Zero-Knowledge Arguments for Lattice-Based PRFs and Applications to E-Cash

Benoît Libert[1,2], San Ling[3], Khoa Nguyen[3], and Huaxiong Wang[3]

[1] CNRS, Laboratoire LIP, France
[2] ENS de Lyon, Laboratoire LIP (U. Lyon, CNRS, ENSL, INRIA, UCBL), France
[3] School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore

Abstract. Beyond their security guarantees under well-studied assumptions, algebraic pseudo-random functions are motivated by their compatibility with efficient zero-knowledge proof systems, which is useful in a number of privacy applications like digital cash. We consider the problem of proving the correct evaluation of lattice-based PRFs based on the Learning-With-Rounding (LWR) problem introduced by Banerjee *et al.* (Eurocrypt'12). Namely, we are interested zero-knowledge arguments of knowledge of triples (y, k, x) such that $y = F_k(x)$ is the correct evaluation of a PRF for a secret input x and a committed key k. While analogous statements admit efficient zero-knowledge protocols in the discrete logarithm setting, they have never been addressed in lattices so far. We provide such arguments for the key homomorphic PRF of Boneh *et al.* (Crypto'13) and the generic PRF implied by the LWR-based pseudo-random generator. As an application of our ZK arguments, we design the first compact e-cash system based on lattice assumptions. By "compact", we mean that the complexity is at most logarithmic in the value of withdrawn wallets. Our system can be seen as a lattice-based analogue of the first compact e-cash construction due to Camenisch, Hohenberger and Lysyanskaya (Eurocrypt'05).

Keywords. Lattices, pseudo-random functions, zero-knowledge arguments, e-cash systems, anonymity.

1 Introduction

Since the seminal results of Ajtai [2] and Regev [85], lattice-based cryptography has been a very active area which undergone quite rapid development, notably with the advent of lattice trapdoors [52,76] and homomorphic encryption [51]. Not only does it enable powerful functionalities, it also offers many advantages over conventional number-theoretic techniques, like simpler arithmetic operations, its conjectured resistance to quantum attacks or a better asymptotic efficiency.

The design of numerous cryptographic protocols appeals to zero-knowledge proofs [55] to prove properties about encrypted or committed values so as to enforce honest behavior on behalf of participants or protect the privacy of users. In the lattice settings, efficient zero-knowledge proofs are non-trivial to construct.

While natural solutions exist for proving knowledge of secret keys [77,73,64,70], they are only known to work for very specific languages. When it comes to proving general circuit satisfiability, the best known methods rely on the ring variants [89,14] of the Learning-With-Errors (LWE) and Short Integer Solution (SIS) problems and are not known to readily carry over to standard lattices. In the standard model, the problem is even trickier as we do not have a lattice-based counterpart of Groth-Sahai proofs [58] and efficient non-interactive proof systems are only available for specific problems [84].

In this paper, we consider the natural problem of proving the correct evaluation of lattice-based pseudo-random functions (PRFs) w.r.t. committed keys and inputs. This problem arises in numerous protocols where a user has to deterministically generate a random-looking value without betraying his identity.

We provide zero-knowledge arguments of correct evaluation for the LWE-based PRF of Boneh, Lewi, Montgomery and Raghunathan (BLMR) [17] as well as the construction generically obtained from pseudo-random generators via the Goldreich-Goldwasser-Micali (GGM) methodology [54]. As an application of our arguments, we provide the first lattice-based realization of the compact e-cash primitive of Camenisch, Hohenberger and Lysyanskaya [22].

Introduced by Chaum [33,34], electronic cash is the digital counterpart of regular money. As envisioned in [33], digital cash involve a bank and several users and merchants. It allows users to withdraw digital coins from the bank in such a way that e-coins can later be spent at merchants. In the on-line setting [33,35,36], merchants contact the bank before accepting any payment so that the bank is involved in all transactions to prevent double-spendings. In the (usually preferred) off-line model [37], the merchant accepts payments without any interaction with the bank: the deposit phase is postponed to a later moment where the merchant can return many coins at once. In all cases, when a merchant returns coins back to the bank, the latter should infer no information as to when and by whom the returned coins were withdrawn. Should the bank collude with the merchant, it remains unable to link a received coin to a particular execution of the withdrawal protocol. Of course, dishonest users should not be able to spend more coins than they withdrew without being identified. While fair e-cash systems [88] resort to an off-line trusted authority to call out cheaters, classical e-cash [37] allows identifying double-spenders without any TTP. In 2005, Camenisch, Hohenberger and Lysyanskaya [22] advocated e-cash solutions with *compactness* property: namely, a compact e-cash scheme allows a user to withdraw a wallet of 2^L coins in such a way that the complexity of spending and withdrawal protocols does not exceed $\mathcal{O}(L + \lambda)$, where λ is the security parameter. The constructions of [22] elegantly combine signature schemes with efficient protocols [25,26], number theoretic pseudo-random functions [44] and zero-knowledge proofs, making it possible to store a wallet using only $\mathcal{O}(L + \lambda)$ bits.

1.1 Our Contributions

Our Results. We describe the first compact e-cash system [22] based on lattice assumptions. Here, consistently with the literature on e-cash, "compactness" refers to schemes where the withdrawal, spending and deposit phases have at most logarithmic complexities in the maximal value of withdrawn wallets (analogously to the solutions of [22] where the term "compact" was introduced). The security of our scheme is proved in the random oracle model [11] under the Short Integer Solution (SIS) and LWE assumptions.

As a crucial ingredient of our solution, we provide zero-knowledge arguments vouching for the correct evaluation of lattice-based pseudo-random functions. More precisely, we construct arguments of knowledge of a committed seed $\mathbf{k}$, a secret input J and an output $\mathbf{y}$ satisfying $\mathbf{y} = F_{\mathbf{k}}(J)$. We describe such arguments for the key-homomorphic PRF of Boneh *et al.* [17] and the PRF obtained by applying the Goldreich-Goldwasser-Micali (GGM) [54] paradigm. As a building block, we provide zero-knowledge arguments for statements related to the Learning-With-Rounding (LWR) problem of Banerjee, Peikert and Rosen [8]. Given a public random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, it requires to tell apart vectors $\lfloor \mathbf{A}^T \cdot \mathbf{s} \rfloor_p = \lfloor (p/q) \cdot \mathbf{A}^T \cdot \mathbf{s} \rfloor \in \mathbb{Z}_p^m$ from the uniform distribution $U(\mathbb{Z}_p^m)$ over $\mathbb{Z}_p^m$, where $q > p \geq 2$. A crucial step of our argument system consists in demonstrating the correct computation of the rounding step: i.e., proving that $\mathbf{y} = \lfloor \mathbf{x} \rfloor_p$, for $\mathbf{x} \in \mathbb{Z}_q^m$ satisfying some additional context-dependent constraints.

We believe that our zero-knowledge arguments can find use cases in many other applications involving PRFs, and where zero-knowledge proofs constrain participants not to deviate from the protocol. Examples include privacy-preserving de-centralized e-cash systems [12,57,39], stateful anonymous credentials [40], n-times periodic anonymous authentication [21], traceable ring signatures [50], anonymous survey systems [59], password-protected secret sharing [61] or unlinkable pseudonyms for privacy-preserving distributed databases [24]. We also think of distributed PRFs [75,80], where servers holding a polynomial share $\mathbf{k}_i$ of the seed $\mathbf{k}$ can prove the correctness of their contribution w.r.t. to their committed share $\mathbf{k}_i$. Our arguments may also prove useful in the context of oblivious PRF evaluations [49,62], where one party holds a PRF key $\mathbf{k}$ and must convince the other party that $\mathbf{k}$ was correctly used in oblivious computations.

Our Techniques. In order to convince a verifier of the correct evaluation of LWR-based PRFs, the first step is to provide evidence that the underlying rounding operation is properly carried out. For dimension $m > 1$ and moduli $q > p \geq 2$, identify $\mathbb{Z}_q$, $\mathbb{Z}_p$ as the set $[0, q-1]$ and $[0, p-1]$, respectively, and consider the function $\lfloor \cdot \rfloor_p : \mathbb{Z}_q^m \to \mathbb{Z}_p^m : \mathbf{x} \mapsto \mathbf{y} = \lfloor (p/q) \cdot \mathbf{x} \rfloor \bmod p$. We observe that, one knows secret vector $\mathbf{x} \in [0, q-1]^m$ such that $\lfloor \mathbf{x} \rfloor_p = \mathbf{y}$ for a given $\mathbf{y} \in [0, p-1]^m$, if and only if one knows $\mathbf{x}, \mathbf{z} \in [0, q-1]^m$ such that

$$p \cdot \mathbf{x} = q \cdot \mathbf{y} + \mathbf{z} \mod pq. \qquad (1)$$

This crucial observation gives us a modular equation where the secret vectors $\mathbf{x}, \mathbf{z}$ are "small" relatively to the modulus pq. To prove that we know such secret vectors (where $\mathbf{x}$ may satisfy additional statements, e.g., it is committed, or certified, or is the output of other algorithms), we exploit Ling *et al.*'s decomposition-extension framework [70], which interacts well with Stern's permuting technique [86]. Specifically, we employ a matrix $\mathbf{H}_{m,q-1} \in \mathbb{Z}_q^{m \times \bar{m}}$, where

$\bar{m} = m\lceil \log q \rceil$, that allows to compute $\tilde{\mathbf{x}}, \tilde{\mathbf{z}} \in \{0,1\}^{\bar{m}}$ such that $\mathbf{H}_{m,q-1} \cdot \tilde{\mathbf{x}} = \mathbf{x}$ and $\mathbf{H}_{m,q-1} \cdot \tilde{\mathbf{z}} = \mathbf{z}$. Then, we let $\mathsf{B}^2_{\bar{m}}$ be the set of all vectors in $\{0,1\}^{2\bar{m}}$ that have fixed Hamming weight $\bar{m}$, and append $\bar{m}$ suitable entries to $\tilde{\mathbf{x}}, \tilde{\mathbf{z}}$ to obtain $\widehat{\mathbf{x}}, \widehat{\mathbf{z}} \in \mathsf{B}^2_{\bar{m}}$. Now, equation (1) is rewritten as:

$$\big(p \cdot [\mathbf{H}_{m,q-1} \mid \mathbf{0}^{m \times \bar{m}}]\big) \cdot \widehat{\mathbf{x}} - [\mathbf{H}_{m,q-1} \mid \mathbf{0}^{m \times \bar{m}}] \cdot \widehat{\mathbf{z}} = q \cdot \mathbf{y} \bmod pq. \tag{2}$$

Note that, one knows $\mathbf{x}, \mathbf{z} \in [0, q-1]^m$ satisfying (1) if and only if one can compute $\widehat{\mathbf{x}}, \widehat{\mathbf{z}} \in \mathsf{B}^2_{\bar{m}}$ satisfying (2). Moreover, as the constraint of $\widehat{\mathbf{x}}, \widehat{\mathbf{z}}$ is invariant under permutation (namely, $\widehat{\mathbf{x}}, \widehat{\mathbf{z}} \in \mathsf{B}^2_{\bar{m}}$ if and only if $\pi_x(\widehat{\mathbf{x}}), \pi_z(\widehat{\mathbf{z}}) \in \mathsf{B}^2_{\bar{m}}$, where π_x, π_z are permutations of $2\bar{m}$ elements), the latter statement can be handled via Stern's technique. Our method is readily extended to prove that the underlying vector $\mathbf{x}$ satisfies additional statements.

