Project PROMETHEUS
PRivacy preserving pOst-quantuM systEms
from advanced crypTograpHic mEchanisms
Using latticeS

D4.1

Survey of existing building blocks for practical advanced protocols

Contractual submission date
Month 10

Actual submisision date
May 2019

Deliverable version

1.0

Main author
Benoît Libert and Fabrice Mouhartem
(ENSL)

http://www.h2020prometheus.eu/

h2020prometheus

PROMETHEUS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 780701. The contents of this deliverable are the responsibility of the PROMETHEUS consortium, and do not necessarily reflect the official views of the European Union.

Document information

Grant agreement no.	780701
Project acronym	PROMETHEUS
Project full title	PRivacy preserving pOst-quantuM systEms from advanced crypTograpHic mEchanisms Using latticeS
Type of action	Research and Innovation Action (RIA)
Topic	H2020-DS-06-2017-Cybersecurity PPP: Cryptography
Project dates	1st January 2018 (Month 1) / 31st December 2021 (Month 48)
Duration	48 months
Project URL	http://www.h2020prometheus.eu/
EU Project Officer	Carmen Ifrim
Work package	WP4 – Building blocks for practical advanced protocols
Deliverable title	Survey of existing building blocks for practical advanced protocols
Deliverable no.	D4.1
Deliverable version	1.0
Deliverable filename	`PROMETHEUS-WP4-D4.1.pdf`
Nature of deliverable	Report
Dissemination level	Public
Number of pages	44
Responsible partner	ENSL (participant number 1)
Author	Benoît Libert and Fabrice Mouhartem (ENSL)

Abstract. This document will give details on the related-work on cryptographic building blocks for advanced protocols, as well as a complete list of open problems that the WP will study. In particular, detailed comparisons will be given among existing lattice-based signatures in the standard model and homomorphic commitment schemes.

Keywords: Lattice-based privacy-preserving cryptography, building blocks.

Signatures

Written by	Benoît Libert and Fabrice Mouhartem	ENSL	May 2019
Reviewed by	Olivier Sanders	ORA	20/03/2019
Reviewed by	Adeline Roux-Langlois	UR1	20/03/2019
Approved by	Benoît Libert as Project coordinator	ENSL	07/05/2019
Approved by	Sébastien Canard as Technical leader	ORA	07/05/2019

Partners

ENSL ENS de Lyon
ORA Orange SA
CWI Stiching Centrum Voor Wiskunde En Informatica
IDC IDC Herzliya
RHUL Royal Holloway, University of London
RUB Ruhr-Universität Bochum
SCYTL Scytl Secure Electronic Voting, S.A.
THA Thales Communications & Security S.A.S.
TNO TNO
UPC Universitat Politècnica de Catalunya · BarcelonaTech
UR1 Université de Rennes 1
WEI Weizmann Institute of Science

1 Introduction

This deliverable gives a survey of the cryptographic building blocks that can be used to build advanced privacy-preserving protocols under lattice assumptions. These include non-interactive cryptographic commitment schemes, digital signatures, public-key encryption, homomorphic encryption and pseudorandom functions.

In order to be useful in the protection of users' privacy, these building blocks have to be compatible with existing zero-knowledge proof systems for lattice-related languages. For example, anonymity primitives like anonymous credentials [Cha85] make use of lattice-based signature schemes that make it possible for a prover to demonstrate possession of a message-signature pair (Msg, sig) while revealing neither Msg nor sig. In e-cash systems like [CHL05], digital coins contain a serial number consisting of a pseudorandom function (PRF) evaluation. Namely, using a digital wallet containing a certified PRF secret key k, a user can spend a coin by computing a serial number $S = F_k(J)$ which is nothing but the evaluation of the PRF for the key k and some counter J. In order to prove the validity of his coin, the spender has to create a zero-knowledge proof that S is indeed the correct evaluation of the PRF for some committed input J and the secret key k for which he possesses a valid certificate.

Other protocols require a prover to guarantee properties about encrypted data. For example, the widely used design principle of group signatures [CVH91] proceeds by having group members verifiably encrypt their group membership certificate under the public key of a tracing authority. This requires a method of efficiently proving that the encrypted value is indeed a valid signature. In group encryption schemes [KTY07], the sender of a ciphertext has to prove that an encrypted message is the witness of some relation. Other similar examples include e-voting protocols, where each voter should provide evidence that he encrypted a valid vote (e.g., a 0 or 1 vote in a referendum). Voting protocols also require additional mechanisms such as distributed threshold decryption procedures [DF89]. Namely, decryption keys are split into N shares given to distinct trustees in such a way that none of these can decrypt individual ballots. Yet, a quorum of at least t-out-of-N trustees should be able to jointly decrypt the final election result (i.e., the "tally" obtained by aggregating individual encrypted votes) without learning individual votes.

In other advanced privacy protocols, it is useful to have encryption schemes endowed with advanced properties that come in handy to protect users' privacy. For example, fully homomorphic encryption (FHE) [RAD78, Gen09] enables the design of multi-party computation (MPC) protocols with a small number of rounds. Namely, assuming a set of trusted public parameters, recent works have shown that FHE makes it possible to realize secure MPC protocols in 3 [AJLA$^+$12] or even 2 rounds [MW16, BP16]. In the plain model (i.e., without assuming a common reference string generated by a trusted entity), secure MPC is known to be possible in 4 rounds [BHP17].

The forthcoming sections will provide an overview of the various lattice-based primitives that have been used so far as building blocks for privacy-enhancing cryptography in the post-quantum setting. Section 2 recalls the lattice assumptions on which the security of these protocols usually relies. Lattice-based cryptographic commitment schemes are discussed in Section 3. The two main families of zero-knowledge proof systems that have been employed in structured/standard lattices are recalled in Section 4. Section 5 provides a succinct state of the art of lattice-based pseudorandom functions as well as extensions (e.g., key-homomorphic or constrained PRFs) thereof. Section 6 is devoted to lattice-based signature schemes that can interact with zero-knowledge protocols for lattice-related statements. It notably discusses signature schemes endowed

with so-called "efficient protocols" [CL02], which allow a user to interact with a signer so as to obtain a signature on a committed message. In Section 7, this deliverable finally presents an overview of lattice-based public-key encryption schemes that can lend themselves to the design of anonymity protocols. In particular, it will cover schemes that either: (i) make it possible to prove properties about encrypted data [Reg05, GPV08]; (ii) enable computations over encrypted data [Gen09]; (iii) support efficient distributed decryption mechanisms [BD10].

2 Background

Vectors are denoted in bold lower-case letters and bold upper-case letters will denote matrices. The Euclidean and infinity norm of any vector $\mathbf{b} \in \mathbb{R}^m$ will be denoted by $\|\mathbf{b}\|$ and $\|\mathbf{b}\|_\infty$, respectively. The Euclidean norm of matrix $\mathbf{B} \in \mathbb{R}^{m \times n}$ with columns $(\mathbf{b}_i)_{i \leq n}$ is $\|\mathbf{B}\| = \max_{i \leq n} \|\mathbf{b}_i\|$. When $\mathbf{B}$ has full column-rank, we let $\widetilde{\mathbf{B}}$ denote its Gram-Schmidt orthogonalization.

When S is a finite set, we denote by $U(S)$ the uniform distribution over S, and by $x \hookleftarrow U(S)$ the action of sampling x according to this distribution. Finally, for any integers $A, B, N \in \mathbb{Z}$, we let $[N]$ and $[A, B]$ denote the sets $\{1, \ldots, N\}$ and $\{A, A+1, \ldots, B\}$, respectively. Finally, $\mathcal{S}_N$ denotes the set of all permutations σ over $[N]$ for any given integer N.

2.1 Lattices

A lattice L is the set of integer linear combinations of linearly independent basis vectors $(\mathbf{b}_i)_{i \leq n}$ living in $\mathbb{Z}^m$. We work with q-ary lattices, for some prime q.

Definition 1 *Let $m \geq n \geq 1$, a prime $q \geq 2$ and $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{u} \in \mathbb{Z}_q^n$, define the lattice $\Lambda_q(\mathbf{A}) := \{\mathbf{e} \in \mathbb{Z}^m \mid \exists \mathbf{s} \in \mathbb{Z}_q^n \text{ s.t. } \mathbf{A}^\top \cdot \mathbf{s} = \mathbf{e} \bmod q\}$ as well as*

$$\Lambda_q^\perp(\mathbf{A}) \quad := \quad \{\mathbf{e} \in \mathbb{Z}^m \mid \mathbf{A} \cdot \mathbf{e} = \mathbf{0}^n \bmod q\},$$

as well as

$$\Lambda_q^{\mathbf{u}}(\mathbf{A}) \quad := \quad \{\mathbf{e} \in \mathbb{Z}^m \mid \mathbf{A} \cdot \mathbf{e} = \mathbf{u} \bmod q\}.,$$

which is a shift of the lattice $\Lambda_q^\perp(\mathbf{A})$ since, for any arbitrary $\mathbf{t} \in \Lambda_q^{\mathbf{u}}(\mathbf{A})$, we have $\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \Lambda_q^\perp(\mathbf{A}) + \mathbf{t}$.

For a lattice L, let $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \exp(-\pi\|\mathbf{x} - \mathbf{c}\|^2/\sigma^2)$ for $x \in L$, a vector $\mathbf{c} \in \mathbb{Z}^m$ and a real $\sigma > 0$. The discrete Gaussian of support L, center $\mathbf{c}$ and parameter σ is

$$D_{L,\sigma,\mathbf{c}}(\mathbf{y}) = \rho_{\sigma,\mathbf{c}}(\mathbf{y})/\rho_{\sigma,\mathbf{c}}(L)$$

for any $\mathbf{y} \in L$, where $\rho_{\sigma,\mathbf{c}}(L) = \sum_{\mathbf{x} \in L} \rho_{\sigma,\mathbf{c}}(\mathbf{x})$. The distribution centered in $\mathbf{c} = \mathbf{0}$ is denoted by $D_{L,\sigma}(\mathbf{y})$.

It is well-known that one can efficiently sample from a Gaussian distribution with lattice support given a sufficiently short basis of the lattice.

Lemma 2.1 ([BLP+13, Le. 2.3]) *There exists a PPT algorithm* GPVSample *that takes as inputs a basis $\mathbf{B}$ of a lattice $L \subseteq \mathbb{Z}^n$ and a rational $\sigma \geq \|\widetilde{\mathbf{B}}\| \cdot \Omega(\sqrt{\log n})$, and outputs vectors $\mathbf{b} \in L$ with distribution $D_{L,\sigma}$.*

We also rely on the trapdoor generation algorithm of Alwen and Peikert [AP09], which refines the technique of Gentry *et al.* [GPV08].

Lemma 2.2 ([AP09, Th. 3.2]) *There is a PPT algorithm* TrapGen *that takes as inputs* 1^n, 1^m *and an integer* $q \geq 2$ *with* $m \geq \Omega(n \log q)$, *and outputs a matrix* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *and a basis* $\mathbf{T_A}$ *of* $\Lambda_q^\perp(\mathbf{A})$ *such that* $\mathbf{A}$ *is within statistical distance* $2^{-\Omega(n)}$ *to* $U(\mathbb{Z}_q^{n \times m})$, *and* $\|\widetilde{\mathbf{T_A}}\| \leq \mathcal{O}(\sqrt{n \log q})$.

We use the basis delegation algorithm [CHKP10] that inputs a trapdoor for $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and produces a trapdoor for any $\mathbf{B} \in \mathbb{Z}_q^{n \times m'}$ containing $\mathbf{A}$ as a submatrix. A technique from Agrawal *et al.* [ABB10] is sometimes used in some proofs.

Lemma 2.3 ([CHKP10, Le. 3.2]) *There is a PPT algorithm* ExtBasis *that inputs* $\mathbf{B} \in \mathbb{Z}_q^{n \times m'}$ *whose first* m *columns span* $\mathbb{Z}_q^n$, *and a basis* $\mathbf{T_A}$ *of* $\Lambda_q^\perp(\mathbf{A})$ *where* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *is a submatrix of* $\mathbf{B}$, *and outputs a basis* $\mathbf{T_B}$ *of* $\Lambda_q^\perp(\mathbf{B})$ *with* $\|\widetilde{\mathbf{T_B}}\| \leq \|\widetilde{\mathbf{T_A}}\|$.

Lemma 2.4 ([ABB10, Th. 19]) *There is a PPT algorithm* SampleRight *that inputs* $\mathbf{A}, \mathbf{C} \in \mathbb{Z}_q^{n \times m}$, *a small-norm* $\mathbf{R} \in \mathbb{Z}^{m \times m}$, *a short basis* $\mathbf{T_C} \in \mathbb{Z}^{m \times m}$ *of* $\Lambda_q^\perp(\mathbf{C})$, *a vector* $\mathbf{u} \in \mathbb{Z}_q^n$ *and a rational* σ *such that* $\sigma \geq \|\widetilde{\mathbf{T_C}}\| \cdot \Omega(\sqrt{\log n})$, *and outputs* $\mathbf{b} \in \mathbb{Z}^{2m}$ *such that* $\left[\begin{array}{c|c} \mathbf{A} & \mathbf{A} \cdot \mathbf{R} + \mathbf{C} \end{array}\right] \cdot \mathbf{b} = \mathbf{u} \bmod q$ *and with distribution statistically close to* $D_{L,\sigma}$ *where* $L = \{\mathbf{x} \in \mathbb{Z}^{2m} : \left[\begin{array}{c|c} \mathbf{A} & \mathbf{A} \cdot \mathbf{R} + \mathbf{C} \end{array}\right] \cdot \mathbf{x} = \mathbf{u} \bmod q\}$.

2.2 Hardness Assumptions in Standard Lattices

Definition 2 *Let* n, m, q, β *be functions of* $\lambda \in \mathbb{N}$. *The Short Integer Solution problem* $\mathsf{SIS}_{n,m,q,\beta}$ *is, given* $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n \times m})$, *find* $\mathbf{x} \in \Lambda_q^\perp(\mathbf{A})$ *with* $0 < \|\mathbf{x}\| \leq \beta$.

The $\mathsf{SIS}^\infty_{n,m,q,\beta}$ is defined in the same way with the difference that the Euclidean norm $\|\mathbf{x}\|$ is replaced by the infinity norm $\|\mathbf{x}\|_\infty$.

If $q \geq \sqrt{n}\beta$ and $m, \beta \leq \mathsf{poly}(n)$, then standard worst-case lattice problems with approximation factors $\gamma = \widetilde{\mathcal{O}}(\beta\sqrt{n})$ reduce to $\mathsf{SIS}_{n,m,q,\beta}$ (see, e.g., [GPV08, Se. 9]).

Definition 3 *Let* $n, m \geq 1$, $q \geq 2$, *and let* χ *be a probability distribution on* $\mathbb{Z}$. *For* $\mathbf{s} \in \mathbb{Z}_q^n$, *let* $A_{\mathbf{s},\chi}$ *be the distribution obtained by sampling* $\mathbf{a} \hookleftarrow U(\mathbb{Z}_q^n)$ *and* $e \hookleftarrow \chi$, *and outputting* $(\mathbf{a}, \mathbf{a}^T \cdot \mathbf{s} + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$. *The Learning With Errors problem* $\mathsf{LWE}_{n,q,\chi}$ *asks to distinguish* m *samples chosen accordingly to* $\mathcal{A}_{\mathbf{s},\chi}$ *(for* $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$*) and* m *samples chosen accordingly to* $U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$.

If q is a prime power, $B \geq \sqrt{n}\omega(\log n)$, $\gamma = \widetilde{\mathcal{O}}(nq/B)$, then there exists an efficient sampleable B-bounded distribution χ (i.e., χ outputs samples with norm at most B with overwhelming probability) such that $\mathsf{LWE}_{n,q,\chi}$ is at least as hard as SIVP_γ (see, e.g., [Reg05, BLP+13]). Similarly, if $\alpha q = \Omega(\sqrt{n})$, standard worst-case lattice problems with approximation factors $\gamma = \mathcal{O}(\alpha/n)$ reduce to $\mathsf{LWE}_{n,q,\alpha}$ [Reg05, BLP+13].

2.3 Hardness Assumptions in Ideal Lattices

The ring of polynomials over the integers is denoted $\mathbb{Z}[X]$. For a degree-N polynomial $f(X)$, the notation $\mathbb{Z}[X]/\langle f(X)\rangle$ stands for the ring of all polynomials modulo $f(X)$. Likewise, the ring of all polynomials with coefficients in $\mathbb{Z}_q$ is denoted $\mathbb{Z}_q[X]$ while the ring $\mathbb{Z}_q[X]/\langle f(X)\rangle$ is defined analogously to $\mathbb{Z}[X]/\langle f(X)\rangle$.

Letting q be a prime and $N = 2^r$ for some $r \in \mathbb{N}^+$, we consider the polynomial rings $R = \mathbb{Z}[X]/\langle X^N + 1\rangle$ and $R_q = \mathbb{Z}_q[X]/\langle X^N + 1\rangle$. Each ring element $f \in R$ (resp. $f \in R_q$) is thus a polynomial $f = \sum_{i=0}^{N-1} f_i X^i$ of degree at most $N-1$ in $\mathbb{Z}[X]$ (resp. $\mathbb{Z}_q[X]$). Each $f \in R$ can be associated with the vector $(f_0, f_1, \ldots, f_{N-1}) \in \mathbb{Z}^N$ containing its coefficients. When speaking of the norm of a polynomial $f \in R$, we mean the norm of its coefficient vector. We thus use the standard norm definitions $\|f\|_1 = \sum_{i=0}^{N-1} |f_i|$, $\|f\|_2 = (\sum_{i=0}^{N-1} f_i^2)^{1/2}$ and $\|f\|_\infty = \max_i |f_i|$.

For any $g \in R_q$ and $g = \sum_i \bar{g}_i X^i$, we identify each $\bar{g}_i$ with an element $g_i \in [-\frac{q-1}{2}, \frac{q-1}{2}]$ such that $\bar{g}_i = g_i \bmod q$. For a positive integer $\alpha > 0$, $S_\alpha = \{a \in R \mid \|a\|_\infty \leq \alpha\}$ denotes the set of all elements in R with ℓ_∞-norm at most α.

Definition 4 ([LS15]) *Let n, m be positive integers and let a real $\beta > 0$. The* **Module-SIS** *(M-SIS$_{q,n,m,\beta}$) problem is, given $\mathbf{A} \hookleftarrow U(R_q^{n\times m})$, to find a non-zero $\mathbf{z} \in R$ such that $\mathbf{A} \cdot \mathbf{z} = \mathbf{0}$ and $0 \leq \|\mathbf{z}\| \leq \beta$.*

Definition 5 ([LS15]) *Let n, m be positive integers and let χ a distribution over R_q. The* **Module-LWE** *(M-LWE$_{q,n,m,\chi}$) problem is to distinguish between m uniform samples $(\mathbf{a}_i, b_i) \hookleftarrow U(R_q^n \times R_q)$ and m samples $(\mathbf{a}_i, b_i) \in R_q^n \times R_q$, where $\mathbf{a}_i \hookleftarrow U(R_q^n)$ and $b_i = \mathbf{a}_i^\top \mathbf{s} + e_i$ for each $i \in [m]$, with $\mathbf{s} \hookleftarrow \chi^n$.*

Some applications of Module-LWE [DPLS18] use a distribution χ which is simply the uniform distribution over $S_1 = \{a \in R_q \mid \|a\|_\infty \leq 1\}$, in which case M-LWE retains its hardness as long as the number of samples is not too large.

