in *CRYPTO*, vol. 9215 of *LNCS*, pp. 63–92, 2015.

[71] K. S. Kedlaya, "A construction of polynomials with squarefree discriminants," *CoRR*, vol. abs/1103.5728, 2011.

[72] W. Castryck, I. Iliashenko, and F. Vercauteren, "Provably weak instances of ring-lwe revisited," in *EUROCRYPT*, vol. 9665 of *LNCS*, pp. 147–167, 2016.

[73] K. Eisenträger, S. Hallgren, and K. E. Lauter, "Weak instances of PLWE," in *SAC*, pp. 183–194, 2014.

[74] K. Laine and K. E. Lauter, "Key recovery for LWE in polynomial time," *Crypt. ePrint Archive, Report 2015/176*, 2015.

[75] H. Chen, K. E. Lauter, and K. E. Stange, "Vulnerable Galois RLWE Families and Improved Attacks," *Crypt. ePrint Archive, Report 2016/193*, 2016.

[76] H. Chen, K. E. Lauter, and K. E. Stange, "Attacks on the search-rlwe problem with small errors," *CoRR*, vol. abs/1710.03739, 2017.

[77] H. Chen, K. E. Lauter, and K. E. Stange, "Security considerations for galois non-dual RLWE families," *CoRR*, vol. abs/1710.03316, 2017.

[78] Z. Brakerski and V. Vaikuntanathan, "Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages," in *CRYPTO*, vol. 6841 of *LNCS*, Springer, 2011.

[79] K. Lauter, M. Naehrig, and V. Vaikuntanathan, "Can homomorphic encryption be practical?," in *ACM CCSW*, pp. 113–124, 2011.

[80] M. R. Albrecht, R. Player, and S. Scott, "On the concrete hardness of Learning with Errors," *J. Mathematical Cryptology*, vol. 9, no. 3, pp. 169–203, 2015.

[81] M. R. Albrecht, B. R. Curtis, A. Deo, A. Davidson, R. Player, E. W. Postlethwaite, F. Virdia, and T. Wunderer, "Estimate All the {LWE, NTRU} Schemes!," in *SCN*, pp. 351–367, 2018.

[82] M. Chase, H. Chen, J. Ding, S. Goldwasser, S. Gorbunov, J. Hoffstein, K. Lauter, S. Lokam, D. Moody, T. Morrison, A. Sahai, and V. Vaikuntanathan, "Security of homomorphic encryption," tech. rep., HomomorphicEncryption.org, Redmond WA, July 2017.

[83] A. Costache and N. P. Smart, "Which ring based somewhat homomorphic encryption scheme is best?," in *CT-RSA*, pp. 325–340, 2016.

[84] A. Lopez-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-Fly Multiparty Computation on the Cloud via Multikey Fully Homomorphic Encryption." Crypt. ePrint Archive, Report 2013/094, 2013.

[85] Z. Brakerski, "Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP," in *CRYPTO* (R. Safavi-Naini and R. Canetti, eds.), vol. 7417 of *LNCS*, pp. 868–886, Springer, 2012.

[86] J. Fan and F. Vercauteren, "Somewhat Practical Fully Homomorphic Encryption." Crypt. ePrint Archive, Report 2012/144, 2012.

[87] J. Bos, K. Lauter, J. Loftus, and M. Naehrig, "Improved Security for a Ring-Based Fully Homomorphic Encryption Scheme," in *Cryptography and Coding* (M. Stam, ed.), vol. 8308 of *LNCS*, pp. 45–64, Springer, 2013.

[88] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," in *IEEE FOCS*, pp. 97–106, 2011.

[89] L. Ducas and D. Micciancio, "FHEW: bootstrapping homomorphic encryption in less than a second," in *EUROCRYPT*, vol. 9056 of *LNCS*, pp. 617–640, 2015.

[90] J. Biasse and L. Ruiz, "FHEW with Efficient Multibit Bootstrapping," in *LATINCRYPT*, vol. 9230 of *LNCS*, pp. 119–135, 2015.

[91] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, "Faster Packed Homomorphic Operations and Efficient Circuit Bootstrapping for TFHE," in *ASIACRYPT*, vol. 10624 of *LNCS*, pp. 377–408, 2017.

[92] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, "TFHE: Fast Fully Homomorphic Encryption over the Torus." Crypt. ePrint Archive, Report 2018/421, 2018.

[93] D. Rathee, P. K. Mishra, and M. Yasuda, "Faster PCA and Linear Regression through Hypercubes in HElib." Crypt. ePrint Archive, Report 2018/801, 2018.

[94] C. Aguilar-Melchor, J. Barrier, S. Guelton, A. Guinet, M.-O. Killijian, and T. Lepoint, "NFLlib: NTT-based fast lattice library," in *CT-RSA*, pp. 341–356, Springer, 2016.

[95] J. Bajard, J. Eynard, M. A. Hasan, and V. Zucca, "A Full RNS Variant of FV Like Somewhat Homomorphic Encryption Schemes," in *SAC*, pp. 423–442, 2016.

[96] S. Halevi, Y. Polyakov, and V. Shoup, "An improved RNS variant of the BFV homomorphic encryption scheme," *Crypt. ePrint Archive, Report 2018/117*, 2018.

[97] A. A. Badawi, Y. Polyakov, K. M. M. Aung, B. Veeravalli, and K. Rohloff, "Implementation and Performance Evaluation of RNS Variants of the BFV Homomorphic Encryption Scheme." Crypt. ePrint Archive, Report 2018/589, 2018.

[98] N. P. Smart and F. Vercauteren, "Fully homomorphic SIMD operations," *Des. Codes Cryptography*, vol. 71, no. 1, pp. 57–81, 2014.

[99] S. Halevi and V. Shoup, "Faster homomorphic linear transformations in helib." Crypt. ePrint Archive, Report 2018/244, 2018.

[100] J. Hoffstein, J. Pipher, and J. H. Silverman, "NTRU: A Ring-Based Public Key Cryptosystem," in *ANTS-III*, pp. 267–288, 1998.

[101] D. J. Bernstein, C. Chuengsatiansup, T. Lange, and C. van Vredendaal, "NTRU prime: Reducing attack surface at low cost," in *SAC*, pp. 235–260, 2017.

[102] B. J. Fino and V. R. Algazi, "Unified Matrix Treatment of the Fast Walsh-Hadamard Transform," *IEEE Trans. on Computers*, vol. C-25, pp. 1142–1146, Nov 1976.

[103] R. K. R. Yarlagadda and J. E. Hershey, *Hadamard Matrix Analysis and Synthesis: With Applications to Communications and Signal/Image Processing*. Norwell, MA, USA: Kluwer Academic Publishers, 1997.

[104] H. J. Nussbaumer, *Fast Fourier Transform and Convolution Algorithms*. Springer, 1982.

[105] C. Gentry, S. Halevi, and N. P. Smart, "Fully homomorphic encryption with polylog overhead." Crypt. ePrint Archive, Report 2011/566, 2011.

[106] N. Dowlin, R. Gilad-Bachrach, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy," in *ICML*, pp. 201–210, 2016.

[107] L. Ducas and D. Micciancio, "FHEW: Bootstrapping Homomorphic Encryption in less than a second," in *EUROCRYPT*, vol. 9056 of *LNCS*, pp. 617–640, Springer, 2015.

[108] L. D. Martin Albrecht, Shi Bai, "A subfield lattice attack on overstretched NTRU assumptions: Cryptanalysis of some FHE and Graded Encoding Schemes." Crypt. ePrint Archive, Report 2016/127, 2016.

[109] D. Micciancio and O. Regev, "Lattice-based Cryptography," in *Post-Quantum Cryptography*, pp. 147–191, Springer, 2009.

[110] R. Lindner and C. Peikert, "Better Key Sizes (and Attacks) for LWE-based Encryption," in *CT-RSA*, pp. 319–339, Springer, 2011.

[111] Y. Chen and P. Nguyen, "BKZ 2.0: Better Lattice Security Estimates," in *ASIACRYPT*, vol. 7073 of *LNCS*, pp. 1–20, Springer, 2011.

[112] "Recommendation for Key Management, Part 1: General." `http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf`.

[113] A. Aysu, C. Patterson, and P. Schaumont, "Low-Cost and Area-Efficient FPGA Implementations of Lattice-Based Cryptography," in *HOST*, pp. 81–86, 2013.

[114] T. Pöppelmann and T. Güneysu, "Towards Practical Lattice-Based Public-Key Encryption on Reconfigurable Hardware," in *SAC 2013*, LNCS, pp. 68–85, Springer, 2014.

[115] S. S. Roy, F. Vercauteren, N. Mentens, D. D. Chen, and I. Verbauwhede, "Compact Ring-LWE Cryptoprocessor," in *CHES*, vol. 8731 of *LNCS*, pp. 371–391, Springer, 2014.

[116] D. D. Chen, N. Mentens, F. Vercauteren, S. S. Roy, R. C. C. Cheung, D. Pao, and I. Verbauwhede, "High-Speed Polynomial Multiplication Architecture for Ring-LWE and SHE Cryptosystems," *IEEE Trans. on Circuits and Systems I*, vol. 62, pp. 157–166, Jan 2015.

[117] Y. Doröz, Y. Hu, and B. Sunar, "Homomorphic AES evaluation using the modified LTV scheme," *Des. Codes Cryptography*, pp. 1–26, 2015.

[118] R. Baillie, "New Primes of the Form $k \cdot 2^n + 1$," *Mathematics of Computation*, vol. 33, no. 148, pp. pp. 1333–1336, 1979.

[119] R. M. Robinson, "A Report on Primes of the Form $k \cdot 2^n + 1$ and On Factors of Fermat Numbers," *Proc. Amer. Math. Soc.*, vol. 9, no. 5, pp. 673–681, 1958.

[120] I. S. Reed, T. K. Truong, Y. S. Kwoh, and E. L. Hall, “Image Processing by Transforms Over a Finite Field,” *IEEE Trans. on Computers*, vol. C-26, pp. 874–881, Sept 1977.

[121] F. Proth, “Théorème relatif à la théorie des nombres,” *Comptes Rendus des Séances de l’Académie des Sciences*, vol. 87, p. p. 926, 1878.

[122] P. J. Davis, *Circulant Matrices*. Providence, Rhode Island: American Mathematical Society, 1994.

[123] D. Stehlé and R. Steinfeld, “Making NTRU as Secure as Worst-Case Problems over Ideal Lattices,” in *EUROCRYPT*, vol. 6632 of *LNCS*, pp. 27–47, 2011.

[124] “GNU Multiple Precision Arithmetic Library.” `www.gmplib.org`.

[125] L. Bluestein, “A Linear Filtering Approach to the Computation of Discrete Fourier Transform,” *IEEE Trans. on Audio and Electro.*, vol. 18, pp. 451–455, Dec 1970.

[126] L. R. Rabiner, R. W. Schafer, and C. M. Rader, “The Chirp Z-Transform Algorithm and Its Application,” *Bell Syst. Tech. J.*, vol. 48, pp. 1249–1292, May 1969.

[127] R. Cramer, I. Damgård, and J. B. Nielsen, “Multiparty Computation from Threshold Homomorphic Encryption,” in *EUROCRYPT*, vol. 2045 of *LNCS*, pp. 280–300, Springer, 2001.

[128] R. Cramer, R. Gennaro, and B. Schoenmakers, “A Secure and Optimally Efficient Multi-Authority Election Scheme,” in *EUROCRYPT*, vol. 1233 of *LNCS*, pp. 103–118, Springer, 1997.

[129] Y. Doröz, G. S. Çetin, and B. Sunar, “On-the-fly Homomorphic Batching/Unbatching,” in *Financial Cryptography and Data Security*, pp. 288–301, 2016.

[130] F. J. Harris, *Multirate Signal Processing for Communication Systems*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2004.

[131] C. Ding, D. Pei, and A. Salomaa, *Chinese Remainder Theorem: Applications in Computing, Coding, Cryptography*. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 1996.

[132] T. Lepoint and M. Naehrig, “A Comparison of the Homomorphic Encryption Schemes FV and YASHE,” in *AFRICACRYPT*, vol. 8469 of *LNCS*, pp. 318–335, Springer, 2014.

[133] R. Lidl and H. Niederreiter, *Introduction to finite fields and their applications*. Cambridge university press, 1994.

[134] B. A. Harrison, “On the Reducibility of Cyclotomic Polynomials over Finite Fields,” *The American Mathematical Monthly*, vol. 114, no. 9, pp. pp. 813–818, 2007.

[135] A. E. Yagle, “Fast Algorithms for Matrix Multiplication Using Pseudo-Number-Theoretic Transforms,” *IEEE Trans. on Signal Proc.*, vol. 43, pp. 71–76, Jan 1995.

[136] M. Mohanty, M. Zhang, M. R. Asghar, and G. Russello, “PANDORA: Preserving Privacy in PRNU-Based Source Camera Attribution,” in *IEEE TrustCom/BigDataSE*, pp. 1202–1207, Aug 2018.

[137] A. Pedrouzo-Ulloa, M. Masciopinto, J. R. Troncoso-Pastoriza, and F. Pérez-González, “Camera Attribution Forensic Analyzer in the Encrypted Domain,” in *IEEE WIFS*, pp. 1–7, 2018.

[138] M. Mohanty, M. Zhang, M. R. Asghar, and G. Russello, "e-PRNU: Encrypted Domain PRNU-Based Camera Attribution for Preserving Privacy," *IEEE Trans. on Dependable and Secure Computing*, pp. 1–1, 2019.

[139] M. Naveed, E. Ayday, E. W. Clayton, J. Fellay, C. A. Gunter, J.-P. Hubaux, B. A. Malin, and X. Wang, "Privacy in the Genomic Era," *ACM Computing Surveys (CSUR)*, vol. 48, no. 1, p. 6, 2015.

[140] A. Johnson and V. Shmatikov, "Privacy-Preserving Data Exploration in Genome-Wide Association Studies," in *ACM KDD*, pp. 1079–1087, Sep. 2013.

[141] F. Yu, S. E. Fienberg, A. B. Slavkovi, and C. Uhler, "Scalable privacy-preserving data sharing methodology for genome-wide association studies," *Journal of Biomedical Informatics*, vol. 50, pp. 133 – 141, 2014. Special Issue on Informatics Methods in Medical Privacy.

[142] E. Ayday, J. Raisaro, and J. Hubaux, "Privacy-Enhancing Technologies for Medical Tests Using Genomic Data," in *NDSS*, Feb. 2013.

[143] M. Namazi, J. R. Troncoso-Pastoriza, and F. Pérez-González, "Dynamic Privacy-Preserving Genomic Susceptibility Testing," in *ACM IH&MMSec*, pp. 45–50, 2016.

[144] S. Kathiresan, O. Melander, D. Anevski, C. Guiducci, N. P. Burtt, C. Roos, J. N. Hirschhorn, G. Berglund, B. Hedblad, L. Groop, D. M. Altshuler, C. Cheh, and M. Orho-Melander, "Polymorphisms Associated with Cholesterol and Risk of Cardiovascular Events," *New England Journal of Medicine*, vol. 358, no. 12, pp. 1240–1249, 2008.

[145] X. Hu, W. Zhang, K. Li, H. Hu, and N. Yu, "Secure Nonlocal Denoising in Outsourced Images," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 12, pp. 40:1–40:23, Mar. 2016.

[146] S. M. SaghaianNejadEsfahani, Y. Luo, and S. c. S. Cheung, "Privacy protected image denoising with secret shares," in *IEEE ICIP*, pp. 253–256, Sept 2012.

[147] A. Buades, B. Coll, and J. M. Morel, "A Review of Image Denoising Algorithms, with a New One," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, 2005.

[148] I. Atkinson, F. Kamalabadi, S. Mohan, and D. L. Jones, "Asymptotically Optimal Blind Estimation of Multichannel Images," *IEEE Trans. on Image Proc.*, vol. 15, pp. 992–1007, April 2006.

[149] G. Phillips, *Interpolation and Approximation by Polynomials*. CMS books in mathematics, New York: Springer, 2003.

[150] S. Winograd, "On the Number of Multiplications Required to Compute Certain Functions," *PNAS*, vol. 58, no. 5, pp. 1840–1842, 1967.

[151] P. Borwein and T. Erdélyi, *Polynomials and Polynomial Inequalities*. Graduate texts in mathematics, New York: Springer, 1995.

[152] M. S. Paterson and L. J. Stockmeyer, "On the Number of Nonscalar Multiplications Necessary to Evaluate Polynomials," *SIAM J. Comput.*, vol. 2, no. 1, pp. 60–66, 1973.

[153] G. H. Owenson and N. J. Savage, "The Tor Dark Net," 9 2015.

[154] M. Goljan and J. Fridrich, "Camera identification from cropped and scaled images," in *Proc. SPIE, Security, Forensics, Steganography, and Watermarking of Multimedia Content X*, vol. 6819, Mar. 2008.

[155] F. Marra, G. Poggi, C. Sansone, and L. Verdoliva, "Blind PRNU-based image clustering for source identification," *IEEE Trans. on Information Forensics and Security*, vol. 12, pp. 2197–2211, September 2017.

[156] F. Gisolf, P. Barens, E. Snel, A. Malgoezar, M. Vos, A. Mieremet, and Z. Geradts, "Common source identification of images in large databases," *Forensic science international*, vol. 244, pp. 222–230, November 2014.

[157] "Nifty website." `http://research.ncl.ac.uk/nifty/`. Accessed: 15 September 2018.

[158] F. Pérez-González, M. Masciopinto, I. González-Iglesias, and P. Comesaña, "Fast sequential forensic detection of camera fingerprint," in *IEEE ICIP*, pp. 3902–3906, Sep. 2016.

[159] C. Meij and Z. Geradts, "Source camera identification using Photo Response Non-Uniformity on WhatsApp," *Digital Investigation*, vol. 24, pp. 142 – 154, 2018.

[160] R. Hurley, S. Prusty, H. Soroush, R. J. Walls, J. Albrecht, E. Cecchet, B. N. Levine, M. Liberatore, B. Lynn, and J. Wolak, "Measurement and analysis of child pornography trafficking on p2p networks," in *International Conference on World Wide Web*, WWW'13, pp. 631–642, ACM, 2013.

[161] G. C. Holst, *CCD Arrays, Cameras, and Displays*. SPIE Optical Engineering Press Bellingham, WA, 2nd ed., 1998.

[162] M. Chen, J. Fridrich, and M. Goljan, "Digital imaging sensor identification (further study)," in *Proc. SPIE, Security, Forensics, Steganography, and Watermarking of Multimedia Content X*, vol. 6505, Feb. 2007.

[163] M. Chen, J. Fridrich, M. Goljan, and J. Lukáš, "Determining image origin and integrity using sensor noise," *IEEE Trans. on Information Forensics and Security*, vol. 3, pp. 74–90, Mar. 2008.

[164] L. Bondi, F. Pérez-González, P. Bestagini, and S. Tubaro, "Design of Projection Matrices for PRNU Compression," in *IEEE WIFS*, pp. 1–6, Dec 2017.

[165] P. Korus and J. Huang, "Multi-scale analysis strategies in prnu-based tampering localization," *IEEE Trans. on Information Forensics and Security*, vol. 12, pp. 809–824, April 2017.

[166] M. Griffin, "Lowest degree of polynomial that removes the first digit of an integer in base p." `https://mathoverflow.net/q/269282`. Accessed: 15 September 2018.

[167] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*. Pearson, 1975.

[168] G. S. Çetin, Y. Doröz, B. Sunar, and W. J. Martin, "An Investigation of Complex Operations with Word-Size Homomorphic Encryption." Crypt. ePrint Archive, Report 2015/1195, 2015.

[169] T. Gloe and R. Böhme, "The 'Dresden Image Database' for benchmarking digital image forensics," in *Proc. of the 25th Symp. On Applied Computing (ACM SAC 2010)*, vol. 2, pp. 1585–1591, Mar. 2010.

[170] D.T. Dang-Nguyen, C. Pasquini, V. Conotter and G. Boato, "RAISE – A raw images dataset for digital image forensics," in *ACM MMSys*, pp. 219–224, Mar. 2015.

[171] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive Verifiable Computing: Outsourcing Computation to Untrusted Workers," in *CRYPTO*, vol. 6223 of *LNCS*, pp. 465–482, 2010.

[172] D. Fiore, R. Gennaro, and V. Pastro, "Efficiently Verifiable Computation on Encrypted Data," in *ACM CCS*, pp. 844–855, 2014.

[173] M. Walfish and A. J. Blumberg, "Verifying Computations Without Reexecuting Them," *Commun. ACM*, vol. 58, pp. 74–84, Jan. 2015.

[174] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," in *CRYPTO*, vol. 8042 of *LNCS*, pp. 75–92, 2013.

[175] F. Bourse, R. D. Pino, M. Minelli, and H. Wee, "FHE Circuit Privacy Almost for Free," in *CRYPTO*, vol. 9815 of *LNCS*, pp. 62–89, 2016.

[176] J.-P. Serre, *Linear Representations of Finite Groups*. Springer-Verlag New York, 1977.

[177] D. Micciancio and O. Regev, "Worst-Case to Average-Case Reductions Based on Gaussian Measures," *SIAM J. Comput.*, vol. 37, pp. 267–302, Apr. 2007.

[178] W. Banaszczyk, "New bounds in some transference theorems in the geometry of numbers," *Mathematische Annalen*, vol. 296, no. 1, pp. 625–635, 1993.