Let us now consider the problem of proving a correct evaluation of the Boneh *et al.* PRF [17]. The function uses public binary matrices $\mathbf{P}_0, \mathbf{P}_1 \in \{0,1\}^{m \times m}$ and a secret seed $\mathbf{k} \in \mathbb{Z}_q^m$ which allows mapping an input $J \in \{0,1\}^L$ to

$$F_{\mathbf{k}}(J) = \big\lfloor \mathbf{P}_{J[L]} \cdot \mathbf{P}_{J[L-1]} \ \cdots \ \mathbf{P}_{J[1]} \cdot \mathbf{k} \big\rfloor_p.$$

We consider the evaluation process iteratively and transform intermediate witnesses using the decomposition-extension framework [70], so that they nicely interact with Stern's permuting technique [86]. Namely, we define a sequence $\{\mathbf{x}_i\}_{i=0}^L$ which is initialized with $\mathbf{x}_0 = \mathbf{k} \in \mathbb{Z}_q^m$, iteratively computed as $\mathbf{x}_i = \mathbf{P}_{J[i]} \cdot \mathbf{x}_{i-1} \in \mathbb{Z}_q^m$, for each $i \in [1, L]$, and eventually yields the output $\mathbf{y} = \lfloor \mathbf{x}_L \rfloor_p$. For each $i \in [1, L]$, we translate the equation $\mathbf{x}_i = \mathbf{P}_{J[i]} \cdot \mathbf{x}_{i-1} \bmod q$ into

$$\mathbf{x}_i = \big[\mathbf{P}_0 \mid \mathbf{P}_1\big] \cdot \mathbf{t}_{i-1} \bmod q, \qquad \text{with} \qquad \mathbf{t}_{i-1} = \begin{pmatrix} \overline{J[i]} \cdot \mathbf{x}_{i-1} \\ J[i] \cdot \mathbf{x}_{i-1} \end{pmatrix}$$

and where $J[i]$ and $\overline{J[i]} = 1 - J[i]$ are part of the witnesses. Using suitable decomposition-extension techniques [70,69] on vectors $\{\mathbf{x}_i\}_{i=0}^L, \{\mathbf{t}_i\}_{i=1}^L$, we manage to express all the L iterative equations by just one equation of the form $\mathbf{M}_1 \cdot \mathbf{w}_1 = \mathbf{u}_1 \bmod q$, for some public matrix $\mathbf{M}_1$ and vector $\mathbf{u}_1$ over $\mathbb{Z}_q$, while $\mathbf{w}_1$ is a binary vector containing secret bits of all the witnesses and fitting a certain pattern. Meanwhile, the rounding step $\mathbf{y} = \lfloor \mathbf{x}_L \rfloor_p$, as discussed above, would yield an equation of the form $\mathbf{M}_2 \cdot \mathbf{w}_2 = \mathbf{u}_2 \bmod pq$, where $\mathbf{w}_2$ is correlated to $\mathbf{w}_1$. Furthermore, our applications require to additionally prove that a binary representation of the seed $\mathbf{x}_0 = \mathbf{k}$ is properly committed or certified, while the commitment or signature scheme may use a different modulus. Thus, we eventually have to handle relations of the form $\mathbf{M}_i \cdot \mathbf{w}_i = \mathbf{u}_i \bmod q_i$ for several moduli $q_1, \ldots, q_N$ when, for distinct $i, j \in [N]$, witnesses $\mathbf{w}_i, \mathbf{w}_j$ may have entries in common. An abstraction of Stern's protocol was recently suggested by Libert *et al.* [68] to address a similar setting when one has to prove a number of linear relations. Unfortunately, their framework, which deals with a unique modulus, does not directly cover our setting. To overcome this problem, we thus put forward a generalization of Libert *et al.*'s framework, so as to handle correlated witnesses across relations modulo distinct integers.

The above techniques thus smoothly interact with the pseudo-random functions of Boneh *et al.* [17] and the PRG of [8]. Unfortunately, we did not manage to extend them to other existing PRFs [8,7,45] based on the hardness of LWR. In the synthesizer-based construction of Banerjee *et al.* [8], the difficulty is the lack of public matrices which would help us reduce the statement to an assertion of the form $\mathbf{M} \cdot \mathbf{w} = \mathbf{u}$, for some witness $\mathbf{w} \in \mathbb{Z}^m$ and public $\mathbf{M} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{u} \in \mathbb{Z}_q^n$. Our zero-knowledge arguments do not appear to carry over to the key homomorphic functions of Banerjee and Peikert [7] either as they rely on a more complex tree-like structure. The fact that our techniques do *not* apply to all known lattice-based PRFs emphasizes that they are far more innovative than just an application of generic zero-knowledge paradigms, which would resort to a circuit decomposition of the evaluation algorithm and proceed in a gate-by-gate manner. Indeed, we process all statements without decomposing the arithmetic operations into a circuit.

Our compact e-cash construction builds on the design principle of Camenisch *et al.* [22] which combines signatures with efficient protocols [25,26], algebraic pseudo-random functions [44] and zero-knowledge proofs. In the lattice setting, we take the same approach by combining (a variant of) the signature scheme with efficient protocols of [68] and the PRF of [17]. While the GGM-based PRF of [8] would allow a more efficient choice of parameters, we chose to instantiate our system with the realization of Boneh *et al.* [17] since it simplifies the description and the security proof (specifically, we do not have to rely on the pseudo-randomness of the function in one of the security properties). However, our scheme can be modified to rely on the PRF built on the LWR-based PRG.

As in [22], the withdrawal phase allows the user to obtain a wallet of value $2^L - 1$ which consists of two PRF seeds, a counter and a signature generated by the bank on committed values. In the withdrawal protocol, the PRF seeds are obliviously signed (and bound to the user's secret key) by the bank using a signature scheme with efficient protocols [25,26]. The first seed $\mathbf{k}$ is used to derive each coin's serial number $\mathbf{y}_S = F_{\mathbf{k}}(J) \in \mathbb{Z}_p^m$ as a pseudo-random function of an L-bit counter $J \in \{0,1\}^L$ which denotes the number of previously spent coins. By spending the same coin twice, the user is thus forced to use the same serial number in two distinct transactions, making the cheating attempt detectable.

The second PRF seed $\mathbf{t}$ is used to compute a security tag $\mathbf{y}_T$ that allows identifying double-spenders. This tag is a vector $\mathbf{y}_T = PK_{\mathcal{U}} + H(\mathtt{info}) \cdot F_{\mathbf{t}}(J) \in \mathbb{Z}_p^m$, where $PK_{\mathcal{U}}$ is the user's public key and $H(\mathtt{info}) \in \mathbb{Z}_p^{m \times m}$ is a matrix generated by hashing some transaction-specific information supplied by the merchant. From two coins that share the same serial number $\mathbf{y}_S$ and distinct security tags $\mathbf{y}_{T,1} = PK_{\mathcal{U}} + H(\mathtt{info}_1) \cdot F_{\mathbf{t}}(J)$ and $\mathbf{y}_{T,2} = PK_{\mathcal{U}} + H(\mathtt{info}_2) \cdot F_{\mathbf{t}}(J)$, the difference $\mathbf{y}_{T,1} - \mathbf{y}_{T,2} = (H(\mathtt{info}_1) - H(\mathtt{info}_2)) \cdot F_{\mathbf{t}}(J)$ allows computing the PRF value $F_{\mathbf{t}}(J) = (H(\mathtt{info}_1) - H(\mathtt{info}_2))^{-1} \cdot (\mathbf{y}_{T,1} - \mathbf{y}_{T,2}) \in \mathbb{Z}_p^m$ (and then $PK_{\mathcal{U}}$) whenever $H(\mathtt{info}_1) - H(\mathtt{info}_2)$ is invertible over $\mathbb{Z}_p$. This property is precisely ensured by the Full-Rank Difference function of Agrawal *et al.* [1], which comes in handy to instantiate $H : \{0,1\}^* \rightarrow \mathbb{Z}_p^{m \times m}$. In contrast with [1], the Full-Rank Difference function is utilized in the scheme while [1] uses it in security proofs.

1.2 Related Work

E-Cash. Chaum's pioneering work [33,34] inspired a large body of research towards efficient e-cash systems [37,82,38,47,83,87] with better properties during two decades. The first compact realization was given by Camenisch *et al.* [22] whose techniques served as a blueprint for many subsequent e-cash systems with additional features such as refined accountability-anonymity tradeoffs [23], coin endorsement [27], or security proofs in the standard model [10]. The authors of [22] extended their schemes with a coin tracing mechanism whereby all the coins of a double-spender can be traced once this user has been identified.

Divisible e-cash [82] allow users to withdraw a wallet of value 2^L in such a way that each spending may involve transactions of variable amounts. While the early constructions [82,83] only provided weaker anonymity properties, Canard and Gouget gave truly anonymous realizations [28] using tree-based techniques which were subsequently made scalable [29,30]. The recent adoption of de-centralized payment systems [79] has triggered a new line of research towards strengthening the privacy of Bitcoin (see [78,12,57] and references therein).

To our knowledge, all truly private compact e-cash systems rely on discrete-logarithm-based techniques, either because of the underlying pseudo-random function [22,10] or via accumulators [5] (or both). In the lattice setting, we are not aware of any compact e-cash realization and neither do we know of any proofs of correct PRF evaluation with or without random oracles. In particular, it remains an open problem to build verifiable random functions [74] from lattices.