We now recall assumptions that generalize the Module-SIS and Module-LWE assumptions. The Search Knapsack problem in the ℓ_2-norm (DSK$^2_{n,k,\beta}$) is exactly the Module-SIS problem in Hermite Normal form.

Definition 6 *The DSK$^2_{n,k,\beta}$ problem is, given a uniform $\mathbf{A}' \in R_q^{n\times(k-n)}$, to find a short vector $\mathbf{y}$ satisfying $[\mathbf{I}_n \mid \mathbf{A}'] \cdot \mathbf{y} = \mathbf{0}^n$. An algorithm $\mathcal{A}$ has advantage ϵ in solving DSK$^2_{n,k,\beta}$ if*

$$\mathbf{Adv}(\mathcal{A}) = \Pr\Big[\mathbf{y} \neq \mathbf{0}^n \ \wedge\ \|\mathbf{y}\|_2 \leq \beta \ \wedge\ [\mathbf{I}_n \mid \mathbf{A}'] \cdot \mathbf{y} = \mathbf{0}^n \mid \mathbf{A}' \hookleftarrow U(R_q^{n\times(k-n)}); \mathbf{y} \leftarrow \mathcal{A}(\mathbf{A}')\Big] \geq \epsilon$$

Baum *et al.* [BDL$^+$18, Lemma 5] gave parameter choices under which DSK$^2_{n,k,\beta}$ is unconditionally hard, which is useful to construct statistically binding commitment schemes. It was shown in [BDL$^+$18] that, if q is a prime congruent to $2d + 1 (\bmod 4d)$ for some power of two $d \in \{1, \ldots, N\}$ such that

$$\begin{aligned} \beta &< q^{1/d} \\ \beta &< \sqrt{N/(2\pi e)} \cdot q^{n/k} \cdot 2^{-128/(k\cdot N)} - \sqrt{N}/2, \end{aligned}$$

even an all-powerful algorithm $\mathcal{A}$ has advantage at most 2^{-128} in solving DSK$^2_{n,k,\beta}$.

In ℓ_∞ norm, the Decisional Knapsack problem is a generalization of the Module-LWE problem.

Definition 7 *The DSK$^\infty_{n,k,\beta}$ problem is to distinguish the distribution $[\mathbf{I}_n \mid \mathbf{A}'] \cdot \mathbf{y}$, for a short $\mathbf{y}$ and a uniform $\mathbf{A}'$, from the uniform distribution. An algorithm $\mathcal{A}$ has advantage ϵ in solving DSK$^\infty_{n,k,\beta}$ if the function*

$$\begin{aligned} \mathbf{Adv}(\mathcal{A}) = |\Pr[b = 1 \mid \mathbf{A}' \hookleftarrow U(R_q^{n\times(k-n)}); \mathbf{y} \hookleftarrow U(S_\beta^k); b \leftarrow \mathcal{A}(\mathbf{A}', [\mathbf{I}_n \mid \mathbf{A}']\cdot\mathbf{y})] \\ - \Pr[b = 1 \mid \mathbf{A}' \hookleftarrow U(R_q^{n\times(k-n)}); \mathbf{u} \hookleftarrow U(R_q^n); b \leftarrow \mathcal{A}(\mathbf{A}', \mathbf{u})]| \geq \epsilon. \end{aligned}$$

The $\mathsf{DSK}^{\infty}_{n,k,\beta}$ is equivalent to the Module-LWE problem when the number of samples is limited.

For applications requiring statistically-hiding commitments, Baum *et al.* [BDL+18, Lemma 4] showed that $\mathsf{DSK}^{\infty}_{n,k,\beta}$ can be made unconditionally hard for an appropriate choice of parameters. They showed that, if q is a prime congruent to $2d+1 (\mathrm{mod} 4d)$ for some power of two $d \in \{1,\ldots,N\}$ such that

$$q^{n/k} \cdot 2^{256/(k\cdot N)} \leq 2\beta < \frac{1}{\sqrt{d}} \cdot q^{1/d},$$

even an all-powerful algorithm $\mathcal{A}$ has advantage at most 2^{-128} in solving $\mathsf{DSK}^{2}_{n,k,\beta}$.

2.4 Vector Decompositions

The decomposition technique from [LNSW13, LLM+16a] is sometimes employed, which allows transforming vectors with infinity norm larger than 1 into vectors with infinity norm 1. We recall this technique below.

For any $B \in \mathbb{Z}_+$, define the number $\delta_B := \lfloor \log_2 B \rfloor + 1 = \lceil \log_2(B+1) \rceil$ and the sequence $B_1,\ldots,B_{\delta_B}$, where $B_j = \lfloor \frac{B+2^{j-1}}{2^j} \rfloor$, $\forall j \in [1,\delta_B]$. This sequence satisfies $\sum_{j=1}^{\delta_B} B_j = B$ and any integer $v \in [0,B]$ can be decomposed to $\mathsf{idec}_B(v) = (v^{(1)},\ldots,v^{(\delta_B)})^\top \in \{0,1\}^{\delta_B}$ such that $\sum_{j=1}^{\delta_B} B_j \cdot v^{(j)} = v$. We describe this decomposition procedure in a deterministic manner as follows:

1. Set $v' := v$; For $j = 1$ to δ_B do:

   If $v' \geq B_j$ then $v^{(j)} := 1$, else $v^{(j)} := 0$; $v' := v' - B_j \cdot v^{(j)}$.

2. Output $\mathsf{idec}_B(v) = (v^{(1)},\ldots,v^{(\delta_B)})^\top$.

For any positive integers $\mathfrak{m}, B$, we define $\mathbf{H}_{\mathfrak{m},B} := \mathbf{I}_{\mathfrak{m}} \otimes [B_1|\ldots|B_{\delta_B}] \in \mathbb{Z}^{\mathfrak{m}\times \mathfrak{m}\delta_B}$ and the following injective functions:

(i) $\mathsf{vdec}_{\mathfrak{m},B} : [0,B]^{\mathfrak{m}} \to \{0,1\}^{\mathfrak{m}\delta_B}$ that maps vector $\mathbf{v} = (v_1,\ldots,v_{\mathfrak{m}})$ to vector $\left(\mathsf{idec}_B(v_1)^\top \| \ldots \| \mathsf{idec}_B(v_{\mathfrak{m}})^\top\right)^\top$. Note that $\mathbf{H}_{\mathfrak{m},B} \cdot \mathsf{vdec}_{\mathfrak{m},B}(\mathbf{v}) = \mathbf{v}$.

(ii) $\mathsf{vdec}'_{\mathfrak{m},B} : [-B,B]^{\mathfrak{m}} \to \{-1,0,1\}^{\mathfrak{m}\delta_B}$ that maps vector $\mathbf{w} = (w_1,\ldots,w_{\mathfrak{m}})$ to vector $\left(\sigma(w_1)\cdot \mathsf{idec}_B(|w_1|)^\top \| \ldots \| \sigma(w_{\mathfrak{m}})\cdot \mathsf{idec}_B(|w_{\mathfrak{m}}|)^\top\right)^\top$, where for each $i = 1,\ldots,\mathfrak{m}$: $\sigma(w_i) = 0$ if $w_i = 0$; $\sigma(w_i) = -1$ if $w_i < 0$; $\sigma(w_i) = 1$ if $w_i > 0$. Note that $\mathbf{H}_{\mathfrak{m},B} \cdot \mathsf{vdec}'_{\mathfrak{m},B}(\mathbf{w}) = \mathbf{w}$.

We also need to define the two following sets, which are left invariant under any permutation and are used in the *decomposition-extension framework* described in Section 4.3.1:

- $\mathsf{B}^2_{\mathfrak{m}}$: the set of vectors in $\{0,1\}^{2\mathfrak{m}}$ with Hamming weight $\mathfrak{m}$;
- $\mathsf{B}^3_{\mathfrak{m}}$: the set of vectors in $\{-1,0,1\}^{3\mathfrak{m}}$ which has exactly $\mathfrak{m}$ coordinates equal to j for each $j \in \{-1,0,1\}$.

3 Lattice-Based Commitment Schemes

Commitment schemes [Blu82] are a key tool in the design of cryptographic protocols and have numerous applications (e.g., threshold cryptosystems [DF89] or e-voting [CFSY96]). In particular, combining commitment schemes and zero-knowledge proofs is a standard technique that prevents malicious parties from significantly deviating from the specification of a protocol.

This section reviews the main commitment schemes based on the hardness of the SIS and LWE assumptions. We begin with the commitment scheme of Kawachi, Tanaka and Xagawa [KTX08] (KTX), which is statistically hiding and computationally binding under the SIS assumption.

3.1 The KTX Commitment Scheme

For a security parameter λ, the commitment scheme of Kawachi *et al.* [KTX08] uses a dimension $n(\lambda)$, a prime modulus $q = \mathcal{O}(\sqrt{L} \cdot n)$, where L is the number of bits to commit, and an integer $m = n(\lceil \log_2 q \rceil + 3)$. We describe several flavors of the scheme.

We first describe the variant that allows committing to $L \leq \mathsf{poly}(n)$ bits at once. In this variant, the commitment key is $(\mathbf{a}_0, \ldots, \mathbf{a}_{L-1}, \mathbf{B}) \hookleftarrow U(\mathbb{Z}_q^{n \times (m+L)})$. To commit to an L-bit message $\mathbf{x} = (x_0, \ldots, x_{L-1}) \in \{0,1\}^L$, the committer samples a random m-bit string $\mathbf{r}$ and outputs

$$\mathbf{c} = \sum_{i=0}^{L-1} \mathbf{a}_i \cdot x_i + \mathbf{B} \cdot \mathbf{r} \bmod q.$$

In order to open the commitment, the committer simply reveals the underlying message $(x_0, \ldots, x_{L-1}) \in \{0,1\}^L$ and the randomness $\mathbf{r} \in \{0,1\}^m$.

Keygen$(1^\lambda, 1^L)$**:** Given a security parameter λ and the desired message length 1^L, choose uniformly random matrices $\mathbf{A} = [\mathbf{a}_0 \mid \ldots \mid \mathbf{a}_{L-1}] \hookleftarrow U(\mathbb{Z}_q^{n \times L})$, $\mathbf{B} \hookleftarrow U(\mathbb{Z}_q^{n \times m})$. Output the commitment key

$$\mathsf{ck} = \{\mathbf{A}, \mathbf{B}\}$$

Commit$(\mathsf{ck}, \mathbf{x})$**:** In order to commit to $\mathbf{x} = (x_0, \ldots, x_{L-1})^\top \in \{0,1\}^L$, choose $\mathbf{r} \hookleftarrow U(\{0,1\}^m)$ and compute

$$\mathbf{c} = \mathbf{A} \cdot \mathbf{x} + \mathbf{B} \cdot \mathbf{r} \ \in \mathbb{Z}_q^n. \tag{1}$$

Output the commitment string $\mathbf{c} \in \mathbb{Z}_q^n$ and retain $\mathbf{r} \in \{0,1\}^m$ as a state information allowing to open the commitment.

Open$(\mathbf{c}, \mathbf{x}, \mathbf{r})$**:** To open a commitment $\mathbf{c} \in \mathbb{Z}_q^n$, the committer simply reveals the randomness $\mathbf{r} \in \{0,1\}^m$ that was used to compute (1). The verifier accepts $(\mathbf{x}, \mathbf{r})$ as a valid opening of $\mathbf{c}$ if $(\mathbf{x}, \mathbf{r}) \in \{0,1\}^L \times \{0,1\}^m$ and if (1) is satisfied. Otherwise, the verifier rejects the pair $(\mathbf{x}, \mathbf{r})$.

If a dishonest committer can come up with a commitment $\mathbf{c} \in \mathbb{Z}_q^n$ for which it can provide compute two valid openings $(x'_0, \ldots, x'_{L-1}, \mathbf{r}')$ and $(x''_0, \ldots, x''_{L-1}, \mathbf{r}'')$ for two distinct messages

$$\mathbf{x}' = (x'_0, \ldots, x'_{L-1}) \neq (x''_0, \ldots, x''_{L-1}) = \mathbf{x}'',$$

then a simple reduction can compute a solution to the $\mathsf{SIS}^{\infty}_{n,m+L,q,1}$ problem associated with the uniformly random matrix $\bar{\mathbf{A}} = [\mathbf{A} \mid \mathbf{B}] \in \mathbb{Z}_q^{n\times(m+L)}$. The scheme is thus computationally binding, assuming the worst-case hardness of $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(\sqrt{L}\cdot n)}$. On the other hand, by the Leftover Hash Lemma (see, e.g., the variant stated in [GTKPV10]), the distribution of a commitment $\mathbf{c}$ is statistically close to the uniform distribution over $\mathbb{Z}_q^n$. This implies that the scheme is statistically hiding.

In the special case when $L = 1$, the scheme becomes a bit commitment scheme, in which case it can use a small modulus $q = \widetilde{\mathcal{O}}(n)$ and rely on a weak SIVP assumption with $\gamma = \widetilde{\mathcal{O}}(n)$.

In the above description, each KTX commitment is computed using a random vector $\mathbf{r}$ which is sampled from a uniform binary distribution. Alternatively, $\mathbf{r} \in \mathbb{Z}^n$ could be sampled from a discrete Gaussian distribution $D_{\mathbb{Z}^m,\sigma}$ for a large enough standard deviation $\sigma > 0$. In this case, the scheme can be modified to become a trapdoor commitment, where a trapdoor equivocation key (which is only known to a simulator in security proofs of cryptographic protocols) makes it possible to open a given commitment to any desired message. To do this, we need to increase m to make it as large as $m = \Omega(n \cdot \lceil \log q \rceil)$, in such a way that the matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$ can be generated with a trapdoor $\mathbf{T_B}$ for the lattice $\Lambda_q^{\perp}(\mathbf{B})$, which can be used as a trapdoor for the commitment scheme.

The KTX scheme allows commitment strings $\mathbf{c} \in \mathbb{Z}_q^n$ to be shorter than the underlying messages $\mathbf{x}$. On the other hand, these messages are restricted to consist of binary vectors, or at least small-norm integer vectors. Indeed, the binding property is lost if the scheme is used to commit to arbitrary vectors over $\mathbb{Z}_q^L$. As a consequence, the KTX commitment is *not* a homomorphic commitment scheme over the message space $\mathbb{Z}_q^L$, making it unsuitable for specific applications that require homomorphic commitments over large message spaces. In the following subsection, we recall how this problem can be addressed by commitment schemes in structured lattices.

3.2 Commitment Schemes in Ideal Lattices

In [BKLP15], Benhamouda *et al.* described a statistically-binding commitment scheme which is computationally-hiding under the ring LWE assumption. Their scheme allows committing to arbitrary messages in the ring $R_q = \mathbb{Z}_q[X]/\langle X^N + 1\rangle$ and is also additively homomorphic over R_q. Benhamouda *et al.* [BKLP15, Section 4] also provide zero-knowledge proofs allowing a committer to prove knowledge of an opening of a commitment. They also provide protocols allowing to prove that committed ring elements satisfy linear and multiplicative relations. In particular, they can prove that committed witnesses satisfy an algebraic circuit. Using the additional structure offered by ideal lattices, their zero-knowledge protocols achieve negligible soundness errors after a single execution.

The commitment scheme of [BKLP15] is only computationally hiding. In the following, we will recall a commitment scheme proposed by Baum *et al.* [BDL$^+$18] which can be tuned to be either statistically-hiding or statistically-binding. The commitment scheme of [BDL$^+$18] further retains the useful properties of [BKLP15] in that: (i) It remains additively homomorphic over R_q; (ii) It enables efficient zero-knowledge protocols that achieve negligible soundness error in one execution.

We now recall the commitment scheme described in [BDL$^+$18]. The table below summarizes the parameters used by the scheme.

Keygen$(1^\lambda, q, 1^\ell)$**:** Given a security parameter λ and the desired message space

$R = \mathbb{Z}[X]/\langle X^N + 1\rangle$	The ring over which the norm of vectors is defined
$R_q = \mathbb{Z}_q[X]/\langle X^N + 1\rangle$	The ring over which the computations take place
q	The prime modulus defining R_q
k	Width (over R_q) of the commitment matrices
n	Height (over R_q) of the commitment matrix $\mathbf{A}_1$
ℓ	Dimension (over R_q) of the message space
β	Norm bound of the committer's randomness in ℓ_∞-norm
$\mathcal{C}$	A subset of S_1 from which challenges are sampled
$\bar{\mathcal{C}}$	The set of non-zero differences $\mathcal{C} - \mathcal{C}$
κ	The maximum ℓ_1 norm of any element in $\mathcal{C}$
$\sigma = 11\kappa\beta\sqrt{kN}$	Standard deviation used in the ZK proof

R_q^ℓ, choose random matrices $\mathbf{A}_1' \hookleftarrow U(R_q^{n\times(k-n)})$, $\mathbf{A}_2' \hookleftarrow U(R_q^{\ell\times(k-n-\ell)})$ and define

$$\begin{aligned}\mathbf{A}_1 &= [\mathbf{I}_n \mid \mathbf{A}_1'] \in R_q^{n\times k}\\ \mathbf{A}_2 &= [\mathbf{0}^{\ell\times n} \mid \mathbf{I}_\ell \mid \mathbf{A}_2'] \in R_q^{\ell\times k}\end{aligned}$$

Output the commitment key

$$\mathsf{ck} = \{\mathbf{A}_1, \mathbf{A}_2\}$$

Commit$(\mathsf{ck}, \mathbf{x})$**:** In order to commit to $\mathbf{x} \in R_q^\ell$, choose $\mathbf{r} \hookleftarrow U(S_\beta^k)$ and compute

$$\mathbf{c} = \begin{bmatrix}\mathbf{c}_1\\ \mathbf{c}_2\end{bmatrix} = \begin{bmatrix}\mathbf{A}_1\\ \mathbf{A}_2\end{bmatrix}\cdot\mathbf{r} + \begin{bmatrix}\mathbf{0}^n\\ \mathbf{x}\end{bmatrix} \in R_q^{n+\ell}. \tag{2}$$

Output the commitment string $\mathbf{c} \in R_q^{n+\ell}$ and retain $\mathbf{r}$ as a state information allowing to open the commitment.

Open$(\mathbf{c}, \mathbf{x}, \mathbf{r})$**:** To open a commitment $\mathbf{c} \in R_q^{n+\ell}$, the committer reveals $\mathbf{r}$ that was used to compute (2). According to a function $f \in \bar{\mathcal{C}}$ (see below), the verifier accepts a triple $(f, \mathbf{x}, \mathbf{r}) \in \bar{\mathcal{C}} \times R_q^\ell \times R_q^k$ as a valid opening of $\mathbf{c}$ if $\mathbf{r} = \begin{bmatrix}r_1\\ \dots\\ r_k\end{bmatrix}$ satisfies $\|r_i\|_2 \le 4\sigma\sqrt{N}$ for all $i \in [k]$ and

$$f\cdot\begin{bmatrix}\mathbf{c}_1\\ \mathbf{c}_2\end{bmatrix} = \begin{bmatrix}\mathbf{A}_1\\ \mathbf{A}_2\end{bmatrix}\cdot\mathbf{r} + f\cdot\begin{bmatrix}\mathbf{0}^n\\ \mathbf{x}\end{bmatrix}. \tag{3}$$

Otherwise, the verifier rejects the opening $(f, \mathbf{x}, \mathbf{r})$.