[179] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge University Press, 1991. Cambridge Books Online.

[180] C. Peikert and A. Rosen, "Lattices that admit logarithmic worst-case to average-case connection factors," in *ACM STOC*, pp. 478–487, 2007.

[181] T. van Erven and P. Harremos, "Rényi Divergence and Kullback-Leibler Divergence," *IEEE Trans. on Information Theory*, vol. 60, pp. 3797–3820, July 2014.

[182] G. O. Michel Misiti, Yves Misiti and J.-M. Poggi, *Wavelets and their applications*. London, UK: ISTE Ltd, 2007.

[183] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé, "Classical Hardness of Learning with Errors," in *ACM STOC*, pp. 575–584, 2013.

[184] C. Ning and Q. Xu, "Multiparty computation for modulo reduction without bit-decomposition and a generalization to bit-decomposition," in *ASIACRYPT*, vol. 6477 of *LNCS*, pp. 483–500, Springer, 2010.

[185] T. Veugen, "Encrypted integer division and secure comparison," *Int. J. Appl. Cryptol.*, vol. 3, pp. 166–180, June 2014.

[186] M. Dahl, C. Ning, and T. Toft, "On Secure Two-Party Integer Division," in *Financial Cryptography and Data Security*, pp. 164–178, Springer, 2012.

[187] N. Dowlin, R. Gilad-Bachrach, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "Manual for using homomorphic encryption for bioinformatics," *Proceedings of the IEEE*, vol. 105, pp. 552–567, March 2017.

[188] T. Lindeberg, *Scale-Space Theory in Computer Vision*. Norwell, MA, USA: Kluwer Academic Publishers, 1994.

[189] J. R. Troncoso-Pastoriza, A. Pedrouzo-Ulloa, and F. Pérez-González, “Secure Genomic Susceptibility Testing based on Lattice Encryption,” in *IEEE ICASSP*, pp. 2067–2071, 2017.

Appendix A

A Security Reduction to Multivariate RLWE

This appendix is adapted with permission from ArXiv: Alberto Pedrouzo-Ulloa, Juan Ramón Troncoso-Pastoriza, and Fernando Pérez-González. On Ring Learning with Errors over the Tensor Product of Number Fields. ArXiv e-prints, CoRR abs/1607.05244v3, February 2018.

A.1. Introduction

This appendix formalizes a generalization of RLWE to the tensor product of number fields, denoted m-RLWE (multivariate Ring Learning With Errors). Indeed, the interest on a multivariate version of RLWE emerges when working with multidimensional structures, such as videos or images [4, 46]. In this scenario, the use of a tensorial decomposition in "coprime" cyclotomic rings (see [45, 41, 40]) is not valid, as these applications require that the modular functions have the same form (e.g., $1 + z^n$). This is the context where m-RLWE [4] was originally introduced as a means to easily deal with encrypted multidimensional structures.

Contributions: The main contribution of this appendix is the formalization of the multivariate Ring Learning With Errors problem, also providing an extended reduction of the original proof by Lyubashevsky *et al.* [40, 41]. Unfortunately, as we can see in Chapters 2 and 5, for some parameterizations the multivariate RLWE problem is reduced from an easier problem than expected at a first glance; which implies a decrease in the effective dimension of the underlying lattices. Even so, the problem still introduces efficiency improvements in many practical signal processing applications (see Chapters 5, 7 and 8, and Appendix B). In Chapters 2 and 5 we study in depth how to securely parameterize the multivariate RLWE problem.

Structure: The rest of the appendix is organized as follows: Section A.2 introduces some algebraic number theory notions needed for the security reduction of m-RLWE. The m-RLWE problem and its reduction is introduced in Section A.3. Finally, Section A.4 discusses the main contributions and draws some conclusions.

A.2. Background

This section presents the fundamental concepts of lattices and algebraic number theory and extends them to the more general case of a tensor of number fields on which m-RLWE is based.

A.2.1. Some Concepts of Lattices and Algebraic Number Theory

Here we include the most relevant concepts required to work with multivariate RLWE such as it was introduced in [4, 22]. To focus interest on the main difference with respect to the univariate counterpart, we omit many of the existing details. We refer to Appendix A.A for a deep review of the required concepts of lattices and algebraic number theory.

Number fields

A number field is defined as a field extension $K = \mathbb{Q}(\varsigma)$ where the element ς is incorporated to the field of rationals. This element ς satisfies $f(\varsigma) = 0$ for an irreducible polynomial $f(x) \in \mathbb{Q}[x]$ called *minimal polynomial* of ς. The degree n of a number field is the degree of its minimal polynomial.

We can also see the number field K as an n-dimensional vector space over $\mathbb{Q}$ where $\{1, \varsigma, \ldots, \varsigma^{n-1}\}$ is called the *power basis* of the field K. Of course, we have an isomorphism between K and $\mathbb{Q}\left[x\right]/f(x)$.

In this Appendix, we have a special interest on cyclotomic fields, which are those fields where $\varsigma = \varsigma_m$ (for some natural number m) is an m-th primitive root of unity and the minimal polynomial of ς_m is the m-th cyclotomic polynomial $\Phi_m(x) = \prod_{i\in\mathbb{Z}_m^*}(x - \omega_m^i) \in \mathbb{Z}\left[x\right]$, where $\omega_m \in \mathbb{C}$ is any primitive m-th complex root of unity (for example $\omega_m = e^{2\pi\sqrt{-1}/m}$). It is important to note that the different powers ω_m^i of $\Phi_m(x)$ are the m-th roots of unity in $\mathbb{C}$ and that the degree of $\Phi_m(x)$ is $n = \phi(m)$, where $\phi(m)$ is the Euler's totient function.

In general, there is no bound to the number of elements that can be added, so we could have $K = \mathbb{Q}(\varsigma_{m_1}, \ldots, \varsigma_{m_l})$, that is isomorphic to the cyclotomic field $\mathbb{Q}(\varsigma_m) = \bigotimes_{i\in[l]}\mathbb{Q}(\varsigma_{m_i})$ when $m = \prod_{i\in[l]} m_i$ has a prime-power decomposition and each ς_{m_i} is a m_i-th primitive root of unity (See [45]).

Tensor product of number fields

Throughout the Appendix, we consider a tensor of number fields $K_{(T)} = \bigotimes_{i\in[l]} K_i$, where each K_i is a number field, not necessarily being all of them different. We also consider the ring R as the tensor of the corresponding ring of integers $\mathcal{O}_{K_i}$, that is, $R = \bigotimes_{i\in[l]}\mathcal{O}_{K_i}$, and an integer modulus $q \geq 2$. Unless otherwise specified, we in general restrict ourselves to the case where each K_i is a cyclotomic field of order $m_i > 2$ and degree $n_i = \phi(m_i)$, so elements of $K_{(T)}$ (resp. of R) can be equivalently described as multivariate polynomials in $\mathbb{Q}[x_1, \ldots, x_l]$ (resp. $\mathbb{Z}[x_1, \ldots, x_l]$) modulo $\Phi_{m_1}(x_1), \ldots, \Phi_{m_l}(x_l)$. The total degree of the field tensor polynomials is therefore $n = \prod_{i=1}^{l} n_i$.

Embeddings and Geometry

A number field $K = Q(\varsigma)$ of degree n has exactly n embeddings $\sigma_i : K \to \mathbb{C}$, where each of these embeddings maps ς to a different complex root of its minimal polynomial f. The number of real embeddings is denoted by s_1 and the number of pairs of complex embeddings by s_2, so we have $n = s_1 + 2s_2$ (the pair (s_1, s_2) is called the signature of the number field).

The canonical embedding is defined as $\sigma : K \to \mathbb{R}^{s_1} \times \mathbb{C}^{2s_2}$, where $\sigma(x) = (\sigma_1(x), \ldots, \sigma_n(x))^T$. Let $\{\sigma_i\}$ with $i = 1, \ldots, s_1$ be the real embeddings and $\sigma_{s_1+s_2+j} = \bar{\sigma}_{s_1+j}$ with $j = 0, \ldots, s_2 - 1$ be the complex embeddings ($\bar{\sigma_j}$ denotes the complex conjugate of σ_j).

In the cyclotomic case of order $m > 2$, $\Phi_m(x)$ has only complex roots, so s_1 is always equal to zero, and $2s_i = \phi(m)$ (being $\phi(\cdot)$ the Euler's totient function). For our purposes, it is useful to redefine the embedding of $\bigotimes_{i\in[l]} K_i$ as in [45] with a particular ordering of the $\sigma_i(x)$, that improve the handling of automorphisms. For each $i \in [l]$ and $j \in [n_i]$, we denote by $(\omega_{i,1}, \ldots, \omega_{i,n_i})$ the $n_i = 2s_i$ roots of $\Phi_{m_i}(x)$, assembled in an order such that $\omega_{i,j} = \bar{\omega}_{i,j+s_i}$ for all $j \in [s_i]$. We also fix the following alphabetical ordering between $[n_1] \times \cdots \times [n_l]$ and $[n]$, where the tuple $(j_1, \ldots, j_l)$ is sequentially numbered as:

$$j = 1 + \sum_{i\in[l]} (j_i - 1) \prod_{d\in[i]} n_{d-1}. \tag{A.1}$$

With this definition in mind, we extend the canonical embeddings of each number field to an explicit tensor embedding σ from $\bigotimes_{i\in[l]} K_i$ to $\mathbb{C}^n$, where $\sigma(P)$ is the vector of evaluations $(\sigma_1(P), \ldots, \sigma_n(P))$ where $\sigma_j(P) = P(\omega_{j_1}, \ldots, \omega_{j_n})$ using the above indexing. This is a ring homomorphism where multiplication and additions are element-wise.

Note also that because each complex root always appears with its conjugate, and the evaluated polynomial is rational, half of the positions of the tensor embedding are just conjugates of the other positions, hence the output tensor embedding lives in a vector subspace of $\mathbb{C}^n$ of dimension n. We denote this space $H_{(T)}$ in the rest of the Appendix.

The trace is extended over the tensor and this can be defined as the sum of the embeddings: $\text{Trace}(P) = \sum_{j=1}^{n} \sigma_j(P)$, its output is always rational.

Similarly, for a prime number q such that $q = 1$ modulo each $m_1, \ldots, m_l$, we define the tensor embedding modulo q the same way, from R/qR to $(\mathbb{Z}/q\mathbb{Z})^n$ by taking the roots of $\Phi_{m_i}(x)$ modulo q instead of the complex roots.

Multivariate Ideals: An (integer) *ideal* $\mathcal{J}$ is a subgroup of R such that $R \cdot \mathcal{J} \subseteq \mathcal{J}$, and a *fractional ideal* $\mathcal{I}$ is a subgroup of $K_{(T)}$ where there is an element $d \in R$ such that $d \cdot \mathcal{I}$ is an ideal of R.

The *dual* of $\mathcal{I}$, noted $\mathcal{I}^\vee$ is by definition $\{a \in K_{(T)}\text{s.t.}\forall b \in I, \text{Trace}(ab) \in \mathbb{Z}\}$. $\mathcal{I}^\vee$ is a fractional ideal of $K_{(T)}$.

We denote $\mathcal{J}_q$ for $\mathcal{J}/q\mathcal{J}$, where $\mathcal{J}$ is a fractional ideal in $K_{(T)}$. Let $R^\vee$ be the dual fractional ideal of R and $\mathbb{T} = K_{(T),\mathbb{R}}/R^\vee$.[1]

[1] $K_{(T),\mathbb{R}}$ is defined as $K_{(T)} \bigotimes_{\mathbb{Q}} \mathbb{R}$.

Lattice background

Ideal Lattices: The image $\sigma(\mathcal{I})$, also noted $\mathcal{L}(\mathcal{I})$, of any fractional ideal by the tensor embedding is a discrete subgroup of $H_{(T)}$, which we denote as *multivariate ideal lattice*. The ideal $\mathcal{I}$ is *full-rank* when the dimension of $\mathcal{L}(\mathcal{I})$, as an $\mathbb{R}$-vector space, is n. Note that unlike in the univariate case, being non-trivial is not a sufficient condition for an ideal to be full rank: for instance, the bi-variate ideal generated by $x - y$ modulo $x^2 + 1, y^2 + 1$ has dimension 2 instead of 4.

When $\mathcal{L}(\mathcal{I})$ is full rank, the dual ideal lattice $\mathcal{L}(\mathcal{I}^\vee)$ is the dual lattice of of $\mathcal{L}(\mathcal{I})$.

Gaussian Measures: We consider the Gaussian function $\rho_r : H_{(T)} \to (0, 1]$ with $r > 0$ as $\rho_r(\boldsymbol{x}) = \exp(-\pi||\boldsymbol{x}||^2/r^2)$ (see Section 2.3.1 from Chapter 2). A continuous Gaussian probability distribution can be obtained by normalizing the previous function, hence obtaining D_r with a density function $r^{-n}\rho_r(\boldsymbol{x})$. Extending it to the non-spherical Gaussian case, we consider the vector $\boldsymbol{r} = \bigotimes_{i\in[l]} \boldsymbol{r}_i$ where $\boldsymbol{r} = (r_1, \dots, r_n) \in (\mathbb{R}^+)^n$ or also each $\boldsymbol{r}_i = (r_{i,1}, \dots, r_{i,n_i}) \in (\mathbb{R}^+)^{n_i}$ and whose components satisfy $r_{i,j+s_1+s_2} = r_{i,j+s_1}$. Finally, a sample from $D_{\boldsymbol{r}}$ is given by $\sum_{i\in[n]} x_i \boldsymbol{h}_i$, where $x_j = \prod_{i\in[l]} x^{(i)}_{j_i}$ and each x_j is drawn independently from the Gaussian distribution D_{r_j} over $\mathbb{R}$; r_j is equal to $\prod_{i\in[l]} r_{i,j_i}$. We are using the mapping between $\{j\}_{j\in[n]}$ and $\{j_i\}_{j_i\in[n_i],i\in[l]}$ in Eq. (A.1).

Due to space constraints, we refer the reader to Appendix A.A for the definition of the smoothing parameter $\eta_\epsilon(\Lambda)$ for a lattice Λ, and for some of the underlying lemmas required at some steps of the proofs.[2]

Definition 14 (Bounded Distance Decoding problem in ℓ_p-norm)**.** *For a full-rank lattice Λ, and a bounded decoding radius $r_p \leq ||\lambda_1(\lambda)/2||_p$, given a target $t = v+e$, where $v \in \Lambda$ and $||e||_p < r_p$, recover v and e.*

Definition 15 (Discrete Gaussian Sampling problem)**.** *Given a lattice Λ and a parameter $s > \eta_\epsilon(\Lambda)$, the goal is to output samples generated from $D_{\Lambda,s}$.*

A.2.2. Automorphisms and Linear Representation Theory

In order to justify the structure and behavior of the new automorphisms in the tensor case, we resort to the theory of Linear Representations [176]. First, we introduce the main concepts needed from this theory and afterwards, we detail the different automorphisms that can be found.

In general, we consider V as a vector space of dimension d over $\mathbb{C}$, and we define GL(V) as the group composed of all the isomorphisms of V onto itself. An element a belonging to GL(V) can be seen as a linear mapping from V to V, and we denote its inverse as a^{-1}. Analogously, we could think of each linear mapping as an invertible square matrix A of size $d \times d$ whose coefficients are complex numbers. Hence, we can see that GL(V) is composed of all the different invertible square matrices of order d.

Let G be a finite group, we define a linear representation of G in V as a homomorphism ρ from G to GL(V). Provided that the group G has the composition operation $(r, s) \to rs$ for $r, s \in G$, we have that $\rho(rs) = \rho(r)\rho(s)$, where $\rho(r)\rho(s)$ represents the matrix multiplication operation between the two matrices associated with r and s, respectively. When $1 \in G$, this implies $\rho(1) = 1$ and $\rho(s^{-1}) = \rho(s)^{-1}$. Commonly, we consider V as a representation space (or simply

[2] These additional details can also be consulted in the Appendix A.A.1.

a representation) of G. Now, we can particularize the previous results to our specific case, for $W = \mathbb{Q}(\varsigma_{m_i}) \subset \mathbb{C}$. Let $G = \mathbb{Z}_{m_i}^*$, and define the composition operation as the product operation between units of $\mathbb{Z}_{m_i}$; we have the following linear representation $\rho_i : \mathbb{Z}_{m_i}^* \to \mathrm{GL}(\mathbb{Q}(\varsigma_{m_i}))$, where $\rho_i(\mathbb{Z}_{m_i}^*) \subseteq \mathrm{GL}(\mathbb{Q}(\varsigma_{m_i}))$ is composed of the different automorphisms $\tau_k = \rho_i(k)$ for $k \in \mathbb{Z}_{m_i}^*$ such that $\tau_k(\varsigma_{m_i}) = \varsigma_{m_i}^k$. Hence, we have $\mathbb{Q}(\varsigma_{m_i})$ as a representation of $\mathbb{Z}_{m_i}^*$. It is worth noting that the effect of τ_k over the embedding is a rotation of the coordinates of the subspace H_i, that is, $\sigma_i(\tau_k(\varsigma_{m_i})) = \sigma_{ik}(\varsigma_{m_i})$, being $i \in \mathbb{Z}_{m_i}^*$.

Outer tensor product of Linear Representations: Consider two groups $(G_1, \cdot)$ and $(G_2, \cdot)$ and consider the direct product $G_1 \times G_2$ with the following "$\cdot$" operation: $(s_1, s_2) \cdot (t_1, t_2) = (s_1 \cdot s_2, t_1 \cdot t_2)$ where $(s_1, s_2), (t_1, t_2) \in G_1 \times G_2$.

If we define $\rho^1 : G_1 \to \mathrm{GL}(V_1)$ and $\rho^2 : G_2 \to \mathrm{GL}(V_2)$ as linear representations of G_1 and G_2, we have a linear representation $\rho^1 \otimes \rho^2 : G_1 \times G_2 \to \mathrm{GL}(V_1 \bigotimes V_2)$ by setting $\left(\rho^1 \otimes \rho^2\right)(s_1, s_2) = \rho^1(s_1) \otimes \rho^2(s_2)$.

This way of dealing with the tensor of different linear representations allows us to define the different automorphisms of the field tensor $K_{(T)} = \bigotimes_{i \in [l]} K_i$ in terms of the automorphisms of each K_i. Then, we have the corresponding homomorphism for $K_{(T)}$ with the tensor of linear representations

$$\bigotimes_{i \in [l]} \rho_i : \bigoplus_{i \in [l]} \mathbb{Z}_{m_i}^* \to \mathrm{GL}\left(\bigotimes_{i \in [l]} \mathbb{Q}(\varsigma_{m_i})\right),$$

where each ρ_i satisfies $\rho_i(k_i) = \tau_{k_i}^{(i)}$, with $k_i \in \mathbb{Z}_{m_i}^*$ and being $\tau_{k_i}^{(i)}$ the corresponding $\phi(m_i)$ automorphisms of the K_i number field.

Finally, in order to map the set of $\prod_{i \in [l]} \phi(m_i)$ automorphisms $\bigotimes_{i \in [l]} \tau_{k_i}^{(i)}$ with only one index we can consider the mapping in Eq. (A.1), in such a way that $k_i \in \mathbb{Z}_{m_i}^* = g^{(i)}([\phi(m_i)])$ and $j_i = \left(g^{(i)}(k_i)\right)^{-1}$.

A.2.3. Chinese Remainder Theorem

This section reformulates the Chinese Remainder Theorem (CRT) for the ring $R = \bigotimes_{i \in [l]} \mathcal{O}_{K_i}$ in the tensor of number fields $K_{(T)} = \bigotimes_{i \in [l]} K_i$, and revisits some important concepts introduced in [41].

Lemma 3 (Chinese Remainder Theorem)**.** *Let $\mathcal{I}_1, \ldots, \mathcal{I}_r$ be pairwise coprime ideals in R, and let $\mathcal{I} = \prod_{i \in [r]} \mathcal{I}_i$. The natural ring homomorphism $R \to \bigoplus_{i \in [r]} (R/\mathcal{I}_i)$ induces a ring isomorphism $R/\mathcal{I} \to \bigoplus_{i \in [r]} (R/\mathcal{I}_i)$.*

We now focus on explaining why the CRT works over multivariate polynomial rings and how the use of the automorphisms in Section A.2.2 affects the decomposition caused by the CRT. First, consider $R = \mathcal{O}_{K_i} = \mathbb{Z}[\varsigma_{m_i}]$, the ring of integers of a number field $\mathbb{Q}(\varsigma_{m_i})$, where ς_{m_i} is the m_i-th primitive root of unity. If we work with the ideal $\langle q \rangle = qR$ and $q \in \mathbb{Z}$ is prime, we have the following factorization $\langle q \rangle = \prod_i \mathfrak{q}_i^e$, where there are $\phi(m_i)/(ef)$ different $\mathfrak{q}_i$ of norm q^f, $e = \phi(q')$, and f is the minimum natural number that satisfies $q^f \equiv 1 \bmod m_i/q'$ with q' the largest power of q that divides m_i.