Lattices and Zero-knowledge Proofs. Existing methods of proving relations appearing in lattice-based cryptosystems belong to two main families. The first family, introduced by Lyubashevsky [73], uses "rejection sampling" techniques, and recently lead to relatively efficient proofs of knowledge of small secret vectors [13,14,9,42,43]. However, due to the nature of "rejection sampling" mechanisms, even the honest prover may fail to convince the verifier with a tiny probability: i.e., protocols in this family do not have perfect completeness. Furthermore, when proving knowledge of vectors having norm bound β, the knowledge extractor of these protocols is only guaranteed to produce witnesses of norm bound $g \cdot \beta$, for some factor $g > 1$. This factor, called the "soundness slack" in [9,42], may have an undesirable consequence: if an extracted witness has to be used in the security proof to solve a challenge SIS instance, we have to rely on the $\mathsf{SIS}_{g\cdot\beta}$ assumption, which is stronger than the SIS_{β} assumption required by the protocol itself. Moreover, in some advanced protocols such as those considered in this work, the coordinates of extracted vectors are expected to be in $\{0,1\}$ and/or satisfy a specific pattern. Such issues seem hard to tackle using this family of protocols.

The second family, initiated by Ling *et al.* [70], rely on "decomposition-extension" techniques in lattice-based analogues [64] of Stern's protocol [86]. Stern-like systems are less efficient than those of the first family because each protocol execution admits a constant soundness error, requiring the protocols to be repeated $\omega(\log \lambda)$ times in order to achieve a negligible soundness error. On the upside, Stern-like protocols do have perfect completeness and are capable of handling a wide range of lattice-based relations [66,71,69,68,67], especially when the witnesses are not only required to be small or binary, but should also have prescribed arrangements of coordinates. Moreover, unlike protocols of the first family, the extractor of Stern-like protocols are able to output witness vectors having exactly the same properties as those expected from valid witnesses. This feature is often crucial in the design of advanced cryptographic constructions involving zero-knowledge proofs. Additionally, the "soundness slack" issue is completely avoided, so that the hardness assumptions are kept "in place".

When it comes to proving the correct evaluation of AES-like secret key primitives, several works [63,48,32] built zero-knowledge proofs upon garbled circuits or multi-party computation [60,53], which may lead to truly practical proofs [53] even for non-algebraic statements. However, the garbled circuit paradigm [63] inherently requires interactive proofs (and cannot be made non-interactive via Fiat-Shamir [46]), making it unsuitable to our applications where coins must carry a non-interactive proof. While Giacomelli *et al.* [53] successfully designed efficient non-interactive proofs for SHA-256 evaluations, these remain of linear length in the circuit size and efficiently combining them with proofs of algebraic statements is non-trivial here: in the e-cash setting, our goal is to prove the correct evaluation of LWE-based symmetric primitives for committed inputs and keys. To our knowledge, known results on the smooth integration of algebraic and non-algebraic statements [32] are obtained by tweaking the approach of Jawurek *et al.* [63], which requires interaction.

Despite the scarcity of truly efficient zero-knowledge proofs in the lattice-setting, a recent body of work successfully designed proof systems in privacy-preserving protocols [64,56,65,13,81,71]. These results, however, only considered ring signatures [19,64], group signatures [56,65,66,13,81,71], group encryption [67] or building blocks [68] for anonymous credentials [35]. As of the time of writing, lattice-based realizations of anonymous e-cash still remain lacking.

2 Background and Definitions

Vectors are denoted in bold lower-case letters and bold upper-case letters will denote matrices. The Euclidean and infinity norm of any vector $\mathbf{b} \in \mathbb{R}^n$ 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.

For any $\mathbf{x} \in \mathbb{Z}_q^m$, the notation $\lfloor \mathbf{x} \rfloor_p$ stands for the result of the rounding operation $\lfloor \mathbf{x} \rfloor_p = \lfloor (p/q) \cdot \mathbf{x} \rfloor \bmod p$. Intuitively, the mapping $\lfloor \cdot \rfloor_p : \mathbb{Z}_q^m \to \mathbb{Z}_p^m$ can be seen as dividing $\mathbb{Z}_q$ into p intervals of size (q/p) and sending each coordinate of $\mathbf{x} \in \mathbb{Z}_q^m$ to the interval it belongs to.

The column concatenation of matrices $\mathbf{A} \in \mathbb{R}^{n \times k}$ and $\mathbf{B} \in \mathbb{R}^{n \times m}$ is denoted by $[\mathbf{A} \,|\, \mathbf{B}] \in \mathbb{R}^{n \times (k+m)}$. When concatenating column vectors $\mathbf{x} \in \mathbb{R}^k$ and $\mathbf{y} \in \mathbb{R}^m$, for simplicity, we often use the notation $(\mathbf{x} \| \mathbf{y}) \in \mathbb{R}^{k+m}$ (instead of $(\mathbf{x}^\top \| \mathbf{y}^\top)^\top$).

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 $\mathbb{R}^m$. 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 $\Lambda_q(\mathbf{A}) := \{\mathbf{e} \in \mathbb{Z}^m \mid \exists \mathbf{s} \in \mathbb{Z}_q^n \;\; s.t. \;\; \mathbf{A}^T \cdot \mathbf{s} = \mathbf{e} \bmod q\}$ as well as*

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

For any arbitrary $\mathbf{t} \in \Lambda_q^{\mathbf{u}}(\mathbf{A})$, we also define the shifted lattice $\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \Lambda_q^{\perp}(\mathbf{A}) + \mathbf{t}$.

For a lattice L, a vector $\mathbf{c} \in \mathbb{R}^m$ and a real number $\sigma > 0$, define the function $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \exp(-\pi\|\mathbf{x}-\mathbf{c}\|^2/\sigma^2)$. The discrete Gaussian distribution of support L, center $\mathbf{c}$ and parameter σ is defined as $D_{L,\sigma,\mathbf{c}}(\mathbf{y}) = \rho_{\sigma,\mathbf{c}}(\mathbf{y})/\rho_{\sigma,\mathbf{c}}(L)$ for any $\mathbf{y} \in L$, where $\rho_{\sigma,\mathbf{c}}(L) = \sum_{\mathbf{x}\in L} \rho_{\sigma,\mathbf{c}}(\mathbf{x})$. We denote by $D_{L,\sigma}(\mathbf{y})$ the distribution centered in $\mathbf{c} = \mathbf{0}^m$ and exploit the fact that samples from $D_{L,\sigma}$ are short w.h.p.

Lemma 1 ([6, Le. 1.5]). *For any lattice $L \subseteq \mathbb{R}^m$ and positive real number $\sigma > 0$, we have $\Pr_{\mathbf{b}\hookleftarrow D_{L,\sigma}}[\|\mathbf{b}\| \leq \sqrt{m}\sigma] \geq 1 - 2^{-\Omega(m)}$.*

It is well-known that Gaussian distributions with lattice support can be efficiently sampled from a sufficiently short basis of the lattice.

Lemma 2 ([20, 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 rely on the trapdoor generation algorithm of Alwen and Peikert [4].

Lemma 3 ([4, Th. 3.2]). *There exists 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 $\mathbf{T}_{\mathbf{A}}$ 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 $\|\widetilde{\mathbf{T}_{\mathbf{A}}}\| \leq \mathcal{O}(\sqrt{n\log q})$.*

We utilize the basis delegation algorithm [31] 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 $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ as a submatrix.

Lemma 4 ([31, Le. 3.2]). *There exists a PPT algorithm that inputs a matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times m'}$ whose first m columns span $\mathbb{Z}_q^n$, and a basis $\mathbf{T}_{\mathbf{A}}$ of $\Lambda_q^{\perp}(\mathbf{A})$ where $\mathbf{A}$ is an $n\times m$ submatrix of $\mathbf{B}$, and outputs a basis $\mathbf{T}_{\mathbf{B}}$ of $\Lambda_q^{\perp}(\mathbf{B})$ with $\|\widetilde{\mathbf{T}_{\mathbf{B}}}\| \leq \|\widetilde{\mathbf{T}_{\mathbf{A}}}\|$.*

Our security proofs use a technique introduced by Agrawal *et al.* [1].

Lemma 5 ([1, Th. 19]). *There exists a PPT algorithm that inputs matrices $\mathbf{A}, \mathbf{C} \in \mathbb{Z}_q^{n\times m}$, a small-norm matrix $\mathbf{R} \in \mathbb{Z}^{m\times m}$, a short basis $\mathbf{T}_{\mathbf{C}} \in \mathbb{Z}^{m\times m}$ of $\Lambda_q^{\perp}(\mathbf{C})$, a vector $\mathbf{u} \in \mathbb{Z}_q^n$ and a rational σ such that $\sigma \geq \|\widetilde{\mathbf{T}_{\mathbf{C}}}\| \cdot \Omega(\sqrt{\log n})$, and outputs vectors $\mathbf{b} \in \mathbb{Z}^{2m}$ such that $\left[\mathbf{A} \mid \mathbf{A}\cdot\mathbf{R} + \mathbf{C}\right] \cdot \mathbf{b} = \mathbf{u} \bmod q$ and with distribution statistically close to $D_{L,\sigma}$ where L denotes the shifted lattice $\{\mathbf{x} \in \mathbb{Z}^{2m} : \left[\mathbf{A} \mid \mathbf{A}\cdot\mathbf{R} + \mathbf{C}\right] \cdot \mathbf{x} = \mathbf{u} \bmod q\}$.*

2.2 Hardness Assumptions

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

Definition 3. *Let q, α be functions of a parameter n. For a secret $\mathbf{s} \in \mathbb{Z}_q^n$, the distribution $A_{q,\alpha,\mathbf{s}}$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ is obtained by sampling $\mathbf{a} \hookleftarrow U(\mathbb{Z}_q^n)$ and a noise $e \hookleftarrow D_{\mathbb{Z},\alpha q}$, and returning $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$. The Learning-With-Errors problem $\mathsf{LWE}_{q,\alpha}$ is, for $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$, to distinguish between arbitrarily many independent samples from $U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$ and the same number of samples from $A_{q,\alpha,\mathbf{s}}$.*

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 $\mathsf{SIS}_{m,q,\beta}$ (see, e.g., [52, Se. 9]). Similarly, if $\alpha q = \Omega(\sqrt{n})$, standard worst-case lattice problems with approximation factors $\gamma = \mathcal{O}(\alpha/n)$ reduce [85,20] to $\mathsf{LWE}_{q,\alpha}$. In the design of deterministic primitives like PRFs, the following variant of LWE comes in handy.

Definition 4 ([8]). *Let q, p, m be functions of a security parameter n such that $q > p \geq 2$ and $m > n$. The Learning-With-Rounding (*LWR*) problem is to distinguish the distribution $\{(\mathbf{A},\ \lfloor \mathbf{A}^T \cdot \mathbf{s} \rceil_p) \mid \mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n \times m}),\ \mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)\}$ from the distribution $\{(\mathbf{A},\ \mathbf{y}) \mid \mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n \times m}),\ \mathbf{y} \hookleftarrow U(\mathbb{Z}_p^m)\}$.*

Banerjee *et al.* [8] proved that LWR is as hard as LWE when q and the modulus-to-error ratio are super-polynomial. Alwen *et al.* [3] showed that, when the number m of samples is fixed in advance, LWR retains its hardness for polynomial moduli. Bogdanov *et al.* [15] generalized the result of [3] to get rid of restrictions on the modulus q. For such parameters, their result implies the security of the LWR-based PRG, which stretches the seed $\mathbf{s} \in \mathbb{Z}_q^n$ into $\lfloor \mathbf{A}^T \cdot \mathbf{s} \rceil_p \in \mathbb{Z}_p^m$.