In the scheme, the opening does not only consist of the randomness $\mathbf{r}$ and message $\mathbf{x}$, but also includes a polynomial $f \in \bar{\mathcal{C}}$. The reason is that, in the zero-knowledge proofs of knowledge of $\mathbf{r}$ and $\mathbf{x}$ satisfying (3), the knowledge extractor does not guarantee the extraction of $f = 1$ from the prover. If the prover is honest, then the extractor will exactly recover $(\mathbf{r}, \mathbf{x})$ satisfying (3) with $f = 1$. Moreover, if an honest

committer only wants to open the commitment without giving a zero-knowledge proof, he can reveal $(\mathbf{r}, \mathbf{x})$ and $f = 1$. We also note that the randomness in the commitment is generated according to a distribution using the ℓ_∞-norm while the opening is using the ℓ_2-norm. The reason for this is that the most efficient lattice-based zero-knowledge proofs prove the knowledge of small vectors in the ℓ_2-norm.

From a security point of view, Baum *et al.* proved [BDL+18, Section 4.2] that their commitment scheme is binding under the $\mathsf{DSK}^2_{n,k,\beta}$ assumption for $\beta = 16\sigma\sqrt{\kappa N}$. They also proved its hiding property under the $\mathsf{DSK}^\infty_{n+\ell,k,\beta}$ assumption.

As explained in Section 2.3, the parameters can be tuned so as to make the scheme statistically-hiding and computationally binding or the other way around. In the statistically-hiding case, security can entirely rely on the SKS^2 (equivalently, M-SIS) problem. In the statistically-binding case, security relies on the DKS^∞ (and thus M-LWE) problem. In the most efficient instantiation, however, the scheme is neither statistically-hiding nor statistically-binding (i.e., it is only computationally hiding and computationally binding). Such a parameter choice was recently used in a construction of group signatures from the Module-LWE assumption [DPLS18].

Open problems. In the context of lattice-based commitment schemes, an important open problem is to come up with a statistically hiding scheme which is simultaneously homomorphic modulo an odd prime and length-reducing (i.e., the commitment string is shorter than the committed message).

4 Zero-Knowledge Proofs

A *Zero-Knowledge proof* [GMR85] (or **ZK proofs**) is an *interactive proof* between a prover and a verifier at the end of which the verifier should be convinced of the truth of a statement (within some probability, called *soundness error*), while the prover is guaranteed that the verifier learns nothing more that the authenticity of the statement.

One of the early applications of ZK proofs in cryptography was the design of identification systems [FS87]. The goal is for a user A to prove the knowledge of a secret (such as a password) to user B without revealing any piece of information about the secret, otherwise user B would be able to impersonate A. Since then, the use of zero-knowledge proofs is now widespread in privacy-preserving protocols like anonymous credentials [Cha85, CL01], (revocable) group signatures [CVH91, NFHF09], e-cash [CHL05], oblivious transfer [CDN09] ...

If these primitives flourish in the context of number-theory-based cryptography (such as RSA or pairing groups), they are still elusive in the lattice world. In this section, we first present Σ-protocols, which can be used to design those ZK proof systems. Then, we will describe two families of ZK proofs that may prove useful in the context of pairing-based and lattice-based cryptography. Namely, Schnorr-like proofs and Stern-like proofs.

4.1 Σ-protocols

A way to construct zero-knowledge proofs – that will be described with more details in Section 4.2 – is a black-box transformation from a Σ-*protocol* and a *commitment scheme*. The resulting proof system remains secure against malicious verifiers.

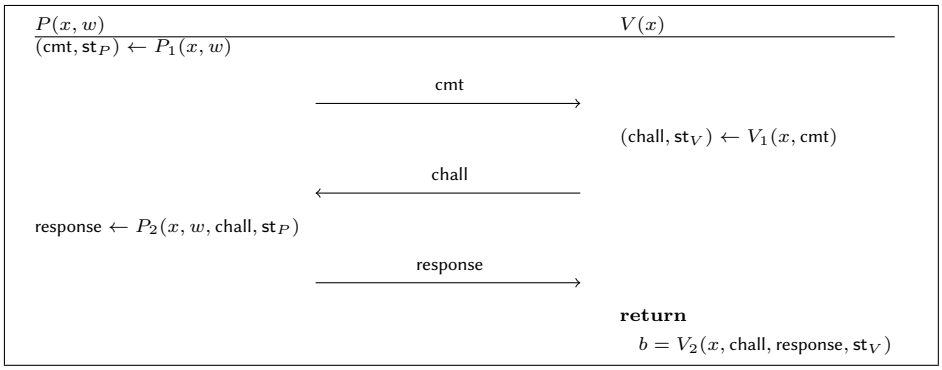

Figure 1: Abstract description of a Σ-protocol.

Definition 8 (Σ-protocol [Cra96]) *Let $R = \{(x, w)\}$ be a binary relation. A Σ-protocol is a 3-move interactive protocol between P and V that follows Figure 1 and verifies the following properties.*

- **Completeness.** *For any $(x, w) \in R$, $P(x, w)$ and $V(x)$ that follows the protocol, the verifier always accepts.*
- **2-Special soundness.** *For any x and any pair of accepting transcripts on input x of the form (cmt, chall, response) and (cmt, chall′, response′), there exists a* PPT *algorithm* extract *that inputs the two aforementioned transcripts and outputs an element w such that $(x, w) \in R$.*
- **Honest-Verifier Zero-Knowledge.** *There exists a* PPT *simulator S, such that the two probability distributions $\{\mathsf{trans}(P(x, w), V(x))\}$ and $\{S(x)\}$ with honest P and V are statistically indistinguishable.*

An example of Σ-protocol is given in Section 4.2, as well as its transformation into a Zero-Knowledge proof using a commitment scheme.

4.2 Schnorr Proofs

Schnorr's methodology [Sch96] to construct proof systems relies on the aforementioned Σ-protocol technique to design zero-knowledge proofs. Its intuition follows: given a commitment scheme (Setup, Commit, Verify), where the randomness r used in Commit is made explicit, the first move of the prover P consists in binding the randomness used in the commitment scheme r using the transmitted value $\rho = g^r$, then the verifier asks the prover to commit to a challenge message c using the randomness carried by ρ, and the prover sends the opening for this commitment open. Finally, the verifier accepts if and only if the commitments opens to its challenge message.

This methodology has been adapted to ideal lattices by Lyubashevsky [Lyu08, Lyu09] along with a technique called *rejection sampling* in order to construct ZK proofs from ideal lattice assumptions and is described in Figure 2. In this description D_y and D_c are the distributions from which $\mathbf{y}$ and $\mathbf{c}$ have to be sampled respectively, and G describes the set of *good* responses $\mathbf{z}$ in order not to leak information about $\mathbf{s}$. The part between brackets is called the *rejection* phase, and ensures that the transmitted $\mathbf{z}_1, \mathbf{z}_2$ will not leak any information about $\mathbf{s}_1, \mathbf{s}_2$ to V. This part induced a noticeable

Common input: A public element $\mathbf{a} \in R$ where $R = \mathbb{Z}_p[\mathbf{x}]/\langle \mathbf{x}^n + 1 \rangle$.

Schnorr's Protocol for Ring-SIS

$P(\mathbf{t} = \mathbf{a} \cdot \mathbf{s}, \mathbf{s})$		$V(\mathbf{t})$
$\mathbf{y} \hookleftarrow D_y \in R$		
$\mathbf{w} = \mathbf{a} \cdot \mathbf{y} \in R$		
	$\xrightarrow{\mathbf{w}}$	
		$\mathbf{c} \hookleftarrow D_c \in R$ (small)
	$\xleftarrow{\mathbf{c}}$	
$\mathbf{z} \leftarrow \mathbf{sc} + \mathbf{y} \in R$		
$\big[\textbf{if } \mathbf{z} \notin G \textbf{ then}$		
$\quad \mathbf{z} \leftarrow \perp\big]$		
	$\xrightarrow{\mathbf{z}}$	
		if $\mathbf{z} \in G \wedge$
		$\quad \mathbf{a} \cdot \mathbf{z} = \mathbf{tc} + \mathbf{w}$ **then**
		$\quad$ **return** 1
		else
		$\quad$ **return** 0

Figure 2: The Schnorr Σ-protocol for Ring-SIS.

error-rate where the prover aborts the proof. As the protocol is proven *witness indistinguishable* [Lyu09], one can run the protocol multiple times in parallel and hope that one of them will not abort [FS90].

One can notice that this is *not* a Σ-protocol in the strict sense as the knowledge extractor outputs *witnesses* that can be up to $\widetilde{\mathcal{O}}(n)$ larger than the actual witness in infinity norm. This behavior is sometimes called "*imperfect soundness*" or "*soundness slack*".

Open problems. However, this method suffers from *limited expressiveness*: the relations that can be proved with this proof system are essentially restricted to be knowledge of a Ring-LWE secret (or adapted to Ring-SIS), which is not sufficient to prove, for instance, the knowledge of a signature on a committed message. Moreover, the gap in the extraction makes it hard, although, to prove that an underlying message under an encryption is binary [DPLNS17]. Still, using some more structures on the way the ring is constructed allows using this proof system for advanced privacy-preserving primitives such as group signatures, as did del Pino, Lyubashevsky and Seiler [DPLS18].

4.3 Stern-like Proofs

Stern's protocol has originally been introduced in the context of code-base cryptography [Ste96]. Initially, it was designed for Syndrome Decoding Problem (SDP): given a matrix $\mathbf{M} \in \mathbb{F}_2^{n \times m}$ and a syndrome $\mathbf{v} \in \mathbb{F}_2^n$, the goal is to find a binary vector $\mathbf{w} \in \mathbb{F}_2^m$

with fixed hamming weight w such that $\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod 2$.

After the initial work of Kawachi, Tanaka and Xagawa [KTX08] to extend Stern's proofs to statements $\bmod q$, the results of Ling, Nguyen, Stehlé and Wang [LNSW13] enable the use of Stern's protocol to prove general SIS or LWE statements (meaning proving knowledge of a solution to these problems). These advances in the expressiveness of Stern-like protocols have been used to further improve them and therefore enable privacy-based primitives for which no constructions previously existed from lattice assumptions, such as dynamic group signatures [LLM+16a], group encryption [LLM+16b], electronic cash [LLNW17], etc. In section 6.2.4, we describe a *signature with efficient protocols* and its companion protocols which rely on Stern-like proof systems which are described in Section 6.2.4.

Unlike Schnorr-like proofs that we described in the previous section, Stern-like proofs are mainly combinatorial and rely on the fact that every permutation on a binary vector $\mathbf{w} \in \{0,1\}^m$ leaves its Hamming weight w invariant. As a consequence, for $\pi \in \mathcal{S}_m$, $\mathbf{w}$ satisfies these conditions if and only if $\pi(\mathbf{x})$ also does. Therefore, the randomness of π is used to verify these two constraints (being binary and having fixed Hamming weight) in a zero-knowledge fashion. We can notice that this can be extended to vectors $\mathbf{w} \in \{-1,0,1\}^m$ having fixed numbers of -1 and 1. This property allowed [LNSW13] to propose the generalization of this protocol to any $\mathsf{ISIS}_{n,m,q,\beta}$ statements. In Section 4.3.2, we describe these permutations while abstracting the set of ZK-provable statements as the set VALID that satisfies conditions (6).

In this Section, we describe in a high-level manner the behavior of Stern-like protocols before detailing it.

4.3.1 The Decomposition-Extension Framework

The original Stern protocol was designed to prove knowledge of an SDP preimage. That is, to prove the knowledge of a vector $\mathbf{w} \in \{0,1\}^m$ that verifies

$$\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod 2. \tag{4}$$

A first improvement by [KTX08] was to extend this protocol using a statistically hiding SIS-based commitment scheme as described in Figure 3 to prove in (statistical) zero-knowledge that

$$\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod q. \tag{5}$$

The details of this proof is given in Section 4.3.2, but it can be summarized in the following Lemma.

Lemma 4.1 ([KTX08, Se. 4]) *There exists a statistical ZKAoK with perfect completeness and soundness error 2/3 to prove the knowledge of a secret vector* $\mathbf{w} \in \{0,1\}^m$ *that verifies relation* (5) *for public input* $(\mathbf{M}, \mathbf{v}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^n$.

Ling, Nguyen, Stehlé and Wang [LNSW13] noticed that the ZKAoK of Lemma 4.1 works in a straightforward manner to prove knowledge of a vector in $\{-1,0,1\}^m$.

To prove the knowledge of an ISIS preimage, i.e. the knowledge of a bounded vector $\mathbf{w} \in [-B, B]^m$ that satisfies relation (5), the goal is to rewrite $\mathbf{w}$ as $\bar{\mathbf{w}} = \mathbf{K} \cdot \mathbf{w} \bmod q$ with a public transformation matrix $\mathbf{K}$ such that $\bar{\mathbf{w}} \in \{-1,0,1\}^{m'}$ and of known numbers of elements equal to j for each $j \in \{-1,0,1\}$. This reduces to use Lemma 4.1 to prove the knowledge of $\bar{\mathbf{w}} \in \{-1,0,1\}^{m'}$ for public input $(\mathbf{M} \cdot \mathbf{K}, \mathbf{v})$.

To construct such a transfer matrix $\mathbf{K}$, [LNSW13] showed that *decomposing* a vector $\mathbf{x} \in [-B, B]^m$ as a vector $\tilde{\mathbf{x}} \in \{-1,0,1\}^{m \cdot \delta_B}$ and *extending* the resulting vector into

1. **Commitment:** Prover samples $\mathbf{r}_w \leftarrow \mathcal{U}(\mathbb{Z}_q^D)$, $\phi \leftarrow \mathcal{U}(\mathcal{S})$ and randomnesses ρ_1, ρ_2, ρ_3 for COM. Then, he sends $\mathrm{CMT} = \big(C_1, C_2, C_3\big)$ to the verifier, where

$$C_1 = \mathsf{COM}(\phi, \mathbf{M} \cdot \mathbf{r}_w \bmod q; \rho_1), \quad C_2 = \mathsf{COM}(\Gamma_\phi(\mathbf{r}_w); \rho_2),$$
$$C_3 = \mathsf{COM}(\Gamma_\phi(\mathbf{w} + \mathbf{r}_w \bmod q); \rho_3).$$

2. **Challenge:** The verifier sends a challenge $Ch \hookleftarrow \mathcal{U}(\{1, 2, 3\})$ to the prover.
3. **Response:** Depending on Ch, the prover sends RSP computed as follows:
   - $Ch = 1$: Let $\mathbf{t}_w = \Gamma_\phi(\mathbf{w})$, $\mathbf{t}_r = \Gamma_\phi(\mathbf{r}_w)$, and $\mathrm{RSP} = (\mathbf{t}_w, \mathbf{t}_r, \rho_2, \rho_3)$.
   - $Ch = 2$: Let $\phi_2 = \phi$, $\mathbf{w}_2 = \mathbf{w} + \mathbf{r}_w \bmod q$, and $\mathrm{RSP} = (\phi_2, \mathbf{w}_2, \rho_1, \rho_3)$.
   - $Ch = 3$: Let $\phi_3 = \phi$, $\mathbf{w}_3 = \mathbf{r}_w$, and $\mathrm{RSP} = (\phi_3, \mathbf{w}_3, \rho_1, \rho_2)$.

Verification: Receiving RSP, the verifier proceeds as follows:

- $Ch = 1$: Check that

$$\mathbf{t}_w \in \mathsf{VALID},$$
$$C_2 = \mathsf{COM}(\mathbf{t}_r; \rho_2), \qquad C_3 = \mathsf{COM}(\mathbf{t}_w + \mathbf{t}_r \bmod q; \rho_3).$$

- $Ch = 2$: Check that

$$C_1 = \mathsf{COM}(\phi_2, \mathbf{M} \cdot \mathbf{w}_2 - \mathbf{v} \bmod q; \rho_1), \qquad C_3 = \mathsf{COM}(\Gamma_{\phi_2}(\mathbf{w}_2); \rho_3).$$

- $Ch = 3$: Check that

$$C_1 = \mathsf{COM}(\phi_3, \mathbf{M} \cdot \mathbf{w}_3; \rho_1), \qquad C_2 = \mathsf{COM}(\Gamma_{\phi_3}(\mathbf{w}_3); \rho_2).$$

In each case, the verifier outputs 1 if and only if all the conditions hold.

Figure 3: Stern-like ZKAoK for the relation $\mathrm{R}_{\mathrm{abstract}}$.

$\bar{\mathbf{x}} \in \mathsf{B}^3_{m\delta_B}$ leads to a new statement that can be proven using the variant of Stern's protocol described in [KTX08]. The resulting matrix $\mathbf{K} = \big[\mathbf{K}_{m,B} \mid \mathbf{0}^{m \times 2m\delta_B}\big] \in \mathbb{Z}^{m \times 3m\delta_B}$, where $\mathbf{K}_{m,B}$ is the $\{-1, 0, 1\}$-decomposition matrix $\mathbf{K}_{m,B} = \mathbf{I}_m \otimes [B_1 \mid \cdots \mid B_{\delta_B}]$ with $B_j = \left\lfloor \frac{B + 2^{j-1}}{2^j} \right\rfloor$, for all $j \in \{1, \ldots, j\}$, can be computed from public parameters.

4.3.2 Abstraction of Stern's Protocol

Let K, D, q be positive integers with $D \geq K$ and $q \geq 2$, and let VALID be a subset of $\mathbb{Z}^D$. Suppose that $\mathcal{S}$ is a finite set such that every element $\phi \in \mathcal{S}$ can be associated with a permutation $\Gamma_\phi \in \mathcal{S}_D$ satisfying the following conditions:

$$\begin{cases} \mathbf{w} \in \mathsf{VALID} \iff \Gamma_\phi(\mathbf{w}) \in \mathsf{VALID}, \\ \text{If } \mathbf{w} \in \mathsf{VALID} \text{ and } \phi \text{ is uniform in } \mathcal{S}, \text{ then } \Gamma_\phi(\mathbf{w}) \text{ is uniform in } \mathsf{VALID}. \end{cases} \tag{6}$$

We aim to construct a statistical Zero-Knowledge Argument of Knowledge (ZKAoK) for the following abstract relation:

$$\mathrm{R}_{\mathrm{abstract}} = \big\{\big((\mathbf{M}, \mathbf{v}), \mathbf{w}\big) \in \mathbb{Z}_q^{K \times D} \times \mathbb{Z}_q^K \times \mathsf{VALID} : \mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod q.\big\}$$

Note that, Stern's original protocol corresponds to the special case when the set $\mathsf{VALID} = \{\mathbf{w} \in \{0,1\}^D : \mathsf{wt}(\mathbf{w}) = k\}$, where $\mathsf{wt}(\cdot)$ denotes the Hamming weight and $k < D$ is a given integer, $\mathcal{S} = \mathcal{S}_D$ is the set of all permutations of D elements and $\Gamma_\phi(\mathbf{w}) = \phi(\mathbf{w})$.