For each ideal, we have $\mathfrak{q}_j = \langle q, F_j(\varsigma_{m_i}) \rangle$ with $\Phi_{m_i}(x) = \prod_j (F_j(x))^e$, being the factorization of $\Phi_{m_i}(x)$ modulo q. As explained in [41], when considering that $q \equiv 1 \bmod m_i$, both e and f are equal to 1, and as there is an m_i-th primitive root of unity w_i in $\mathbb{Z}_q$, then $\Phi_{m_i}(x) = \prod_{j\in\mathbb{Z}^*_{m_i}}(x - w_i^j)$. Therefore, we finally have $\langle q\rangle = \prod_{j\in\mathbb{Z}^*_{m_i}} \mathfrak{q}_j$, with $\mathfrak{q}_j = \langle q, \varsigma_{m_i} - w_i^j\rangle$. Additionally, we can use the automorphism $\tau_k^{(i)}$ to exchange the contents between two different prime ideals $\mathfrak{q}_j$ of qR; that is, we can do $\tau_k^{(i)}(\mathfrak{q}_j) = \mathfrak{q}_{j/k}$ (see Lemma 2.16 in [41]).

Now, resorting to Lemma 3, we have an isomorphism from
$\mathbb{Z}[\varsigma_{m_i}]/\langle q\rangle$ to $\bigoplus_{j\in\mathbb{Z}^*_{m_i}} \mathbb{Z}[\varsigma_{m_i}]/\langle q, \varsigma_{m_i} - w_i^j\rangle$, that is in fact also isomorphic to $\mathbb{Z}_q^{\phi(m_i)}$.

Multivariate extension: The multivariate case $R = \bigotimes_{i\in[l]} \mathcal{O}_{K_i}$ can be formulated as the tensor product between the previously considered univariate rings, that is, $\bigotimes_{i\in[l]} \mathbb{Z}[\varsigma_{m_i}]/\langle q\rangle$, where q has to satisfy $q \equiv 1 \bmod m_i$ for all $i \in [l]$. This is isomorphic to the tensor product of the respective direct sum in terms of the different prime ideals $\bigotimes_{i\in[l]}\left(\bigoplus_{j\in\mathbb{Z}^*_{m_i}} \mathbb{Z}[\varsigma_{m_i}]/\langle q, \varsigma_{m_i} - w_i^j\rangle\right)$; as the tensor and direct product commute, we have $\bigoplus_{j\in[\prod_{i\in[l]}\phi(m_i)]}\left(\bigotimes_{k\in[l]} \mathbb{Z}[\varsigma_{m_k}]/\langle q, \varsigma_{m_k} - w_k^{j_k}\rangle\right)$; the mapping between the set $\{j_1, \ldots, j_l\}$ and j is defined by Eq. (A.1). This ring is in fact isomorphic to $\mathbb{Z}_q^{\prod_{i\in[l]}\phi(m_i)}$.

By virtue of the ring isomorphism $\varsigma_{m_i} \to x_i$ for $i \in [l]$, we have that
$\bigoplus_{i\in[l], j_i\in\mathbb{Z}^*_{m_i}} \mathbb{Z}_q[x_1, \ldots, x_l]/\langle x_1 - w_1^{j_1}, \ldots, x_l - w_l^{j_l}\rangle$. Thanks to the mapping introduced in Eq. (A.1), we consider $\mathfrak{q}_j = \mathfrak{q}_{j_1,\ldots,j_l} = \langle x_1 - w_1^{j_1}, \ldots, x_l - w_l^{j_l}\rangle$, with $j \in \left[\prod_{i\in[l]}\phi(m_i)\right]$. First, it can be easily shown that each $\mathfrak{q}_j$ is an ideal and, as there is an isomorphism from $\mathbb{Z}_q[x_1, \ldots, x_l]/\mathfrak{q}_j$ to the finite field $\mathbb{Z}_q$, $\mathfrak{q}_j$ is a maximal ideal and also a prime ideal, as every maximal ideal over a ring is also a prime ideal. In order to show that all the $\mathfrak{q}_j$ are comaximal ideals, we use the following *reductio ad absurdum* argument: consider two different maximal ideals $\mathfrak{q}_j$ and $\mathfrak{q}_k$, with $k \neq j$; by definition, $\mathfrak{q}_k + \mathfrak{q}_j$ is also an ideal; we have three possible cases: (a) $\mathfrak{q}_k + \mathfrak{q}_j = \mathfrak{q}_k$, (b) $\mathfrak{q}_k + \mathfrak{q}_j = \mathfrak{q}_j$, and (c) there is another maximal ideal $\mathfrak{q}_k + \mathfrak{q}_j$. The first two cases are not true because $\mathfrak{q}_k$ and $\mathfrak{q}_j$ are different, and the third case is impossible because each ideal is maximal, therefore $\mathfrak{q}_k + \mathfrak{q}_j = \mathbb{Z}_q[x_1, \ldots, x_l]$, which is the definition of comaximal ideals.

As $\mathfrak{q}_j$ for $j \in \left[\prod_{i\in[l]}\phi(m_i)\right]$ is a set of comaximal ideals, we can use Lemma 3 to show that there exists an isomorphism

$$\mathbb{Z}_q[x_1, \ldots, x_l]/\langle\Phi_{m_1}(x_1), \ldots, \Phi_{m_l}(x_l)\rangle \cong \bigoplus_{j\in[\prod_{i\in[l]}\phi(m_i)]} \left(\mathbb{Z}_q[x_1, \ldots, x_l]/\mathfrak{q}_j\right);$$

that is, we can compute the corresponding CRT, and its properties also apply. After this, we can present a similar result to Lemma 2.16 in [41], but adapted to our more general case:

Lemma 4 (Lyubashevsky *et al.* [41] Lemma 2.16)**.** *For any* $\mathfrak{q}_j = \mathfrak{q}_{j_1,\ldots,j_l}$ *and* $\mathfrak{q}_{j'} = \mathfrak{q}_{j'_1,\ldots,j'_l}$ *(by Equation* (A.1)*), we have a linear representation (automorphism)* $\otimes_{i\in[l]}\rho_i(k_1, \ldots, k_l) = \otimes_{i\in[l]}\tau_{k_i}^{(i)}$ *where* $k_i \in \mathbb{Z}^*_{m_i}$ *satisfies* $\otimes_{i\in[l]}\tau_{k_i}^{(i)}(\mathfrak{q}_j) = \mathfrak{q}_{j'}$.

In the following sections, we go over the two main blocks of the security reduction.

A.3. Multivariate RLWE

The objective of this section is to adapt and generalize the techniques of Lyubashevsky *et al.* [40, 41] so as to achieve a reduction to a class of multivariate RLWE (m-RLWE) from hardness problems over ideal lattices. For the sake of completeness, we present a generalized version of the multivariate polynomial RLWE problem (see Definition 2 from Chapter 2) which admits any type of cyclotomic polynomial as modular function instead of only those with power-of-two degree.[3]

A.3.1. Main Definitions for Multivariate Ring-LWE

Definition (Definition 2 from Chapter 2: Multivariate Ring LWE distribution)**.** *For $s \in R_q^\vee$ and an error distribution ψ over $K_{(T),\mathbb{R}}$, a sample from the m-RLWE distribution $A_{s,\psi}$ over $R_q \times \mathbb{T}$ is generated by $a \leftarrow R_q$ uniformly at random, $e \leftarrow \psi$, and outputting $(a, b = (a \cdot s)/q + e \bmod R^\vee)$.*

We refer the reader to Chapter 2 for a complete description of the *Search* and *Average-Case Decision* problems (respectively, Definitions 3 and 4), together with their corresponding error distributions (respectively, Definitions 5 and 6).

Our *main Theorem 7* is obtained by combining the theorems from Sections A.3.3 and A.3.4.

Theorem 7 (Extended version to m-RLWE of Lyubashevsky *et al.* [41] Theorem 3.6)**.** *Let $K_{(T)} = \bigotimes_{i\in[l]} K_i$ be the tensor product of l cyclotomic fields of dimension $n_i = \phi(m_i)$ each, and $R = \bigotimes_{i\in[l]} \mathcal{O}_{K_i}$ the tensor of their corresponding ring of integers. Let $\alpha < \sqrt{\log n/n}$, and let $q = q(n) \geq 2$, $q \equiv 1 \bmod m_i$, for all i, be a poly(n)-bounded prime such that $\alpha q \geq \omega(\sqrt{\log n})$, where $\omega(f(n))$ denotes a function that asymptotically grows faster than $f(n)$. Then, there is a polynomial-time quantum reduction from $\tilde{\mathcal{O}}(\sqrt{n}/\alpha)$-approximate SIVP (or SVP) on (tensor) ideal lattices in $K_{(T)}$ to m-R-DLWE$_{q,\Upsilon_\alpha}$. Alternatively, for any $l \geq 1$, we can replace the target problem by the problem of solving m-R-DLWE$_{q,D_\xi}$ given only l samples, where $\xi = \alpha \cdot (nl/\log nl)^{1/4}$.*

Our proof follows the techniques introduced by Lyubashevsky *et al.* [41, 40], adapting and extending their proof in order to deal with the new inconveniences that this tensor case introduces. For example, the considered number field tensor is not even a field (for instance, considering $(\mathbb{Q}[x, y] \bmod 1 + x^n) \bmod 1 + y^n$, the polynomial $x - y$ does not have inverse). We discuss and show how the different peculiarities of the tensor case can be tackled.

Therefore, we first justify that the main properties required by the techniques used by Lyubashevsky *et al.* are preserved in the multivariate case: (a) we show that the ring homomorphism between the finite field tensor and the subspace $H_{(T)}$ (defined in Section A.2.1) exists (even though the finite field tensor is not a field); and we define the Gaussian measures over this tensor space, (b) we explain the structure of the automorphisms which can be used in this tensor case and how to address and work with them; and (c) we explain the use of the CRT (Chinese Remainder Theorem) and its effects over the corresponding automorphisms.

Additionally, we carefully readapt and revise the tools introduced by Lyubashevsky *et al.* to the multivariate case; those steps that need further treatment or corrections are further detailed (see Appendix A.C). Nevertheless, due to space limitations, only the main definitions and lemmas are included here. For a detailed explanation of the different proofs we refer the reader to the Appendices A.B, A.C and A.D.

Finally, the spherical Gaussian is in general easier to use than the distribution Υ_α, especially in the context of FHE where m-RLWE samples can be the result of long chains of computations. We also provide a simpler worst-case to search-to-decision theorem for the decision version of

[3] See Definition 1 for a particularized version of the problem.

m-RLWE with Spherical Gaussian error, but which is not connected to the general multivariate ideal lattice problems.

Theorem 8 (Extended Theorem 5.3 of Lyubashevsky *et al.* [41])**.** *Let R, q and α be as in Theorem 10. There exists a randomized polynomial-time reduction from solving m-R-LWE$_{q,D_\alpha}$ to solving m-R-DLWE$_{q,D_\alpha}$.*

A.3.2. Proof sketch of reduction to the multivariate Ring Learning with Errors problem

The security of m-RLWE is based on three different reductions:

- *Quantum reduction:* A polynomial time quantum reduction between the worst case discrete Gaussian sampling (DGS) (equivalent to the SIVP) problem on multivariate ideal lattices in $K_{(T)}$ to the worst case guaranteed distance decoding (GDD) problem in the same class of lattices.

- *Search hardness:* A polynomial time classical reduction between worst-case GDD to worst-case multivariate search-LWE.[4]

- *Pseudorandomness:* A polynomial time classical reduction from worst-case search m-RLWE to the average case multivariate decisional m-RLWE.

The third reduction proves the equivalence between the search version of m-RLWE, which consists in recovering the secret key s, and the decision version of the same problem, which consists in breaking the semantical security.

The first two reductions connect the average case of the above problems to worst case problems on multivariate ideal lattices with the same module. This, as usual, means that if no algorithm can efficiently solve worst case instances of multivariate GDD or DGS problems, cryptosystems based on m-RLWE are semantically secure.

Quantum reduction: The quantum reduction of Regev [39] proves that given an oracle for the guaranteed distance decoding problem on a lattice, one can obtain small discrete Gaussian samples on its dual lattice. This proof can be specialized to any class of lattices that is stable by duality: in [41], this was applied to the case of univariate ideal lattices. Similarily, by definition of the dual of a multivariate ideal, the class of multivariate ideal lattices is stable by duality.

Search hardness: The main contribution here is to extend the tools from to the more general case of the tensor of cyclotomic fields (or even the tensor of more general fields).

For this purpose, we use Regev's iterative quantum reduction for general lattices together with the corresponding tools that we can find on algebraic number theory; i.e., the Chinese Remainder Theorem and the canonical embedding that were used in the original RLWE reduction, but adapted to our multivariate case.

[4] In this chain of reductions, a worst-case instance is enough, but by increasing the parameters of this reduction, we can reduce it to the average case problem.

Pseudorandomness of m-RLWE: The main purpose of this second part is to show that there exists a reduction from the search problem, discussed in the first part, to the decision variant of the hardness problem. Two different versions are discussed: one for the decision problem with a non-spherical distribution in the canonical embedding, and another one for the decision problem with a spherical distribution but with a bounded number of samples. Additionally, when assuming the hardness of the search problem with a fixed spherical Gaussian error distribution, we also have hardness of the decision version with the same error distribution.

We remark that the main contribution of this Appendix relies on proving that the multivariate samples following the m-RLWE distribution are pseudorandom, therefore generalizing the results of [41] to the case of multivariate elements. The main needed properties are those related to the decomposition of $\langle q \rangle$ into n prime ideals and the use of the automorphisms allowing us to permute the prime ideals.

A.3.3. Hardness Search-LWE

Along this section, let $K_{(T)} = \bigotimes_{i\in[l]} K_i$ of degree n denote the tensor of l arbitrary number fields, and $R = \bigotimes_{i\in[l]} \mathcal{O}_{K_i}$ the corresponding tensor of rings of integers. The results can be applied to an arbitrary number field, so in this section we do not have to consider the specific case of cyclotomic fields.

Theorem 9 (Extended Theorem 4.1 of Lyubashevsky *et al.* [41])**.** *Let $K_{(T)}$ be a tensor of arbitrary number fields with degree n_i each and R the tensor of the corresponding ring of integers. Let $\alpha = \alpha(n) > 0$, and let $q = q(n) \geq 2$ be such that $\alpha q \geq 2 \cdot \omega(\sqrt{\log n})$, where $\omega(f(n))$ denotes a function that asymptotically grows faster than $f(n)$. For some negligible $\epsilon = \epsilon(n)$, there is a probabilistic polynomial-time quantum reduction from $K_{(T)}$-DGS$_\gamma$ to m-R-LWE$_{q,\Psi_{\leq\alpha}}$, where*

$$\gamma = \max\{\eta_\epsilon(\mathcal{I}) \cdot (\sqrt{2}/\alpha) \cdot \omega(\sqrt{\log n}), \sqrt{2n}/\lambda_1(\mathcal{I}^\vee)\}.$$

K$_{(T)}$-DGS$_\gamma$ denotes the discrete Gaussian sampling problem [39, 41], where given an ideal $\mathcal{I}$ in $K_{(T)}$ and a number $s \geq \gamma = \gamma(\mathcal{I})$, we have to generate samples from $D_{\mathcal{I},s}$. Regev [39] showed reductions from standard lattice problems to DGS. As Lyubashevsky *et al.* [41] assert, combining their lemmas 2.2 and 2.4 (from [41]) we have $\eta_\epsilon(\mathcal{I}) \leq \lambda_n(\mathcal{I}) \cdot \omega(\sqrt{\log n})$ (being η_ϵ the smoothing parameter) for any fractional ideal $\mathcal{I}$ and negligible $\epsilon(n)$; we also have that samples from $D_{\mathcal{I},\gamma}$ have length at most $\gamma\sqrt{n}$ with overwhelming probability. This also applies in our tensor case.

Analogously, an oracle for K$_{(T)}$-DGS$_\gamma$ with $\gamma = \eta_\epsilon(\mathcal{I}) \cdot \tilde{\mathcal{O}}(1/\alpha)$ implies an oracle for $\tilde{\mathcal{O}}(\sqrt{n}/\alpha)$-approximate SIVP on ideal lattices in the field tensor $K_{(T)}$.

When each K_i is a cyclotomic field, we also have $\lambda_n(\mathcal{I}) = \lambda_1(\mathcal{I})$ for any fractional ideal $\mathcal{I}$, as for each shortest non-zero $v \in \mathcal{I}$, if we multiply it by different combinations of $\varsigma_{m_1}^{e_1-1} \otimes \ldots \otimes \varsigma_{m_l}^{e_l-1}$ with $e_i \in [\phi(m_i)]$, it yields a total of n independent elements of equal length; that is, we have an oracle for $\tilde{\mathcal{O}}(\sqrt{n}/\alpha)$-approximate SVP. It is worth noting that as the error distribution is added modulo $R^\vee$ in the definition of m-RLWE, the condition $\alpha < \eta_\epsilon(R^\vee)$ must be satisfied for all negligible $\epsilon(n)$ for the problem to be solvable.

A.3.4. Pseudorandomness of m-RLWE

In this section, we particularize again $K_{(T)} = \bigotimes_{i\in[l]} K_i$ and $R = \bigotimes_{i\in[l]} \mathcal{O}_{K_i}$ for the cyclotomic case $K_i = \mathbb{Q}(\varsigma_{m_i})$ with ς_{m_i} a primitive m_i-th root of unity. We also consider the prime

$q \equiv 1 \bmod m_i$ for all $i \in [l]$ and we have that it is poly(n)-bounded, where $n = \prod_{i\in[l]} \phi(m_i)$ is the degree of the considered multivariate polynomials. We recall that $K_{(T)}$ has a set of n different automorphisms τ_j with $j \in [n]$ (see Eq. (A.1)); when working over q, then $\langle q \rangle = \prod_{i\in[n]} \mathfrak{q}_i$ splits into a product of prime ideals $\mathfrak{q}_i$ where the automorphisms satisfy $\otimes_{i\in[l]} \tau^{(i)}_{k_i}(\mathfrak{q}_j) = \mathfrak{q}_{j'}$ for any prime ideals $\mathfrak{q}_j$, $\mathfrak{q}_{j'}$ where $k_i \in \mathbb{Z}^*_{m_i}$ and $j, j' \in [n]$ (see Section A.2.1).

We now present the main theorems about the reductions from the search version of m-RLWE (see Definition 3 and Theorem 9 about the reduction over worst-case lattice problems) to the average-case decision problem m-R-DLWE (see Definition 4).

Theorem 10 (Extended Theorem 5.1 of Lyubashevsky *et al.* [41])**.** *Let R and q be as shown previously and let $\alpha q \geq \eta_\epsilon(R^\vee)$ for some negligible $\epsilon = \epsilon(n)$. Then, there is a randomized polynomial-time reduction from m-R-LWE$_{q,\Psi_{\leq\alpha}}$ to m-R-DLWE$_{q,\Upsilon_\alpha}$.*

In order to prove the previous theorem we need four more reductions described in the following discussion.

$$\begin{tikzcd}
\text{LWE}_{q,\Psi} \arrow[r, "Automorphisms", "\text{Lemma 16}"'] & \mathfrak{q}_i\text{-LWE}_{q,\Psi} \arrow[r, "Search/Decision", "\text{Lemma 18}"'] & \text{WDLWE}^i_{q,\Psi}
\end{tikzcd}$$

$$\begin{tikzcd}
\text{WDLWE}^i_{q,\Psi} \arrow[r, "Worst/Average", "\text{Lemma 19}"'] & \text{DLWE}^i_{q,\Upsilon} \arrow[r, "Hybrid", "\text{Lemma 20}"'] & \text{DLWE}_{q,\Upsilon}
\end{tikzcd}$$

The details of the proof follow the steps of Lyubashevsky *et al.* [41], which, conversely, follows similar steps to the reductions of [39], the main point being the use of the automorphisms to recover the secret key s when only knowing the secret key relative to one prime ideal $\mathfrak{q}_i$ (Lemma 16). An additional needed step is the randomization of the error distribution (sampled from Υ) such that the error is invariant under the different field automorphisms (see Lemma 19) because the different $\psi \in \Psi_{\leq\alpha}$ are not necessarily invariant under the field automorphisms. Equivalently, if this reduction randomizing the error distribution is not desirable, we can apply a bound on the number of samples for considering a result about pseudorandomness of m-RLWE with a fixed spherical noise distribution.

Theorem 11 (Extended Theorem 5.2 of Lyubashevsky *et al.* [41])**.** *Let R, q and α be as in Theorem 10 and let $l \geq 1$. There is a randomized polynomial-time reduction from solving m-R-LWE$_{q,\Psi_{\leq\alpha}}$ to solving m-R-DLWE$_{q,D_\xi}$ given only l samples, where $\xi = \alpha \cdot (nl/\log(nl))^{1/4}$.*

In this case, we have a similar reduction to the one in Theorem 10 but considering a different lemma (Lemma 22 instead of Lemma 19 in one of the steps).

$$\begin{tikzcd}
\text{WDLWE}^i_{q,\Psi} \arrow[r, "Worst/Average", "\text{Lemma 22}"'] & \text{DLWE}^i_{q,D_\xi} \arrow[r, "Hybrid", "\text{Lemma 20}"'] & \text{DLWE}_{q,D_\xi}
\end{tikzcd}$$

It is worth noting that if we assume hardness of the search version with a spherical error distribution LWE$_{q,D_\xi}$, then there is also a reduction for the pseudorandomness with a spherical error, simplifying Lemma 19 instead of resorting to sampling from the Υ distribution.

A.4. Discussion and conclusions

This appendix formalizes the multivariate Ring Learning With Errors problem, including an extended reduction of the original Lyubashevsky *et al.* proof [40, 41] from hardness assumptions over ideal lattices.