2.3 Syntactic Definitions for Off-line Compact E-Cash

An off-line e-cash system involves a bank $\mathcal{B}$, many users $\mathcal{U}$ and merchants $\mathcal{M}$. In the syntax defined by Camenisch, Hohenberger and Lysyanskaya [22], all these parties interact together via the following algorithms and protocols:

ParGen(1^λ)**:** inputs a security parameter 1^λ and outputs public parameters par.

In the following, we assume that par are available to all parties although we sometimes omit them from the inputs of certain algorithms.

BKeygen(par)**:** generates a bank's key pair $(SK_{\mathcal{B}}, PK_{\mathcal{B}})$ which allows $\mathcal{B}$ to issue wallets of value $2^L \in \mathsf{poly}(\lambda)$ (we assume that L is part of par).

UKeygen(par)**:** generates a user key pair $(SK_{\mathcal{U}}, PK_{\mathcal{U}})$.

Withdraw$\big(\mathcal{U}(PK_{\mathcal{B}}, SK_{\mathcal{U}}), \mathcal{B}(PK_{\mathcal{U}}, SK_{\mathcal{B}})\big)$**:** is an interactive protocol between a user $\mathcal{U}$ and the bank $\mathcal{B}$. The user $\mathcal{U}$ obtains either a wallet $\mathcal{W}$ of 2^L coins or an error message $\perp$. The bank outputs some state information $\mathsf{T}_{\mathcal{W}}$ which allows identifying $\mathcal{U}$, should he overspend.

Spend$\big(\mathcal{U}(\mathcal{W}, PK_{\mathcal{B}}, PK_{\mathcal{M}}, \texttt{info}), \mathcal{M}(SK_{\mathcal{M}}, PK_{\mathcal{B}}, 2^L)\big)$**:** is a protocol whereby the user $\mathcal{U}$, on input of public keys $PK_{\mathcal{M}}, PK_{\mathcal{B}}$ and some transaction-specific meta data `info`, spends a coin from his wallet $\mathcal{W}$ to merchant $\mathcal{M}$. The merchant obtains a coin *coin* comprised of a serial number and a proof of validity. $\mathcal{U}$'s output is an updated wallet $\mathcal{W}'$.

VerifyCoin$(\mathsf{par}, PK_{\mathcal{M}}, PK_{\mathcal{B}}, coin)$**:** is a non-interactive coin verification algorithm. On input of a purported coin and the public keys $PK_{\mathcal{M}}$, $PK_{\mathcal{B}}$ of the bank and the merchant, it outputs 0 or 1.

Deposit$\big(\mathcal{M}(SK_{\mathcal{M}}, coin, PK_{\mathcal{B}}), \mathcal{B}(PK_{\mathcal{M}}, SK_{\mathcal{B}}, \mathsf{state}_{\mathcal{B}}))\big)$**:** is a protocol allowing the merchant $\mathcal{M}$ to deposit a received coin *coin* into its account at the bank $\mathcal{B}$. $\mathcal{M}$ outputs $\perp$ if the protocol fails and nothing if it succeeds. The bank $\mathcal{B}$ outputs "accept" and updates its state $\mathsf{state}_{\mathcal{B}}$ by adding an entry $(PK_{\mathcal{M}}, coin)$ if $\mathsf{VerifyCoin}(\mathsf{par}, PK_{\mathcal{M}}, PK_{\mathcal{B}}, coin) = 1$ and no double-spending is detected. Otherwise, if $\mathsf{VerifyCoin}(\mathsf{par}, PK_{\mathcal{M}}, PK_{\mathcal{B}}, coin) = 1$ and $\mathsf{state}_{\mathcal{B}}$ already contains a coin with the same serial number, it outputs "user". If $\mathsf{VerifyCoin}(\mathsf{par}, PK_{\mathcal{M}}, PK_{\mathcal{B}}, coin) = 0$ or $\mathsf{state}_{\mathcal{B}}$ already contains an entry $(PK_{\mathcal{M}}, coin)$, it outputs "merchant".

Identify$\big(\mathsf{par}, PK_{\mathcal{B}}, coin_1, coin_2\big)$**:** is an algorithm that allows the bank $\mathcal{B}$ to identify a double-spender on input of two coins $coin_1$, $coin_2$ with identical serial numbers. The bank outputs the double-spender's public key $PK_{\mathcal{U}}$ and a proof Π_G that $\mathcal{U}$ indeed overspent.

Like [22], we assume that wallets $\mathcal{W}$ contain a counter J, initialized to 0, which indicates the number of previously spent coins. We also assume that each coin contains a serial number S, a proof of validity π as well as some information on the merchant's public key $PK_{\mathcal{M}}$ and some meta-data `info`.

Following [22], we say that an off-line e-cash system is *compact* if the bitlength of the wallet $\mathcal{W}$ and the communication/computational complexities of all protocols is at most logarithmic in the value of the wallet (i.e., linear in L).

We use the security definitions of [22], which formalize security requirements called *anonymity*, *balance*, *double-spender identification* and *exculpability.*

Informally, the *balance* property considers colluding users interacting with a honest bank and attempting to spend more coins than they withdraw. This property is broken if the adversary manages to spend a coin of which the serial number does not match the serial number of any legally withdrawn coin. *Double-spender identification* complements the balance property by requiring that a malicious user be unable to output two coins with the same serial number without being caught by the Identify algorithm. *Anonymity* mandates that, when a merchant returns a received coin to the bank, even if they collude, they cannot infer anything as to when and by whom the coin was withdrawn. The *exculpability* property captures that honest users cannot be falsely accused of being double-spenders: the adversary controls the bank and wins if it outputs two coins with the same serial number and such that Identify points to a well-behaved user. The formal definitions of these properties are recalled in the full version of the paper.

3 Warm-up: Permutations, Decompositions, Extensions

This section presents various notations and techniques that appeared (in slightly different forms) in earlier works on Stern-like protocols [70,66,71,69,68], and that will be used extensively throughout this work.

Permutations. For any positive integer $\mathfrak{m}$, we define the following sets.

- $\mathcal{S}_{\mathfrak{m}}$: the set of all permutations of $\mathfrak{m}$ elements.
- $\mathsf{B}^2_{\mathfrak{m}}$: the set of binary vectors in $\{0,1\}^{2\mathfrak{m}}$ with Hamming weight $\mathfrak{m}$. Note that for any $\mathbf{v} \in \mathbb{Z}^{2\mathfrak{m}}$ and $\pi \in \mathcal{S}_{2m}$, we have:

$$\mathbf{v} \in \mathsf{B}^2_{\mathfrak{m}} \Longleftrightarrow \pi(\mathbf{v}) \in \mathsf{B}^2_{\mathfrak{m}}. \tag{3}$$

- $\mathsf{B}^3_{\mathfrak{m}}$: the set of vectors in $\{-1,0,1\}^{3\mathfrak{m}}$ that have exactly $\mathfrak{m}$ coordinates equal to j, for every $j \in \{-1,0,1\}$. Note that for any $\mathbf{w} \in \mathbb{Z}^{3\mathfrak{m}}$ and $\phi \in \mathcal{S}_{3m}$:

$$\mathbf{w} \in \mathsf{B}^3_{\mathfrak{m}} \Longleftrightarrow \phi(\mathbf{w}) \in \mathsf{B}^3_{\mathfrak{m}}. \tag{4}$$

For bit $c \in \{0,1\}$ and integer vector $\mathbf{v}$ of any dimension $\mathfrak{m}$, we denote by $\mathsf{Expand}(c,\mathbf{v})$ the vector $\begin{pmatrix} \bar{c} \cdot \mathbf{v} \\ c \cdot \mathbf{v} \end{pmatrix} \in \mathbb{Z}^{2\mathfrak{m}}$, where $\bar{c}$ denotes the bit $1-c$.

For any positive integer $\mathfrak{m}$, bit $b \in \{0,1\}$, and permutation $\pi \in \mathcal{S}_{\mathfrak{m}}$, we denote by $T_{b,\pi}$ the permutation that transforms the vector $\mathbf{v} = \begin{pmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \end{pmatrix} \in \mathbb{Z}^{2\mathfrak{m}}$, where $\mathbf{v}_0, \mathbf{v}_1 \in \mathbb{Z}^{\mathfrak{m}}$, into the vector $T_{b,\pi}(\mathbf{v}) = \begin{pmatrix} \pi(\mathbf{v}_b) \\ \pi(\mathbf{v}_{\bar{b}}) \end{pmatrix}$. Namely, $T_{b,\pi}$ first rearranges the 2 blocks of $\mathbf{v}$ according to b (it keeps the arrangement of blocks if $b=0$ and swaps them if $b=1$), then it permutes each block according to π.

Observe that the following equivalence holds for all $\mathfrak{m} \in \mathbb{Z}_+$, $b, c \in \{0,1\}$, $\pi \in \mathcal{S}_{\mathfrak{m}}$, $\mathbf{v} \in \mathbb{Z}^{\mathfrak{m}}$:

$$\mathbf{z} = \mathsf{expand}(c,\mathbf{v}) \iff T_{b,\pi}(\mathbf{z}) = \mathsf{expand}(c \oplus b, \pi(\mathbf{v})), \tag{5}$$

where $\oplus$ denotes the addition operation modulo 2.