The conditions in (6) play a crucial role to prove in zero-knowledge that $\mathbf{w} \in \mathsf{VALID}$. To this end, the prover samples a random $\phi \hookleftarrow \mathcal{U}(\mathcal{S})$ and lets the verifier check that $\Gamma_\phi(\mathbf{w}) \in \mathsf{VALID}$ without learning any additional information about $\mathbf{w}$ due to the randomness of ϕ. Furthermore, to prove in a zero-knowledge manner that the linear equation is satisfied, the prover samples a masking vector $\mathbf{r}_w \hookleftarrow \mathcal{U}(\mathbb{Z}_q^D)$, and convinces the verifier instead that $\mathbf{M} \cdot (\mathbf{w} + \mathbf{r}_w) = \mathbf{M} \cdot \mathbf{r}_w + \mathbf{v} \bmod q$.

The interaction between prover $\mathcal{P}$ and verifier $\mathcal{V}$ is described in Figure 3. The protocol uses a statistically hiding and computationally binding string commitment scheme COM (e.g., the SIS-based scheme from [KTX08]). As described in Figure 3, the basic protocol uses a ternary challenge space and has soundness error $2/3$. Hence, $O(\lambda)$ iterations of this basic protocols are necessary (about 200 to achieve 128-bit security) to make the soundness error negligible.

Theorem 4.1 *The protocol in Figure 3 is a statistical* ZKAoK *with perfect completeness, soundness error* $2/3$*, and communication cost* $\mathcal{O}(D \cdot \log q)$*. Namely:*

- *There exists a polynomial-time simulator that, on input* $(\mathbf{M}, \mathbf{v})$*, outputs an accepted transcript statistically close to that produced by the real prover.*
- *There exists a polynomial-time knowledge extractor that, on input a commitment* CMT *and* 3 *valid responses* $(\mathrm{RSP}_1, \mathrm{RSP}_2, \mathrm{RSP}_3)$ *to all* 3 *possible values of the challenge* Ch*, outputs* $\mathbf{w}' \in \mathsf{VALID}$ *such that* $\mathbf{M} \cdot \mathbf{w}' = \mathbf{v} \bmod q$*.*

The proof of the theorem relies on standard simulation and extraction techniques for Stern-like protocols [KTX08, LNSW13, LLM+16a].

Open problems. In the context of zero-knowledge proofs, a major open problem is to find a way to combine the expressiveness of Stern-like proofs and the efficiency Schnorr-like proofs. While the former makes it possible to prove relatively expressive statements, it suffers from a significant lack of efficiency induced by the $2/3$-soundness error of the basic protocol. In order to make the soundness error negligible, this basic protocol thus has to be repeated $O(\lambda)$ times, which blows up its communication and computational complexity by a similar factor. The "Fiat-Shamir with aborts" technique [Lyu09] can achieve a negligible soundness error in one short (i.e., without repeating the basic protocol $O(\lambda)$ times) but it currently remains an open problem to use it in order to prove expressive statements (at least without decomposing them into a circuit as in [BKLP15]).

Another major open problem in theoretical cryptography is to determine whether general non-interactive zero-knowledge proofs [BFM88] (in the common reference string model and without relying on the random oracle methodology [BR93]) can solely rely on quantum-resistant assumptions. While initial steps have been taken in this direction (e.g., [PW08, RSS19, KW18, CLW]), it remains a long-standing open problem whether NIZK proofs can rely on the LWE assumption alone.

5 Lattice-Based Pseudorandom Functions

A pseudorandom function (PRF) family [GGM86] is a set $\mathcal{F}$ of keyed functions with common domain Dom and range Rng such that no PPT adversary can distinguish a real experiment, where it has oracle access to a random member $f \hookleftarrow \mathcal{F}$ of the PRF family, from an ideal experiment where it is interacting with a truly random function $R : \mathsf{Dom} \to \mathsf{Rng}$. To be useful, a PRF should be efficiently computable – meaning that $F_{\mathbf{s}}(x)$ must be deterministically computable in polynomial time given the key $\mathbf{s}$ and the input $x \in \mathsf{Dom}$ – and the key size must be polynomial.

PRFs are fundamental objects in cryptography as most central tasks of symmetric cryptography (like secret-key encryption or message authentication) can be efficiently realized from a secure PRF family. Moreover, algebraic pseudorandom functions (e.g., [NR97]) come in handy to build privacy-preserving cryptographic protocols, like e-cash systems [CHL05], as they are easier to combine with zero-knowledge protocols. Algebraic PRFs naturally appear in many protocols where a prover has to deterministically generate a random-looking value without revealing his identity while proving that this value has been correctly evaluated.

Goldreich, Goldwasser and Micali (GGM) [GGM86] showed how to build a PRF from any length-doubling pseudorandom generator (PRG). In turn, PRGs are known [HILL99] to exist under the sole assumption that one-way functions exist. However, much more efficient constructions can be obtained by relying on specific number theoretic assumptions like the Decision Diffie-Hellman assumption [NR97].

In lattice-based cryptography, the noisy nature of LWE makes it non-trivial to design efficient PRF families. In order to design PRFs with small-depth evaluation circuits, several works [BPR12, BLMR13, BP14] rely on the Learning-With-Rounding (LWR) technique [BPR12], which is a "de-randomization" of LWE where noisy vectors $\mathbf{A}\cdot\mathbf{s}+\mathbf{e}$ are replaced by rounded vectors $\lfloor \mathbf{A}\cdot\mathbf{s} \rceil_p = \lfloor (p/q)\cdot(\mathbf{A}\cdot\mathbf{s}) \rfloor \in \mathbb{Z}_p^m$ for a smaller modulus $p < q$. For suitable parameters (e.g., $q = p^2$ and $m = 4n$), the LWR problem immediately implies a length-doubling PRG $G : \mathbb{Z}_q^n \to \mathbb{Z}_p^{4n}$ where $G(\mathbf{s}) = \lfloor \mathbf{A} \cdot \mathbf{s} \rceil_p$. This provides a PRF based on the LWR problem via the GGM construction [GGM86]. In turn, this implies an LWE-based PRF via the results of Banerjee *et al.* [BPR12] who showed the polynomial hardness of LWR when q/p is super-polynomial.

The PRF constructions of [BPR12, BLMR13, BP14] all rely on lattice assumptions with super-polynomial approximation factors. To our knowledge, the only lattice-based PRF with a security proof under a standard assumption with polynomial approximation factor is the GGM-based construction implied by the LWE-to-LWR reduction of Alwen *et al.* [AKPW13], which works for polynomial moduli and inverse-error rates. This construction, however, inherits the disadvantage of all GGM-based constructions as its evaluation depth is linear in the input length. So far, it remains an open problem to come up with a lattice-based PRF that simultaneously features a small-depth (e.g., NC1 or even NC2) evaluation circuit and a security proof under standard lattice assumptions with polynomial approximation factors. Indeed, the NC1-circuit construction of Banerjee *et al.* [BPR12] requires an exponential approximation factor in the input length. Subsequent works [DS15, JKP18] gave variants under weaker lattice assumptions, but they still need super-polynomial moduli and inverse-error rates.

An appealing advantage of lattice-based techniques is that they enable the design of *key-homomorphic* PRF families in the standard model [BLMR13, BP14].[1] Namely, as-

[1] In the random oracle model, key-homomorphic PRFs were already known to exist under the Decision Diffie-Hellman assumption [NPR99].

suming that their range and key space form an additive group, for any input x and keys $\mathbf{s}, \mathbf{t}$, we have $F_{\mathbf{s}+\mathbf{t}}(x) \approx F_{\mathbf{s}}(x) + F_{\mathbf{t}}(x)$. In turn, key-homomorphic PRFs provide simple and non-interactive constructions of distributed pseudorandom functions [NPR99]. Lattices actually enable the design of PRFs with other advanced properties, such as constrained pseudorandom functions [BW13, KPTTZ13, BGI14], key-homomorphic constrained PRFs [BFP+15, BV15] or watermarkable PRFs [KW17]. Constrained PRFs (CPRFs) are relevant to this project as they can be used in the design of *divisible* e-cash protocols [Oka95, CG07]. In short, CPRFs are pseudorandom functions where the secret key SK allows deriving sub-keys that can be used to evaluate the function on a restricted portion of its domain. The LWE assumption enables CPRF constructions [BV15, BTVW17, PS18a] for arbitrary circuits: namely, a sub-key SK_C corresponds to a Boolean circuit C and allows evaluating the function on any input x such that $C(x) = 1$. Some LWE-based CPRFs [CC17, BKM17, BTVW17, PS18a] additionally support sub-keys SK_C that hide the underlying constraint C. One caveat is that all known LWE-based constructions (with or without constraint privacy) are only secure in the single-key setting: namely, security is only guaranteed as long as the adversary obtains a single sub-key SK_C for a circuit C of its choice. It remains an open problem to build CPRFs that remain secure under standard assumptions when the adversary obtains a polynomial number of sub-keys SK_C.

In order to lend themselves to the design of privacy-preserving protocols, PRFs often need to be compatible with zero-knowledge protocols. This is the reason why e-cash systems [CHL05], for example, appeal to algebraic pseudorandom functions (e.g., [NR97]) which make it easier to prove that a given value is the correct PRF evaluation for committed inputs and keys. It was shown in [LLNW17] that the GGM-based PRF of Banerjee, Peikert and Rosen [BPR12] can be combined with Stern-like protocols to construct anonymous e-cash systems by adapting the design principle of Camenisch *et al.* [CHL05].

Open problems. One problem is that the many parallel repetitions of Stern-like proof (which are necessary to achieve negligible soundness error) make the constructions of [LLNW17] really far from being practical. It thus remains a challenging open problem to find pseudorandom functions that can smoothly interact with more efficient zero-knowledge protocols based on the "Fiat-Shamir with aborts" technique [Lyu09].

6 Signatures Schemes

Signature schemes have been initially designed to provide message authenticity, message integrity and signer's non repudiation. But for a number of new usages, it is necessary to add some extensions to basic signature schemes. This is typically the case in a privacy-preserving setting. In this setting, we review some of these extensions, but we first give some words about basic signatures.

6.1 Basic Signature Schemes

Formally speaking, a signature scheme is composed of three PPT algorithms, namely, Keygen to generate the key pair, Sign to provide a signature on a message under a private key, and Verify to verify the previously computed signature, given the message and the verification public key.

A lot of basic signature schemes based on lattices now exist. Among them, several ones have been submitted to the NIST call for proposal and after the end of the first round, it remains three of them:

- CRYSTALS-DILITHIUM [DKL+18], based on the Fiat-Shamir paradigm and the module SIS/LWE problem;
- FALCON [FHK+17], based on the hash and sign paradigm and the SIS problem over NTRU lattices;
- qTESLA [ABB+19], based on the Fiat-Shamir paradigm and the ring SIS/LWE problem.

As our first purpose is to work on privacy-preserving cryptographic protocols, one option can be to use or to transform those signatures to fit our needs.

6.2 Signature Schemes with Efficient Protocols

To be meaningful in privacy-preserving cryptography, signatures have to interact with other cryptographic building blocks. For these signatures to be meaningful, it is crucial to be able to prove relations between a message and its signature. Many solutions exist, for instance, in the pairing setting there are the so-called structure-preserving signatures [AFG+10], and Camenisch and Lysyanskaya [CL01] formalized *signature with efficient protocols*. This primitive is a signature scheme empowered with two companion protocols allowing: (i) A user to obtain a signature on a committed value; (ii) To prove in a zero-knowledge manner the possession of a message-signature pair. This building block basically captures most of the usage where a signature is needed for in privacy-preserving cryptography and this is supported by several applications in privacy-preserving cryptography, as in e-cash [Cha82a], group signatures [CVH91] and anonymous credentials [Cha85].

More formally, a signature with efficient protocols is a triple of PPT algorithms $(\mathsf{Keygen}, \mathsf{Sign}, \mathsf{Verify})$ which goes along with two companion (interactive) protocols $(\mathsf{Issue} \leftrightarrow \mathsf{Obtain}, \mathsf{Prove})$ that respectively allow obtaining a signature on a committed message, and proving the knowledge of a message-signature pair in a zero-knowledge fashion.

A list of signature schemes is given in Table 1 and some of them are detailed in the following sections.

6.2.1 Gentry-Peikert-Vaikuntanathan Signature Scheme

Gentry, Peikert and Vaikuntanathan introduced in [GPV08] a lattice-based signature scheme in the random oracle model, which is fairly efficient and serves as a basis for the other lattice-based signature schemes. This construction is presented in this section.

Keygen(1^λ)**:** Given a security parameter λ, this algorithm selects parameters n, m and $q \in \mathbb{N}$ as well as a hash function $H : \{0,1\}^\star \to \mathbb{Z}_q^n$. Then it runs $\mathsf{TrapGen}(1^n, 1^m, q)$ to obtain $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a short basis $\mathbf{T_A}$ of $\Lambda_q^\perp(\mathbf{A})$ with a Gaussian parameter σ that defines the bound $B = \Omega(\sigma\sqrt{n})$. Finally, it sets $PK := (\mathbf{A}, H)$ and $SK := \mathbf{T_A}$.

Scheme	Pub key $R_q^{1\times k}$ mat.	Secret key $R_q^{k\times k}$ mat.	Signature R_q^k vec.	SIS param β
[LM08]	1	1	$\log n$	$\tilde{\Omega}(n^2)$
[CHKP10]	n	n	n	$\tilde{\Omega}(n^{3/2})$
[Boy10, MP12]	n	n	1	$\tilde{\Omega}(n^{7/2}), \tilde{\Omega}(n^{5/2})$
[BHJ$^+$15]	1	1	d	$\tilde{\Omega}(n^{5/2})$
[DM14]	d	1	1	$\tilde{\Omega}(n^{7/2})$
[AS15]	1	1	1	$\tilde{\Omega}(d^{2d} \cdot n^{11/2})$

The comparison is made in the ring setting as some of the above schemes ([LM08, DM14]) have no realization in the general lattice setting. For schemes using the *confined guessing* technique in their security proof, d is a value satisfying $2Q^2/\varepsilon < 2^{\lfloor c^d \rfloor}$ for an arbitrary constant $c > 1$ (which controls the trade-off between public key size and the reduction loss).

Table 1: Comparison table between *standard model* lattice-based signature schemes in the ring setting

Sign$\big(SK, \mathsf{Msg}\big)$**:** Given a secret key SK parsed as $\mathbf{T_A}$ and a message $\mathsf{Msg} \in \{0,1\}^\star$, first check if m have already been signed. If so, then return the corresponding signature Sig. Else, run the $\mathsf{GPVSample}$ algorithm using the trapdoor $\mathbf{T_A}$ to get a vector $\mathbf{u} \in \mathbb{Z}^m$ such that:

$$\mathbf{A} \cdot \mathbf{u} = H(\mathsf{Msg}) \bmod q$$

Output the signature $sig := \mathbf{u}$.

Verify$\big(PK, \mathsf{Msg}, sig\big)$**:** To verify a signature $sig \in \mathbb{Z}^m$ on a message Msg, this algorithm accepts if and only if $\|sig\| \leq B$ and $\mathbf{A} \cdot \mathbf{u} = H(\mathsf{Msg}) \bmod q$.

6.2.2 Boyen's Lattice-Based Signature Scheme

We first recall a signature scheme proposed by Boyen [Boy10], which was used as a building block for several group signatures. In this section, we present the improved variant due to Micciancio and Peikert [MP12].

Keygen$(1^\lambda, 1^L)$**:** On input of a security parameter $\lambda > 0$ and a message length $\ell = \mathcal{O}(\lambda)$, choose $n = \mathcal{O}(\lambda)$, a prime modulus $q = \mathsf{poly}(\lambda)$, a dimension $m > 2n\lceil \log q \rceil$; and parameters $\sigma = \Omega(\sqrt{\ell n \log q} \log n)$. Define the message space as $\mathcal{M} = \{0,1\}^\ell$.

1. Run $\mathsf{TrapGen}(1^n, 1^m, q)$ to get $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a short basis $\mathbf{T_A}$ of $\Lambda_q^\perp(\mathbf{A})$. This basis allows computing short vectors in $\Lambda_q^\perp(\mathbf{A})$ with a Gaussian parameter σ. Next, choose $\ell + 1$ random $\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_\ell \hookleftarrow U(\mathbb{Z}_q^{n\times m})$.
2. Choose a random vector $\mathbf{u} \hookleftarrow U(\mathbb{Z}_q^n)$.

The private signing key consists of $SK := \mathbf{T_A}$ while the public key is comprised of $PK := \big(\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^\ell, \mathbf{u}\big)$.

Sign$\big(SK, \mathsf{Msg}\big)$**:** To sign a message $\mathsf{Msg} = \mathsf{Msg}[1] \ldots \mathsf{Msg}[\ell] \in \{0,1\}^\ell$,

1. Compute a short basis $\mathbf{T}_{\mathsf{Msg}} \in \mathbb{Z}^{2m \times 2m}$ for the lattice $\Lambda_q^{\perp}(\mathbf{A}_{\mathsf{Msg}})$, where

$$\mathbf{A}_{\mathsf{Msg}} = [\mathbf{A} \mid \mathbf{A}_0 + \sum_{j=1}^{\ell} \mathsf{Msg}[j] \cdot \mathbf{A}_j] \in \mathbb{Z}_q^{n \times 2m}. \quad (7)$$

2. Using the delegated basis $\mathbf{T}_{\mathsf{Msg}} \in \mathbb{Z}^{2m \times 2m}$, sample a vector $\mathbf{v} \in \mathbb{Z}^{2m}$ in $D_{\Lambda_q^{\mathbf{u}}(\mathbf{A}_{\mathsf{Msg}}),\sigma}$.

Output the signature $sig = \mathbf{v} \in \mathbb{Z}^{2m}$.

Verify$\big(PK, \mathsf{Msg}, sig\big)$**:** Given a message $\mathsf{Msg} \in \{0,1\}^{\ell}$ and a purported signature $sig = \mathbf{v} \in \mathbb{Z}^{2m}$, compute the matrix $\mathbf{A}_{\mathsf{Msg}} \in \mathbb{Z}_q^{n \times 2m}$ as per (7). Then, return 1 if $\|\mathbf{v}\| < \sigma\sqrt{2m}$ and

$$\mathbf{A}_{\mathsf{Msg}} \cdot \mathbf{v} = \mathbf{u} \mod q. \quad (8)$$

The scheme initially proposed by Boyen [Boy10] is identical to the above construction with two differences. First, the vector $\mathbf{u} \in \mathbb{Z}_q^n$ was initially chosen to be the zero vector $\mathbf{u} = \mathbf{0}^n$. However, choosing $\mathbf{u} \neq \mathbf{0}^n$ has the advantage of making the scheme strongly unforgeable (whereas, in the original system [Boy10], any valid signature $\mathbf{v} \in \mathbb{Z}^{2m}$ allows publicly computing $-\mathbf{v} \in \mathbb{Z}^{2m}$, which is also a valid signature on the same message). The second difference is that, in [Boy10], messages were initially encoded as $\mathsf{Msg} \in \{-1,1\}^{\ell}$ instead of $\mathsf{Msg} \in \{0,1\}^{\ell}$.

Micciancio and Peikert [MP12] proved that the above variant of Boyen's signature is unforgeable under adaptive chosen-message attacks if the $\mathsf{SIS}_{n,m,q,\beta'}$ problem is hard, for a SIS parameter $\beta' = \tilde{\Omega}(n^{5/2})$.