As we have briefly mentioned in Section A.1, the multivariate RLWE problem is especially sensitive to the choice of modular functions. In fact, for "non-coprime" cyclotomic rings it is reduced from an easier problem than expected at a first glance; which implies a decrease in the effective dimension of the underlying lattice. Even so, the problem still introduces efficiency improvements in many practical signal processing applications (see Chapters 5, 7 and 8, and Appendix B) and also in many basic cryptographic primitives (see Chapter 3). In Chapters 5 and 2 we study in depth how we can search for *secure* instantiations of the multivariate RLWE problem (by *secure* we mean that there is no decrease on the effective dimension of the RLWE sample). This allow us to preserve both the efficiency and security originally claimed in [4].

A.A. Lattices and Algebraic Number Theory

This appendix reviews in depth the required concepts of lattices and algebraic number theory for the more general case of a tensor of number fields.

A.A.1. Gaussian Measures

We include several known results about Gaussian distributions that are needed (we refer the reader to Section 2.3.1 from Chapter 2 for more details).

Definition 16 (Smoothing parameter)**.** *The smoothing parameter $\eta_\epsilon(\Lambda)$ for a lattice Λ and real $\epsilon > 0$ is defined as the smallest r such that $\rho_{1/r}(\Lambda^*\backslash\{\mathbf{0}\}) \leq \epsilon$.*

In addition, several important lemmas from [41], [177], [39] and [178] about the relation between the smoothing parameter and properties of lattices are included below.

Lemma 5 (Lyubashevsky *et al.* [41] Lemma 2.2, Micciancio and Regev [177] Lemmas 3.2 and 3.3)**.** *For any n-dimensional lattice Λ, we have $\eta_{2^{-2n}}(\Lambda) \leq \sqrt{n}/\lambda_1(\Lambda^*)$ and $\eta_\epsilon(\Lambda) \leq \sqrt{\ln(n/\epsilon)}\lambda_n(\Lambda)$ for all $0 < \epsilon < 1$.*

Lemma 6 (Lyubashevsky *et al.* [41] Lemma 2.3, Micciancio and Regev [177] Lemma 4.1, Regev [39] Claim 3.8)**.** *For any lattice Λ, $\epsilon > 0$, $r \geq \eta_\epsilon(\Lambda)$, and $\boldsymbol{c} \in H_{(T)}$, the statistical distance*[5] *between $(D_r + \boldsymbol{c}) \mod \Lambda$ and the uniform distribution modulo Λ is at most $\epsilon/2$. Alternatively, we have $\rho_r(\Lambda + \boldsymbol{c}) \in \left[\frac{1-\epsilon}{1+\epsilon}, 1\right] \cdot \rho_r(\Lambda)$.*

Let a lattice Λ, a point $\boldsymbol{u} \in H_{(T)}$ and $r > 0$ with $r \in \mathbb{R}$, the discrete Gaussian probability distribution over $\Lambda + \boldsymbol{u}$ with parameter r can be defined as $D_{\Lambda+\boldsymbol{u},r}(\boldsymbol{x}) = \frac{\rho_r(\boldsymbol{x})}{\rho_r(\Lambda+\boldsymbol{u})}$ for all $\boldsymbol{x} \in \Lambda + \boldsymbol{u}$.

Lemma 7 (Banaszczyk [178], Lemma 1.5 (i))**.** *For any n-dimensional lattice Λ and $r > 0$, a sample point from $D_{\Lambda,r}$ has Euclidean norm at most $r\sqrt{n}$, except with probability at most 2^{-2n}.*

Lemma 8 (Regev [39])**.** *Let Λ be a lattice, let $\boldsymbol{u} \in H_{(T)}$ be any vector, and let $r, s > 0$ be reals. Assume that $1/\sqrt{1/r^2 + 1/s^2} \geq \eta_\epsilon(\Lambda)$ for some $\epsilon < 1/2$. Consider the continuous distribution Y on $H_{(T)}$ obtained by sampling from $D_{\Lambda+\boldsymbol{u},r}$ and then adding an element drawn independently from D_s. Then, the statistical distance between Y and $D_{\sqrt{r^2+s^2}}$ is at most 4ϵ.*

[5] The statistical distance $\Delta(X, Y)$ between two continuous random variables X and Y over $\mathbb{R}^n$ with probability density functions T_1 and T_2 is defined as $\Delta(X, Y) = \frac{1}{2}\int_{\mathbb{R}^n} |T_1(r) - T_2(r)|dr$. For more details we refer the reader to [177] and [39].

A.A.2. Algebraic Number Theory background

This section covers the main concepts related to number fields that are used in the works [41] and [45]; we highlight the theorems and lemmas that are fundamental to our proof and cannot fit in the main body of the appendix; even when they have already been presented in the literature, we include them here for completeness. We also particularize some of the results to the case of cyclotomic fields; for further details, we refer the reader to the two cited works or to any introductory book on the subject (e.g., [133]).

The concepts about algebraic number theory presented here are necessary to show which are the main changes needed to extend the proof of Lyubashevsky *et al.* to the generic multidimensional case (not only "coprime" factors), as explained in Section A.3.2.

Full-Rank Tensor Embedding: We can work with the embedding over the space H (see Appendix A.2.1) of any type of cyclotomic field. As a cyclotomic field can be decomposed in the tensor of power prime cyclotomic fields, it is easily shown that for that particular case of tensor of cyclotomic fields the embedding exists.

In our more general case, this relation with cyclotomic fields does not necessarily hold, so we cannot justify the existence of the tensor embedding by solely resorting to the existence of the embedding in an isomorphic cyclotomic field.

We can see that the embedding of a cyclotomic field (respectively, its corresponding ring of integers or the corresponding reduction modulo q) is equivalent to an invertible linear transformation from $\mathbb{Q}^{\phi(m_i)}$ (respectively, $\mathbb{Z}^{\phi(m_i)}$ or $\mathbb{Z}_q^{\phi(m_i)}$) to the corresponding subspace $H_i \subseteq \mathbb{C}^{n_i}$, where $n_i = \phi(m_i)$.

There are two properties of Kronecker products that allow us to justify the existence of the embeddings: (a) $\det(A \bigotimes B) = \det(B \bigotimes A) = (\det(A))^n (\det(B))^m$ where A and B are square matrices of size $n \times m$ and $m \times m$, respectively. This property states that $A \bigotimes B$ is non-singular (and therefore invertible) if and only if A and B are non-singular. (b) $(A \bigotimes B)^{-1} = A^{-1} \bigotimes B^{-1}$, which defines this inverse. See [179] for further details on the properties of the Kronecker product.

Our embedding can be defined as the Kronecker product of different invertible linear transformations (which correspond to the different embeddings for each cyclotomic field). Hence, by resorting to the properties of the Kronecker product, there exists the corresponding tensor embedding between the tensor of cyclotomic fields and the subspace $H_{(T)} = \bigotimes_{i \in [l]} H_i$ (see Appendix A.2.1).

Trace and Norm

Here we present the basic concepts of trace and norm over number fields that were proposed in previous works. Section A.2.2 in the main body of the Appendix A highlights which are the changes needed and how we can work with them when we have the tensor product of non coprime cyclotomic fields.

The trace $\text{Tr} = \text{Tr}_{K/\mathbb{Q}} : K \to \mathbb{Q}$ and norm $N = N_{K/\mathbb{Q}} : K \to \mathbb{Q}$ are defined as:

$$\text{Tr}(x) = \sum_{i \in [n]} \sigma_i(x), \ N(x) = \prod_{i \in [n]} \sigma_i(x). \tag{A.2}$$

In addition, the trace is a linear function in $\mathbb{Q}$ because $\text{Tr}(a+b) = \text{Tr}(a) + \text{Tr}(b)$ and $\text{Tr}(ca) = c\text{Tr}(a)$ for all $a, b \in K$ and $c \in \mathbb{Q}$. It is also important to note that $\text{Tr}(a \cdot b) = \sum_i \sigma_i(a)\sigma_i(b)$.

Even though we will put more emphasis on this later, we note that when working with tensor products $K_{(T)} = \bigotimes_i K_i$, resorting to the fact that $\sigma(\otimes_i a_i) = \otimes_i \sigma^{(i)}(a_i)$ the corresponding trace satisfies $\text{Tr}_{K_{(T)}/\mathbb{Q}}(\otimes_i a_i) = \prod_i \text{Tr}_{K_i/\mathbb{Q}}(a_i)$.

Tensor Ring of Integers and its Ideals

This section revises some basic properties of the ring of integers of a number field and its ideals. Although we are considering cyclotomic number fields $K_i = \mathbb{Q}(\varsigma_{m_i})$, these results apply to more general number fields. The ring of integers of a number field is denoted $\mathcal{O}_{K_i}$ and it is defined as the set of elements belonging to K_i that satisfy a monic polynomial $f(x)$ with coefficients belonging to the integers, that is, elements $a \in K_i$ such that $f(a) = 0$.

It can be seen that $\mathcal{O}_{K_i}$ is a free $\mathbb{Z}$-module[6] with rank the degree of K_i (when working with cyclotomic fields this degree is $\phi(m_i)$), and that its $\mathbb{Z}$-basis $B_i = \{b_1^{(i)}, \ldots, b_n^{(i)}\} \subset \mathcal{O}_{K_i}$ results to be a $\mathbb{Q}$-basis for K_i and also an $\mathbb{R}$-basis for $K_i \bigotimes \mathbb{R}$.

We work with the result of the tensor product of the different rings of integers which corresponds to each number field, that is, for the tensor of number fields $K_{(T)} = \bigotimes_{i\in[l]} K_i$ we consider the tensor ring of integers $R = \bigotimes_{i\in[l]} \mathcal{O}_{K_i}$. All the properties introduced for the ring of integers in [41] are also valid when working with ideals of the new multivariate polynomial ring R.

Firstly, we could see R as a $\mathbb{Z}$-module with rank $n = \prod_{i\in[l]} \phi(m_i)$ and its $\mathbb{Z}$-basis would be $\bigotimes_{i\in[l]} B_i \subset R$ that also results to be a $\mathbb{Q}$-basis for $K_{(T)}$ and an $\mathbb{R}$-basis for $K_{(T),\mathbb{R}}$.

Next, we include some important facts about the ideals of R. An integral ideal (a.k.a. ideal) of R is an additive subgroup that is closed under multiplication by R, that is, $r \cdot x \in \mathcal{I}$ for any $r \in R$ and $x \in \mathcal{I}$. In order to generate an ideal $\mathcal{I}$ of R, it can be shown that there exist two different elements $g_1, g_2 \in R$ whose R-linear combinations generate $\mathcal{I} = \langle g_1, g_2 \rangle$. An ideal is also a free $\mathbb{Z}$-module of rank n, so we have some basis $\{u_1, \ldots, u_n\} \subset R$.

The norm of an ideal is its corresponding index as an additive subgroup, that is, $N(\mathcal{I}) = |R : \mathcal{I}|$. The sum $\mathcal{I} + \mathcal{J}$ is also an ideal whose elements are all the pairs $x + y$ with $x \in \mathcal{I}$ and $y \in \mathcal{J}$, the product ideal $\mathcal{I}\mathcal{J}$ is the set of all finite sums of pairs xy with $x \in \mathcal{I}$ and $y \in \mathcal{J}$. The norm of ideals generalizes the previous definition of norm in the following way $N(\langle x \rangle) = |N(x)|$ with $x \in R$ and $N(\mathcal{I}\mathcal{J}) = N(\mathcal{I})N(\mathcal{J})$.

We say that two ideals $\mathcal{I}$ and $\mathcal{J}$ are coprime (or relatively prime) if $\mathcal{I} + \mathcal{J} = R$. An ideal $\mathfrak{p} \subsetneq R$ is prime if whenever $ab \in \mathfrak{p}$ for some $a, b \in R$, then $a \in \mathfrak{p}$ or $b \in \mathfrak{p}$. An ideal $\mathfrak{p}$ of R is prime if and only if it is maximal. The ring R has unique factorization on ideals, that is, every ideal of R can be expressed as a unique product of powers of prime ideals.

A fractional ideal $\mathcal{I} \subset K$ satisfies $d\mathcal{I} \subseteq R$ where $d\mathcal{I}$ is an integral ideal for some $d \in R$. Its norm is defined as $N(\mathcal{I}) = N(d\mathcal{I})/|N(d)|$.

[6] A free module is a module which has a basis.

Ideal Lattices

This Appendix relies on the lattices embedded by the fractional ideals in $K_{(T)}$ under the canonical embedding. Next, we describe some of their properties. A fractional ideal $\mathcal{I}$ has a $\mathbb{Z}$-basis $U = \{u_1, \ldots, u_n\}$. Then, under the canonical embedding σ, the ideal yields a rank-n ideal lattice $\sigma(\mathcal{I})$ with basis $\{\sigma(u_1), \ldots, \sigma(u_n)\} \subset H_{(T)}$. The lattice embedded by an ideal is commonly identified by the ideal, so we consider the minimum distance $\lambda_1(\mathcal{I})$ of an ideal.

The absolute discriminant Δ_K is defined for a field K. We generalize this term to the field tensor $K_{(T)}$, considering $\Delta_{K_{(T)}}$ as the square of the fundamental volume of the embedded lattice $\sigma(R)$. We also have $\Delta_{K_{(T)}} = |\text{det}(\text{Tr}(b_i \cdot b_j))|$, where $\{b_1, \ldots, b_n\}$ is an integral basis of R. Therefore, we can define the fundamental volume of an ideal lattice $\sigma(\mathcal{I})$ as $N(\mathcal{I}) \cdot \sqrt{\Delta_{K_{(T)}}}$.

The following lemma that gives upper and lower bounds on the minimum distance of an ideal lattice.

Lemma 9 (Extended version of Lyubashevsky *et al.* [41] Lemma 2.9, Peikert and Rosen [180] detailed proof)**.** *For any fractional ideal $\mathcal{I}$ in a field tensor $K_{(T)}$ of degree n, and in any l_p-norm for $p \in [1, \infty]$,*

$$n^{1/p} \cdot N(\mathcal{I})^{1/n} \overset{(a)}{\leq} \lambda_1(\mathcal{I}) \overset{(b)}{\leq} n^{1/p} \cdot N(\mathcal{I})^{1/n} \cdot \sqrt{\Delta^{1/n}_{K_{(T)}}}. \tag{A.3}$$

The proof of the previous Lemma 9 follows analogously to the proofs of the Lemmas 6.1 (upper bound) and 6.2 (lower bound) in [180].

First, we start with the upper bound (b) following the guidelines of [180]. Considering $||x||_p \leq n^{1/p}||x||_\infty$ for $x \in K_{(T)}$, we only need to prove the bound for the $p = \infty$ norm. For this purpose, we resort to Minkowski's Theorem 12 to bound the distance of λ_1^∞:

Theorem 12 (Minkowski's Theorem)**.** *Let Λ be any lattice of rank n and $\mathcal{B} \subseteq$ span (Λ) be any convex body symmetric about the origin having n-dimensional volume vol $(\mathcal{B}) > 2^n \cdot$ det (Λ). Then $\mathcal{B}$ contains some nonzero $\boldsymbol{x} \in \Lambda$.*

Now, we consider the n-dimensional closed $\mathcal{C} = \{\boldsymbol{x} \in H_{(T)} : ||\boldsymbol{x}||_\infty \leq 1\}$, and each $\phi(m_i)$-dimensional closed $\mathcal{C}^{(i)} = \{\boldsymbol{x} \in H_i : ||\boldsymbol{x}||_\infty \leq 1\}$. Knowing that $H_i \subseteq \mathbb{R}^{s_1^{(i)}} \times \mathbb{C}^{2s_2^{(i)}}$, it can be shown that the volume of $\mathcal{C}^{(i)}$ is $2^{\phi(m_i)} \cdot (\pi/2)^{s_2^{(i)}}$, where $\phi(m_i) = s_1^{(i)} + s_2^{(i)}$ and finally being $2^n \cdot (\pi/2)^{\prod_{i\in[l]} s_2^{(i)}}$ the volume of $\mathcal{C}$.

Proceeding as in [180], we have for any $\beta > N^{1/n}(\mathcal{I}) \cdot \sqrt{\Delta^{1/n}_{K_{(T)}}} \cdot (2/\pi)^{\prod_{i\in[l]} s_2^{(i)}/n}$

$$\text{vol}\,(\beta\mathcal{C}) = \beta^n \text{vol}\,(\mathcal{C}) > 2^n \cdot N\,(\mathcal{I}) \cdot \sqrt{\Delta_{K_{(T)}}} = 2^n \cdot \text{det}\,(\sigma\,(\mathcal{I}))\,,$$

where by Minkowski's Theorem 12, we know that $\beta\mathcal{C}$ contains a nonzero point of $\sigma\,(\mathcal{I})$, therefore $\lambda_1^\infty \leq \beta$; consequently, it also satisfies the upper bound (b) of Lemma 9.

Regarding the lower bound (a), we follow the steps of the proof for Lemma 6.2 in [180]. For $1 \leq p \leq \infty$, by the arithmetic mean/geometric mean inequality we have:

$$||x||_p^p = \sum_{i\in[n]} |\sigma_i\,(x)\,|^p \geq n \cdot \left(\prod_{i\in[n]} |\sigma_i\,(x)\,|^p\right)^{1/n} = n \cdot |N\,(x)\,|^{p/n};$$

by taking the p-root in both sides, this expression yields the lower bound (a), by considering that $|N(x)| \geq N(\mathcal{I})$ for any nonzero $x \in \mathcal{I}$ (for more details of both proofs we refer the reader to [180]). Here, it is important to note that by resorting to the concepts presented in Section A.2.1, we can deal with the different embeddings, even when working with the tensor of number fields.

Duality

For any lattice $\mathcal{L}$ in $K_{(T)}$ (this is the $\mathbb{Z}$-span of any $\mathbb{Q}$-basis of $K_{(T)}$), its dual is defined as:

$$\mathcal{L}^\vee = \{x \in K_{(T)} : \mathrm{Tr}(x\mathcal{L}) \subseteq \mathbb{Z}\}. \tag{A.4}$$

As in the "traditional" (non-tensor) number field case, using the canonical embedding, $\mathcal{L}^\vee$ embeds as the complex conjugate of the dual lattice, that is, $\sigma(\mathcal{L}^\vee) = \bar{\sigma}^*_{\mathcal{L}}$. Taking this into account and considering also that $\mathcal{L} = \bigotimes_{i\in[l]} \mathcal{L}_i$ and the dual operation commutes with tensoring, we have:

$$\begin{aligned}\sigma(\mathcal{L}^\vee) &= \sigma(\otimes_i \mathcal{L}_i^\vee) = \otimes_i \sigma(\mathcal{L}_i^\vee) = \otimes_i \bar{\sigma}^*(\mathcal{L}_i)\\ &= \overline{\otimes_i \sigma^*(\mathcal{L}_i)} = \overline{(\otimes_i \sigma(\mathcal{L}_i))^*} = \overline{\sigma^*(\otimes_i \mathcal{L}_i)} = \overline{\sigma^*(\mathcal{L})}.\end{aligned}$$

It is also easy to check that $(\mathcal{L}^\vee)^\vee = \mathcal{L}$ (tensoring commutes with dual), and that if $\mathcal{L}$ is a fractional ideal, its dual is also fractional. An important fact is that an ideal and its inverse are related by multiplication with the dual ideal of the ring: for any fractional ideal $\mathcal{I}$, its dual ideal is $\mathcal{I}^\vee = \mathcal{I}^{-1} \cdot R^\vee$. The factor $R^\vee$ (often called codifferent) is a fractional ideal whose inverse $(R^\vee)^{-1}$, called the different ideal, is integral and of norm $N((R^\vee)^{-1}) = \Delta_{K_{(T)}}$, the discriminant of $K_{(T)}$.

Ideal Lattice Problems

We revise here the computational problems over ideal lattices related to RLWE, and, by extension, to m-RLWE: the Shortest Vector Problem (SVP), Shortest Independent Vectors Problem (SIVP), and the Bounded Distance Decoding (BDD) Problem. The three problems can be restricted to the case of integral ideals over R (the tensor of ring of integers $\mathcal{O}_{K_i}$), analogously to the argument followed by Lyubashevsky *et al.* [40], [41] in the non-tensor case: if $\mathcal{I}$ is a fractional ideal with denominator $d \in R$ (such that $d\mathcal{I} \subseteq R$ is a integral ideal), then the ideal $N(d) \cdot \mathcal{I} \subseteq R$, because $N(d) \in \langle d \rangle$.