Decompositions. For any $B \in \mathbb{Z}_+$, define $\delta_B := \lfloor \log_2 B \rfloor + 1 = \lceil \log_2(B+1) \rceil$ and the sequence $B_1, \ldots, B_{\delta_B}$, where $B_j = \lfloor \frac{B+2^{j-1}}{2^j} \rfloor$, for each $j \in [1, \delta_B]$. As observed in [72,70], it 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. $v' := v$
2. For $j = 1$ to $\delta_B$ do:
     (i) If $v' \geq B_j$ then $v^{(j)} := 1$, else $v^{(j)} := 0$;
     (ii) $v' := v' - B_j \cdot v^{(j)}$.
3. Output $\mathsf{idec}_B(v) = (v^{(1)}, \ldots, v^{(\delta_B)})^\top$.
```

Next, for any positive integers $\mathfrak{m}, B$, we define the matrix:

$$\mathbf{H}_{\mathfrak{m},B} := \begin{bmatrix} B_1 \ldots B_{\delta_B} & & & \\ & B_1 \ldots B_{\delta_B} & & \\ & & \ddots & \\ & & & B_1 \ldots B_{\delta_B} \end{bmatrix} \in \mathbb{Z}^{\mathfrak{m} \times \mathfrak{m}\delta_B}, \tag{6}$$

and the following injective functions:

(i) $\mathsf{vdec}_{\mathfrak{m},B} : [0, B]^{\mathfrak{m}} \to \{0,1\}^{\mathfrak{m}\delta_B}$ that maps the vector $\mathbf{v} = (v_1, \ldots, v_{\mathfrak{m}})$ to $\big(\mathsf{idec}_B(v_1) \| \ldots \| \mathsf{idec}_B(v_{\mathfrak{m}})\big)$. 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 decomposes $\mathbf{w} = (w_1, \ldots, w_{\mathfrak{m}})$ into the vector $\big(\sigma(w_1) \cdot \mathsf{idec}_B(|w_1|) \| \ldots \| \sigma(w_{\mathfrak{m}}) \cdot \mathsf{idec}_B(|w_{\mathfrak{m}}|)\big)$ such that, for each $i \in [\mathfrak{m}]$, we have: $\sigma(w_i) = 0$ if $w_i = 0$; $\sigma(w_i) = -1$ if $w_i < 0$; $\sigma(w_i) = 1$ if $w_i > 0$. Note that $\mathbf{H}_{\mathfrak{m},B} \cdot \mathsf{vdec}'_{\mathfrak{m},B}(\mathbf{w}) = \mathbf{w}$.

Extensions. We define following extensions of matrices and vectors.

- For any $\mathfrak{m}, B \in \mathbb{Z}_+$, define $\widehat{\mathbf{H}}_{\mathfrak{m},B} \in \mathbb{Z}^{\mathfrak{m} \times 2\mathfrak{m}\delta_B}$, $\breve{\mathbf{H}}_{\mathfrak{m},B} \in \mathbb{Z}^{\mathfrak{m} \times 3\mathfrak{m}\delta_B}$ as follows:

$$\widehat{\mathbf{H}}_{\mathfrak{m},B} := \big[\, \mathbf{H}_{\mathfrak{m},B} \,\big|\, \mathbf{0}^{\mathfrak{m} \times \mathfrak{m}\delta_B} \,\big]; \qquad \breve{\mathbf{H}}_{\mathfrak{m},B} := \big[\, \mathbf{H}_{\mathfrak{m},B} \,\big|\, \mathbf{0}^{\mathfrak{m} \times 2\mathfrak{m}\delta_B} \,\big].$$

- Given $\mathbf{v} \in \{0,1\}^{\mathfrak{m}}$, define $\mathsf{TwoExt}(\mathbf{v}) := (\mathbf{v} \| \mathbf{0}^{\mathfrak{m}-n_0} \| \mathbf{1}^{\mathfrak{m}-n_1}) \in \mathsf{B}^2_{\mathfrak{m}}$, where n_0, n_1 are the number of coordinates in $\mathbf{v}$ equal to 0 and 1, respectively.
- Given $\mathbf{v} \in [-1, 0, 1]^{\mathfrak{m}}$, define

$$\mathsf{ThreeExt}(\mathbf{v}) := (\mathbf{v} \| \mathbf{0}^{\mathfrak{m}-n_0} \| \mathbf{1}^{\mathfrak{m}-n_1} \| {-\mathbf{1}}^{\mathfrak{m}-n_{-1}}) \in \mathsf{B}^3_{\mathfrak{m}},$$

where n_0, n_1, n_{-1} are the number of coordinates in $\mathbf{v}$ equal to 0, 1, and -1, respectively.

Note that, if $\mathbf{x} \in [0, B]^m$ and $\mathbf{y} \in [-B, B]^m$, then we have:

$$\mathsf{TwoExt}\big(\mathsf{vdec}_{m,B}(\mathbf{x})\big) \in \mathsf{B}^2_{m\delta_B} \text{ and } \widehat{\mathbf{H}}_{m,B} \cdot \mathsf{TwoExt}\big(\mathsf{vdec}_{m,B}(\mathbf{x})\big) = \mathbf{x}, \tag{7}$$

$$\mathsf{ThreeExt}\big(\mathsf{vdec}'_{m,B}(\mathbf{y})\big) \in \mathsf{B}^3_{m\delta_B} \text{ and } \breve{\mathbf{H}}_{m,B} \cdot \mathsf{ThreeExt}\big(\mathsf{vdec}'_{m,B}(\mathbf{y})\big) = \mathbf{y}. \tag{8}$$

In the framework of Stern-like protocols [86,64,70,71,68], the above techniques are useful when it comes proving in zero-knowledge the possession of integer vectors satisfying several different constraints:

Case 1: $\mathbf{x} \in [0, B]^m$. We equivalently prove $\hat{\mathbf{x}} = \mathsf{TwoExt}\big(\mathsf{vdec}_{m,B}(\mathbf{x})\big) \in \mathsf{B}^2_{m\delta_B}$. To do this, pick $\pi \hookleftarrow U(\mathcal{S}_{2m\delta_B})$, and convince the verifier that $\pi(\hat{\mathbf{x}}) \in \mathsf{B}^2_{m\delta_B}$.

Case 2: $\mathbf{x} \in [-B, B]^m$. We equivalently prove $\breve{\mathbf{x}} = \mathsf{ThreeExt}\big(\mathsf{vdec}'_{m,B}(\mathbf{y})\big) \in \mathsf{B}^3_{m\delta_B}$. To do this, pick $\pi \hookleftarrow U(\mathcal{S}_{3m\delta_B})$, and convince the verifier that $\pi(\breve{\mathbf{x}}) \in \mathsf{B}^3_{m\delta_B}$.

Case 3: $\mathbf{x} = \mathsf{expand}(c, \mathbf{v})$, where $\mathbf{v}$ satisfies one of the above two constraints. To hide $\mathbf{v}$, we use the respective decomposition-extension-permutation technique. To hide the bit c, we pick a "one-time pad" $b \hookleftarrow U(\{0,1\})$ and exploit the equivalence observed in (5). Looking ahead, this technique will be used in Section 4.3 to hide the bits of the PRF input J and those of a signature component $\tau \in \{0,1\}^\ell$ in Section 5.

4 Zero-Knowledge Arguments for Lattice-Based PRFs

Here, we first give an abstraction of Stern's protocol [86]. With this abstraction in mind, we then present our techniques for achieving zero-knowledge arguments for the BLMR PRF [17].

In the full version of the paper, we adapt these techniques to the PRF generically implied by the GGM [54] paradigm. While slightly more complex to describe, the GGM-based construction allows for a better choice of parameters since, owing to the result of Bogdanov *et al.* [15], it allows instantiating the LWR-based PRG with polynomial-size moduli.

4.1 An Abstraction of Stern's Protocol

1. **Commitment:** $\mathcal{P}$ samples $\phi \leftarrow U(\mathcal{S})$, $\mathbf{r}_1 \leftarrow U(\mathbb{Z}_{q_1}^{d_1}), \ldots, \mathbf{r}_N \leftarrow U(\mathbb{Z}_{q_N}^{d_N})$, and computes $\mathbf{r} = (\mathbf{r}_1 \| \ldots \| \mathbf{r}_N)$, $\mathbf{z} = \mathbf{w} \boxplus \mathbf{r}$.
   Then $\mathcal{P}$ samples randomness ρ_1, ρ_2, ρ_3 for COM, and sends $\mathrm{CMT} = \big(C_1, C_2, C_3\big)$ to $\mathcal{V}$, where $C_1 = \mathsf{COM}(\phi, \{\mathbf{M}_i \cdot \mathbf{r}_i \bmod q_i\}_{i \in [N]};\, \rho_1)$, and
   $$C_2 = \mathsf{COM}(\varGamma_\phi(\mathbf{r});\, \rho_2), \quad C_3 = \mathsf{COM}(\varGamma_\phi(\mathbf{z});\, \rho_3).$$
2. **Challenge:** $\mathcal{V}$ sends a challenge $Ch \leftarrow U(\{1, 2, 3\})$ to $\mathcal{P}$.
3. **Response:** $\mathcal{P}$ sends RSP computed according to Ch, as follows:
   - $Ch = 1$: $\mathrm{RSP} = (\mathbf{t}, \mathbf{s}, \rho_2, \rho_3)$, where $\mathbf{t} = \varGamma_\phi(\mathbf{w})$ and $\mathbf{s} = \varGamma_\phi(\mathbf{r})$.
   - $Ch = 2$: $\mathrm{RSP} = (\pi, \mathbf{x}, \rho_1, \rho_3)$, where $\pi = \phi$ and $\mathbf{x} = \mathbf{z}$.
   - $Ch = 3$: $\mathrm{RSP} = (\psi, \mathbf{y}, \rho_1, \rho_2)$, where $\psi = \phi$ and $\mathbf{y} = \mathbf{r}$.

Verification: Receiving RSP, $\mathcal{V}$ proceeds as follows:

- $Ch = 1$: Check that $\mathbf{t} \in \mathsf{VALID}$, and $C_2 = \mathsf{COM}(\mathbf{s}; \rho_2)$, $\quad C_3 = \mathsf{COM}(\mathbf{t} \boxplus \mathbf{s}; \rho_3)$.
- $Ch = 2$: Parse $\mathbf{x} = (\mathbf{x}_1 \| \ldots \| \mathbf{x}_N)$, where $\mathbf{x}_i \in \mathbb{Z}_{q_i}^{d_i}$ for all $i \in [N]$, and check that
  $$C_1 = \mathsf{COM}(\pi, \{\mathbf{M}_i \cdot \mathbf{x}_i - \mathbf{u}_i \bmod q_i\}_{i \in [N]}; \rho_1), \quad C_3 = \mathsf{COM}(\varGamma_\pi(\mathbf{x}); \rho_3).$$
- $Ch = 3$: Parse $\mathbf{y} = (\mathbf{y}_1 \| \ldots \| \mathbf{y}_N)$, where $\mathbf{y}_i \in \mathbb{Z}_{q_i}^{d_i}$ for all $i \in [N]$, and check that
  $$C_1 = \mathsf{COM}(\psi, \{\mathbf{M}_i \cdot \mathbf{y}_i \bmod q_i\}_{i \in [N]}; \rho_1), \quad C_2 = \mathsf{COM}(\varGamma_\psi(\mathbf{y}); \rho_2).$$

In each case, $\mathcal{V}$ outputs 1 if and only if all the conditions hold.

Fig. 1: Our abstract protocol.