The main disadvantage of Boyen's signature is its large public key, which has to contain $\mathcal{O}(\lambda)$ matrices in $\mathbb{Z}_q^{n \times m}$. Böhl *et al.* [BHJ+15] described a variant that only requires a constant number of matrices in the public key at the expense of increasing the signature length: in [BHJ+15], each signature contains $\mathcal{O}(\log \lambda)$ vectors. Using ideal lattices, Ducas and Micciancio [DM14] decreased the signature length to $\mathcal{O}(1)$ vectors while retaining relatively short public keys comprised of a logarithmic number of vectors containing ring elements. Alperin-Sheriff [AS15] subsequently showed how to simultaneously obtain signatures made of $\mathcal{O}(1)$ vectors and public keys containing $\mathcal{O}(1)$ matrices. However, the security proof of [AS15] requires a stronger $\mathsf{SIS}_{n,m,q,\beta}$ assumption, where $\beta = \tilde{\Omega}(d^{2d} \cdot n^{5.5})$, where $d = \log n$. Katsumata and Yamada [KY16] described a signature scheme with shorter public keys and proved it secure under the ring-SIS assumption in the standard model. Subsequently, Yamada gave a similar construction [Yam17] using standard lattices. He notably described signature schemes based on the SIS assumption where each signature consists of one vector and public keys only contain $O(\log^2 \lambda)$ matrices.

The signature schemes [BHJ+15, AS15] do not appear to interact well with existing zero-knowledge proof systems in the lattice setting. Ling *et al.* [LNWX18] showed how to prove knowledge of a Ducas-Micciancio signature [DM14] using Stern-like protocols. One of the constructions proposed by Yamada [Yam17] may be compatible with Stern-like zero-knowledge proofs depending on which error correcting code is used to encode the message in the construction of [Yam17].

6.2.3 Ducas-Micciancio ring-based signature scheme

In this section, we recall the Micciancio-Ducas [DM14] signature scheme based on ideal lattices. The notations are similar to those of Section 3.

Similarly, to the general lattice setting as reminded in Lemma 2.2, there is a PPT algorithm $\mathsf{TrapGen}_R$ [MP12] that takes as input an integer w, a tag $\mathbf{H} \in R_q$ and a parameter $s > \omega(\sqrt{\ln nw})$ and outputs a matrix $\mathbf{A} \in R_q^{1\times k}$ along with a trapdoor $\mathbf{R} \in R_q^{w\times k}$, such that its spectral norm $s_1(\mathbf{R}) := \sup_{\mathbf{x}} \|\mathbf{A}/\mathbf{x}\|/\|\mathbf{x}\| = s \cdot \mathcal{O}(\sqrt{w} + \sqrt{k} + \omega(\sqrt{\log n}))$, for it. This algorithm is usually accompanied by a PPT algorithm $\mathsf{SamplePre}$ which is similar to the $\mathsf{GPVSample}$ algorithm from Lemma 2.1. Namely, $\mathsf{SamplePre}$ takes as input a matrix $\mathbf{A} \in R_q^{1\times(w+k)}$, a syndrome $\mathbf{u} \in R_q$, a trapdoor $\mathbf{R} \in R_q^{w\times k}$ and an invertible tag $\mathbf{H} \in R_q$ and a parameter $s > \omega(\sqrt{\log n}) \cdot s_1(\mathbf{R})$ and outputs a sample statistically close to $D_{\Lambda^{\perp}_{\mathbf{u}}(\mathbf{A}),s}$.

Keygen(1^λ)**:** On input of a security parameter $\lambda > 0$, select an integer $n = \mathcal{O}(\lambda)$ which is a power of 2 and a modulus q assumed to be a power of 3, which defines the ring $R_q = \mathbb{Z}_q[X]/\langle X^n + 1\rangle$ (as n is a power of two, $X^n + 1$ coincide with $\Phi_{2n}(X)$). Also select $w = 2\lceil \log_2 q\rceil + 2$, $m = w + k$, $s = n^{3/2}\omega(\log n)^{3/2}$, $d = \log n$, $\sigma = \omega(\sqrt{\log n})$ and a collection of tags. Then:

1. Run $\mathsf{TrapGen}_R(w, \mathbf{I}, \sigma)$ to get $\mathbf{A} \in R_q^{1\times k}$ and a trapdoor $\mathbf{R} \in R_q^{w\times k}$. This trapdoor allows computing short vectors in $\Lambda^{\perp}_{\mathbf{u}}(\mathbf{A})$ with a Gaussian parameter σ. Next choose $d+2$ matrices $\mathbf{A}_0, \ldots, \mathbf{A}_d, \mathbf{U} \in R_q^{1\times k}$ uniformly at random.
2. Choose a random element $\mathbf{v} \hookleftarrow U(R_q)$.

The private signing key consists of $SK := \mathbf{R}$ while the public key is comprised of $PK := \big(\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^d, \mathbf{U}, \mathbf{v}\big)$. This public key implicitly defines a collection of matrices $\mathbf{A}_\tau = [\mathbf{A} \mid \mathbf{A}_0 + \sum_{i=1}^d \tau_i \cdot \mathbf{A}_i]$ indexed by the tags $\tau \in \mathcal{T}$.

Sign$\big(SK, \mathsf{Msg}\big)$**:** To sign a message $\mathsf{Msg} \in \{0,1\}^{nk} \subsetneq R_q^k$, first parse the message as a vector of R_q^k splitting the nk bits into k binary polynomials.

1. Sample a uniformly random tag $\tau \hookleftarrow U(\mathcal{T})$ and compute a matrix $\mathbf{A}_\tau$ and the ring element $\mathbf{u}_{\mathsf{Msg}} = \mathbf{U} \cdot \mathsf{Msg} + \mathbf{v}$.
2. With trapdoor $\mathbf{R}$, sample a short vector $\mathbf{s} \leftarrow \mathsf{SamplePre}(\mathbf{A}_\tau, \mathbf{u}_{\mathsf{Msg}}, \mathbf{R}, \sigma)$.

Output the signature $sig = (\tau, \mathbf{s}) \in \mathcal{T} \times \mathbb{Z}^k$.

Verify$\big(PK, \mathsf{Msg}, sig\big)$**:** Given a message $\mathsf{Msg} \in \{0,1\}^{nk}$ and a signature $sig = (\tau, \mathbf{s}) \in \mathcal{T} \times \mathbb{Z}^k$, compute the matrix $\mathbf{A}_\tau \in R_q^{2\times k}$ and the ring element $\mathbf{u}_{\mathsf{Msg}}$. Then, return 1 if and only if $\|\mathbf{s}\| < \sigma\sqrt{nm}$ and

$$\mathbf{A}_\tau \cdot \mathbf{s} = \mathbf{u}_{\mathsf{Msg}}. \tag{9}$$

6.2.4 The LLMNW Signature Scheme

We now describe a signature scheme proposed by Libert, Ling, Mouhartem, Nguyen and Wang [LLM+16a] who extended the Böhl *et al.* signature [BHJ+15] in order to sign messages comprised of multiple blocks while keeping the scheme compatible with zero-knowledge proofs.

Keygen$(1^\lambda, 1^{N_b})$**:** Given a security parameter $\lambda > 0$ and the number of blocks $N_b = \mathsf{poly}(\lambda)$, choose $n = \mathcal{O}(\lambda)$, a prime modulus $q = \widetilde{\mathcal{O}}(N \cdot n^4)$, a dimension $m = 2n\lceil \log q \rceil$; an integer $\ell = \mathsf{poly}(n)$ and Gaussian parameters $\sigma = \Omega(\sqrt{n \log q} \log n)$. Define the message space as $\mathcal{M} = (\{0,1\}^{m_I})^{N_b}$.

1. Run $\mathsf{TrapGen}(1^n, 1^m, q)$ to get $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a short basis $\mathbf{T}_\mathbf{A}$ of $\Lambda_q^\perp(\mathbf{A})$. This basis allows computing short vectors in $\Lambda_q^\perp(\mathbf{A})$ with a Gaussian parameter σ. Next, choose $\ell + 1$ random $\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_\ell \hookleftarrow U(\mathbb{Z}_q^{n \times m})$.
2. Choose random $\mathbf{D} \hookleftarrow U(\mathbb{Z}_q^{n \times m/2})$, $\mathbf{D}_0 \hookleftarrow U(\mathbb{Z}_q^{n \times m})$, $\mathbf{D}_j \hookleftarrow U(\mathbb{Z}_q^{n \times m_I})$ for $j \in \{1, \ldots, N_b\}$, as well as a random vector $\mathbf{u} \hookleftarrow U(\mathbb{Z}_q^n)$.

The private signing key consists of $SK := \mathbf{T}_\mathbf{A}$ while the public key is comprised of $PK := \big(\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell}, \mathbf{D}, \{\mathbf{D}_k\}_{k=0}^{N_b}, \mathbf{u}\big)$.

Sign$\big(SK, \mathsf{Msg}\big)$**:** To sign an N_b-block $\mathsf{Msg} = (\mathfrak{m}_1, \ldots, \mathfrak{m}_{N_b}) \in (\{0,1\}^{m_I})^{N_b}$,

1. Choose a random string $\tau \hookleftarrow U(\{0,1\}^\ell)$. Using $SK := \mathbf{T}_\mathbf{A}$, compute a short basis $\mathbf{T}_\tau \in \mathbb{Z}^{2m \times 2m}$ for the lattice $\Lambda_q^\perp(\mathbf{A}_\tau)$, where

$$\mathbf{A}_\tau = [\mathbf{A} \mid \mathbf{A}_0 + \sum_{j=1}^{\ell} \tau[j]\mathbf{A}_j] \in \mathbb{Z}_q^{n \times 2m}. \tag{10}$$

2. Sample $\mathbf{r} \hookleftarrow D_{\mathbb{Z}^m, \sigma}$. Compute the vector $\mathbf{c}_M \in \mathbb{Z}_q^n$ as a chameleon hash of $(\mathfrak{m}_1, \ldots, \mathfrak{m}_{N_b})$. Namely, compute $\mathbf{c}_M = \mathbf{D}_0 \cdot \mathbf{r} + \sum_{k=1}^{N_b} \mathbf{D}_k \cdot \mathfrak{m}_k \in \mathbb{Z}_q^n$, which is used to define $\mathbf{u}_M = \mathbf{u} + \mathbf{D} \cdot \mathsf{vdec}_{n,q-1}(\mathbf{c}_M) \in \mathbb{Z}_q^n$. Using the delegated basis $\mathbf{T}_\tau \in \mathbb{Z}^{2m \times 2m}$, sample a vector $\mathbf{v} \in \mathbb{Z}^{2m}$ in $D_{\Lambda_q^{\mathbf{u}_M}(\mathbf{A}_\tau), \sigma}$.

Output the signature $sig = (\tau, \mathbf{v}, \mathbf{r}) \in \{0,1\}^\ell \times \mathbb{Z}^{2m} \times \mathbb{Z}^m$.

Verify$\big(PK, \mathsf{Msg}, sig\big)$**:** Given $\mathsf{Msg} = (\mathfrak{m}_1, \ldots, \mathfrak{m}_{N_b}) \in (\{0,1\}^{m_I})^{N_b}$ and

$$sig = (\tau, \mathbf{v}, \mathbf{r}) \in \{0,1\}^\ell \times \mathbb{Z}^{2m} \times \mathbb{Z}^m,$$

return 1 if $\|\mathbf{v}\| < \sigma\sqrt{2m}$, $\|\mathbf{r}\| < \sigma\sqrt{m}$ and

$$\mathbf{A}_\tau \cdot \mathbf{v} = \mathbf{u} + \mathbf{D} \cdot \mathsf{vdec}_{n,q-1}(\mathbf{D}_0 \cdot \mathbf{r} + \sum_{k=1}^{N_b} \mathbf{D}_k \cdot \mathfrak{m}_k) \bmod q. \tag{11}$$

In [LLM+16a], the authors also presents two companion protocols for signing a committed value and proving possession of a signature in the Camenisch and Lysyanskaya fashion [CL02]. These protocols are described below.

Companion Protocols for Signing a Committed Value and Proving Possession of a Signature. We now show a two-party protocol whereby a user can interact with the signer in order to obtain a signature on a committed message.

In order to prove that the scheme still guarantees unforgeability for obliviously signed messages, we will assume that each message block $\mathfrak{m}_k \in \{0,1\}^{2m}$ is obtained by encoding the actual message $M_k = M_k[1] \ldots M_k[m] \in \{0,1\}^m$ as $\mathfrak{m}_k =$

$\mathsf{Encode}(M_k) = (\bar{M}_k[1], M_k[1], \ldots, \bar{M}_k[m], M_k[m])$. Namely, each 0 (respectively each 1) is encoded as a pair $(1,0)$ (resp. $(0,1)$). The correctness of this encoding can be efficiently proved using Stern-like [Ste96] protocols.

To make this construction usable in the definitional framework of Camenisch *et al.* [CKL+15], we assume common public parameters (i.e., a common reference string) and encrypt all witnesses of which knowledge is being proved under a public key included in the common reference string. The resulting ciphertexts thus serve as statistically binding commitments to the witnesses. To enable this, the common public parameters comprise public keys $\mathbf{G}_0 \in \mathbb{Z}_q^{n\times\ell}$, $\mathbf{G}_1 \in \mathbb{Z}_q^{n\times 2m}$ for multi-bit variants of the dual Regev cryptosystem [GPV08] and all parties are denied access to the underlying private keys. The flexibility of Stern-like protocols allows us to prove that the content of a perfectly hiding commitment $\mathbf{c}_\mathfrak{m}$ is consistent with encrypted values.

Global-Setup: Let $B = \sqrt{n}\omega(\log n)$ and let χ be a B-bounded distribution. Let $p = \sigma \cdot \omega(\sqrt{m})$ upper-bound entries of vectors sampled from the distribution $D_{\mathbb{Z}^{2m},\sigma}$. Generate two public keys for the dual Regev encryption scheme in its multi-bit variant. These keys consist of a public random matrix $\mathbf{B} \hookleftarrow (\mathbb{Z}_q^{n\times m})$ and random matrices $\mathbf{G}_0 = \mathbf{B}\cdot\mathbf{E}_0 \in \mathbb{Z}_q^{n\times\ell}$, $\mathbf{G}_1 = \mathbf{B}\cdot\mathbf{E}_1 \in \mathbb{Z}_q^{n\times 2m}$, where $\mathbf{E}_0 \in \mathbb{Z}^{m\times\ell}$ and $\mathbf{E}_1 \in \mathbb{Z}^{m\times 2m}$ are short Gaussian matrices with columns sampled from $D_{\mathbb{Z}^m,\sigma}$. These matrices will be used to encrypt integer vectors of dimension ℓ and $2m$, respectively. Finally, generate public parameters $CK \coloneqq \{\mathbf{D}_k\}_{k=0}^N$ consisting of uniformly random matrices $\mathbf{D}_k \hookleftarrow (\mathbb{Z}_q^{2n\times 2m})$ for a statistically hiding commitment to vectors in $(\{0,1\}^{2m})^N$. Return public parameters consisting of

$$\mathsf{par} \coloneqq \{\mathbf{B} \in \mathbb{Z}_q^{n\times m}, \mathbf{G}_0 \in \mathbb{Z}_q^{n\times\ell}, \mathbf{G}_1 \in \mathbb{Z}_q^{n\times 2m}, CK\}.$$

Issue $\leftrightarrow$ Obtain : The signer S, who holds a key pair $PK \coloneqq \{\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^\ell, \mathbf{D}, \mathbf{u}\}$, $SK \coloneqq \mathbf{T}_\mathbf{A}$, interacts with the user U who has a message $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N)$, in the following interactive protocol.

1. U samples $\mathbf{s}' \hookleftarrow D_{\mathbb{Z}^{2m},\sigma}$ and computes $\mathbf{c}_\mathfrak{m} = \mathbf{D}_0\cdot\mathbf{s}' + \sum_{k=1}^N \mathbf{D}_k\cdot\mathfrak{m}_k \in \mathbb{Z}_q^{2n}$ which is sent to S as a commitment to $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N)$. In addition, U encrypts $\{\mathfrak{m}_k\}_{k=1}^N$ and $\mathbf{s}'$ under the dual-Regev public key $(\mathbf{B}, \mathbf{G}_1)$ by computing for all $k \in \{1,\ldots,N\}$:

$$\begin{aligned}\mathbf{c}_k &= (\mathbf{c}_{k,1}, \mathbf{c}_{k,2}) \\ &= \left(\mathbf{B}^T\cdot\mathbf{s}_k + \mathbf{e}_{k,1},\ \mathbf{G}_1^T\cdot\mathbf{s}_k + \mathbf{e}_{k,2} + \mathfrak{m}_k\cdot\lfloor q/2\rfloor\right) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{2m}\end{aligned} \tag{12}$$

for randomly chosen $\mathbf{s}_k \hookleftarrow \chi^n$, $\mathbf{e}_{k,1} \hookleftarrow \chi^m$, $\mathbf{e}_{k,2} \hookleftarrow \chi^{2m}$, and

$$\begin{aligned}\mathbf{c}_{s'} &= (\mathbf{c}_{s',1}, \mathbf{c}_{s',2}) \\ &= \left(\mathbf{B}^T\cdot\mathbf{s}_0 + \mathbf{e}_{0,1},\ \mathbf{G}_1^T\cdot\mathbf{s}_0 + \mathbf{e}_{0,2} + \mathbf{s}'\cdot\lfloor q/p\rfloor\right) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{2m}\end{aligned} \tag{13}$$

where $\mathbf{s}_0 \hookleftarrow \chi^n$, $\mathbf{e}_{0,1} \hookleftarrow \chi^m$, $\mathbf{e}_{0,2} \hookleftarrow \chi^{2m}$. The ciphertexts $\{\mathbf{c}_k\}_{k=1}^N$ and $\mathbf{c}_{s'}$ are sent to S along with $\mathbf{c}_\mathfrak{m}$.

Then, U generates an interactive zero-knowledge argument to convince S that $\mathbf{c}_\mathfrak{m}$ is a commitment to $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N)$ with the randomness $\mathbf{s}'$ such

that $\{\mathfrak{m}_k\}_{k=1}^N$ and $\mathbf{s}'$ were honestly encrypted to $\{\mathbf{c}_k\}_{i=1}^N$ and $\mathbf{c}_{s'}$, as in (12) and (13). The complete argument system is described in [LLM+16a], where it is demonstrated that, together with other zero-knowledge protocols, it can be derived from a Stern-like [Ste96] protocol.