Definition 17 (SVP and SIVP). *Let $K_{(T)}$ be a tensor of number fields endowed with some geometric norm (e.g, the l_2-norm), and let $\gamma \geq 1$. The $K_{(T)}$-SVP$_\gamma$ problem in the given norm is posed as: given a fractional ideal $\mathcal{I}$ in $K_{(T)}$, find some nonzero $x \in \mathcal{I}$ such that $||x|| \leq \gamma \cdot \lambda_1(\mathcal{I})$. The $K_{(T)}$-SIVP$_\gamma$ problem is defined similarly, where the goal is to find n linearly independent elements in $\mathcal{I}$ whose norms are all at most $\gamma \cdot \lambda_n(\mathcal{I})$.*

Definition 18 (BDD). *Let $K_{(T)}$ be a tensor of number fields endowed with some geometric norm (e.g, the l_2 norm), let $\mathcal{I}$ be a fractional ideal in $K_{(T)}$, and let $d < \lambda_1(\mathcal{I})/2$. The $K_{(T)}$-BDD$_{\mathcal{I},d}$ problem in the given norm is: given $\mathcal{I}$ and y of the form $y = x + e$ for some $x \in \mathcal{I}$ and $||e|| \leq d$, find x.*

Chinese Remainder Theorem

The next lemma states that when the ring isomorphism from Lemma 3 exists, we can compute a CRT basis C for the set of pairwise coprime ideals $\mathcal{I}_1, \dots, \mathcal{I}_r$. The basis is composed by elements $c_1, \dots, c_r \in R$ that satisfy $c_i = 1 \bmod \mathcal{I}_i$ and $c_i = 0 \bmod \mathcal{I}_j$ when $i \neq j$. We can use that basis in order to invert the CRT isomorphism as follows: for any $w = (w_1, \dots, w_r) \in \bigoplus_i (R/\mathcal{I}_i)$, we have that $v = \sum_i w_i \cdot c_i \bmod \mathcal{I}$ is the unique element in $R/\mathcal{I}$ that maps to w with that ring isomorphism.

Lemma 10 (Efficient computable basis for isomorphism)**.** *There is a deterministic polynomial-time algorithm that, given coprime ideals $\mathcal{I}, \mathcal{J} \subseteq R$ (represented by $\mathbb{Z}$-bases), outputs some $c \in \mathcal{J}$ such that $c = 1 \bmod \mathcal{I}$. More generally, there is a deterministic polynomial-time algorithm that, given pairwise coprime ideals $\mathcal{I}_1, \dots, \mathcal{I}_r$, outputs a CRT basis $c_1, \dots, c_r \in R$ for those ideals.*

The following two lemmas enable an efficiently computable bijection between the quotient groups $\mathcal{I}/q\mathcal{I}$ and $\mathcal{J}/q\mathcal{J}$ for any fractional ideals $\mathcal{I}, \mathcal{J}$. They are important for clearing out the arbitrary ideal $\mathcal{I}$ in the BDD-to-LWE reduction:

Lemma 11 (Lyubashevsky *et al.* [41] Lemma 2.14)**.** *Let $\mathcal{I}$ and $\mathcal{J}$ be ideals in R. There exists $t \in \mathcal{I}$ such that the ideal $t \cdot \mathcal{I}^{-1} \subseteq R$ is coprime to $\mathcal{J}$. Moreover, such t can be found efficiently given $\mathcal{I}$ and the prime ideal factorization of $\mathcal{J}$.*

Lemma 12 (Lyubashevsky *et al.* [41] Lemma 2.15)**.** *Let $\mathcal{I}$ and $\mathcal{J}$ be ideals in R, let $t \in \mathcal{I}$ be such that $t \cdot \mathcal{I}^{-1}$ is coprime with $\mathcal{J}$, and let $\mathcal{M}$ be any fractional ideal in $K_{(T)}$. Then, the function $\theta_t : K_{(T)} \to K_{(T)}$ defined as $\theta_t(u) = t \cdot u$ induces an isomorphism from $\mathcal{M}/\mathcal{J}\mathcal{M}$ to $\mathcal{I}\mathcal{M}/\mathcal{I}\mathcal{J}\mathcal{M}$, as R-modules. Moreover, this isomorphism may be efficiently inverted given $\mathcal{I}$, $\mathcal{J}$, $\mathcal{M}$ and t.*

The proof of Lemma 12 for the case where $K_{(T)}$ is a tensor of cylotomic fields follows with the same techniques considered in [41], by taking into account that θ_t induces a homomorphism of R-modules because it represents a multiplication by a $t \in R$, so we do not include it here.

A.B. Proof of Theorem 9

This appendix presents the proof of Theorem 9. It is based on the iterative use of the following lemma:

Lemma 13 (Extended version of Lemma 4.2 Lyubashevsky *et al.* [41])**.** *Let $\alpha > 0$ and $q \geq 2$ be an integer. There exists an efficient quantum algorithm that, given a fractional ideal $\mathcal{I}$ in $K_{(T)}$, a number $r \geq \sqrt{2}q \cdot \eta_\epsilon(\mathcal{I})$ for some negligible $\epsilon = \epsilon(n)$ such that $r' = r \cdot \omega(\sqrt{\log n})/(\alpha q) > \sqrt{2n}/\lambda_1(\mathcal{I}^\vee)$, an oracle to m-R-LWE$_{q,\Psi_{\leq\alpha}}$, and a list of samples from the discrete Gaussian distribution $D_{\mathcal{I},r}$ (as many as required by the m-R-LWE$_{q,\Psi_{\leq\alpha}}$ oracle), outputs an independent sample from $D_{\mathcal{I},r'}$.*

Theorem 9 is proven as follows: we start with a value $r \geq 2^{2n}\lambda_n(\mathcal{I})$, in such a way that we can classically generate any polynomial number of samples from $D_{\mathcal{I},r}$. Given the samples from $D_{\mathcal{I},r}$, Lemma 13 can be used iteratively a polynomial number of times (using the same samples) to obtain a polynomial number of independent samples from $D_{\mathcal{I},r'}$ with $r' = r/2$ at each iteration. Repeating this process, we can obtain samples from increasingly narrower distributions, until we have samples from a distribution with parameter $s \geq \gamma$.

Lemma 13 is obtained thanks to the following two results (Lemmas 14 and 15):

Lemma 14 (Extended version of Lemma 4.3 of Lyubashevsky *et al.* [41], proof in Section 4.2)**.** *Let $\alpha > 0$, let $q \geq 2$ be an integer with known factorization, let $\mathcal{I}$ be a fractional ideal in $K_{(T)}$, and let $r \geq \sqrt{2}q \cdot \eta_\epsilon(\mathcal{I})$ for some negligible $\epsilon = \epsilon(n)$. Given an oracle for the discrete Gaussian distribution $D_{\mathcal{I},r}$, there is a probabilistic polynomial-time (classical) reduction from BDD$_{\mathcal{I}^\vee,d}$ in the l_∞ norm to m-R-LWE$_{q,\Psi_{\leq\alpha}}$, where $d = \alpha q/(\sqrt{2}r)$.*

Details for the proof of the lemma 14 follow the same steps of Lyubashevsky *et al.* for Lemma 4.3 in [41], so we do not replicate it here. However, we have to take into account that we are working with ideals over the tensor of the ring of integers, so instead of considering Lemmas 2.14 and 2.15 from [41] we have to use the redefined lemmas already presented in our work as Lemmas 11 and 12.

Lemma 15 (Extended version of Lemma 4.4 of Lyubashevsky *et al.* [41])**.** *There is an efficient quantum algorithm that, given any n-dimensional lattice Λ, a number $d' < \lambda_1(\Lambda^\vee)/2$ (where λ_1 is with respect to the l_2 norm), and an oracle that solves BDD on $\Lambda^\vee$ except with negligible probability for points whose offset from $\Lambda^\vee$ is sampled from $D_{d'/\sqrt{2n}}$, outputs a sample from $D_{\Lambda,\sqrt{n}/(\sqrt{2}d')}$. In particular, since a sample from $D_{d'/\sqrt{2n}}$ has l_∞ norm at most $d' \cdot \omega(\sqrt{\log n})/\sqrt{n}$ except with negligible probability, it suffices if the oracle solves BDD$_{\mathcal{I}^\vee,d}$ in the l_∞ norm, where $d = d' \cdot \omega(\sqrt{\log n})/\sqrt{n}$.*

The sketch of the proof for Lemma 13 is the following: starting with samples from $D_{\mathcal{I},r}$ and an oracle for m-R-LWE$_{q,\Psi_{\leq\alpha}}$ and resorting to Lemma 14 we can obtain an algorithm for BDD on $\mathcal{I}^\vee$ to within distance $d = \alpha q/(\sqrt{2}r)$ in the l_∞ norm. Next, considering Lemma 15 with $d' = d\sqrt{n}/\omega(\sqrt{\log n}) = \sqrt{n/2}/r' < \lambda_1(\mathcal{I}^\vee)/2$, we obtain a quantum procedure that outputs samples from the discrete Gaussian distribution $D_{\mathcal{I},r'}$.

A.C. Proofs of Theorems 10, 11 and 8

This section includes the proofs for the main results involving the security reductions of m-RLWE, as stated in Theorems 10, 11 and 8.

A.C.1. Search to Worst-Case Decision

Here we explain the two first reductions of Theorems 10 and 11. Next, we introduce the main definitions of the intermediate problems and the corresponding lemmas, and we also highlight the differences due to working with the tensor of rings of integers.

Definition 19 (Extended version of the $\mathfrak{q}_i$-LWE$_{q,\Psi}$ problem, Definition 5.4 from Lyubashevsky *et al.* [41])**.** *The $\mathfrak{q}_i$-LWE$_{q,\Psi}$ problem is defined as: given access to $A_{s,\psi}$ for some arbitrary $s \in R_q^\vee$ and $\psi \in \Psi$, find $s \bmod \mathfrak{q}_i R^\vee$.*

Lemma 16 (LWE to $\mathfrak{q}_i$-LWE, entending Lemma 5.5 of Lyubashevsky *et al.* [41])**.** *Suppose that the family Ψ is closed under all the automorphisms of $K_{(T)}$ (see Lemma 17), that is, $\psi \in \Psi$ implies that $\tau_k(\psi) \in \Psi$ for all $k \in [n]$. Then, for every $i \in [n]$, there exists a deterministic polynomial-time reduction from LWE$_{q,\Psi}$ to $\mathfrak{q}_i$-LWE$_{q,\Psi}$.*

The proof is based on the fact that by having an oracle for $\mathfrak{q}_i$-LWE and resorting to the different field automorphisms, we can recover s modulo $\mathfrak{q}_j R^\vee$ for every $j \in [n]$ and we can use the CRT for recovering s modulo $R^\vee$.

The reduction works in the following way: Let $(a, b) \leftarrow A_{s,\psi}$ and apply an automorphism $(\tau_k(a), \tau_k(b))$ that satisfies $\tau_k(\mathfrak{q}_j) = \mathfrak{q}_i$. Now, we use the $\mathfrak{q}_i$-LWE oracle with the transformed samples and we apply the reverse automorphism $\tau_k(t)^{-1} \in R^\vee/\mathfrak{q}_j R^\vee$ to its output $t \in R^\vee/\mathfrak{q}_i R^\vee$.

In order to see that $\tau_k(t)^{-1}$ has the desired value $s \bmod \mathfrak{q}_j R^\vee$, we operate with the pair $(\tau_k(a), \tau_k(b))$, with $\tau_k(b) = \tau_k(a) \cdot \tau_k(s)/q + \tau_k(e) \bmod R^\vee$ where we see that the pair follows the $A_{\tau_k(s),\tau_k(\psi)}$ distribution (we know that $\tau_k(\psi) \in \Psi$, see Lemma 17). Therefore, the oracle outputs $t = \tau_k(s) \bmod \mathfrak{q}_i R^\vee$ and Lemma 16 is proven.

Lemma 17 (Extended version of Lemma 5.6 of Lyubashevsky *et al.* [41])**.** *For any $\alpha > 0$, the family $\Psi_{\leq\alpha}$ is closed under every automorphism τ of $K_{(T)}$, that is, $\psi \in \Psi_{\leq\alpha}$ implies that $\tau(\psi) \in \Psi_{\leq\alpha}$.*

In order to see that for $\psi \in \Psi$ any possible automorphism also belongs to Ψ, we proceed as follows: each automorphism is the tensor of the existing automorphisms for each cyclotomic field, that is, $\otimes_{i\in[l]}\tau^{(i)}_{k_i}$ with $k_i \in \mathbb{Z}^*_{m_i}$. Hence, resorting to the definition of our error distributions (see Section A.2.1), we have $\psi = D_{\otimes_{i\in[l]}\boldsymbol{r}_i} \in \Psi_{\leq\alpha}$ where the elements of $\otimes_{i\in[l]}\boldsymbol{r}_i$ are bounded by α. As the effect of the automorphism simply permutes the coordinates of each $\boldsymbol{r}_i$, we can clearly see that $\otimes_{j\in[l]}\tau^{(j)}_{k_j}\left(D_{\otimes_{i\in[l]}\boldsymbol{r}_i}\right) = D_{\otimes_{i\in[l]}\boldsymbol{r}'_i}$ for $k_j \in \mathbb{Z}^*_j$, which also belongs to $\Psi_{\leq\alpha}$ because the value of the different elements follow being at most α (they have only been permuted).

Before stating Lemma 18 for the second reduction of the proof, we introduce two definitions for the intermediate problems:

Definition 20 (Extended Hybrid LWE Distribution of Lyubashvesky *et al.*[41])**.** *For $j \in [n]$, $s \in R^\vee_q$, and a distribution ψ over $K_{(T),\mathbb{R}}$, the distribution $A^j_{s,\psi}$ over $R_q \times \mathbb{T}$ is defined as follows: choose $(a, b) \leftarrow A_{s,\psi}$ and output $(a, b+h/q)$ where $h \in R^\vee_q$ is uniformly random and independent modulo $\mathfrak{q}_i R^\vee$ for all $i \leq j$, and is equal to zero modulo all the remaining $\mathfrak{q}_i R^\vee$. We also define $A^0_{s,\psi} = A_{s,\psi}$.*

Definition 21 (Extended WDLWE$^j_{q,\Psi}$ (Worst-Case Decision LWE Relative to $\mathfrak{q}_j$) of Lyubashevsky *et al.* [41])**.** *For $j \in [n]$ and a family of distributions Ψ, the WDLWE$^j_{q,\Psi}$ problem is defined as follows: given access to $A^i_{s,\psi}$ for arbitrary $s \in R^\vee_q$, $\psi \in \Psi$, and $i \in \{j-1, j\}$, find i.*

Lemma 18 (Extended version of Search to Decision of Lyubashvesky *et al.* [41])**.** *For any $j \in [n]$, there exists a probabilistic polynomial-time reduction from $\mathfrak{q}_j$-LWE$_{q,\Psi}$ to WDLWE$^j_{q,\Psi}$.*

The proof of the reduction is based on trying each of the different possible values of s modulo $\mathfrak{q}_j R^\vee$ in such a way that after modifying the samples from $A_{s,\psi}$, we have that (a) for the correct value, the samples are distributed following $A^{j-1}_{s,\psi}$ and (b) for the rest of possible values, they follow $A^j_{s,\psi}$.

We can try all different values for $s \bmod \mathfrak{q}_j R^\vee$ because the norm of $\mathfrak{q}_j$ for all j satisfies $N(\mathfrak{q}_j) = q = \text{poly}(n)$, so we can enumerate all the combinations. Finally, we can use the oracle WDLWE$^j_{q,\Psi}$ for distinguishing between the distributions $A^{j-1}_{s,\psi}$ and $A^j_{s,\psi}$.

Following an analogous procedure as the one in [41], given a sample $(a, b) \leftarrow A_{s,\psi}$, we have:

$$(a', b') = (a + v, b + (h + vg)/q) \in R_q \times \mathbb{T},$$

where $v \in R_q$ satisfies that it is uniformly random modulo $\mathfrak{q}_j$ and zero modulo other different prime ideal, $h, g \in R^\vee_q$, where h is uniformly random and independent modulo any $\mathfrak{q}_i R^\vee$ when

$i < j$, and it is zero for the rest of possible values of i. Finally, we have:

$$b' = (a's + h + v(g - s))/q + e,$$

with $e \leftarrow \psi$.

Now, choosing different values for g we have the following results: (a) if $g = s \bmod \mathfrak{q}_j R^\vee$, the distribution of (a', b') is $A^{j-1}_{s,\psi}$, and (b) if $g \neq s \bmod \mathfrak{q}_j R^\vee$, the distribution of (a', b') is $A^{j}_{s,\psi}$. Hence, we only have to enumerate different g values which satisfy different conditions modulo $\mathfrak{q}_j R^\vee$ (the values modulo other $\mathfrak{q}_i R^\vee$ with $i \neq j$ are not important) to achieve the reduction.

A.C.2. Worst-Case Decision to Average-Case Decision

The objective of this part is to cover the two last reductions of Theorems 10 and 11. For this purpose, we present some definitions and lemmas that allow us to reduce the worst-case decision $\text{WDLWE}^j_{q,\Psi}$ problem to an average-case problem $\text{DLWE}_{q,\Upsilon}$ where the goal is to distinguish between $A_{s,\psi}$ and uniform samples where the parameters of the error distribution are also secret and drawn from Υ.

Definition 22 (Extended version of Average-Case Decision LWE Relative to $\mathfrak{q}_j$ ($\text{DLWE}^j_{q,\Upsilon}$) of Lyubashevsky *et al.* [41])**.** *For $j \in [n]$ and a distribution Υ over error distributions, we say that an algorithm solves the $\text{DLWE}^j_{q,\Upsilon}$ problem if with a non negligible probability over the choice of a random $(s, \psi) \leftarrow U(R_q^\vee) \times \Upsilon$, it has a non negligible difference in acceptance probability on inputs from $A^j_{s,\psi}$ versus inputs from $A^{j-1}_{s,\psi}$.*

Lemma 19 (Extended version of Worst-Case to Average-Case Lemma 5.12 of Lyubashevsky *et al.* [41])**.** *For any $\alpha > 0$ and every $j \in [n]$, there is a randomized polynomial-time reduction from $\text{WDLWE}^j_{1,\Psi_{\leq\alpha}}$ to $\text{DLWE}^j_{q,\Upsilon_\alpha}$.*

In order to prove the previous lemma, let $s' \in R_q^\vee$, $\boldsymbol{r}' \in (\mathbb{R}^+)^n$, $k \in [n]$, and the pair (a, b), and consider the transformation $(a, b + (a \cdot s' + h)/q + e')$ where e' is drawn from $D_{\boldsymbol{r}'}$, $h \in R_q^\vee$ and h satisfies that $h \bmod \mathfrak{q}_i R^\vee$ are uniformly random and independent for $i \leq k$, and zero for all other i. Then, when the input is $A^j_{s,\psi}$, this transformation outputs $A^{\max\{k,j\}}_{s+s',\psi+D_{\boldsymbol{r}'}}$.

Now, to achieve the reduction, we repeat the following process a polynomial number of times: we draw $s' \in R_q^\vee$, and we have $\boldsymbol{r}' \in (\mathbb{R}^+)^n$ where $\boldsymbol{r}' = \bigotimes_{i\in[l]} \boldsymbol{r}'_i$ (as it was presented in Section A.2.1) and $r'_{i,j} = r'_{i,j+\phi(m_i)/2}$ with $i \in [l]$ and $j \in [\phi(m_i)]$. We also have $r'^2_j = \alpha^2\sqrt{n}x_j$ for all $j \in [n]$ and where the x_j are chosen not necessarily independently from $\Gamma(2,1)$. A sample from $D_{\boldsymbol{r}'}$ is given by $\sum_{j\in[n]} \sqrt{2^{l-1}}x'_j \boldsymbol{h}_j$ where the x'_j are Gaussian variables with parameter $\alpha^2\sqrt{n}\Gamma(\frac{2}{2^{l-1}}, 1)$ (l comes from the expression $n = \prod_{i\in[l]} n_i$), and the distribution of the gaussian variances r'^2_j is preserved thanks to the properties of the Γ distribution which satisfies $\sum_{i\in[2^{l-1}]} \Gamma(\frac{2}{2^{l-1}}, 1) = \Gamma(2,1)$.

Next, we estimate the acceptance probability of the oracle for two different input distributions: (a) applying to the input the previous transformation with parameters s', $\boldsymbol{r}'$ and $j-1$; (b) applying to the input the previous transformation with parameters s', $\boldsymbol{r}'$ and j. Finally, after a polynomial number of repetitions we output $j-1$ if there is a non negligible difference between the two acceptance probabilities; on the contrary, we output j.

Let us assume that the input distribution is $A^{j-1}_{s,D_{\boldsymbol{r}}}$ for some $\boldsymbol{r}$ where all $r_i \in [0, \alpha]$ for $i \in [n]$. Then, we have to estimate the acceptance probability of the oracle on $A^{j-1}_{s+s',D_{\boldsymbol{r}}+D_{\boldsymbol{r}'}}$ and

$A^j_{s+s',D_{\boldsymbol{r}}+D_{\boldsymbol{r}'}}$, and we notice that $D_{\boldsymbol{r}} + D_{\boldsymbol{r}'} = D_{\boldsymbol{r}''}$ where ${r''}^2_i = {r'}^2_i + r^2_i$. If we denote by S the set of pairs (s, ψ) for which the oracle has non negligible difference in acceptance probability between $A^{j-1}_{q,\psi}$ and $A^j_{q,\psi}$, we have by assumption (the measure of S under $U(R_q^\vee) \times \Upsilon_\alpha$ is non negligible) and by claim 2 below that $(s+s', D_{\boldsymbol{r}} + D_{\boldsymbol{r}'}) \in S$ with non negligible probability, and the proof of Lemma 19 is complete.

Our Claim 2 is a variant of Claim 5.11 presented by Lyubashevsky *et al.* [41]. For our case, we need a similar result, but it must hold not only for independent variables following a $\Gamma(2,1)$ distribution, because in our more general case, for $i \in [n]$ we can have that more than two x_i are equal. Therefore, we present a modification for vectors of coefficients distributed as $\Gamma(2,1)$, where they do not have to be independent, and we justify its validity.