In [86], Stern proposed a zero-knowledge protocol for the Syndrome Decoding problem, in which the main idea is to use a random permutation over coordinates of a secret vector to prove that the latter satisfies a given constraint (e.g., having fixed Hamming weight). Later on, Stern's protocol was adapted to the lattice setting by Kawachi *et al.* [64] and refined by Ling *et al.* [70] to handle statements related to the SIS and LWE problems. Subsequently, the protocol was further developed to design several lattice-based systems [66,71,69]. Recently, Libert *et al.* [68] suggested an abstraction that addresses the setting where one has to prove knowledge of small secret vectors satisfying a number of modular linear equations with respect to one modulus. While their generalization subsumes many relations that naturally appear in privacy-preserving protocols involving lattices, it is not immediately applicable to the statements considered in this paper since we have to work with more than one modulus.

We thus put forward a new abstraction of Stern's protocol [86] that handles modular equations with respect to $N \geq 1$ moduli $q_1, \ldots, q_N$, where secret witnesses may simultaneously appear across multiple equations.

Let n_i and $d_i \geq n_i$ be positive integers, and let $d = d_1 + \cdots + d_N$. Suppose that VALID is a subset of $\{-1,0,1\}^d$ and $\mathcal{S}$ is a finite set such that every $\phi \in \mathcal{S}$ can be associated with a permutation $\varGamma_\phi$ of d elements satisfying the conditions

$$\begin{cases} \mathbf{w} \in \mathsf{VALID} \iff \varGamma_\phi(\mathbf{w}) \in \mathsf{VALID}; \\ \text{If } \mathbf{w} \in \mathsf{VALID} \text{ and } \phi \text{ is uniform in } \mathcal{S}, \text{ then } \varGamma_\phi(\mathbf{w}) \text{ is uniform in } \mathsf{VALID}. \end{cases} \tag{9}$$

In our abstract protocol, for public matrices $\{\mathbf{M}_i \in \mathbb{Z}_{q_i}^{n_i \times d_i}\}_{i \in [N]}$ and vectors $\mathbf{u}_i \in \mathbb{Z}_{q_i}^{n_i}$, the prover argues in zero-knowledge the possession of integer vectors $\{\mathbf{w}_i \in \{-1,0,1\}^{d_i}\}_{i \in [N]}$ such that:

$$\mathbf{w} = \big(\mathbf{w}_1 \| \ldots \| \mathbf{w}_N\big) \in \mathsf{VALID}, \tag{10}$$

$$\forall i \in [N]:\ \mathbf{M}_i \cdot \mathbf{w}_i = \mathbf{u}_i \bmod q_i. \tag{11}$$

Looking ahead, all the statements considered in Sections 4.3, 5 will be reduced to the above setting, wherein secret vectors $\mathbf{w}_1, \ldots, \mathbf{w}_N$ are mutually related, e.g., some entries of $\mathbf{w}_i$ also appear in $\mathbf{w}_j$.

The main ideas driving our protocol are as follows. To prove (10), the prover samples $\phi \leftarrow U(\mathcal{S})$ and provides evidence that $\varGamma_\phi(\mathbf{w}) \in \mathsf{VALID}$. The verifier should be convinced while learning nothing else, owing to the aforementioned properties of the sets VALID and $\mathcal{S}$. Meanwhile, to prove that equations (11) hold, the prover uses masking vectors $\{\mathbf{r}_i \leftarrow U(\mathbb{Z}_{q_i}^{d_i})\}_{i \in [N]}$ and demonstrates instead that $\mathbf{M}_i \cdot (\mathbf{w}_i + \mathbf{r}_i) = \mathbf{u}_i + \mathbf{M}_i \cdot \mathbf{r}_i \bmod q_i$.

The interaction between prover $\mathcal{P}$ and verifier $\mathcal{V}$ is described in Figure 1. The common input consists of $\{\mathbf{M}_i \in \mathbb{Z}_{q_i}^{n_i \times d_i}\}_{i \in [N]}$ and $\mathbf{u}_i \in \mathbb{Z}_{q_i}^{n_i}$, while $\mathcal{P}$'s secret input is $\mathbf{w} = \big(\mathbf{w}_1 \| \ldots \| \mathbf{w}_N\big)$. The protocol makes use of a statistically hiding and computationally binding string commitment scheme COM such as the SIS-based commitment of [64]. For simplicity of presentation, for vectors $\mathbf{w} = \big(\mathbf{w}_1 \| \ldots \| \mathbf{w}_N\big) \in \mathbb{Z}^d$ and $\mathbf{r} = \big(\mathbf{r}_1 \| \ldots \| \mathbf{r}_N\big) \in \mathbb{Z}^d$, we denote by $\mathbf{w} \boxplus \mathbf{r}$ the operation that computes $\mathbf{z}_i = \mathbf{w}_i + \mathbf{r}_i \bmod q_i$ for all $i \in [N]$, and outputs d-dimensional integer vector $\mathbf{z} = \big(\mathbf{z}_1 \| \ldots \| \mathbf{z}_N\big)$. We note that, for all $\phi \in \mathcal{S}$, if $\mathbf{t} = \varGamma_\phi(\mathbf{w})$ and $\mathbf{s} = \varGamma_\phi(\mathbf{r})$, then we have $\varGamma_\phi(\mathbf{w} \boxplus \mathbf{r}) = \mathbf{t} \boxplus \mathbf{s}$.

The properties of our protocol are summarized in the following theorem.

Theorem 1. *Suppose that* COM *is a statistically hiding and computationally binding string commitment. Then, the protocol of Figure 1 is a zero-knowledge argument of knowledge for the given statement, with perfect completeness, soundness error* $2/3$*, and communication cost* $\mathcal{O}\big(\sum_{i=1}^{N} d_i \cdot \log q_i\big)$*. In particular:*

- *There exists an efficient simulator that, on input $\{\mathbf{M}_i, \mathbf{u}_i\}_{i\in[N]}$, outputs an accepted transcript statistically close to that produced by the real prover.*
- *There exists an efficient knowledge extractor that, on input a commitment* CMT *as well as valid responses* $(\mathrm{RSP}_1, \mathrm{RSP}_2, \mathrm{RSP}_3)$ *to all three possible values of the challenge Ch, outputs a witness $\mathbf{w}' = (\mathbf{w}'_1\|\ldots\|\mathbf{w}'_N) \in \mathsf{VALID}$ such that $\mathbf{M}_i \cdot \mathbf{w}'_i = \mathbf{u}_i \bmod q_i$, for all $i \in [N]$.*

The proof of Theorem 1 employs standard simulation and extraction techniques of Stern-like protocols [64,70,69,68], and is deferred to the full version of the paper.

4.2 Transforming the LWR Relation

Let $q \geq p \geq 2$, $m \geq 1$, and let $\mathbb{Z}_q = [0, q-1]$ and $\mathbb{Z}_p = [0, p-1]$. Consider the LWR rounding function: $\lfloor\cdot\rceil_p : \mathbb{Z}_q^m \to \mathbb{Z}_p^m \; : \; \mathbf{x} \mapsto \mathbf{y} = \lfloor (p/q)\cdot\mathbf{x}\rceil \bmod p$.

On the road towards zero-knowledge arguments for LWR-based PRFs, we have to build a sub-protocol that allows proving knowledge of a secret vector $\mathbf{x} \in \mathbb{Z}_q^m$ satisfying, among other statements, the property of rounding to a given $\mathbf{y} \in \mathbb{Z}_p^m$: i.e., $\lfloor\mathbf{x}\rceil_p = \mathbf{y}$. To our knowledge, such a sub-protocol is not available in the literature for the time being and must be designed from scratch.

Our crucial observation is that one knows $\mathbf{x} \in [0, q-1]^m$ such that $\lfloor\mathbf{x}\rceil_p = \mathbf{y}$, if and only if one can compute $\mathbf{x}, \mathbf{z} \in [0, q-1]^m$ such that:

$$p \cdot \mathbf{x} = q \cdot \mathbf{y} + \mathbf{z} \bmod pq. \tag{12}$$

This observation allows us to transform the LWR relation into an equivalent form that can be handled using the Stern-like techniques provided in Section 3. Let $\widehat{\mathbf{x}} = \mathsf{TwoExt}\big(\mathsf{vdec}_{m,q-1}(\mathbf{x})\big)$ and $\widehat{\mathbf{z}} = \mathsf{TwoExt}\big(\mathsf{vdec}_{m,q-1}(\mathbf{z})\big)$. Then we have $\mathbf{x} = \widehat{\mathbf{H}}_{m,q-1} \cdot \widehat{\mathbf{x}}$ and $\mathbf{z} = \widehat{\mathbf{H}}_{m,q-1} \cdot \widehat{\mathbf{z}}$, so that equation (12) can be written as:

$$(p \cdot \widehat{\mathbf{H}}_{m,q-1}) \cdot \widehat{\mathbf{x}} - \widehat{\mathbf{H}}_{m,q-1} \cdot \widehat{\mathbf{z}} = q \cdot \mathbf{y} \bmod pq. \tag{13}$$

Note that one knows $\mathbf{x}, \mathbf{z} \in [0, q-1]^m$ satisfying (12) if and only if one can compute $\widehat{\mathbf{x}}, \widehat{\mathbf{z}} \in \mathsf{B}^2_{m\delta_{q-1}}$ satisfying (13). Furthermore, Stern's framework allows proving the latter in zero-knowledge using random permutations.

4.3 Argument of Correct Evaluation for the BLMR PRF

We now consider the problem of proving the correct evaluation of the BLMR pseudo-random function from [17]. Namely, we would like to prove that a given $\mathbf{y} = \big\lfloor \prod_{i=1}^{L} \mathbf{P}_{J[L+1-i]} \cdot \mathbf{k} \big\rceil_p \in \mathbb{Z}_p^m$ is the correct evaluation for a committed seed $\mathbf{k} \in \mathbb{Z}_q^m$ and a secret input $J[1]\ldots J[L] \in \{0,1\}^L$, where $\mathbf{P}_0, \mathbf{P}_1 \in \{0,1\}^{m\times m}$ are public binary matrices, while revealing neither $\mathbf{k}$ nor $J[1]\ldots J[L]$. We assume public matrices $\mathbf{D}_0 \in \mathbb{Z}_{q_s}^{n\times m_0}$, $\mathbf{D}_1 \in \mathbb{Z}_{q_s}^{n\times \bar{m}}$, for some modulus q_s and integers m_0 and $\bar{m} = m\delta_{q-1}$, which are used to compute a KTX commitment [64] $\mathbf{c} = \mathbf{D}_0 \cdot \mathbf{r} + \mathbf{D}_1 \cdot \tilde{\mathbf{k}} \in \mathbb{Z}_{q_s}^n$ to the decomposition $\tilde{\mathbf{k}} = \mathsf{vdec}_{m,q-1}(\mathbf{k}) \in \{0,1\}^{\bar{m}}$ of the seed $\mathbf{k}$, where $\mathbf{r} \in [-\beta, \beta]^{m_0}$ is a discrete Gaussian vector (for some small integer β), and $\tilde{\mathbf{k}}$ satisfies $\mathbf{H}_{m,q-1} \cdot \tilde{\mathbf{k}} = \mathbf{k}$.