2. If the argument of step 1 properly verifies, S samples $\mathbf{s}'' \hookleftarrow D_{\mathbb{Z}^{2m},\sigma_0}$ and computes a vector $\mathbf{u}_\mathfrak{m} = \mathbf{u} + \mathbf{D} \cdot \{0,1\}\big(\mathbf{c}_\mathfrak{m} + \mathbf{D}_0 \cdot \mathbf{s}''\big) \in \mathbb{Z}_q^n$. Next, S randomly picks $\tau \hookleftarrow \{0,1\}^\ell$ and uses $\mathbf{T}_\mathbf{A}$ to compute a delegated basis $\mathbf{T}_\tau \in \mathbb{Z}^{2m\times 2m}$ for the matrix $\mathbf{A}_\tau \in \mathbb{Z}_q^{n\times 2m}$ of (10). Using $\mathbf{T}_\tau \in \mathbb{Z}^{2m\times 2m}$, S samples a short vector $\mathbf{v} \in \mathbb{Z}^{2m}$ in $D^{\mathbf{u}_M}_{\Lambda^\perp(\mathbf{A}_\tau),\sigma}$. It returns the vector $(\tau, \mathbf{v}, \mathbf{s}'') \in \{0,1\}^\ell \times \mathbb{Z}^{2m} \times \mathbb{Z}^{2m}$ to U.
3. U computes $\mathbf{s} = \mathbf{s}' + \mathbf{s}''$ over $\mathbf{Z}$ and verifies that

$$\mathbf{A}_\tau \cdot \mathbf{v} = \mathbf{u} + \mathbf{D} \cdot \{0,1\}\big(\mathbf{D}_0 \cdot \mathbf{s} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k\big) \bmod q.$$

If so, it outputs $(\tau, \mathbf{v}, \mathbf{s})$. Otherwise, it outputs $\perp$.

Note that, if both parties faithfully run the protocol, the user obtains a valid signature $(\tau, \mathbf{v}, \mathbf{s})$ for which the distribution of $\mathbf{s}$ is $D_{\mathbf{Z}^{2m},\sigma_1}$, where $\sigma_1 = \sqrt{\sigma^2 + \sigma_0^2}$.

The following protocol allows proving possession of a message-signature pair.

Prove: On input of a signature $(\tau, \mathbf{v} = (\mathbf{v}_1^T \mid \mathbf{v}_2^T)^T, \mathbf{s}) \in \{0,1\}^\ell \times \mathbf{Z}^{2m} \times \mathbf{Z}^{2m}$ on the message $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N)$, the user does the following.

1. Using $(\mathbf{B}, \mathbf{G}_0)$ and $(\mathbf{B}, \mathbf{G}_1)$ generate perfectly binding commitments to $\tau \in \{0,1\}^\ell$, $\{\mathfrak{m}_k\}_{k=1}^N$, $\mathbf{v}_1, \mathbf{v}_2 \in \mathbf{Z}^m$ and $\mathbf{s} \in \mathbf{Z}^{2m}$. Namely, compute

$$\begin{aligned}
\mathbf{c}_\tau &= (\mathbf{c}_{\tau,1}, \mathbf{c}_{\tau,2}) \\
&= \big(\mathbf{B}^T \cdot \mathbf{s}_\tau + \mathbf{e}_{\tau,1},\ \mathbf{G}_0^T \cdot \mathbf{s}_\tau + \mathbf{e}_{\tau,2} + \tau \cdot \lfloor q/2 \rfloor\big) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^\ell, \\
\mathbf{c}_k &= (\mathbf{c}_{k,1}, \mathbf{c}_{k,2}) \\
&= \big(\mathbf{B}^T \cdot \mathbf{s}_k + \mathbf{e}_{k,1},\ \mathbf{G}_1^T \cdot \mathbf{s}_k + \mathbf{e}_{k,2} + \mathfrak{m}_k \cdot \lfloor q/2 \rfloor\big) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{2m} \\
&\qquad\qquad \forall k \in \{1, \ldots, N\}
\end{aligned}$$

where $\mathbf{s}_\tau, \mathbf{s}_k \hookleftarrow \chi^n$, $\mathbf{e}_{\tau,1}, \mathbf{e}_{k,1} \hookleftarrow \chi^m$, $\mathbf{e}_{\tau,2} \hookleftarrow \chi^\ell$, $\mathbf{e}_{k,2} \hookleftarrow \chi^{2m}$, as well as

$$\begin{aligned}
\mathbf{c}_\mathbf{v} &= (\mathbf{c}_{\mathbf{v},1}, \mathbf{c}_{\mathbf{v},2}) \\
&= \big(\mathbf{B}^T \cdot \mathbf{s}_\mathbf{v} + \mathbf{e}_{\mathbf{v},1},\ \mathbf{G}_1^T \cdot \mathbf{s}_\mathbf{v} + \mathbf{e}_{\mathbf{v},2} + \mathbf{v} \cdot \lfloor q/p \rfloor\big) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{2m} \\
\mathbf{c}_s &= (\mathbf{c}_{s,1}, \mathbf{c}_{s,2}) \\
&= \big(\mathbf{B}^T \cdot \mathbf{s}_0 + \mathbf{e}_{0,1},\ \mathbf{G}_1^T \cdot \mathbf{s}_0 + \mathbf{e}_{0,2} + \mathbf{s} \cdot \lfloor q/p \rfloor\big) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{2m},
\end{aligned}$$

where $\mathbf{s}_\mathbf{v}, \mathbf{s}_0 \hookleftarrow \chi^n$, $\mathbf{e}_{\mathbf{v},1}, \mathbf{e}_{0,1} \hookleftarrow \chi^m$, $\mathbf{e}_{\mathbf{v},2}, \mathbf{e}_{0,2} \hookleftarrow \chi^{2m}$.

2. Prove in zero-knowledge that $\mathbf{c}_\tau$, $\mathbf{c}_s$, $\mathbf{c}_\mathbf{v}$, $\{\mathbf{c}_k\}_{k=1}^N$ encrypt a valid message-signature pair. In [LLM+16a], the authors show that this involved zero-knowledge protocol can be derived from the statistical zero-knowledge argument of knowledge for a simpler, but more general relation. The proof system can be made statistically ZK for a malicious verifier using standard techniques (assuming a common reference string, we can use [Dam00]). In the random oracle model, it can be made non-interactive using the Fiat-Shamir heuristic [FS87].

The above construction is proven secure under the $\mathsf{SIS}_{n,2m,q,\hat{\beta}}$ assumption, where $\hat{\beta} = N\sigma(2m)^{3/2} + 4\sigma_1 m^{3/2}$, and the protocols are secure for obtaining a signature on a committed message and proving possession of a valid message-signature pair.

Open problems. However, it remains some work to do to obtain the same flexibility as pairing-based constructions for signature schemes with efficient protocols. For example, in the e-cash setting, or to design some group signature schemes, recent constructions [PST17, CS18] are based on the randomization property of some pairing based signature schemes [CL04, PS18b]. In the lattice-based setting, some components of existing signatures have (Gaussian) distributions on integers, which makes it less trivial to randomize than working modulo q. It is then currently necessary to perform some flooding, which implies a larger modulus and affects the efficiency. It thus remains an open problem to find such kind of efficient construction in lattices.

6.3 Leveled Homomorphic Signatures

In ordinary signature schemes, a message-signature pair (M, sig) becomes invalid if the message M is modified by a single bit. In some applications [Des93, JMSW02], it may be useful to tolerate specific *public* modifications on signed data. In network coding, for example, it is useful to have linearly homomorphic signatures [BFKW09] that enable linear transformations over authenticated data. Other applications (see [ABC$^+$12, ALP12, ALP13] and references therein) may require different kinds of transformations, such as extracting a substring of a signed message together with a valid signature on it.

In the context of lattice-based cryptography, the first homomorphic signature schemes were put forth by Boneh and Freeman [BF11b, BF11a]. Their first solution [BF11b] was a linearly homomorphic scheme allowing signing messages in binary fields (while earlier constructions were limited to sign vectors over large prime fields). In [BF11a], they used ideal lattices to construct a scheme allowing evaluating small-degree polynomials over signed data.

The construction of Boneh and Freeman [BF11a] was limited to the evaluation of small-depth circuits (in fact, constant-degree polynomials). Gorbunov, Vaikuntanathan and Wichs [GVW15b] showed how to remove the latter restriction. Using standard lattices, they described a leveled fully homomorphic signature scheme, which makes it possible to evaluate circuits of any (a priori bounded) polynomial depth using a given evaluation key. Namely, given original signatures on some data set M, a cloud server can publicly evaluate a circuit $C(M)$ on the dataset and derive a signature that authenticates the evaluation result $C(M)$ (importantly, the verifier is able to verify the derived signature on $C(M)$ *without* knowing the original dataset M). In particular, their scheme allows a cloud server to authenticate the evaluation of any polynomial-depth circuit in order to convince a client that a given circuit was correctly evaluated over the client's data. The construction of [GVW15b] is actually based on the ideas of the Gentry-Sahai-Waters homomorphic encryption scheme [GSW13], which is recalled in Section 7.4.2.

Open problems. In this context, a major open problem left is to turn the scheme of [GVW15b] into a truly fully homomorphic signature, where a given evaluation key does not *a priori* restrict the depth of circuits that can be evaluated over signed data. Another open problem is to base the security of (leveled) homomorphic signatures

on lattice assumptions with smaller approximation factors without decreasing the homomorphic capabilities of [GVW15b].

6.4 Blind Signatures

A blind signature scheme [Cha82b] permits a user to obtain from a signer a signature on a message of his choice. The main property is that the signer cannot later recognize the signature he has provided nor the message he has signed. This way, the user publishing the message signature pair is anonymous among the set of users having obtained a signature from this signer, this protecting this way his privacy.

More formally, a blind signature scheme consists of three algorithms (Keygen, Sign, Verify) but in which Sign is now an interactive protocol between a signer $\mathcal{S}$ and a user $\mathcal{U}$. More precisely,

- Keygen outputs a private signing key sk and a public verification key pk;
- Sign describes a joint execution between $\mathcal{S}$ (with signing key sk) and $\mathcal{U}$ (qith private message m) and where the user finally outputs a signature σ on the message m;
- Verify is the verification algorithm outputting 1 if σ is a valid signature on m under pk and 0 otherwise.

Such signature scheme should provide *blindness* (the authority cannot make the link between a (message,signature) pair and its transcript of the signing protocol) and *one-more unforgeability* (a user cannot output more valid (message,signature) pairs than the number of times he has interacted with the authority).

A variant of blind signatures, called *partially* blind signatures, has also been introduced. In this variant, the two parties agree on a common and public information added to the message during the blind signature process.

Relying on standard assumptions, there exists a lot of blind signature schemes, e.g., taking as a basis RSA-based blind signatures or Schnorr-based blind signatures. In [AO00], Abe and Okamoto have moreover proposed a generic transformation from a basic blind signature scheme to a partially blind one. But the result necessitates to increase the number of elements exchanged between the user and the authority to include a common information. In the lattice-based cryptography setting, the literature is much less rich. To the best of our knowledge, there exists one single proposal due to Rückert [Rüc10], later improved in [ZJZ+18]. Finally, a partially blind variant has been proposed in [TZW16].

Open problems. All existing lattice-based blind signature schemes include some trigger restarts in their protocol, making the result quite unpractical. Moreover the partially blind construction given in [TZW16] is adapted from [AO00], with the same disadvantages.

7 Public-Key Encryption Schemes Usable in Privacy-Preserving Protocols

The well-known purpose of an encryption scheme is to protect the confidentiality of a message to be transmitted to the legitimate reader. Formally speaking, it is composed

of three PPT algorithms, namely, Keygen to generate the key pair, Enc to encrypt a message under the public key of the legitimate receiver, and Dec which, on input the corresponding private key outputs the initial message.

Several lattice-based encryption signature schemes have been proposed and we will not review all of them in this document. Among them, nine schemes have passed the first round of the NIST call for proposal on Key Encapsulation Message, either based on the standard LWE problem (FrodoKEM), the module LWE problem (CRYSTALS-KYBER), the module LWR problem (SABER), the ring LWE one (Round5, LAC, NewHope), the NTRU lattices (NTRU, NTRU Prime) or some other problems (Three Bears).

In the sequel, we focus on some scheme that are suitable in the privacy context since they are suitable with zero-knowledge proofs. We then give some words about advanced encryption tools such as identity-based encryption or homomorphic encryption.

7.1 Public-Key Encryption Schemes from the LWE Assumption

In this section, we first recall Regev's public-key encryption scheme [Reg05] and its dual variant suggested by Gentry, Peikert and Vaikuntanathan [GPV08]. Regev's cryptosystem and its dual variant can be combined with Stern-like zero-knowledge proofs [Ste96] to construct privacy-preserving protocols. For example, they were used in [LLNW16, LLM$^+$16a] to build lattice-based group signatures in standard lattices.

We start by recalling the multi-bit variant of Regev suggested by Peikert, Vaikuntanathan and Waters [PVW08].

Keygen$(1^\lambda, 1^L)$**:** On input of a security parameter λ and the desired message length L, the key generation algorithm samples a uniformly random matrix $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n\times m})$ with $n \in \mathsf{poly}(\lambda)$ and $m \geq 2n\lceil \log q \rceil$. It chooses a uniformly random matrix $\mathbf{S} \hookleftarrow U(\mathbb{Z}_q^{n\times L})$ and computes $\mathbf{P} = \mathbf{A}^\top \cdot \mathbf{S} + \mathbf{E}$, where $\mathbf{E} \hookleftarrow \chi^{m\times L}$ is sampled from a noise distribution χ. It defines the public key

$$\mathsf{pk} := \left(\mathbf{A} \in \mathbb{Z}_q^{n\times m},\ \mathbf{P} \in \mathbb{Z}_q^{m\times L}\right)$$

and the corresponding secret key $\mathsf{sk} := \mathbf{S} \in \mathbb{Z}_q^{n\times L}$.

Encrypt(pk, μ)**:** In order to encrypt an L-bit message $\mu \in \{0,1\}^L$, the encryption algorithm samples a uniform vector $\mathbf{r} \hookleftarrow U(\{0,1\}^m)$ and computes

$$\begin{aligned} \mathbf{c}_0 &= \mathbf{A}\cdot\mathbf{r} \in \mathbb{Z}_q^n, \\ \mathbf{c}_1 &= \mathbf{P}^\top\cdot\mathbf{r} + \mu\cdot\lfloor q/2\rfloor \in \mathbb{Z}_q^L. \end{aligned}$$

Then, it outputs the ciphertext $\mathbf{c} = (\mathbf{c}_0, \mathbf{c}_1) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^L$.

Decrypt$(\mathsf{sk}, \mathbf{c})$**:** On input of $\mathbf{c} = (\mathbf{c}_0, \mathbf{c}_1) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^L$ and $\mathsf{sk} := \mathbf{S} \in \mathbb{Z}_q^{n\times L}$, do the following:

1. Compute $\mathbf{w} = (\mathbf{w}[1], \ldots, \mathbf{w}[L])^\top = \mathbf{c}_1 - \mathbf{S}^\top\mathbf{c}_0 \in \mathbb{Z}_q^L$.
2. For each $i \in [L]$, if $|\mathbf{w}[i]|$ is closer to 0 than to $\lfloor \frac{q}{2} \rfloor$, define $\mu_i = 0$. Otherwise, set $\mu_i = 1$.

Finally, output $\mu = \mu_1 \ldots \mu_L \in \{0,1\}^L$.

In the special case where $L = 1$ (i.e., for one-bit messages), the above scheme is identical to Regev's cryptosystem [Reg05]. We note that, in both variants, the randomness $\mathbf{r}$ of the encryption algorithm is sampled uniformly in $\{0,1\}^m$. In a variant suggested in [GPV08, Section 8.1], the vector $\mathbf{r}$ is sampled from a discrete Gaussian distribution $D_{\mathbb{Z}^m,\sigma}$ with a suitable standard deviation σ.

We now recall the dual Regev cryptosystem due to Gentry, Peikert and Vaikuntanathan [GPV08, Section 7.1]. As for the primal Regev case, we describe the "packed" variant that allows encrypting L bits at once. The scheme is parameterized by a Gaussian parameter $\sigma > 0$ and an error distribution χ.

Keygen$(1^\lambda, 1^L)$**:** On input of a security parameter λ and the desired message length L, the key generation algorithm samples a uniformly random matrix $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n\times m})$ with $n \in \mathsf{poly}(\lambda)$ and $m \geq 2n\lceil \log q \rceil$. It samples a small-norm matrix $\mathbf{V} \hookleftarrow (D_{\mathbb{Z}^m,\sigma})^L$ whose columns are independently sampled from the discrete Gaussian distribution $D_{\mathbb{Z}^m,\sigma}$. Then, it computes $\mathbf{U} = \mathbf{A} \cdot \mathbf{V} \in \mathbb{Z}_q^{n\times L}$. It defines the public key

$$\mathsf{pk} := \left(\mathbf{A} \in \mathbb{Z}_q^{n\times m},\ \mathbf{U} \in \mathbb{Z}_q^{n\times L}\right)$$

and the corresponding secret key $\mathsf{sk} := \mathbf{V} \in \mathbb{Z}^{m\times L}$.

Encrypt(pk, μ)**:** In order to encrypt an L-bit message $\mu \in \{0,1\}^L$, the encryption algorithm samples a uniform vector $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$ and noise terms $\mathbf{e}_0 \hookleftarrow \chi^m$, $\mathbf{e}_1 \hookleftarrow \chi^L$. Then, it computes

$$\begin{aligned} \mathbf{c}_0 &= \mathbf{A}^\top \mathbf{s} + \mathbf{e}_0 \in \mathbb{Z}_q^m, \\ \mathbf{c}_1 &= \mathbf{U}^\top \mathbf{s} + \mu \cdot \lfloor q/2 \rfloor + \mathbf{e}_1 \in \mathbb{Z}_q^L. \end{aligned}$$

Then, it outputs the ciphertext $\mathbf{c} = (\mathbf{c}_0, \mathbf{c}_1) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^L$.

Decrypt$(\mathsf{sk}, \mathbf{c})$**:** On input of $\mathbf{c} = (\mathbf{c}_0, \mathbf{c}_1) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^L$ and $\mathsf{sk} := \mathbf{V} \in \mathbb{Z}^{m\times L}$, do the following:

1. Compute $\mathbf{w} = (\mathbf{w}[1], \ldots, \mathbf{w}[L])^\top = \mathbf{c}_1 - \mathbf{V}^\top \mathbf{c}_0 \in \mathbb{Z}_q^L$.
2. For each $i \in [L]$, if $|\mathbf{w}[i]|$ is closer to 0 than to $\lfloor \frac{q}{2} \rfloor$, define $\mu_i = 0$. Otherwise, set $\mu_i = 1$.

Output $\mu = \mu_1 \ldots \mu_L \in \{0,1\}^L$.

7.2 Identity-Based Encryption

Unlike the primal Regev scheme, the dual Regev system of [GPV08] features a "dense" public key space. Namely, the distribution of its public keys $\mathbf{U}$ is statistically close to the uniform distribution over $\mathbb{Z}_q^{n\times L}$. This property was used in [GPV08, Section 7.2] to build an identity-based encryption (IBE) scheme under the LWE assumption (in the random oracle model). In order to construct an IBE system, the idea is to involve a trusted authority that generates a statistically uniform matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ together with a trapdoor $\mathbf{T_A}$ for the lattice $\Lambda^\perp(\mathbf{A})$. The authority can derive a secret key $\mathsf{sk}_{\mathsf{id}}$ for any identity id by computing $\mathbf{U}_{\mathsf{id}} = H(\mathsf{id}) \in \mathbb{Z}_q^{n\times L}$ using a random oracle $H : \{0,1\}^* \to \mathbb{Z}_q^{n\times L}$ and using the trapdoor $\mathbf{T_A}$ to sample a small-norm

matrix $\mathsf{sk}_{\mathsf{id}} = \mathbf{V}_{\mathsf{id}} \in \mathbb{Z}^{m \times L}$ with Gaussian entries such that $\mathbf{A} \cdot \mathbf{V}_{\mathsf{id}} = \mathbf{U}_{\mathsf{id}} \bmod q$. Ciphertexts can then be encrypted under the identity id by computing

$$\begin{aligned} \mathbf{c}_0 &= \mathbf{A}^\top \cdot \mathbf{s} + \mathbf{e}_0 \in \mathbb{Z}_q^m, \\ \mathbf{c}_1 &= \mathbf{U}_{\mathsf{id}}^\top \cdot \mathbf{s} + \mu \cdot \lfloor q/2 \rfloor + \mathbf{e}_1 \in \mathbb{Z}_q^L. \end{aligned}$$

in such a way that the receiver can decrypt by computing $\mathbf{w} = \mathbf{c}_1 - \mathbf{V}_{\mathsf{id}}^\top \cdot \mathbf{c}_0$.