Claim 2 (Extended Claim 5.11 from [41])**.** *Let P be the distribution* $\Gamma(2,1)^n$ *and Q be the distribution* $(\Gamma(2,1)-z_1)\times\cdots\times(\Gamma(2,1)-z_n)$ *for some* $0 \le z_1,\ldots,z_n \le 1/\sqrt{n}$ *where the different* $\Gamma(2,1)$ *of both P and Q do not have to be independent and some of them can be equal to each other. Then, any set* $A \subseteq \mathbb{R}^n$ *whose measure under P is non negligible also has non negligible measure under Q.*

The proof of the claim follows the next scheme: first, let $P, Q : \mathbb{R}^n \to \mathbb{R}^+$, where when $Q(\boldsymbol{x}) = 0$ we also have $P(\boldsymbol{x}) = 0$, and we define $R\left(P||Q\right) = \int_{\mathbb{R}^n} \frac{P(\boldsymbol{x})^2}{Q(\boldsymbol{x})} d\boldsymbol{x}$, considering that the fraction is zero when both the numerator and the denominator are zero.[7] By Cauchy-Schwarz inequality, we have for any non empty set $A \subseteq \mathbb{R}^n$,

$$\frac{\left(\int_A P(\boldsymbol{x})d\boldsymbol{x}\right)^2}{\int_A Q(\boldsymbol{x})d\boldsymbol{x}} \le \int_A \frac{P(\boldsymbol{x})^2}{Q(\boldsymbol{x})} d\boldsymbol{x} \le R\left(P||Q\right).$$

Thus, if we have a set A with non negligible measure under P and $R\left(P||Q\right) \le \text{poly}(n)$ holds, we can say that the set A has non negligible measure under Q.

For the particular setting of the Claim 2, when $z > 0$ we have

$$R\left(\Gamma(2,1)||\Gamma(2,1)-z\right) = e^z \left(1 - z + z^2 e^z \int_z^\infty x^{-1} e^{-x} dx\right),$$

and when z is small, this expression reduces to $1 + z^2 \log\left(1/z\right) + \mathcal{O}(z^2)$.

The difference regarding the proof of [41] relies on the following fact: if we compute $R\left(P||Q\right)$, we have:

$$\begin{aligned}&R\left(\Gamma(2,1)^n||\left(\Gamma(2,1)-z_1\times\cdots\times\Gamma(2,1)-z_n\right)\right)\\ &\le R\left(\Gamma(2,1)||\Gamma(2,1)-z_1\right)\cdot\ldots\cdot R\left(\Gamma(2,1)||\Gamma(2,1)-z_n\right),\end{aligned}$$

where the equality is achieved when all the components of each vector are independent. When some of the $\Gamma(2,1)$ variables are equal, we can see that the ratio of the corresponding distributions is equal to the ratio of only one of the variables of P and Q respectively.

Now, as we know that the second term of the expression is bounded by $\text{poly}(n)$, the claim is proven because for the setting of the claim our expression is bounded by the second term.

Lemma 20 (Extended version of Lemma 5.14 Hybrid by Lyubashevsky *et al.* [41])**.** *Let* Υ *be a distribution over noise distributions satisfying that for any* ψ *in the support of* Υ *and any* $s \in$

[7] The logarithm of $R(P||Q)$ is the Rényi divergence of order 2 [181].

$R_q^\vee$, the distribution $A^n_{s,\psi}$ is within negligible statistical distance from uniform. Then for any oracle solving the DLWE$_{q,\Upsilon}$ problem, there exists a $j \in [n]$ and an efficient algorithm that solves DLWE$^j_{q,\Upsilon}$ using the oracle.

The proof works as follows: consider a pair (s, ψ) for which the oracle can distinguish between $A_{s,\psi}$ and uniform distribution with a non negligible advantage. By Markov's inequality, the probability measure of those pairs is non negligible. Knowing that $A^0_{s,\psi} = A_{s,\psi}$ and that $A^n_{s,\psi}$ is negligibly far from the uniform distribution (see Lemma 21), we see that for each (s, ψ) we must have a $j \in [n]$ for which the oracle distinguishes between $A^j_{q,\psi}$ and $A^{j-1}_{q,\psi}$ with non negligible advantage. Finally, the lemma is proven if we take the j that is associated to the set of pairs (s, ψ) with the highest probability. With the proof of this lemma, the proof of the Theorem 10 is complete.

Lemma 21 (Adapted version of lemma 5.13 of Lyubashevsky *et al.* [41])**.** *Let $\alpha \geq \eta_\epsilon(R^\vee)/q$ for some $\epsilon > 0$. Then, for any ψ in the support of Υ_α and $s \in R_q^\vee$, the distribution $A^n_{s,\psi}$ is within statistical distance $\epsilon/2$ of the uniform distribution over $(R_q, \mathbb{T})$.*

The proof of this lemma is obtained by following the steps in [41] and taking into account the considered changes in our setting together with our Lemma 6.

Finally, we introduce the needed lemma for the reductions of Theorem 11.

Lemma 22 (Extended version of Lemma 5.16 of Lyubashevsky *et al.* [41] Worst-Case to Average–Case with Spherical Noise)**.** *For any $\alpha > 0$, $l \geq 1$, and every $j \in [n]$, there exists a randomized polynomial-time reduction from solving WDLWE$^j_{q,\Psi_{\leq\alpha}}$ to solving DLWE$^j_{q,D_\xi}$ given only l samples, where $\xi = \alpha(nl/\log{(nl)})^{1/4}$.*

In order to prove Lemma 22, we consider the transformation that we have already used for the proof of Lemma 19, but in this case the transformation has l different inputs. So, let $s' \in R_q^\vee$, $k \in [n]$, and $e_i \in \mathbb{T}$ for $i \in [l]$. Now, consider for the following l samples (a_i, b_i) the mentioned transformation $(a_i, b_i + (a_i \cdot s' + h_i)/q + e_i)$, where $h_i \in R_q^\vee$ and $i \in [l]$. It is important to note that all the h_i satisfy that they are independent and uniform modulo $\mathfrak{q}_d R^\vee$ for all $d \leq k$, and they are zero when d does not satisfy the previous relation. Therefore, if we take l independent inputs drawn from $A^j_{s,\psi}$ and we apply the transformation to all of them considering that all e_i are independently drawn from $D_{\boldsymbol{r}'}$, we have as output distribution $\left(A^{\max\{k,j\}}_{s+s',\psi+D_{\boldsymbol{r}'}}\right)^l$.

Now, the reduction repeats the following process a polynomial number of times: we consider $s' \in R_q^\vee$ and a set of independent e_i drawn from D_ξ. Next, we estimate the acceptance probability of the oracle for two different input distributions: (a) applying to the input the previous transformation with parameters s', e_i and $j-1$; (b) applying to the input the previous transformation with parameters s', e_i and j. After a polynomial number of repetitions, we output $j-1$ whenever a non negligible difference between the two acceptance probabilities is observed; otherwise, we output j.

Assuming the input distribution is $A^{j-1}_{s,D_{\boldsymbol{r}}}$, where all the coefficients of $\boldsymbol{r}$ are in $[0, \alpha]$ for the two previous cases, we have two different output distributions: $\left(A^{j-1}_{s+s',\psi+D_{\boldsymbol{r}'}}\right)^l$ and $\left(A^{j}_{s+s',\psi+D_{\boldsymbol{r}'}}\right)^l$. We also consider that the coefficients of $\boldsymbol{r}'$ verify $r'^2_i = \xi^2 - r_i^2$, so we have $D_{\boldsymbol{r}} + D_{\boldsymbol{r}'} = D_\xi$.

As with Lemma 19, let S be the set of all tuples $(s, e_1, \ldots, e_l)$ for which the oracle has a non negligible difference in acceptance probability on $\left(A^{j-1}_{s+s',\psi+D_{\boldsymbol{r}'}}\right)^l$ and $\left(A^{j}_{s+s',\psi+D_{\boldsymbol{r}'}}\right)^l$. By our assumption and a Markov argument, the measure of S under $U\left(R_q^\vee\right) \times (D_{\mathbf{r}'})^l$ is non negligible, and we have

$$1 \leq \frac{\xi}{\sqrt{\xi^2 - r_i^2}} \leq \frac{\xi}{\sqrt{\xi^2 - \alpha^2}} \leq 1 + \sqrt{\frac{\log(nl)}{nl}},$$

where thanks to Claim 3 below, we can assert that the measure of S is also non negligible under $U\left(R_q^\vee\right) \times (D_\xi)^l$, and where we can derive the condition $\xi = \alpha(nl/\log(nl))^{1/4}$, hence completing the proof of Lemma 22 and Theorem 11.

Claim 3 (Claim 5.15 from [41])**.** *Let $r_1, \ldots, r_n \in \mathbb{R}^+$ and $s_1, \ldots, s_n \in \mathbb{R}^+$ be such that for all i, $|s_i/r_i - 1| < \sqrt{(\log n)/n}$. Then any set $A \subseteq \mathbb{R}^n$ whose measure under the Gaussian distribution $D_{r_1} \times \cdots \times D_{r_n}$ is non negligible, also has non negligible measure under $D_{s_1} \times \cdots \times D_{s_n}$.*

A.D. Definition 4 over χ error distribution

Here, we present a variant of Definition 4 that we call m-R-DLWE$_{q,\chi}$ where we have a given number of samples from χ instead of ψ, and we have the problem of distinguishing between samples from $A_{s,\chi}$ and uniform samples from $R_q \times R_q^\vee$.

In order to guarantee the hardness of the discrete version, we follow the procedure described in [45]. We revisit the main lemmas together with some relevant explanations about the considerations needed for our multivariate case.

The following lemma states that if m-R-DLWE$_{q,\psi}$ is hard with l samples, then m-R-DLWE$_{q,\chi}$ is also hard for the same number of samples, being χ the distribution obtained from $\lfloor p \cdot \psi \rceil_{w+pR^\vee}$ and p and q coprime integers ($\lfloor \cdot \rceil$ denotes a valid discretization to cosets of $pR^\vee$, see [45]).

Lemma 23 (Extended version of Lemma 2.23 in [45])**.** *Let p and q coprime integers, and $\lfloor \cdot \rceil$ a valid discretization to cosets of $pR^\vee$. There exists an efficient transformation that on input $w \in R_p^\vee$ and a pair in $(a', b') \in R_q \times K_{(T),\mathbb{R}}/qR^\vee$ outputs $(a = pa' \bmod qR, b) \in R_q \times R_q^\vee$ with the following considerations: if the input pair is uniformly distributed then so is the output pair; and if the input pair is distributed according to the multivariate Ring-LWE distribution $A_{s,\psi}$ for some unknown $s \in R^\vee$ and distribution ψ over $K_{(T),\mathbb{R}}$, then the output is distributed according to $A_{s,\chi}$ where we have that $\chi = \lfloor p \cdot \psi \rceil_{w+pR^\vee}$.*

In order to show that the variant with short error (R-DLWE$_{q,\chi}$) is as hard as the original R-DLWE$_{q,\psi}$, [45] follows the technique of [58]. Their results can be adapted to our more general case, so we include below the relevant lemma:

Lemma 24 (Extended version of Lemma 2.24 in [45])**.** *Let p and q be positive coprime integers, $\lfloor \cdot \rceil$ be a valid discretization to cosets of $pR^\vee$, and w be an arbitrary element in $R_p^\vee$. If m-R-DLWE$_{q,\psi}$ is hard given some number l of samples, then so is the variant of m-R-DLWE$_{q,\chi}$ where the secret is sampled from $\chi = \lfloor p \cdot \psi \rceil_{w+pR^\vee}$, given $l - 1$ samples.*

The proof of the previous lemma relies on how to use an oracle of the second problem to solve the first one. The difference with respect the proof presented in [45] lies on how to compute the fraction of invertible elements of R_q. In order to resolve this, we resort to the following claim about cyclotomic fields:

Claim 4 (Claim 2.25 in [45])**.** *Consider the m-th cyclotomic field of degree $n = \phi(m)$ for some $m \geq 2$. Then, for any $q \geq 2$, the fraction of invertible elements in R_q is at least $1/poly\,(n, \log q)$.*

Proof. Our case deals with the tensor of cyclotomic fields $K_{(T)} = \bigotimes_{i \in [l]} K_i$; for each cyclotomic field K_i, the fraction of irreducible elements in $\mathcal{O}_{K_i}/\langle q \rangle$ is at least $1/\text{poly}\left(\phi(m_i), \log q\right)$ with $q \geq 2$ and with $q \equiv 1 \bmod m_i$ for all $i \in [l]$. When working in the tensor of the different polynomial rings over $\mathbb{Z}_q$, if an element is invertible, the corresponding elements belonging to each $\mathcal{O}_{K_i}$ must be invertible too (same explanation as for the Kronecker product of matrices, Section A.2.1). Then, the fraction of invertible elements in $R_q = \bigotimes_{i \in [l]} \mathcal{O}_{K_i}/\langle q \rangle$ is at least the product of the fractions of each ring of integers $1/\text{poly}\left(\prod_{i \in [l]} \phi(m_i), \log q\right) = 1/\text{poly}\left(n, \log q\right)$. □

Appendix B

Block-Processing

This appendix is adapted with permission from ArXiv: Alberto Pedrouzo-Ulloa, Juan Ramón Troncoso-Pastoriza, and Fernando Pérez-González. Multivariate Cryptosystems for Secure Processing of Multidimensional Signals. ArXiv e-prints, CoRR abs/1712.00848, December 2017.

B.1. Introduction

This appendix analyzes the applications of the m-RLWE problem on signal processing scenarios (initially introduced in the conference paper [4]), and shows why it better suits multidimensional signals (e.g., 2-D and 3-D images, video, ...). By rooting the used SHE cryptosystems in this hard problem, we show that we can achieve a reduction of both the computational cost and the cipher expansion along with an increase in the security when working with multidimensional signals and a *secure* multivariate RLWE instantiation (see Chapters 2 and 5 for a detailed discussion on the security of multivariate RLWE). This is so due to the more compact and efficient representation of the signals that outperforms the direct use of packing and unpacking steps in RLWE-based cryptosystems. Furthermore, we show that the use of m-RLWE is compatible with other methods, so it can be combined with packing techniques and CRT (Chinese Remainder Theorem) [131], which can be leveraged for parallelizing cleartext operations under encryption [50, 98, 29]. We therefore achieve our *first goal of efficient and practical encrypted processing of multidimensional signals*.

Besides its benefits for multidimensional signals, it must be noted that the m-RLWE problem yields further degrees of freedom which can be leveraged to exploit additional structures (not necessarily related to the dimensions of the data) in the data or operations. These structures can be recognized, for example, when processing several signals in parallel or when applying block-wise operations. Therefore, we can achieve *performance and security gains with respect to univariate RLWE in a variety of applications*, especially comprising multi-scale approaches [46, 182]; these are used, among others, in disciplines like geology, astrophysics, biology, imagery, medicine; being the latter one of the most relevant due to its privacy constraints. Furthermore, m-RLWE also *enables a new type of homomorphic operations which are independent of the dimensions presented on both the signals and scenarios*.

Contributions: Here we summarize and briefly describe the contributions of this appendix:

- We present a toolset of multidimensional secure operations enabled by the m-RLWE problem (see Section B.3), comprising: (a) better encrypted packing of information, (b) unattended encrypted divisions without resorting to interactive protocols, and (c) multi-scale approaches as wavelet transforms and pyramids.
- We analyze the use of pre- and post-processing to enable unattended packed and block-processing operations. Additionally, NTTs (Number Theoretic Transforms) are proposed as a means to optimize the encrypted operations (see Section B.4).
- We develop strategies to homomorphically modify the structure of ciphertexts by incorporating some additional information, and without the need of an interactive protocol with the secret key owner, hence enabling different types of unattended secure operations (see Section B.5).

Structure: The rest of the appendix is structured as follows: Section B.2 briefly revisits some basic concepts of homomorphic cryptosystems and the underlying hard problems. Section B.3 introduces a set of possible encrypted unattended applications for which m-RLWE brings about notable optimizations; Section B.4 includes the description of the main tools proposed in this appendix, and Section B.5 proposes an optimization which enables to homomorphically update the ciphertext structure.

B.2. Preliminaries

The state of the art in FHE is based on the Learning with Errors (LWE) [183] and Ring Learning with Errors (RLWE) problems [41], which have proven security reductions from hard lattice problems. Both RLWE leveled cryptosystems [50], which enable the homomorphic execution of a bounded-degree polynomial function, and scale-invariant leveled cryptosystems based on RLWE produce the currently most efficient FHE systems [85, 86, 87].

Both RLWE and LWE have a similar formulation, that Brakerski *et al.* generalized to a common General Learning with Errors (GLWE) problem [50]. We recall a slightly adapted informal definition of GLWE, as the basis for our schemes introduced in the next sections:

Definition 23 (GLWE problem [50]). *Given a security parameter λ, an integer dimension $l = l(\lambda)$, two univariate polynomial rings $R[x] = \mathbb{Z}[x]/(f(x))$, $R_q[x] = \mathbb{Z}_q[x]/(f(x))$ with $f(x) = x^n+1$, $q = q(\lambda)$ a prime integer, $n = n(\lambda)$ a power of two, and an error distribution $\chi[x] \in R_q[x]$ that generates small-norm random univariate polynomials in $R_q[x]$, GLWE$_{l,f,q,\chi}$ relies upon the computational indistinguishability between pairs of samples $(\boldsymbol{a}_i, b_i = \boldsymbol{a}_i \cdot \boldsymbol{s} + t \cdot e_i)$ and $(\boldsymbol{a}_i, u_i)$, where $\boldsymbol{a}_i \leftarrow R_q^l[x]$, $u_i \leftarrow R_q[x]$ are chosen uniformly at random, $\boldsymbol{s} \leftarrow \chi^l[x]$ and $e_i \leftarrow \chi[x]$ are drawn from the error distribution, and t is an integer relatively prime to q.*

When $n = 1$, GLWE becomes the standard LWE$_{l,q,\chi}$, and when $l = 1$ it reduces to RLWE$_{q,f,\chi}$. LWE-based cryptosystems yield huge expansion factors and are computationally demanding, reason why RLWE was defined as an algebraic version of LWE, trading subspace dimensionality for polynomial ring order (using an ideal ring), and achieving huge efficiency improvements. As for the generic GLWE ($n > 1$ and $l > 1$), Brakerski *et al.* [50] speculate that it is hard for $n \cdot l = \Omega\left(\lambda \log(q/B)\right)$, where B is a bound on the length of the elements output by $\chi[x]$. It must be noted that despite the efficiency improvement, there are no known attacks in RLWE that get a substantial advantage with respect to attacks to LWE.[1] Hence, the currently most efficient homomorphic cryptosystems are based on RLWE, particularly BGV [50, 78] and NTRU [84], together with their scale-invariant counterparts FV [86] and YASHE [87]; depending on the requirements of the specific application, the optimal choice of the used RLWE-based cryptosystem can be different as analyzed by Costache and Smart in [83].

In [4, 22] we proposed a generalization of RLWE as a new problem called m-RLWE (multivariate Ring Learning with Errors), providing an exemplary new cryptosystem based on it, especially designed for encrypted image filtering. The m-RLWE hardness assumption is especially useful for working with multidimensional signals; for simplicity of the exposition, this appendix works with cryptosystems extending Lauter's cryptosystem [79] (a simpler non-leveled version of BGV), but the same methodology can be applied to any other RLWE-based cryptosystem as those previously cited.

It is worth noting that the contributions of this appendix are exemplified considering the m-RLWE problem instantiated with power-of-two cyclotomic modular functions. The formulation of this variant of the m-RLWE problem can be found in Chapters 2 and 5 (see Definition 1).

B.3. Applications of m-RLWE for Secure Computation

This section discusses how the m-RLWE problem can enable encrypting multidimensional information while still preserving its structure. As we show, this can be achieved with only a small overhead on cipher expansion with respect to the version in the clear, enabling additive and multiplicative homomorphisms.

We briefly recall first the example cryptosystem presented in [4] and the use of m-RLWE for performing encrypted multidimensional linear convolutions. Next, we introduce a set of practical scenarios where the m-RLWE problem can produce effective solutions. These methods are not exclusive for multidimensional signals, so they can also be of benefit to unidimensional signals. Among the proposed solutions, we find a better way to pack the information, we enable encrypted divisions without an interactive protocol, and we implement encrypted versions of several multiscale algorithms (e.g., wavelet transforms and pyramids) which are widely used in both computer vision and signal processing applications. We provide here a high level description for these solutions, and detail the proposed underlying mechanisms in Section B.4.

B.3.1. An example of an m-RLWE based Cryptosystem

Any cryptosystem whose security is based on RLWE (e.g., [50, 78, 79, 84, 86, 87]) could be extended to m-RLWE (see Chapter 5). In [4], we extended Lauter *et al.*'s [79], due to its efficiency and security, as a basis to exemplify the main properties of a semantically secure m-RLWE-based cryptosystem. Table B.1 summarizes its parameters and primitives. There are currently more efficient choices like FV [86] or BGV [50], but we prefer to abstract the peculiarities of the high level cryptosystem functions and focus on the actual functionalities that our proposed mechanisms enable. Our results can be straightforwardly extended to more efficient cryptosystems in case it is required.

The cryptosystem in Table B.1 supports both additions (the smallest ciphertext is previously

[1] For a formal definition of the GLWE problem and proofs of security reductions for RLWE and LWE, we refer the reader to [41, 50, 183].