We first note that, in the evaluation process of $\mathbf{y} = \big\lfloor \prod_{i=1}^{L} \mathbf{P}_{J[L+1-i]} \cdot \mathbf{k} \big\rceil_p$, one works with vectors $\{\mathbf{x}_i \in \mathbb{Z}_q^m\}_{i=0}^{L}$ such that $\mathbf{x}_0 = \mathbf{k}$, $\mathbf{x}_i = \mathbf{P}_{J[i]} \cdot \mathbf{x}_{i-1} \bmod q$ for each $i \in \{1, \ldots, L\}$, and $\mathbf{y} = \lfloor \mathbf{x}_L \rceil_p$. We further observe that the iterative equation $\mathbf{x}_i = \mathbf{P}_{J[i]} \cdot \mathbf{x}_{i-1} \bmod q$ is equivalent to:

$$\mathbf{x}_i = \mathbf{P}_0 \cdot (\overline{J[i]} \cdot \mathbf{x}_{i-1}) + \mathbf{P}_1 \cdot (J[i] \cdot \mathbf{x}_{i-1}) = \big[\mathbf{P}_0 \,|\, \mathbf{P}_1\big] \cdot \begin{pmatrix} \overline{J[i]} \cdot \mathbf{x}_{i-1} \\ J[i] \cdot \mathbf{x}_{i-1} \end{pmatrix} \bmod q. \quad (14)$$

Intuitively, this observation allows us to move the secret bit $J[i]$ from the "matrix side" to the "vector side" in order to make the equation compatible with Stern-like protocols. Next, for each $i \in \{0, \ldots, L\}$, we form the vector $\widehat{\mathbf{x}}_i = \mathsf{TwoExt}\big(\mathsf{vdec}_{m,q-1}(\mathbf{x}_i)\big) \in \mathsf{B}^2_{\bar{m}}$. Equation (14) can then be written as:

$$\widehat{\mathbf{H}}_{m,q-1} \cdot \widehat{\mathbf{x}}_i = \big[\mathbf{P}_0 \cdot \widehat{\mathbf{H}}_{m,q-1} \,|\, \mathbf{P}_1 \cdot \widehat{\mathbf{H}}_{m,q-1}\big] \cdot \mathsf{expand}(J[i], \widehat{\mathbf{x}}_{i-1}) \bmod q.$$

Let $\mathbf{P} = \big[\mathbf{P}_0 \cdot \widehat{\mathbf{H}}_{m,q-1} \,|\, \mathbf{P}_1 \cdot \widehat{\mathbf{H}}_{m,q-1}\big] \in \mathbb{Z}_q^{m \times 4\bar{m}}$, and $\{\mathbf{s}_{i-1} = \mathsf{expand}(J[i], \widehat{\mathbf{x}}_{i-1})\}_{i=1}^{L}$, we have the L equations:

$$\begin{cases} \mathbf{P} \cdot \mathbf{s}_0 - \widehat{\mathbf{H}}_{m,q-1} \cdot \widehat{\mathbf{x}}_1 = \mathbf{0} \bmod q, \\ \qquad\qquad \vdots \\ \mathbf{P} \cdot \mathbf{s}_{L-1} - \widehat{\mathbf{H}}_{m,q-1} \cdot \widehat{\mathbf{x}}_L = \mathbf{0} \bmod q, \end{cases} \quad (15)$$

Regarding the rounding step $\lfloor \mathbf{x}_L \rceil_p = \mathbf{y} \in \mathbb{Z}_p^m$, using the transformations of Section 4.2, we obtain the following equation for $\widehat{\mathbf{z}} \in \mathsf{B}^2_{\bar{m}}$:

$$(p \cdot \widehat{\mathbf{H}}_{m,q-1}) \cdot \widehat{\mathbf{x}}_L - \widehat{\mathbf{H}}_{m,q-1} \cdot \widehat{\mathbf{z}} = q \cdot \mathbf{y} \bmod pq, \quad (16)$$

As for the commitment relation, we have the equation $\mathbf{c} = \mathbf{D}_0 \cdot \mathbf{r} + \mathbf{D}_1 \cdot \tilde{\mathbf{k}} \bmod q_s$, where $\mathbf{H}_{m,q-1} \cdot \tilde{\mathbf{k}} = \mathbf{x}_0$. We let $\check{\mathbf{r}} = \mathsf{ThreeExt}\big(\mathsf{vdec}'_{m_0,\beta}(\mathbf{r})\big) \in \mathsf{B}^3_{m_0\delta_\beta}$ and remark that $\mathsf{TwoExt}\big(\tilde{\mathbf{k}}\big) = \widehat{\mathbf{x}}_0$. Then, we have:

$$[\mathbf{D}_0 \cdot \check{\mathbf{H}}_{m_0,\beta}] \cdot \check{\mathbf{r}} + [\mathbf{D}_1 \mid \mathbf{0}^{n \times \bar{m}}] \cdot \widehat{\mathbf{x}}_0 = \mathbf{c} \bmod q_s. \quad (17)$$

Our goal is now reduced to proving the possession of $J[1] \ldots J[L] \in \{0,1\}^L$, $\widehat{\mathbf{x}}_0, \ldots, \widehat{\mathbf{x}}_L, \widehat{\mathbf{z}} \in \mathsf{B}^2_{\bar{m}}$ and $\check{\mathbf{r}} \in \mathsf{B}^3_{m_0\delta_\beta}$, satisfying equations (17), (15) and (16). Next, we let $q_1 = q_s$, $q_2 = q$, $q_3 = pq$, and proceed as follows.

Regarding equation (17), letting $\mathbf{M}_1 = \big[\mathbf{D}_0 \cdot \check{\mathbf{H}}_{m_0,\beta} \,\big|\, \mathbf{D}_1 \mid \mathbf{0}^{n \times \bar{m}}\big]$, $\mathbf{u}_1 = \mathbf{c}$ and $\mathbf{w}_1 = \big(\check{\mathbf{r}} \| \widehat{\mathbf{x}}_0\big)$, the equation becomes:

$$\mathbf{M}_1 \cdot \mathbf{w}_1 = \mathbf{u}_1 \bmod q_1.$$

Next, we unify the L equations in (15). To this end, we define

$$\mathbf{M}_2 = \begin{bmatrix} \mathbf{P} & -\widehat{\mathbf{H}}_{m,q-1} & & & \\ & \ddots & \ddots & & \\ & & & \mathbf{P} & -\widehat{\mathbf{H}}_{m,q-1} \end{bmatrix}, \quad \mathbf{u}_2 = \mathbf{0},$$

and $\mathbf{w}_2 = \big(\mathbf{s}_0 \| \widehat{\mathbf{x}}_1 \| \cdots \| \mathbf{s}_{L-1} \| \widehat{\mathbf{x}}_L\big)$. Then, (15) can be equivalently written as:

$$\mathbf{M}_2 \cdot \mathbf{w}_2 = \mathbf{u}_2 \bmod q_2.$$

As for equation (16), let $\mathbf{M}_3 = \big[(p \cdot \widehat{\mathbf{H}}_{m,q-1}) | -\widehat{\mathbf{H}}_{m,q-1}\big]$, $\mathbf{u}_3 = q \cdot \mathbf{y}$ and $\mathbf{w}_3 = \big(\widehat{\mathbf{x}}_L \| \widehat{\mathbf{z}}\big)$. Then, we obtain:

$$\mathbf{M}_3 \cdot \mathbf{w}_3 = \mathbf{u}_3 \bmod q_3.$$

Now, we let $d_1 = 3m_0\delta_\beta + 2\bar{m}$, $d_2 = 6L\bar{m}$ and $d_3 = 4\bar{m}$ be the dimensions of $\mathbf{w}_1, \mathbf{w}_2$ and $\mathbf{w}_3$, respectively, and $d = d_1 + d_2 + d_3$. We form the vector $\mathbf{w} = (\mathbf{w}_1 \| \mathbf{w}_2 \| \mathbf{w}_3) \in \{-1, 0, 1\}^d$, which has the form:

$$\mathbf{w} = \big(\check{\mathbf{r}} \,\|\, \widehat{\mathbf{x}}_0 \,\|\, \mathbf{s}_0 \,\|\, \widehat{\mathbf{x}}_1 \,\|\, \cdots \,\|\, \mathbf{s}_{L-1} \,\|\, \widehat{\mathbf{x}}_L \,\|\, \widehat{\mathbf{x}}_L \,\|\, \widehat{\mathbf{z}}\big). \tag{18}$$

At this point, we have come close to reducing our statement to an instance of the one considered in Section 4.1. Next, let us specify the set VALID containing $\mathbf{w}$, the set $\mathcal{S}$ and the associated permutation $\mathit{\Gamma}_\phi$ satisfying conditions in 9.

Let VALID be the set of all vectors in $\{-1, 0, 1\}^d$ having the form (18), where

- $\check{\mathbf{r}} \in \mathsf{B}^3_{m_0\delta_\beta}$, and $\widehat{\mathbf{x}}_0, \ldots, \widehat{\mathbf{x}}_L, \widehat{\mathbf{z}} \in \mathsf{B}^2_{\bar{m}}$.
- $\{\mathbf{s}_{i-1} = \mathsf{expand}(J[i], \widehat{\mathbf{x}}_{i-1})\}_{i=1}^L$, for some $J[1] \ldots J[L] \in \{0, 1\}^L$.

It can be seen that our vector $\mathbf{w}$ belongs to this tailored set VALID.

Now, we define $\mathcal{S} := \mathcal{S}_{3m_0\delta_\beta} \times (\mathcal{S}_{2\bar{m}})^{L+2} \times \{0, 1\}^L$. Then, for any set element $\phi = (\phi_r, \phi_0, \phi_1, \ldots, \phi_L, \phi_z, b_1 \ldots b_L) \in \mathcal{S}$, let $\mathit{\Gamma}_\phi$ be the permutation that transforms vector $\mathbf{w} \in \mathbb{Z}^d$ of the form (18) to vector $\mathit{\Gamma}_\phi(\mathbf{w})$ of the form:

$$\begin{aligned}\mathit{\Gamma}_\phi(\mathbf{w}) = \big(\phi_r(\check{\mathbf{r}}) \,\|\, \phi_0(\widehat{\mathbf{x}}_0) \,\|\, T_{b_1,\phi_0}(\mathbf{s}_0) \,\|\, \phi_1(\widehat{\mathbf{x}}_1) \,\|\, \cdots \,\|\, T_{b_L,\phi_{L-1}}(\mathbf{s}_{L-1}) \,\|\, \phi_L(\widehat{\mathbf{x}}_L)\\ \|\phi_L(\widehat{\mathbf{x}}_L) \,\|\, \phi_z(\widehat{\mathbf{z}})\big).\end{aligned}$$

Thanks to the equivalences (3), (4), (5) from Section 3, we have $\mathbf{w} \in \mathsf{VALID}$ if and only if $\mathit{\Gamma}_\phi(\mathbf{w}) \in \mathsf{VALID}$. Furthermore, if $\phi \leftarrow U(\mathcal{S})$, then $\mathit{\Gamma}_\phi(\mathbf{w})$ is uniform in VALID. Said otherwise, the conditions in (9) are satisfied.