Interestingly, the resulting IBE scheme is simultaneously semantically secure and *anonymous* in that the ciphertext $(\mathbf{c}_0, \mathbf{c}_1)$ computationally hides the message $\mu \in \{0,1\}^L$ *and* the receiver's identity id. In particular, it is computationally infeasible to distinguish a ciphertext encrypted under the identity id from a random element of the ciphertext space $\mathbb{Z}_q^m \times \mathbb{Z}_q^L$. This property is useful in the design of *searchable public key encryption* (a.k.a. "public-key encryption with keyword search", or PEKS) [BDCOP04]. In short, PEKS is a public-key encryption primitive where a keyword-specific trapdoor td_W makes it possible to efficiently recognize any encryption of a particular keyword W without learning anything else. It was actually shown [BDCOP04, ABC+05] that any anonymous IBE scheme can be used to generically construct a PEKS system. The scheme of [GPV08] thus implies a PEKS scheme under the LWE assumption in the random oracle model.

Later on, the dual Regev cryptosystem [GPV08] was used to construct anonymous IBE schemes in the standard model (see, e.g., [CHKP10, ABB10, Yam17] and references therein), which also imply PEKS constructions that provably rely on the LWE assumption in the standard model.

7.3 Attribute-Based Encryption

The dual Regev system also enabled the realization of attribute-based encryption (ABE) schemes for circuits [BGG+14]. Attribute-based encryption [SW05, GPSW06] is a powerful generalization of identity-based encryption where ciphertexts are labeled with an attribute set S and secret keys sk_P correspond to Boolean predicates P: the ABE functionality allows the decryptor to obtain the plaintext μ as long as his secret key sk_P corresponds to a predicate P such that $P(S) = 1$. Until 2013, all known ABE schemes were limited to access policies consisting of special cases of NC1 circuits. This situation changed when Gorbunov, Vaikuntanathan and Wee [GVW13] used the LWE assumption (with subexponential approximation factors) to construct an ABE system where policies may be arbitrary circuits. In 2014, Boneh *et al.* [BGG+14] described an improved ABE scheme for all circuits, where the secret key size $|\mathsf{sk}_P|$ only depends on the depth (rather than its size) of the circuit P. On the other hand, their scheme still relies on a strong LWE assumption with subexponential approximation factor.

Open problems. It still remains an open problem to construct an ABE system for all circuits under an LWE assumption with polynomial approximation factors. So far, the only known lattice-based ABE schemes that rely on such a mild assumption [GV15] are restricted to predicates P that can be described as branching programs of (a priori bounded) polynomial length.

7.4 Homomorphic Encryption

A fully homomorphic encryption (FHE) scheme is an encryption scheme that enables the evaluation of arbitrarily complex functions on encrypted data. In *compact* FHE

schemes, the ciphertexts do not grow in size with each homomorphic operation. This section summarizes the main achievements in the area since Gentry's result [Gen09].

7.4.1 Brief Overview

The problem of designing compact FHE schemes was first suggested by Rivest, Adleman and Dertouzos [RAD78] in 1978. Still, the first plausible candidate was only given in 2009 by Gentry [Gen09]. His scheme involved new and relatively untested cryptographic assumptions in ideal lattices. Improved solutions with a better efficiency were suggested by Smart and Vercauteren [SV10] and by Stehlé and Steinfeld [SS10].

The main building block in Gentry's construction was a so-called "somewhat" homomorphic encryption scheme which enables the homomorphic evaluation of any function whose polynomial representation has bounded degree. Ciphertexts actually contain a noise that grows importantly during homomorphic multiplications, thus restricting the scheme to the evaluation of low-degree polynomials. Gentry showed that, as long as a leveled FHE scheme[2] can homomorphically evaluate its own decryption circuit, it can be bootstrapped (by publicizing encryptions of the secret key bits) into a fully homomorphic system where arbitrary circuits may be evaluated using a fixed-size evaluation key. In order to obtain a leveled scheme of which the decryption circuit fits within its homomorphic capabilities, Gentry used "squashing step" which decreases the depth of the decryption circuit at the cost of making an additional very strong hardness assumption: namely, the hardness of the (average-case) sparse subset-sum problem. Brakerski and Vaikuntanathan [BV11b] subsequently gave a much simpler somewhat homomorphic construction under the ring-LWE assumption.

In independent works, Gentry and Halevi [GH11] and Brakerski and Vaikuntanathan [BV11a] described different techniques to avoid the squashing step and the sparse subset sum assumption. Brakerski and Vaikuntanathan [BV11a] managed to base the security of their leveled FHE scheme entirely on the hardness of the LWE problem (for sub-exponential approximation factors) in standard (i.e., non-ideal) lattices. Starting with the results of Brakerski, Gentry and Vaikuntanathan [BGV12], several works using different approaches [Bra12, GSW13] have reduced the required factor of approximation to quasi-polynomial approximation factors.

Many FHE schemes make use of a relatively involved multiplication procedure. In the LWE-based schemes of [BV11a, BGV12], the ciphertext $\mathbf{c}$ and secret key $\mathbf{s}$ are n-dimensional vectors whose inner product $\langle \mathbf{c}, \mathbf{s} \rangle$ equals the plaintext μ up to some small error term that is removed by rounding. Multiplication proceeds by tensoring ciphertexts $\mathbf{c}_1 \otimes \mathbf{c}_2$ in such a way that a tensor product $\mathbf{s} \otimes \mathbf{s}$ of the secret key with itself can be used to decrypt $\mathbf{c}_1 \otimes \mathbf{c}_2$ to $\mu_1 \cdot \mu_2$. Since tensoring blows up the ciphertexts size from $O(n)$ to $O(n^2)$ elements, the evaluator must relinearize [BV11a] the ciphertext via a procedure that takes the long ciphertext that encrypts $\mu_1 \cdot \mu_2$ under the long key $\mathbf{s} \otimes \mathbf{s}$ and compresses it into a normal-sized n-dimensional ciphertext encrypting $\mu_1 \cdot \mu_2$ under some n-dimensional key $\mathbf{t}$. To relinearize, the evaluator multiplies the long ciphertext vector by a special relinearization matrix, which is part of the homomorphic evaluation key. While ingenious, the relinearization step is somewhat expensive as it requires $\Omega(n^3)$ operations, each of which has complexity polynomial in L. Also, each relinearization matrix has size $\Omega(n^3)$ and the public key must contain L of them to evaluate circuits of multiplicative depth L. Gentry, Sahai and Waters [GSW13] (GSW) described a scheme with a much simpler multiplication procedure which eliminates the

[2] A *leveled* FHE scheme is one that can evaluate Boolean circuits of bounded depth with an evaluation key of linear size in the maximal depth.

need for a relinearization step and does not require any dimension/modulus-switching. In their scheme, each ciphertext is an $n \times m$ matrix and their multiplication algorithm computes a product of two matrices. The Gentry-Sahai-Waters scheme turns out to be quite powerful as it was used as a crucial building block in a number of advanced cryptographic constructions (see, e.g., [BGG+14, BV15, GVW15a, GVW15b, Yam16, MW16, BV16, BP16, KW17, Yam17, PS18a, BGG+18]). Since it is also the simplest known FHE system to date, we recall its description in the next subsection.

7.4.2 The GSW FHE

This section recalls the leveled homomorphic encryption scheme of Gentry, Sahai and Waters [GSW13] in its simplified variant described in [ASP14]. The description makes use of the "gadget matrix" introduced by Micciancio and Peikert [MP12].

Micciancio and Peikert [MP12] proved that, for any $m \geq n\lceil \log q \rceil$, there exists an efficiently computable matrix $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$ and an efficiently computable deterministic "*short preimage*" function $\mathbf{G}^{-1}(\cdot)$ with the following property: on input of a matrix $\mathbf{M} \in \mathbb{Z}_q^{n \times m'}$, for any integer $m' > 0$, the function $\mathbf{G}^{-1}(\mathbf{M})$ outputs a binary matrix $\mathbf{G}^{-1}(\mathbf{M}) \in \{0,1\}^{m \times m'}$ such that $\mathbf{G}\mathbf{G}^{-1}(\mathbf{M}) = \mathbf{M}$. Note that $\mathbf{G}^{-1}(\cdot)$ is not a matrix itself but rather a function.

We can think of $\mathbf{G}$ as a special matrix with a "*public trapdoor*" that allows sampling short integer vectors $\mathbf{v} \in \mathbb{Z}^m$ such that $\mathbf{G} \cdot \mathbf{v} = \mathbf{0}^n$. For example, by defining

$$\mathbf{G} = \left[\mathbf{I}_n \otimes (1, 2, 4, \ldots, 2^{\lceil \log q \rceil})^\top \mid \mathbf{0}^{n \times \lceil \log q \rceil}\right] \in \mathbb{Z}_q^{n \times m},$$

where $m = 2n\lceil \log q \rceil$, we can define $\mathbf{G}^{-1}(\cdot)$ to be the entry-wise binary decomposition function whose outputs are padded with zeroes until they reach the desired dimension.

Keygen$(1^\lambda, 1^d)$**:** On input of a security parameter λ and a maximal circuit depth, the key generation algorithm samples the following elements:

- A uniformly random matrix $\bar{\mathbf{A}} \hookleftarrow U(\mathbb{Z}_q^{(n-1) \times m})$ with $n \in \mathsf{poly}(\lambda)$ and $m = 2n\lceil \log q \rceil$;
- A uniformly random vector $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^{n-1})$;
- A small-norm vector $\mathbf{e} \hookleftarrow \chi^m$, which is sampled from the error distribution χ.

It defines the public key

$$\mathsf{pk} := \mathbf{A} = \begin{pmatrix} \bar{\mathbf{A}} \\ \mathbf{s}^\top \cdot \bar{\mathbf{A}} + \mathbf{e}^\top \end{pmatrix} \in \mathbb{Z}_q^{n \times m}$$

and the corresponding secret key $\mathsf{sk} := \mathbf{t} = \begin{pmatrix} -\mathbf{s} \\ 1 \end{pmatrix} \in \mathbb{Z}_q^n$.

Encrypt(pk, μ)**:** In order to encrypt a bit $\mu \in \{0,1\}$, the encryption algorithm samples a uniform matrix $\mathbf{R} \hookleftarrow U(\{0,1\}^{m \times m})$ and computes

$$\mathbf{C} = \mathbf{A} \cdot \mathbf{R} + \mu \cdot \mathbf{G} \quad \in \mathbb{Z}_q^{n \times m},$$

where $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$ is the gadget matrix of [MP12].

Decrypt$(\mathsf{sk}, \mathbf{C})$**:** On input of a ciphertext $\mathbf{C} \in \mathbb{Z}_q^{n \times m}$ and $\mathsf{sk} := \mathbf{t} \in \mathbb{Z}_q^n$, do the following:

1. Define $\mathbf{w} = (0, \ldots, 0, \lfloor q/2 \rfloor)^\top \in \mathbb{Z}_q^n$ and compute
$$\mathbf{v} = \mathbf{C} \cdot \mathbf{G}^{-1}(\mathbf{w}) \in \mathbb{Z}_q^n.$$
2. If $|\mathbf{t}^\top \mathbf{v}|$ is close to 0, output $\mu = 0$. If $|\mathbf{t}^\top \mathbf{v}|$ is close to $\lfloor q/2 \rfloor$, output $\mu = 1$.

Eval$(\mathsf{pk}, \mathbf{C}_1, \mathbf{C}_2, op)$**:** Given two ciphertexts $\mathbf{C}_1, \mathbf{C}_2 \in \mathbb{Z}_q^{n \times m}$ and an operation $op \in \{+, \times\}$ to be evaluated over $\mathbf{C}_1$ and $\mathbf{C}_2$, do the following:

- If $op =$ " $+$ ", output $\mathbf{C}^+ := \mathbf{C}_1 + \mathbf{C}_2 \in \mathbb{Z}_q^{n \times m}$.
- If $op =$ " $\times$ ", output $\mathbf{C}^\times := \mathbf{C}_1 \cdot \mathbf{G}^{-1}(\mathbf{C}_2)$.

If two fresh ciphertexts $\mathbf{C}_1 = \mathbf{A} \cdot \mathbf{R}_1 + \mu_1 \cdot \mathbf{G}$ and $\mathbf{C}_2 = \mathbf{A} \cdot \mathbf{R}_2 + \mu_2 \cdot \mathbf{G}$ encrypt $\mu_1 \in \{0, 1\}$ and $\mu_2 \in \{0, 1\}$, respectively, then

$$\begin{aligned} \mathbf{C}^\times = \mathbf{C}_1 \cdot \mathbf{G}^{-1}(\mathbf{C}_2) &= \mathbf{A} \cdot \big(\mathbf{R}_1 \cdot \mathbf{G}^{-1}(\mathbf{C}_2) + \mu_1 \cdot \mathbf{R}_2\big) + \mu_1 \cdot \mu_2 \cdot \mathbf{G} \\ &= \mathbf{A} \cdot \mathbf{R}^\times + \mu_1 \cdot \mu_2 \cdot \mathbf{G} \end{aligned}$$

is indeed an encryption of $\mu_1 \cdot \mu_2 \in \{0, 1\}$ with the small-norm matrix

$$\mathbf{R}^\times = \mathbf{R}_1 \cdot \mathbf{G}^{-1}(\mathbf{C}_2) + \mu_1 \cdot \mathbf{R}_2. \tag{14}$$

Unlike $\mathbf{C}^\times$, the sum of two ciphertexts $\mathbf{C}^+ = \mathbf{A} \cdot (\mathbf{R}_1 + \mathbf{R}_2) + (\mu_1 + \mu_2) \cdot \mathbf{G}$ may not be an encryption of a bit since we may have $\mu_1 + \mu_2 = 2$. However, it is not a problem since we can still evaluate a circuit composed of NAND gates by computing

$$\mathbf{C}^{\mathrm{NAND}} = \mathbf{G} - \mathbf{C}_1 \cdot \mathbf{G}^{-1}(\mathbf{C}_2) = \mathbf{A}(-\mathbf{R}^\times) + (\mu_1 \text{ NAND } \mu_2) \cdot \mathbf{G},$$

which allows evaluating any Boolean circuit of *a priori* bounded depth and still end up with an encryption of a Boolean value. Using Gentry's bootstrapping technique [Gen09], the GSW scheme can be modified to enable the evaluation of any Boolean circuit with a given choice of public parameters.

As the matrix $\mathbf{R}^\times$ inevitably has larger entries than $\mathbf{R}_1$ and $\mathbf{R}_2$, the modulus q should be large enough to maintain the property that evaluated ciphertexts correctly decrypt with high probability. In its basic variant, the somewhat homomorphic scheme thus requires q to be exponentially large in the depth d of the considered circuit. Brakerski and Vaikuntanathan [BV14] suggested a technique to prevent the noise matrix $\mathbf{R}^\times$ from growing too large. They observed from (14) that the magnitude of $\mathbf{R}^\times$ can be minimized by evaluating "sequentialized" circuits, in such a way that $\mathbf{R}_1$ is always the noise matrix of a fresh GSW ciphertext. In particular, if the circuit to be evaluated is a branching program of polynomial length L, the noise matrices only increase by a factor $L \cdot \mathsf{poly}(n)$. By combining this observation with Barrington's theorem [Bar86] and other ideas, Brakerski and Vaikuntanathan [BV14] showed how to evaluate any NC1 circuit using a polynomial-size modulus $q = \mathsf{poly}(n)$. Since the decryption circuit of the scheme is itself in NC1, Gentry's bootstrapping theorem [Gen09] could be applied in [BV14] to construct an FHE scheme with $q = \mathsf{poly}(n)$ under lattice assumptions with polynomial approximation factors.

7.5 Threshold Cryptosystems

In threshold cryptosystems, secret keys are broken into N shares $\mathbf{s}_1, \ldots, \mathbf{s}_N$, each of which is given to a different server. Using its secret key share $\mathbf{s}_i$, the i-th server can locally compute a partial secret-key operation (e.g., a partial decryption or a partial signature). A dedicated server can then gather at least $t \leq N$ correct partial results in order to reconstruct the final result of the secret-key operation. In some applications, it is desirable to have non-interactive threshold protocols where servers should be able to generate their partial evaluations without interacting with one another.

The benefit of threshold cryptography [DF89] is that setting $t < N$ (a common scenario is $t = N/2 - 1$, when an honest majority is assumed) allows for fault-tolerant systems which can keep running when some server crashes. Second, the adversary is forced to break into t servers to compromise the security of the whole scheme.

A typical application of threshold cryptosystems is the design of e-voting protocols. They prevent individual shareholders from decrypting individual votes. At the same time, a quorum of at least t-out-of-N trustees should be able to jointly decrypt the final result of an election without affecting the privacy of individual votes.

Lattice-based threshold protocols were studied by Bendlin and Damgård back in 2010 [BD10]. They gave a threshold variant of Regev's CPA-secure encryption scheme and Myers *et al.* [MSs11] applied the technique to fully homomorphic encryption. N-out-of N threshold FHE systems were used in the context of multiparty computation protocols [AJLA+12, MW16] but they only considered the case $t = N$ rather than arbitrary thresholds. Xie *et al.* [XXZ11] put forth a chosen-ciphertext-secure threshold cryptosystem using lossy trapdoor functions. Their scheme can be instantiated under the LWE [PW08] but the size of ciphertexts is at least linear in the number of servers. Bendlin *et al.* described threshold Gaussian sampling protocols [BKP13] which can be used to realize threshold signatures and IBE schemes where the public key and the size of the signatures are all independent of the number of servers. A limitation of their schemes is that the servers can only carry out an *a priori* bounded number of online non-interactive private key operations before they must perform an interactive protocol.

Boneh *et al.* [BGGK17, BGG+18] showed how to generically compile cryptographic functionalities into threshold functionalities using distributed FHE schemes as a building block. In particular, they obtain non-interactive threshold signatures and chosen-ciphertext-secure [RS91] public-key encryption schemes with non-interactive threshold decryption protocols. As a consequence of relying on FHE, their constructions require strong LWE assumptions with subexponential approximation factors.

Several other works considered threshold systems from lattice assumptions. In the context of pseudorandom functions, Boneh *et al.* [BLMR13] obtained threshold distributed PRFs from key-homomorphic PRFs [BLMR13, BP14], where the key sizes are independent of the number of evaluators and the evaluation process is also non-interactive. A recent work by Boneh *et al.* [BGG+18] described a generic method allowing to thresholdize several cryptographic functionalities, including pseudorandom functions, digital signatures and chosen-ciphertext-secure public-key encryption. They also gave a t-out-of-N threshold decryption mechanism in the Gentry-Sahai-Waters FHE.