Table B.1: Proposed Cryptosystem: Parameters and Primitives.

Parameters		
Let $R_t[x_1, \dots, x_m]$ be the cleartext ring and $R_q[x_1, \dots, x_m]$ as ciphertext's. The noise distribution $\chi[x_1, \dots, x_m]$ in $R_q[x_1, \dots, x_m]$ takes its coefficients from a spherically-symmetric truncated i.i.d Gaussian $\mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$. q is a prime $q \equiv 1 \mod 2 \max\{n_1, \dots, n_m\}$ (with $n = \prod n_i$), and $t < q$ is relatively prime to q.		
Cryptographic Primitives		
SH.KeyGen	Process	$s, e \leftarrow \chi[x_1, \dots, x_m]$, $a_1 \leftarrow R_q[x_1, \dots, x_m]$ $sk = s$ and $pk = (a_0 = -(a_1 s + te), a_1)$
SH.Enc	Input	$pk = (a_0, a_1)$ and $m \leftarrow R_t[x_1, \dots, x_m]$
	Process	$u, f, g \leftarrow \chi[x_1, \dots, x_m]$ and the fresh ciphertext is $\boldsymbol{c} = (c_0, c_1) = (a_0 u + tg + m, a_1 u + tf)$
SH.Dec	Input	sk and $\boldsymbol{c} = (c_0, c_1, \dots, c_{\gamma-1})$
	Process	$m = \left(\left(\sum_{i=0}^{\gamma-1} c_i s^i\right) \mod q\right) \mod t$
SH.Add	Input	$\boldsymbol{c}_0 = (c_0, \dots, c_{\beta-1})$ and $\boldsymbol{c}_1 = (c'_0, \dots, c'_{\gamma-1})$
	Process	$\boldsymbol{c}_{add} = (c_0 + c'_0, \dots, c_{\max(\beta,\gamma)-1} + c'_{\max(\beta,\gamma)-1})$
SH.Mult	Input	$\boldsymbol{c}_0 = (c_0, \dots, c_{\beta-1})$ and $\boldsymbol{c}_1 = (c'_0, \dots, c'_{\gamma-1})$
	Process	Using a symbolic variable v their product is $\left(\sum_{i=0}^{\beta-1} c_i v^i\right) \cdot \left(\sum_{i=0}^{\gamma-1} c'_i v^i\right) = \sum_{i=0}^{\beta+\gamma-2} c''_i v^i$

zero-padded) and multiplications between ciphertexts which are composed by $\gamma \geq 2$ ring elements from $R_q[x_1, \dots, x_m]$. This encryption size increases with each multiplication (see Table B.1), and it can be brought back to the size of a fresh cipher by means of a relinearization step, which involves using partial encryptions of the secret key (more details can be found in [50, 79], and Section B.5).

Security and Correctness: The security of the cryptosystem is based on the computational difficulty of reducing the n-dimensional lattice ($n = \prod n_i$) generated by the secret key, and on the semantic security guaranteed by the underlying m-RLWE problem (two encryptions of the same or different plaintexts cannot be distinguished). As we have discussed in other chapters, the hardness of the m-RLWE problem can vary substantially depending on the chosen modular functions. For the worst-case (see Chapter 5), it is roughly equivalent to the RLWE problem considering the maximum univariate degree; that is, for Definition 1 the effective dimension is $\max\{n_1, \dots, n_m\}$. Even so, all the proposed solutions in this appendix can be adapted to work with *secure* m-RLWE instantiations which do not suffer a decrease on the effective lattice dimension.[2]

As for correctness, q must be set such that enough "space" is guaranteed to avoid decryption errors produced by wrap-arounds of the performed homomorphic operations. Due to the analogous (not isomorphic) polynomial structure of m-RLWE with $n = \prod n_i$ and n-degree RLWE, bounds for the error norm [79] are preserved when switching from RLWE to m-RLWE, by adjusting the increased dimensionality of the ring elements: for D successive products between fresh ciphertexts and A sums, the needed q for correct decryption is lower-bounded by

$$q \geq 4(2t\sigma^2\sqrt{n_1 n_2 \dots n_m})^{D+1}(2n_1 n_2 \dots n_m)^{D/2}\sqrt{A}. \tag{B.1}$$

[2] We refer the reader to Chapters 2 and 3 for a detailed discussion on the choice of secure multivariate RLWE instantiations and how to adapt the results of this appendix to them.

B.3.2. Encrypted Multidimensional Linear Convolutions

Unlike RLWE-based cryptosystems, which lack support for multidimensional signals, the proposed cryptosystem [4] introduces a natural way to work with multidimensional linear operations. Additionally, it achieves a more compact representation of the data, as it can effectively encrypt one signal value per coefficient of the encryption polynomial. We exemplify here the implementation of different representative encrypted processing operations like convolutions, correlations or filtering, showing the advantages of the proposed cryptosystem compared to its RLWE-based counterpart. Unless otherwise stated, we always consider that all the used signals and filters are encrypted, to fully conceal all the involved elements in an untrustworthy environment.

Convolutions, correlations and filtering can all be expressed as a linear convolution between two m-dimensional signals $\boldsymbol{X}$ and $\boldsymbol{H}$, namely $\boldsymbol{Y}[n_1, \ldots, n_m] = \boldsymbol{X}[n_1, \ldots, n_m] * \boldsymbol{H}[n_1, \ldots, n_m]$, which is equivalent to the ring product of the signals represented as multivariate polynomials $y(z_1, \ldots, z_m) = x(z_1, \ldots, z_m) \cdot h(z_1, \ldots, z_m)$. Using the original RLWE-based scheme, an encrypted convolution would comprise encoding each dimension of the two signals separately as elements of the univariate polynomial ring $R_t[z]$, resulting in two $(m-1)$-dimensional elements $\boldsymbol{X}_{n_1,\ldots,n_{m-1}}(z)$ and $\boldsymbol{H}_{n_1,\ldots,n_{m-1}}(z)$ of $R_t^{m-1}[z]$. If $N_{n_i,y}$ is the number of samples in dimension n_i for the signal y, the number of involved polynomial products is $\prod_{i=1}^{m-1} N_{n_i,x} N_{n_i,h}$ (i.e., $N^{2(m-1)}$ if $N_{n_i,x} = N_{n_i,h} = N$).

Contrarily, with our proposed cryptosystem the convolution can be done through a single polynomial product of the encryptions, homomorphic to the polynomial product of the clear text. In particular, an encrypted image convolution with the proposed cryptosystem would translate into the product of two bivariate polynomial encryptions.

Complex signals: m-RLWE also enables to naturally incorporate one extra variable to cope with complex signals, represented in the polynomial ring $\mathbb{Z}_t[w]/(w^2+1)$, isomorphic to the complex integers ring, where the variable w plays the role of the imaginary unit.

Edge Detection Algorithms: As an example of multidimensional convolutions, the Sobel operator is frequently used in image processing and computer vision applications as part of edge detection algorithms. Resorting to the homomorphic product property of the m-RLWE cryptosystem, we can easily convolve the Sobel kernel (any other different type of kernel could be considered) with the encrypted image (even a 3D image).

Additionally, if the kernel operator is public, it can be in the clear when convolving it with the encrypted image, hence being its homomorphic execution even more efficient.

B.3.3. Better Encrypted Packing

It can be seen that for practical image processing scenarios it is not so common to filter the whole image. In fact, images are usually divided in different blocks and independent operations are applied to each block.

The approach introduced in [4] applied to this scenario would encrypt each block separately. However, this would not benefit from the use of 2-RLWE (m-RLWE with bivariate polynomials) because we would not be encrypting the whole image in only one ciphertext.

In order to preserve the same security (related to the dimension of the underlying bivariate lattices) as in [4], we propose different mechanisms to pack the information by exploiting the block-structure of the operation and restructuring the signals into "virtual" dimensions that can be leveraged by an m-RLWE encryption.

Instead of encrypting each block independently, we can consider one additional polynomial variable for representing the image like a video sequence where each frame corresponds to a different image block (see Figure B.1). Therefore, we can get an optimal packing of the information while preserving and exploiting the block structure in the encrypted domain.

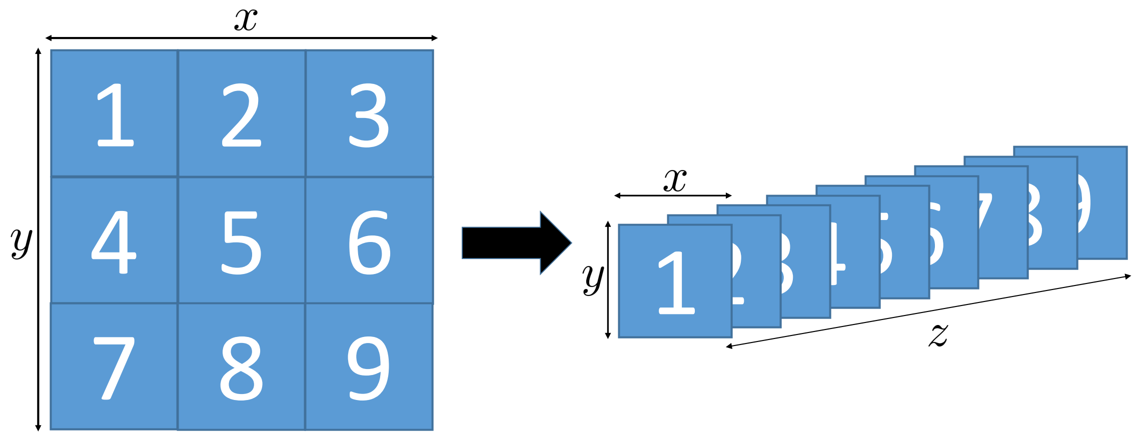


Figure B.1: Indexing a set of blocks with 3-RLWE.

The division of an image into blocks is not the unique additional dimension that we can consider and, in any real application, we can also take into account the number of plaintext signals which we want to work with. One would traditionally have to encrypt each signal in a different ciphertext, ending up with as many ciphertexts as plaintext signals in the process. To this end, we can use m-RLWE as an optimization which enables packing several signals in only one ciphertext, therefore using a smaller number of ciphertexts or even just one ciphertext.[3]

For this purpose, we only have to consider one additional polynomial variable that indexes the different messages which are encrypted inside the ciphertext. For example, when dealing with simple images we would use 3-RLWE (for 3D-images we would resort to 4-RLWE) in such a way that two polynomial variables would define the content of each image, and the third variable would define the "index" of the chosen image (see Figure B.1).

It is easy to find further scenarios where this strategy can be applied. For example, when considering the different color layers of the images we can encrypt each layer in a different polynomial variable; hence having a total of 7 dimensions (R, G, B layers, horizontal and vertical spatial dimensions, the block structure of the images and the number of images). This highlights the versatility of m-RLWE.

The difficulty of the implementation in the encrypted domain can vary depending on the operations performed on each block or signal. For example, the computational cost will be smaller or higher depending on whether all the operations are, respectively, the same or different for each block. All the details of the underlying primitives are explained in Section B.4.

[3] Depending on the choice of the modular functions we also can benefit of an increase in security (higher dimensionalities in the underlying lattices; see Chapter 5).

An example of Block Image Processing: A paradigmatic example of block processing in computer vision can be found in the JPEG compression method, where one step requires to divide the image in blocks of 8×8 pixels and apply a DCT (Discrete Cosine Transform) to each block.

Chapter 4 describes how to efficiently apply a known linear transform to a signal which has been previously encrypted by a RLWE-based cryptosystem. These techniques can also be applied to this block-wise processing scenario, however the size of needed relinearization matrix would become $2n^2\lceil \log_t q \rceil$ coefficients modulo q (considering a single layer image with n pixels).

Our proposed strategy encodes the block structure with an additional variable. This enables a reduction in the size of the required relinearization matrix, resulting in $128n\lceil \log_t q \rceil$ coefficients modulo q for an image with n pixels (we would have to generate the vectors $\boldsymbol{a}, \boldsymbol{b}$ which are composed of $64\lceil \log_t q \rceil$ polynomials with n coefficients).

B.3.4. Unattended Encrypted Divisions and Homomorphic Modular Reductions

A recurrent problem in Secure Signal Processing is the cipher blow-up of the obtained results after several encrypted operations in iterative processes, as a result of the accumulation of the multiplicative factor whenever the encryptions are not "refreshed" after each iteration [19]. For mitigating the effect of this overflow we could increase the available space for the encrypted messages (the modulo t in an m-RLWE cryptosystem; see Table B.1), or consider a homomorphic integer division or quantization after each encrypted iteration (removing the accumulated factor).

In the literature we can find several approaches for computing a secure integer division $\lfloor \frac{a}{b} \rfloor$, but all of them resort to interactive protocols (e.g., [184, 185, 186]), and they commonly consider that the denominator b in the division is public ([186] keeps it private).

We briefly discuss how to tackle non-interactive encrypted quantizations by resorting to the flexibility of the m-RLWE formulation, by including additional (i.e., virtual) polynomial variables. This enables the execution of both real and integer divisions, at the cost of an increase on the cipher expansion.

First, we deal with unattended encrypted integer divisions (always considering that the denominator is public), and then we address how to encode real numbers.

Integer Divisions

We can add one variable representing the binary encoding of the different messages (either signal samples or pixels when dealing with images). This implies an increase on the cipher expansion as we encode each value using one polynomial instead of only one coefficient. Thanks to this increase in the cipher expansion (and with the use of binary masks), we enable encrypted integer divisions with a denominator power of 2.

For performing these integer divisions we can leverage the tools from [29], where we show how to perform shifts and element-wise products between two encrypted messages in an unattended way; the secret key owner only has to generate several relinearization matrices which allow the server to recover the original structure of the ciphertexts after the different operations. Thus, if we work with the binary representation of the different values, we only have to apply a mask which discards the bit(s) with the smallest significance, and afterwards, homomorphically perform the corresponding binary shift.

The efficiency of such scheme is severely limited by the use of a binary decomposition, so we can look for a tradeoff that enhances the performance: instead of encoding each value using its binary decomposition, we can use a representation in any other base $b > 2$. This considerably reduces the cipher expansion while still being able to perform a reduced set of integer divisions by powers of the new base.

Additionally, it is worth noting that the encryption does not hold information about the carries in each position (when they have been previously undergone another homomorphic operations), so the performed divisions could contain errors. To address this, we can adapt the homomorphic threshold function presented in [46] to homomorphically compute the existing carries in each position, therefore correcting the results.

Working with Real Numbers

The same additional variable used in the previous paragraphs can be used for a fixed point representation of real numbers. For example, we can use the binary encoding of [187]. Hence, the polynomial $b_0+b_1v+\ldots+b_{N+}v^{N_+}-b_{-1}v^{n_v-1}-b_{-2}v^{n_v-2}-\ldots-b_{N_-}v^{n_v-N_-}$ that belongs to the ring $R_2[v] = \mathbb{Z}_2[v]/v^{n_v}+1$ encodes the real number $b_{N_+}\ldots b_1b_0.b_{-1}b_{-2}\ldots b_{-N_-}$ in base two. After a product of two polynomials encoding two real numbers, if the number of coefficients in the polynomial is big enough for storing the new integer and decimal parts, we obtain a polynomial that encodes the desired result.

This encoding enables multiplications between real numbers and also real divisions in fixed-point. After an encrypted division between real numbers, we can apply a mask for rounding the corresponding result, hence achieving a better control on the increase of the encrypted values after the homomorphic operations. Analogously, as in the case of integer divisions, we can consider a base $b > 2$ for the real fixed point representation.

B.3.5. Multi-Scale Approaches

Both signal processing and computer vision make extensive use of multi-scale representations to work with the content of a signal or image [188]. In essence, they aim at finding describing structures of the content by means of representing the information as a one-parameter family of smoothed signals which we call the scale-space representation.

Among the most widespread multi-scale approaches, we can highlight pyramids (e.g., Gaussian and Laplacian pyramids) and wavelet transforms (e.g., Gabor and Haar wavelets). In general, both cases require the use of a chain of downsampling and filtering operations. The use of 2-RLWE to perform wavelet-based operations was introduced in [46] (we revise it in Chapter 7), where we exemplify how to homomorphically perform the denoising of an image in an unattended way. By combining m-RLWE-based cryptosystems with the tools introduced in [29], which enable the computation of changes on the sampling rate, we can efficiently perform multi-scale processing like wavelet filters and pyramids.

The set of possible applications [182] enabled by these techniques is really wide and covers some very diverse applications. Among all of them, applications related to medical scenarios are more amenable for the presented solutions, due to their intrinsic privacy constraints. In these scenarios, we can consider several applications dealing with highly sensitive data, like Electrocardiograms - ECG, Electroencephalograms - EEG, Computer Tomography scans, Magnetic Resonance Imaging - MRI, fMRI, among others.

B.4. Encrypted Toolset based on m-RLWE

As mentioned above, image processing commonly relies heavily on block-wise processing. This section explains in detail how the block structure of these operations can be incorporated into m-RLWE ciphertexts to take advantage of the multivariate structure and the m-RLWE formulation. It is worth noting that while we exemplify solutions for image processing scenarios due to their typical block-wise operations, all the results are equally valid and applicable for any scenario dealing with multidimensional signals.

B.4.1. Block Processing

First, we consider the case where the same processing is applied to each block. The straightforward approach would be to encrypt each block separately and filter each encrypted block independently, effectively considering every block as a different signal. However, we can leverage the m-RLWE structure and, instead of encrypting each block separately, we include one additional variable to the encrypted polynomials which assigns one block per coefficient and enables processing different blocks in parallel without separating them (for the case of images that are divided in several blocks, the equivalent would be to use 3-RLWE for coding the image as a video where each frame is one of the different blocks). That is, incorporating an "index" variable to address the block structure, we can work with only one ciphertext for all the blocks or signals.

If we apply under encryption a filter defined in those variables that represent the dimensions of the blocks, we can effectively work with ciphertexts whose underlying lattice dimensionality is much higher than the ciphertexts of the straightfoward approach, so the security can be considerably increased. We remind the reader that the security of m-RLWE can vary substantially depending on the chosen modular functions, but even in the worst-case scenario from Definition 1 we have an effective dimension of $\max\{n_1, \ldots, n_m\}$ with an additional increase in the error variance by a factor of $\frac{n}{\max\{n_1,\ldots,n_m\}}$ when using the Bootland *et al.*'s attack [44] (see Chapter 5).[4]

In addition, efficiency is not reduced, as the expansion is not significantly increased, and one encrypted operation is equivalent to processing several blocks in parallel. We address now the case in which each block has to be processed by a different filter.

Modifying encryption and decryption primitives

When a different filter has to be applied to each block of the multidimensional signal, it is not enough to have one additional variable for coding the pointer to the block structure. This case would be analogous to having a set of independent multidimensional signals, and the corresponding filter has to be applied to each of them. Hence, we need an efficient and secure packing of several independently operable multidimensional signals into only one ciphertext.

To this end, we consider a pre- and post-processing inside the encryption and decryption primitives, respectively, that we explain below, highlighting the differences that have to be accounted for with respect to the univariate primitives of the cryptosystem presented in [4].

[4] The most favorable case is that of preserving an effective dimension equal to the product $\prod_i n_i$. We cover this situation in Chapters 2 and 3.

DFT/IDFT as pre-/post-processing: In order to obtain independent blocks, we apply a transform (DFT, Discrete Fourier Transform) along the additional variable defined as the block index. The convolution theorem states that the transform of a cyclic convolution between two signals in the temporal domain is equivalent to the element-wise product of the transforms of the two original signals:

$$\text{DFT}(x[l] \circledast y[l]) = \text{DFT}(x[l]) \circ \text{DFT}(y[l]).$$

This means that the operations applied along the variable l will be "component-wise" and independent for each coefficient slot. Hence, we represent the m-dimensional signals by means of multivariate polynomials with $m+1$ variables

$$x(z_1, \ldots, z_m, z) = \sum_{l_1,\ldots,l_m,l} x[l_1, \ldots, l_m, l] z_1^{l_1} \ldots z_m^{l_m} z^l,$$

considering $x(\boldsymbol{z}, z)$ where $\boldsymbol{z} = (z_1, \ldots, z_m)$ and $\boldsymbol{l} = (l_1, \ldots, l_m)$; z is the variable that indexes the different blocks of x, so we compute the DFT with respect to the coefficients (each coefficient represents an m-dimensional block) encoded in the variable z (we consider the modular function $1 + z^N$, that is, N blocks). We have the following:

$$\text{DFT}(x[\boldsymbol{l}, l]) = \sum_{l=0}^{N-1} x[\boldsymbol{l}, l] e^{\frac{-i2\pi kl}{N}}.$$

If we apply the cyclic convolution (by means of one homomorphic product between ciphertexts) between $\boldsymbol{X}[\boldsymbol{l}, k]$ and $\boldsymbol{H}[\boldsymbol{l}, k]$ with respect to the variable k, and afterwards the corresponding IDFT with respect to k, we are effectively computing the block-wise linear convolution between the blocks that form $x(\boldsymbol{z}, z)$ and $h(\boldsymbol{z}, z)$ (provided that the results of the linear convolutions do not overflow).

Therefore, if we apply the unidimensional DFT/IDFT across the index variable as pre-/post-processing, we can perform the block-wise linear convolution between all the blocks that form both signals by means of just one homomorphic convolution between $\boldsymbol{X}$ and $\boldsymbol{H}$.