Open problems. To our knowledge, for arbitrary thresholds $1 < t < N$ (the most interesting case being $t \approx N/2$), it remains an open problem to come up with non-interactive threshold signatures, threshold cryptosystems or threshold PRFs under

standard lattice assumptions with polynomial approximation factors.

8 Conclusion

In the context of cryptographic building blocks for practical advanced protocols, a lot of work has already been done in the lattice setting, as shown in this document. But it remains several important open problems to solve. Within WP4 of the project, the purpose of PROMETHEUS is then to design and implement better lattice-based signatures, encryption, commitment schemes and zero-knowledge proof systems that can easily be combined altogether in higher-level protocols, and prove their security even against side-channel attacks.

References

[ABB10] S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT 2010*, 2010.

[ABB+19] Erdem Alkim, Paulo S. L. M. Barreto, Nina Bindel, Patrick Longa, and Jefferson E. Ricardini. The lattice-based digital signature scheme qtesla. 2019.

[ABC+05] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In *Crypto*, 2005.

[ABC+12] J.-H. Ahn, D. Boneh, J. Camenisch, S. Hohenberger, a. shelat, and B. Waters. Computing on authenticated data. In *TCC*, 2012.

[AFG+10] Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Structure-preserving signatures and commitments to group elements. In *CRYPTO 2010*, volume 6223 of *LNCS*, pages 209–236. Springer, 2010.

[AJLA+12] G. Asharov, A. Jain, A. Lopez-Alt, E. Tromer, V. Vaikuntanathan, and D. Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. In *Eurocrypt*, 2012.

[AKPW13] J. Alwen, S. Krenn, K. Pietrzak, and D. Wichs. Learning with rounding, revisited - new reduction, properties and applications. In *Crypto*, 2013.

[ALP12] N. Attrapadung, B. Libert, and T. Peters. Computing on authenticated data: New privacy definitions and constructions. In *Asiacrypt*, 2012.

[ALP13] N. Attrapadung, B. Libert, and T. Peters. Efficient completely context-hiding quotable and linearly homomorphic signatures. In *PKC*, 2013.

[AO00] Masayuki Abe and Tatsuaki Okamoto. Provably secure partially blind signatures. In *CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 271–286. Springer, 2000.

[AP09] J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. In *STACS 2009*, volume 3 of *LIPIcs*, pages 75–86. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2009.

[AS15] J. Alperin-Sheriff. Short signatures with short public keys from homomorphic trapdoor functions. In *PKC*, 2015.

[ASP14] J. Alperin-Sheriff and C. Peikert. Faster bootstrapping with polynomial error. In *Crypto*, 2014.

[Bar86] D. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in nc1. In *STOC*, 1986.

[BD10] R. Bendlin and I. Damgård. Threshold decryption and zero-knowledge proofs for lattice-based cryptosystems. In *TCC*, 2010.

[BDCOP04] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In *Eurocrypt*, 2004.

[BDL+18] C. Baum, I. Damgård, V. Lyubashevsky, S. Oechsner, and C. Peiket. More efficient commitments from structured lattice assumptions. In *SCN*, 2018.

[BF11a] D. Boneh and D. M. Freeman. Homomorphic signatures for polynomial functions. In *Eurocrypt*, 2011.

[BF11b] D. Boneh and D. M. Freeman. Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures. In *PKC*, 2011.

[BFKW09] D. Boneh, D. Freeman, J. Katz, and B. Waters. Signing a linear subspace: Signature schemes for network coding. In *PKC*, 2009.

[BFM88] M. Blum, M. Feldman, and S. Micali. Non-interactive zero-knowledge and its applications. In *STOC*, 1988.

[BFP+15] A. Banerjee, G. Fuchsbauer, C. Peikert, K. Pietrzak, and S. Stevens. Key-homomorphic constrained pseudorandom functions. In *TCC*, 2015.

[BGG+14] D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, and D. Vinayagamurthy. Fully key-homomorphic encryption, arithmetic ABE and compact garbled circuits. In *Eurocrypt*, 2014.

[BGG+18] D. Boneh, R. Gennaro, S. Goldfeder, A. Jain, S. Kim, P. Rasmussen, and A. Sahai. Threshold cryptosystems from threshold fully homomorphic encryption. In *Crypto*, 2018.

[BGGK17] D. Boneh, R. Gennaro, S. Goldfeder, and S. Kim. A lattice-based universal thresholdizer for cryptographic systems. Cryptology ePrint Archive: Report 2017/251, September 2017.

[BGI14] E. Boyle, S. Goldwasser, and I. Ivan. Functional signatures and pseudorandom functions. In *PKC*, 2014.

[BGV12] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *ITCS*, 2012.

[BHJ+15] F. Böhl, D. Hofheinz, T. Jager, J. Koch, and C. Striecks. Confined guessing: New signatures from standard assumptions. *Journal of Cryptology*, 28(1):176–208, 2015.

[BHP17] Z. Brakerski, S. Halevi, and A. Polychroniadou. Four round secure computation without setup. In *TCC*, 2017.

[BKLP15] F. Benhamouda, S. Krenn, V. Lyubashevsky, and K. Pietrzak. Efficient zero-knowledge proofs for commitments from learning with errors over rings. In *ESORICS 2015*, volume 9326 of *LNCS*, pages 305–325. Springer, 2015.

[BKM17] D. Boneh, S. Kim, and H. Montgomery. Private puncturable prfs from standard lattice assumptions. In *Eurocrypt*, 2017.

[BKP13] R. Bendlin, S. Krehbiel, and C. Peikert. How to share a lattice trapdoor: Threshold protocols for signatures and (H)IBE. In *ACNS*, 2013.

[BLMR13] D. Boneh, K. Lewi, H. Montgomery, and A. Raghunathan. Key-homomorphic PRFs and their applications. In *Crypto*, 2013.

[BLP+13] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. On the classical hardness of learning with errors. In *STOC*, 2013.

[Blu82] M. Blum. Coin flipping by telephone: A protocol for solving impossible problems. In *COMPCON – 24th IEEE Computer Society International Conference*, 1982.

[Boy10] X. Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In *PKC*, 2010.

[BP14] A. Banerjee and C. Peikert. New and improved key-homomorphic pseudorandom functions. In *Crypto*, 2014.

[BP16] Z. Brakerski and R. Perlman. Lattice-based fully dynamic multi-key fhe with short ciphertexts. In *Crypto*, 2016.

[BPR12] A. Banerjee, C. Peikert, and A. Rosen. Pseudorandom functions and lattices. In *Eurocrypt*, 2012.

[BR93] M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *ACM-CCS 1993*, pages 62–73. ACM, 1993.

[Bra12] Z. Brakerski. Fully homomorphic encryption without modulus switching from classical GapSVP. In *Crypto*, 2012.

[BTVW17] Z. Brakerski, R. Tsabary, V. Vaikuntanathan, and H. Wee. Private constrained PRFs (and more) from lattices. In *TCC*, 2017.

[BV11a] Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *FOCS*, 2011.

[BV11b] Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In *Crypto*, 2011.

[BV14] Z. Brakerski and V. Vaikuntanathan. Lattice-based FHE as secure as PKE. In *ITCS*, 2014.

[BV15] Z. Brakerski and V. Vaikuntanathan. Constrained key-homomorphic PRFs from standard lattice assumptions (or: How to secretly embed a circuit in your PRF). In *TCC*, 2015.

[BV16] Z. Brakerski and V. Vaikuntanathan. Circuit-ABE from LWE: Unbounded attributes and semi-adaptive security. In *Crypto*, 2016.

[BW13] D. Boneh and B. Waters. Constrained pseudorandom functions and their applications. In *Asiacrypt*, 2013.

[CC17] R. Canetti and Y. Chen. Constraint-hiding constrained PRFs for NC1 from LWE. In *Eurocrypt*, 2017.

[CDN09] J. Camenisch, M. Dubovitskaya, and G. Neven. Oblivious transfer with access control. In *ACM-CCS 2009*, pages 131–140, 2009.

[CFSY96] R. Cramer, M. Franklin, B. Schoenmakers, and M. Yung. Multi-authority secret-ballot elections with linear work. In *Eurocrypt*, 1996.

[CG07] S. Canard and A. Gouget. Divisible e-cash systems can be truly anonymous. In *Eurocrypt*, 2007.

[Cha82a] David Chaum. Blind Signatures for Untraceable Payments. In *Crypto*, LNCS, pages 199–203. Springer, 1982.

[Cha82b] David Chaum. Blind signatures for untraceable payments. In *CRYPTO 1982*, pages 199–203. Plenum Press, New York, 1982.

[Cha85] David Chaum. Security without Identification: Transactions System to Make Big Brother Obsolete. *Communications of the ACM*, 28(10):1030–1044, 1985.

[CHKP10] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *Eurocrypt*, 2010.

[CHL05] J. Camenisch, S. Hohenberger, and A. Lysyanskaya. Compact e-cash. In *Eurocrypt*, 2005.

[CKL+15] J. Camenisch, S. Krenn, A. Lehmann, G.-L. Mikkelsen, G. Neven, and M.-0. Pedersen. Formal treatment of privacy-enhancing credential systems. In *SAC 2015*, LNCS, pages 3–24. Springer, 2015.

[CL01] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT 2001*, number 2045 in LNCS, pages 93–118. Springer, 2001.

[CL02] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In *SCN 2002*, number 2576 in LNCS, pages 268–289. Springer, 2002.

[CL04] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *CRYPTO 2004*, number 3152 in LNCS, pages 56–72. Springer, 2004.

[CLW] R. Canetti, A. Lombardi, and D. Wichs. Non-interactive zero knowledge and correlation intractability from circular-secure fhe. Cryptology ePrint Archive: Report 2018/1248.

[Cra96] Ronald Cramer. *Modular Design of Secure, yet Practical Cryptographic Protocols*. PhD thesis, University of Amsterdam, 1996.

[CS18] Remi Clarisse and Olivier Sanders. Short group signature in the standard model. *IACR Cryptology ePrint Archive*, 2018:1115, 2018.

[CVH91] David Chaum and Eugène Van Heyst. Group signatures. In *EUROCRYPT 1991*, volume 547 of *LNCS*, pages 257–265. Springer, 1991.

[Dam00] I. Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. In *Eurocrypt*, 2000.

[Des93] Y. Desmedt. Computer security by redefining what a computer is. In *New Security Paradigms Workshop (NSPW)*, 1993.

[DF89] Y. Desmedt and Y. Frankel. Threshold cryptosystems. In *Crypto*, 1989.

[DKL+18] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-dilithium: A lattice-based digital signature scheme. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1):238–268, 2018.

[DM14] L. Ducas and D. Micciancio. Improved short lattice signatures in the standard model. In *Crypto*, 2014.

[DPLNS17] R. Del Pino, V. Lyubashevsky, G. Neven, and G. Seiler. Practical quantum-safe voting from lattices. In *ACM-CCS*, 2017.

[DPLS18] R. Del Pino, V. Lyubashevsky, and G. Seiler. Lattice-based group signatures and zero-knoweldge proofs of automorphism stability. In *ACM-CCS*, 2018.

[DS15] N. Döttling and D. Schröder. Efficient pseudorandom functions via on-the-fly adaptation. In *Crypto*, 2015.

[FHK+17] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. Falcon: Fast-fourier lattice-based compact signatures over ntru. In *Specification 1.0*, 2017.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Crypto*, 1987.

[FS90] Uriel Feige and Adi Shamir. Witness Indistinguishable and Witness Hiding Protocols. In ACM, editor, *STOC*, pages 416–426, 1990.

[Gen09] C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC 2009*, pages 169–178. ACM, 2009.

[GGM86] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. In *J. of ACM*, volume 33, 1986.

[GH11] C. Gentry and S. Halevi. Fully homomorphic encryption without squashing using depth-3 arithmetic circuits. In *FOCS*, 2011.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems. In *STOC 1985*, pages 291–304. ACM, 1985.

[GPSW06] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM-CCS*, 2006.

[GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, 2008.

[GSW13] C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Crypto*, 2013.

[GTKPV10] S. Goldwasser, Y. Tauman Kalai, C. Peikert, and V. Vaikuntanathan. Robustness of the learning with errors assumption. In *ICS*, page 0, 2010.

[GV15] S. Gorbunov and D. Vinayagamurthy. Riding on asymmetry: Efficient ABE for branching programs. In *Asiacrypt*, 2015.

[GVW13] S. Gorbunov, V. Vaikuntanathan, and H. Wee. Attribute-based encryption for circuits. In *STOC*, 2013.

[GVW15a] S. Gorbunov, V. Vaikuntanathan, and H. Wee. Predicate encryption for circuits from LWE. In *Crypto*, 2015.

[GVW15b] S. Gorbunov, V. Vaikuntanathan, and D. Wichs. Leveled fully homomorphic signatures from standard lattices. In *STOC*, 2015.

[HILL99] J. Hastad, R. Impagliazzo, L. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 8(4):1364–1396, 1999.

[JKP18] T. Jager, R. Kurek, and J. Pan. Simple and more efficient PRFs with tight security from LWE and matrix-DDH. In *Asiacrypt*, 2018.

[JMSW02] R. Johnson, D. Molnar, D. Song, and D. Wagner. Homomorphic signature schemes. In *TC-RSA*, 2002.

[KPTTZ13] A. Kiayias, S. Papadopoulos, N. Triandopoulos, and T. Thomas Zacharias. Delegatable pseudorandom functions and applications. In *ACM-CCS*, 2013.

[KTX08] A. Kawachi, K. Tanaka, and K. Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *Asiacrypt*, 2008.

[KTY07] Aggelos Kiayias, Yiannis Tsiounis, and Moti Yung. Group encryption. In *ASIACRYPT 2007*, number 4833 in LNCS, pages 181–199. Springer, 2007.

[KW17] S. Kim and D. Wu. Watermarking cryptographic functionalities from standard lattice assumptions. In *Crypto*, 2017.

[KW18] S. Kim and D. Wu. Multi-theorem preprocessing nizks from lattices. In *Crypto*, 2018.

[KY16] S. Katsumata and S. Yamada. Partitioning via non-linear polynomial functions: More compact ibes from ideal lattices and bilinear maps. In *Asiacrypt*, 2016.

[LLM+16a] B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang. Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In *Asiacrypt*, 2016.

[LLM+16b] B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang. Zero-knowledge arguments for matrix-vector relations and lattice-based group encryption. In *Asiacrypt*, 2016.

[LLNW16] B. Libert, S. Ling, K. Nguyen, and H. Wang. Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In *Eurocrypt*, 2016.

[LLNW17] B. Libert, S. Ling, K. Nguyen, and H. Wang. Zero-knowledge arguments for lattice-based PRFs and applications to e-cash. In *Asiacrypt*, 2017.

[LM08] V. Lyubashevsky and D. Micciancio. Asymptotically efficient lattice-based digital signatures. In *TCC*, 2008.

[LNSW13] San Ling, Khoa Nguyen, Damien Stehlé, and Huaxiong Wang. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In *PKC*, 2013.

[LNWX18] S. Ling, K. Nguyen, H. Wang, and Y. Xu. Constant-size group signatures from lattices. In *PKC*, 2018.

[LS15] A. Langlois and D. Stehlé. Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography*, 75(3), 2015.

[Lyu08] Vadim Lyubashevsky. Lattice-based identification schemes secure under active attacks. In *PKC 2008*, volume 4939 of *LNCS*, pages 162–179. Springer, 2008.

[Lyu09] Vadim Lyubashevsky. Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures. In *Asiacrypt*, pages 598–616. Springer, 2009.

[MP12] D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Eurocrypt*, 2012.

[MSs11] S. Myers, M. Sergi, and a. shelat. Threshold fully homomorphic encryption and secure computation. Cryptology ePrint Archive, Report 2011/454, 2011.

[MW16] P. Mukherjee and D. Wichs. Two round mutliparty computation via multi-key FHE. In *Eurocrypt*, 2016.

[NFHF09] Toru Nakanishi, Hiroki Fujii, Yuta Hira, and Nobuo Funabiki. Revocable Group Signature Schemes with Constant Costs for Signing and Verifying. In *PKC*, LNCS, pages 463–480. Springer, 2009.

[NPR99] M. Naor, B. Pinkas, and O. Reingold. Distributed pseudo-random functions and KDCs. In *Eurocrypt*, 1999.

[NR97] M. Naor and O. Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *FOCS*, 1997.

[Oka95] T. Okamoto. An efficient divisible electronic cash scheme. In *Crypto*, 1995.

[PS18a] C. Peikert and S. Shiehian. Privately constraining and programming PRFs, the LWE way. In *PKC*, 2018.

[PS18b] David Pointcheval and Olivier Sanders. Reassessing security of randomizable signatures. In *CT-RSA 2018*, volume 10808 of *LNCS*, pages 319–338. Springer, 2018.

[PST17] David Pointcheval, Olivier Sanders, and Jacques Traoré. Cut down the tree to achieve constant complexity in divisible e-cash. In *PKC 2017*, volume 10174 of *LNCS*, pages 61–90. Springer, 2017.

[PVW08] C. Peikert, V. Vaikuntanathan, and B. Waters. A framework for efficient and composable oblivious transfer. In *Crypto*, 2008.

[PW08] C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In *STOC*, 2008.

[RAD78] R. Rivest, L. Adleman, and M. Dertouzos. On data banks and privacy homomorphisms. In *FOCS*, 1978.

[Reg05] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, 2005.

[RS91] C. Rackoff and D. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Crypto*, 1991.

[RSS19] R. Rothblum, A. Sealfon, and K. Sotiraki. Towards non-interactive zero-knowledge for np from lwe. In *PKC*, 2019.

[Rüc10] Markus Rückert. Lattice-based blind signatures. In *ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 413–430. Springer, 2010.

[Sch96] Claus Peter Schnorr. Security of 2^t-Root Identification and Signatures. In *Crypto*, LNCS, pages 143–156. Springer, 1996.

[SS10] D. Stehlé and R. Steinfeld. Faster fully homomorphic encryption. In *Asiacrypt*, 2010.

[Ste96] Jacques Stern. A new paradigm for public key identification. *IEEE Transactions on Information Theory*, 42(6):1757–1768, 1996.

[SV10] N. Smart and F. Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In *PKC*, 2010.

[SW05] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *Eurocrypt*, 2005.

[TZW16] Haibo Tian, Fangguo Zhang, and Baodian Wei. A lattice-based partially blind signature. *Security and Communication Networks*, 9(12):1820–1828, 2016.

[XXZ11] X. Xie, R. Xue, and R. Zhang. Efficient threshold encryption from lossy trapdoor functions. In *PQCrypto*, 2011.

[Yam16] S. Yamada. Adaptively secure identity-based encryption from lattices with asymptotically shorter public parameters. In *Eurocrypt*, 2016.

[Yam17] S. Yamada. Asymptotically compact adaptively secure lattice ibes and verifiable random functions via generalized partitioning techniques. In *Crypto*, 2017.

[ZJZ+18] Pingyuan Zhang, Han Jiang, Zhihua Zheng, Peichu Hu, and Qiuliang Xu. A new post-quantum blind signature from lattice assumptions. *IEEE Access*, 6:27251–27258, 2018.