Circular Convolution inside the Cryptosystem: The correctness of the result of the linear convolution only requires that there be enough coefficients to store it, but the convolution property of the DFT requires a cyclic convolution. It must be noted that the cryptosystem only allows to perform multiplications between polynomials modulo $f(z) = 1 + z^n$ for each variable, so we can only perform nega-cyclic convolutions homomorphically.

Several works (see for example [104]) show how to implement operations modulo $1 + z^n$ by means of cyclic convolutions. Here, we can apply the reverse process (presented in [54] and generalized in [29]), for enabling cyclic convolutions using operations between polynomials modulo $f(z) = 1 + z^N$ (see Chapters 4 and 5).

- First, we have to do a pre-processing before encryption

$$x'[\boldsymbol{l}, l] = x[\boldsymbol{l}, l](-1)^{\frac{l}{N}},$$
$$h'[\boldsymbol{l}, l] = h[\boldsymbol{l}, l](-1)^{\frac{l}{N}},$$

  for $l = 0, \ldots, N-1$.

- Next, we can homomorphically evaluate

$$y'(\boldsymbol{z}, z) = x'(\boldsymbol{z}, z)h'(\boldsymbol{z}, z) \mod 1 + z^N.$$

- Finally, we have to do the post-processing for the resulting $y'(\boldsymbol{z}, z)$ after decryption

$$y[\boldsymbol{l}, l] = y'[\boldsymbol{l}, l](-1)^{\frac{-l}{N}},$$

  for $l = 0, \ldots, N - 1$, and we obtain a homomorphic cyclic convolution.

It is important to note that the presented pre- and post-processing steps require the use of complex numbers to represent the complex roots of 1 and -1. As mentioned in Section B.3.2, complex numbers can be accommodated in the used cryptosystem by adding one additional variable with a modular function $f(w) = 1 + w^2$. The main drawback of this solution stems from the need for quantizing the non-integer complex roots represented in fixed-point with sufficient precision; this introduces rounding errors and implies an increase in the needed modulo for representing the signals, therefore increasing also the cipher expansion. In order to remove this constraint and avoid rounding errors, we can replace the DFT by its finite ring counterpart as explained in the next section.

B.4.2. Optimizations: Using the NTT to remove rounding errors

Instead of the complex-valued DFT, we resort to the DFT over finite rings, that is, the NTT (Number Theoretic Transform) [104, 29]. Additionally, we use a finite N-th root of -1 in $\mathbb{Z}_t$ for the pre- and post-processing of the cyclic convolution. This allows us to avoid both the rounding problems and the need of doubling the size of the used polynomials. It can only be applied for certain values of t and N.

The use of the NTT as a method both for efficiently performing encrypted operations and as an encrypted operation inside an RLWE based cryptosystem was introduced by the authors in [29], and exemplified as a pre-/post-processing in [189] for the univariate case. Hence, here we briefly discuss the particularities of the NTT when applied to the multivariate case, and we refer the reader to Chapters 4 and 5 for further details.

The existence conditions for an NTT with size N in $\mathbb{Z}_t$ are the same included in Chapter 4. The expressions for the calculation of the NTT and the INTT are the following:

$$\boldsymbol{X}[\boldsymbol{l}, k] = \sum_{l=0}^{N-1} x[\boldsymbol{l}, l]\alpha^{lk} \mod t,$$

for $k = 0, 1, \ldots, N - 1$ and

$$x[\boldsymbol{l}, l] = N^{-1} \sum_{k=0}^{N-1} \boldsymbol{X}[\boldsymbol{l}, k]\alpha^{-lk} \mod t,$$

for $l = 0, 1, \ldots, N - 1$. In Section B.A we analyze the impact of these pre- and post-processing steps in the computational cost and we show that it is negligible compared with the cost of the (regular) encryption/decryption primitives. In addition, when the case requires it, it is also possible to offload these pre- and post-processing operations to be performed under encryption (without the intervention of the secret key owner) by applying the methods proposed in [29] to the multivariate case, at the cost of an increase in the computational load at the evaluator.

This concludes the basic mechanisms for efficiently operating on m-RLWE encryptions. The next section introduces methods to perform on-the-fly changes in the ciphertext structure in an unattended way, which enables homomorphic updates on the available encrypted operations.

B.5. Updatable Ciphertext Structure

The previous sections show how the possibility of adding some extra structure to the encrypted information together with the use of some pre- and post-processing can enable a unattended encrypted processing in a wide set of practical scenarios. However, once data are encrypted, m-RLWE imposes a specific fixed structure optimized for a determined processing, and it is easy to imagine scenarios where the ability to change the underlying ciphertext structure is very convenient (if a chain of processes has to be applied unattendedly).

The straightforward approach would be to send the ciphertext to the secret key owner with the aim of decrypting and reencrypting under the new structure. This introduces several problems: (a) the user can see some part of the required steps for the execution of the algorithm implemented by the server, and (b) this has an increase in the total response time because of the delay caused by the communication between the server and the user. In order to address these two problems, we propose a new mechanism which allows the third party to change the ciphertext structure in an unattended way (without interaction with the secret key owner). To this end, we apply a modification of the *relinearization* procedure [29].

For simplicity on the exposition we exemplify the process with m-RLWE as it was introduced in Definition 1 (power-of-two cyclotomic functions). It is important to remark that the same idea can be extended to work with more general (*secure*) multivariate RLWE instatiations as the ones discussed in Chapter 2.

B.5.1. Relinearization

The basic relinearization operation is intended to process encryptions after a homomorphic product. After a product, the encryptions become a function of powers of the secret key s. The relinearization builds key homomorphisms that relate s^2 to s and is used to produce a 2-component fresh-like encryption from a three-component one. For our purposes, we present a more generic version of the relinearization, which defines key homomorphisms between two keys s and s'. Let us consider a ciphertext (c_0, c_1) with decryption circuit $c_0 + c_1 s$. If we apply the relinearization algorithm to (c_0, c_1) to express it as a function of the new key s', we have:

$$c_0^{relin} = c_0 + \sum_{i=0}^{\lceil \log_T q \rceil - 1} c_{1,i} b_i \quad \text{and} \quad c_1^{relin} = \sum_{i=0}^{\lceil \log_T q \rceil - 1} c_{1,i} a_i,$$

where the set of polynomials $c_{1,i}$ with $i = 0, \ldots, \lceil \log_T q \rceil - 1$ is the base-T decomposition of c_1 for a given $0 < T < q$.[5] The different b_i and a_i come from the key homomorphism $h_i = (a_i, b_i = -(s'a_i + Te_i) + T^i s)$ with $i = 0, \ldots, \lceil \log_T q \rceil - 1$; these homomorphisms can be seen as "pseudoencryptions" of the key s under s'. For the sake of exposition, the decryption circuit of (c_0, c_1) can be represented in matrix notation as $\boldsymbol{c}_0 + \boldsymbol{C}_1 \boldsymbol{s}$, where $\boldsymbol{C}_1$ is a block skew circulant matrix of the polynomial c_1 [29]. The matrix notation allows to see the decryption

[5] We assume that $T = t$ unless otherwise stated.

equation as a sum of external products of restructured versions of the polynomial c_1 times each of the coefficients of the key: $c_0 + \sum_{j=0}^{n-1} c_1^{(j)} s_j$ where the different $c_1^{(j)}$ are polynomials whose coefficients are the elements of the j-th column of the skew circulant matrix $\boldsymbol{C}_1$. In general, if we consider the concatenation of n key homomorphisms $h_i^{(j)}$ with $i = 0, \ldots, \lceil \log_T q \rceil - 1$ and $j = 0, \ldots, n-1$, where $h_i^{(j)}$ has the coefficient s_j "pseudo-encrypted" with the secret key s', we can obtain a new ciphertext $(c_0^{relin}, c_1^{relin})$ without changing its content (we refer to Chapter 4 for more details).

B.5.2. Changing the polynomial structure

The introduced representation of the decryption circuit ($c_0 + \sum_{j=0}^{n-1} c_1^{(j)} s_j$) already sheds some light about the approach we follow to change the polynomial structure through a relinearization operation: we simply encode the different polynomials that form the $h_i^{(j)}$ along with c_0 and $c_1^{(j)}$ under the desired polynomial structure.

In order to incorporate this new structure, we first define a family of $n!$ different reversible polynomial ring mappings $f_{\boldsymbol{n},\boldsymbol{m}}^{(w)} : R_q[z_1, \ldots, z_l] \to R_q[x_1, \ldots, x_k]$ with w belonging to the set $\{1, \ldots, n!\}$ where $\boldsymbol{n} = (n_1, \ldots, n_l)$, $\boldsymbol{m} = (m_1, \ldots, m_k)$ and $n = \prod_{i=1}^{l} n_i = \prod_{i=1}^{k} m_i$ (the modular functions of the polynomial rings are $f_i(z_i) = z_i^{n_i} + 1$ with $i = 1, \ldots, l$, and $f_j(x_j) = x_j^{m_j} + 1$ with $j = 1, \ldots, k$).

This mapping takes as input a polynomial element that belongs to the ring $R_q[z_1, \ldots, z_l]$ and produces as output a polynomial element that belongs to the ring $R_q[x_1, \ldots, x_k]$ and whose coefficients are the same as the coefficients of the polynomial input but rearranged in one of the $n!$ different ways (w indicates the specific reordering used).

Now, we need a set of key homomorphisms $h_i^{(j)}$ with $j = 0, \ldots, n-1$ where all the used polynomials belong to the output polynomial ring, that is $a_i, e_i \leftarrow R_q[x_1, \ldots, x_k]$, and where instead of using $s \in R_q[z_1, \ldots, z_l]$ we are "pseudo-encrypting" the coefficients s_j with the secret key $f_{\boldsymbol{n},\boldsymbol{m}}^{(w)}(s) \in R_q[x_1, \ldots, x_k]$.

Equipped with these tools, we perform a relinearization in which we consider the use of $f_{\boldsymbol{n},\boldsymbol{m}}^{(w)}(c_0)$, $f_{\boldsymbol{n},\boldsymbol{m}}^{(w)}(c_1^{(j)})$ for $j = 0, \ldots, n-1$ instead of c_0 and $c_1^{(j)}$. By doing this, we obtain a new ciphertext $(c_0^{relin}, c_1^{relin})$ that is the encryption of $f_{\boldsymbol{n},\boldsymbol{m}}^{(w)}(m) \in R_t[x_1, \ldots, x_k]$ (the corresponding reordering of the original message $m \in R_t[z_1, \ldots, z_l]$) with the secret key $f_{\boldsymbol{n},\boldsymbol{m}}^{(w)}(s) \in R_q[x_1, \ldots, x_k]$ and where $c_0^{relin}, c_1^{relin} \in R_q[x_1, \ldots, x_k]$.

For example, if both c_0 and c_1 are polynomials that belong to $\mathbb{Z}_q[z]/(1 + z^n)$ and we want to divide the encrypted signal in blocks of length n_x (e.g., to obtain an image whose rows are the different blocks), we consider the ring $(\mathbb{Z}_q[x, y]/(1 + x^{n_x}))/(1 + y^{n_y})$ with $n_x n_y = n$; being n_x and n_y powers of 2. As we know which is the new position of each coefficient of the encrypted message in the new multivariate structure, we apply the explained method considering that the polynomials belong to the bivariate ring $(\mathbb{Z}_q[x, y]/(1 + x^{n_x}))/(1 + y^{n_y})$.

The presented strategy can be applied to change the structure of the encrypted messages to all types of multivariate polynomials depending on what we need.

Security considerations: The security of this process is guaranteed by the underlying m-RLWE problems involved in the execution of the algorithm. Consider that we have a chain of structure changes defined by a composition of L mappings $f^{(w_1)}_{\boldsymbol{n}^{(1)},\boldsymbol{n}^{(2)}} \circ f^{(w_2)}_{\boldsymbol{n}^{(2)},\boldsymbol{n}^{(3)}} \circ \ldots \circ f^{(w_L)}_{\boldsymbol{n}^{(L)},\boldsymbol{n}^{(L+1)}}$, where each w_i belongs to the set $\{1,\ldots,n!\}$ with $i = 1,\ldots,L$ and each $\boldsymbol{n}^{(j)} = \left(n^{(j)}_1,\ldots,n^{(j)}_{k_j}\right)$ with $j = 1,\ldots,L+1$ is a vector composed of k_j natural numbers satisfying $n = \prod_{i=1}^{k_1} n_i^{(1)} = \prod_{i=1}^{k_2} n_i^{(2)} = \ldots = \prod_{i=1}^{k_{L+1}} n_i^{(L+1)}$. Then, the security of the proposed algorithm is based on the hardness of the underlying multivariate RLWE problems defined over the $L+1$ rings $R_q[z^{(j)}_1,\ldots,z^{(j)}_{k_j}]$, where the different modular functions are defined as in the previous section, that is, $f_{k_j}(z^{(j)}_{k_j}) = (z^{(j)}_{k_j})^{n^{(j)}_{k_j}} + 1$. Additionally, the security is also based on the circular security of the different involved multivariate RLWE based cryptosystems (see Section B.3), hence guaranteeing that releasing encryptions of the secret key is secure.

B.6. Conclusions

This appendix presents novel uses of Multivariate Ring Learning with Errors (m-RLWE), which enable efficient encrypted processing of images and multidimensional signals (3-D images, video,...). Cryptosystems based on this problem can flexibly fit the input signal structure, therefore producing an extremely efficient encryption with very low processing overhead and cipher expansion. We have also produced novel techniques to deal with non-interactive transformations between different structures, enabling for the first time block-based multidimensional encrypted signal processing in a non-interactive way. This is especially relevant in privacy-aware scenarios like outsourced medical imaging (ECG, EEG, CT scans, MRI,...), opening up a wide range of novel encrypted processing applications supporting secure unattended outsourced processing of signals of almost any kind.

B.A. Computational cost for modified encryption and decryption

We have proposed a modification for encryption and decryption by introducing pre- and post-processing in them (see Section B.4.1). We now analyze the impact of such pre- and post-processing in terms of computational cost. Considering an example of filtering between I images with size $N \times N$ and filters of size $F \times F$, the cost for the product between polynomials from our cryptosystem is

$$\text{Cost}_{Poly.Prod} \approx (N+F-1)^4 h^2 I^2.$$

On the other hand, the cost of a pre- or post-processing operation would be $(N^2 + F^2)\text{Cost}_{DFT(Ipoints)}$ because we have to perform N^2 DFTs of size I for the images and F^2 DFTs of size I for the filters. If we use a fast algorithm like FFT for computing the polynomial products and the DFT, we have a total cost of

$$\begin{aligned}\text{Cost} \approx \\ & N_{Poly.Prod.} C_{FFT} (N+F-1)^2 hI \log_2\left((N+F-1)^2 hI\right) \\ & + (N^2+F^2) C_{FFT} I \log_2 I,\end{aligned}$$

where $N_{Poly.Prod.}$ is the number of polynomial products needed for performing the considered cryptographic primitive (in this case, encryption or decryption), and C_{FFT} is the linear constant of the used FFT algorithm.

Using a slack value of $h = 1$, we can obtain the ratio betweeworkn the cost for the pre- or post-processing and the respective encryption/decryption primitive (with no pre-/post-processing):

$$\text{Ratio}_{Cost} \approx \frac{(N^2 + F^2)C_{FFT} I \log_2 I}{N_{Poly.Prod.} C_{FFT}(N + F - 1)^2 I \log_2((N + F - 1)^2 I)},$$

where Ratio_{Cost} achieves its highest value when $F = 1$.

Now, let us express the asymptotic Ratio_{Cost} when $F = 1$ and $N \to \infty$:

$$\begin{aligned}\lim_{N\to\infty} \text{Ratio}_{Cost} &= \lim_{N\to\infty} \frac{(N^2 + 1)C_{FFT} I \log_2 I}{N_{Poly.Prod.} C_{FFT} N^2 I \log_2(N^2 I)} \\ &= \lim_{N\to\infty} \frac{(1 + \frac{1}{N^2})C_{FFT} I \log_2 I}{N_{Poly.Prod.} C_{FFT} I \log_2(N^2 I)} \\ &= 0.\end{aligned}$$

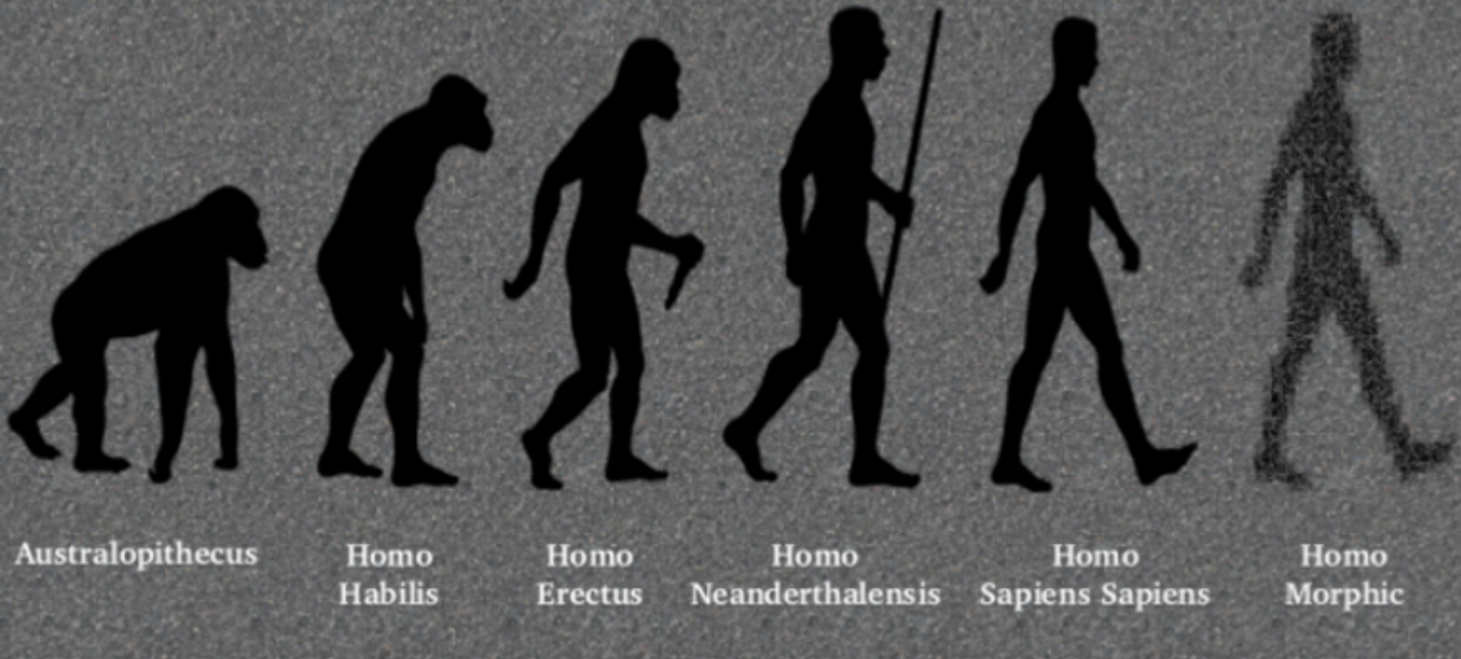
Therefore, when increasing the size of the images, the additional cost for the primitives becomes negligible. Additionally, it is also interesting to calculate the maximum increase in computational cost that the use of pre- and post-processing can incur on. With this aim, we study the case when $I \to \infty$ and $F = 1$:

$$\begin{aligned}\lim_{I\to\infty} \text{Ratio}_{Cost} &= \\ &= \lim_{I\to\infty} \frac{(N^2 + 1)C_{FFT} I \log_2 I}{N_{Poly.Prod.} C_{FFT} N^2 I \log_2(N^2 I)} \\ &= \lim_{I\to\infty} \frac{N^2 + 1}{N_{Poly.Prod.} N^2 \frac{\log_2 IN^2}{\log_2 I}} \\ &= \lim_{I\to\infty} \frac{N^2 + 1}{N_{Poly.Prod.} N^2 (\frac{\log_2 N^2}{\log_2 I} + 1)} \\ &= \frac{N^2 + 1}{N_{Poly.Prod.} N^2},\end{aligned}$$

that is approximately $\frac{1}{N_{Poly.Prod.}}$ when N is big enough.

Hence, the worst-case computational cost of the modified encryption and decryption primitives with respect to the original one is $\text{Cost}_{Orig.Primitive} \cdot (1 + \frac{1}{N_{Poly.Prod.}})$. The encryption conveys 2 polynomial products, so $\frac{3}{2}\text{Cost}_{Encryption}$ ($\text{Cost}_{Encryption}$ represents the computational cost of the original encryption), and for the decryption it depends on both the number of polynomial elements comprising the ciphertexts and the computation of the powers of the secret key. Assuming that the powers of the secret key have been precomputed, we would have $\text{Cost}_{Decryption} \cdot (1 + \frac{1}{\text{Num. of Elements}-1}) = \text{Cost}_{Decryption} \frac{\text{Num. of Elements}}{\text{Num. of Elements}-1}$ ($\text{Cost}_{Decryption}$ represents the computational cost of the original decryption).

Summarizing, we can see that the cost increase due to the use of the pre and post-processing is very small, and in fact, it becomes negligible for practical cases.


Australopithecus
Homo Habilis
Homo Erectus
Homo Neanderthalensis
Homo Sapiens Sapiens
Homo Morphic