Given the above transformations and specifications, we can now run the abstract protocol of Figure 1 to prove knowledge of $\mathbf{w} = (\mathbf{w}_1 \| \mathbf{w}_2 \| \mathbf{w}_3) \in \mathsf{VALID}$ satisfying $\{\mathbf{M}_i \cdot \mathbf{w}_i = \mathbf{u}_i \bmod q_i\}_{i=1,2,3}$, where public matrices/vectors $\{\mathbf{M}_i, \mathbf{u}_i\}_{i=1,2,3}$ are as constructed above. As a result, we obtain a statistical zero-knowledge argument of knowledge for the statement described at the beginning of this section. For simulation, we run the simulator of Theorem 1 with public input $\{\mathbf{M}_i, \mathbf{u}_i\}_{i=1,2,3}$. For extraction (see also the full version of the paper), we first run the knowledge extractor of Theorem 1, to obtain $\mathbf{w}' = (\mathbf{w}_1' \| \mathbf{w}_2' \| \mathbf{w}_3') \in \mathsf{VALID}$ such that $\{\mathbf{M}_i \cdot \mathbf{w}_i' = \mathbf{u}_i \bmod q_i\}_{i=1,2,3}$ and then reverse the witness transformations to get $\mathbf{k}' \in \mathbb{Z}_q^m$, $J'[1] \ldots J'[L] \in \{0, 1\}^L$ and $\mathbf{r}' \in [-\beta, \beta]^{m_0}$, $\tilde{\mathbf{k}}' \in \{0, 1\}^{\bar{m}}$ satisfying:

$$\mathbf{y} = \Big\lfloor \prod_{i=1}^{L} \mathbf{P}_{J'[L+1-i]} \cdot \mathbf{k}' \Big\rfloor_p, \quad \mathbf{c} = \mathbf{D}_0 \cdot \mathbf{r}' + \mathbf{D}_1 \cdot \mathbf{k}' \bmod q_s, \quad \mathbf{H}_{m,q-1} \cdot \tilde{\mathbf{k}}' = \mathbf{k}'.$$

The protocol has communication cost $\mathcal{O}(d_1 \cdot \log q_1 + d_2 \cdot \log q_2 + d_3 \cdot \log q_3)$. For a typical setting of parameters (as in Section 5), this cost is of order $\widetilde{\mathcal{O}}(\lambda \cdot L)$, where λ is the security parameter (and L is the input length of the PRF).

5 Description of Our Compact E-cash System

This section describes our e-cash system. We do not present a general construction from lower level primitives because such a construction is already implicit in the work of Camenisch *et al.* [22] of which we follow the blueprint. To avoid repeating it, we directly show how to apply the same design principle in lattices using carefully chosen primitives that interact with our zero-knowledge proofs.

Like [22], our scheme combines signatures with efficient protocols and pseudorandom functions which support proofs of correct evaluation. Our e-cash system builds on the signature scheme with efficient protocols of Libert *et al.* [68]. The latter is a variant of the SIS-based signatures described by Boyen [18] and Böhl *et al.* [16]. We actually use a simplified version of their scheme which is recalled in the full version of the paper and dispenses with the need to encode messages in a special way.

As in [22], our withdrawal protocol involves a step where the bank and the user jointly compute a seed $\mathbf{k} = \mathbf{k}_0 + \mathbf{k}_1 \in \mathbb{Z}_q^m$, which will be uniform over $\mathbb{Z}_q^m$ as long as one of the two parties is honest. The reason is that the identification of double-spenders can only be guaranteed if two distinct small-domain PRFs with independent random keys never collide, except with negligible probability. To jointly generate the PRF seed $\mathbf{k}$, the protocol of [22] relies on the homomorphic property of the commitment scheme used in their oblivious signing protocol. In our setting, one difficulty is that the underlying KTX commitment [64] has message space $\{0,1\}^{m\lceil \log q \rceil}$ and is not homomorphic over $\mathbb{Z}_q^m$. To solve this problem, our withdrawal protocol lets the user obtain the bank's signature on a message containing the binary decompositions of $\mathbf{k}_0$ and $\mathbf{k}_1$, so that the sum $\mathbf{k} = \mathbf{k}_0 + \mathbf{k}_1$ is only reconstructed during the spending phase.

At the withdrawal step, the user also chooses a second PRF seed $\mathbf{t} \in \mathbb{Z}_q^m$ of its own. The withdrawal protocol ends with the user obtaining a signature on the committed messages $(\mathbf{e}_u, \tilde{\mathbf{k}}_0, \tilde{\mathbf{k}}_1, \tilde{\mathbf{t}})$, where $(\tilde{\mathbf{k}}_0, \tilde{\mathbf{k}}_1, \tilde{\mathbf{t}})$ are bitwise decompositions of PRF seeds and $\mathbf{e}_u$ is the user's private key for which the corresponding public key is a GPV syndrome $PK_{\mathcal{U}} = \mathbf{F} \cdot \mathbf{e}_u \in \mathbb{Z}_p^m$, for a random matrix $\mathbf{F} \in \mathbb{Z}_p^{m \times m\lceil \log q \rceil}$.

In each spent coin, the user computes a serial number $\mathbf{y}_S = F_{\mathbf{k}}(J) \in \mathbb{Z}_p^m$ consisting of a PRF evaluation under $\mathbf{k} \in \mathbb{Z}_p^m$ and generates a NIZK argument that $\mathbf{y}_S$ is the correct evaluation for the secret index J and the key $\mathbf{k} = \mathbf{k}_0 + \mathbf{k}_1$ contained in the certified wallet. Note that the argument does not require a separate commitment to $\mathbf{k}$ since the bank's signature $sig_{\mathcal{B}}$ on the message $(\mathbf{e}_u, \tilde{\mathbf{k}}_0, \tilde{\mathbf{k}}_1, \tilde{\mathbf{t}})$ already contains a commitment to the bits of $(\mathbf{k}_0, \mathbf{k}_1)$. Since $sig_{\mathcal{B}}$ and $(\mathbf{e}_u, \tilde{\mathbf{k}}_0, \tilde{\mathbf{k}}_1, \tilde{\mathbf{t}})$ are part of the witnesses that the user argues knowledge of, it is eventually the bank's public key that commits the user to the seed $\mathbf{k}$.

In each coin, the identification of double-spenders is enabled by a security tag $\mathbf{y}_T = PK_{\mathcal{U}} + H_{\mathrm{FRD}}(R) \cdot F_{\mathbf{t}}(J) \in \mathbb{Z}_p^m$, where $H_{\mathrm{FRD}}(R)$ is a Full-Rank Difference function [1,41] of some transaction-specific information. If two coins share the same serial number $\mathbf{y}_S$, the soundness of the argument system implies that the two security tags $\mathbf{y}_{T,1}, \mathbf{y}_{T,2}$ hide the same $PK_{\mathcal{U}}$. By the Full Rank Difference property, subtracting $\mathbf{y}_{T,1} - \mathbf{y}_{T,2}$ exposes $F_{\mathbf{t}}(J) \in \mathbb{Z}_p^m$ and, in turn, $PK_{\mathcal{U}} \in \mathbb{Z}_p^m$.

The details of the underlying argument system are given in Section 5.2, where we show that the considered statement reduces to an instance of the abstraction given in Section 4.1. On the way, we use a combination our transformation techniques for the BLMR PRF from Section 4.3 and the Stern-like techniques for the signature signature scheme of [68].

5.1 Description

In the description below, we use the injective function $\mathsf{vdec}_{\mathsf{n},\mathsf{q}-1}(\cdot)$ defined in Section 3, which maps a vector $\mathbf{v} \in \mathbb{Z}_{\mathsf{q}}^{\mathsf{n}}$ to the vector $\mathsf{vdec}_{\mathsf{n},\mathsf{q}-1}(\mathbf{v}) \in \{0,1\}^{\mathsf{n}\lceil \log_2 \mathsf{q} \rceil}$.

$\mathsf{ParGen}(1^\lambda, 1^L)$: Given a security parameter $\lambda > 0$ and an integer $L > 0$ such that 2^L is the desired value of wallets, public parameters are chosen as follows.

1. Choose a lattice parameter $n = \mathcal{O}(\lambda)$. Choose parameters that will be used by the BLMR pseudo-random function [17]: an LWE parameter $\alpha = 2^{-\omega(\log^{1+c}(n))}$ for some constant $c > 0$; moduli $p = 2^{\log^{1+c}(n)}$ and $q = \mathcal{O}(\sqrt{n}/\alpha)$ such that p divides q; and dimension $m = \lceil n \log q \rceil$. Pick another prime modulus $q_s = \widetilde{\mathcal{O}}(n^4)$ to be used by the signature scheme. Pick an integer $\ell = \Theta(\lambda)$, a Gaussian parameter $\sigma = \Omega(\sqrt{n \log q_s} \log n)$, and an infinity norm bound $\beta = \sigma \cdot \omega(\log n)$. Let $\delta_{q_s-1} = \lceil \log_2(q_s) \rceil$, $\delta_{q-1} = \lceil \log_2(q) \rceil$, $\delta_{p-1} = \lceil \log_2(p) \rceil$.

   We will use an instance of the signature scheme with efficient protocols from [68], where matrices $(\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell}, \mathbf{D})$, $\{\mathbf{D}_k\}_{k=0}^{4}$ do not all have the same number of columns. Specifically, let $m_s = m_0 = 2n\delta_{q_s-1}$ and define the length of message blocks to be $m_1 = m_2 = m_3 = m_4 = \bar{m} = m\delta_{q-1}$. We also use an additional matrix $\mathbf{F} \in \mathbb{Z}_p^{m \times m_f}$, where $m_f = \bar{m} = m\delta_{q-1}$.
2. Choose a commitment key CK for a statistically hiding commitment where the message space is $\{0,1\}^{m_1} \times \{0,1\}^{m_2} \times \{0,1\}^{m_3}$. This commitment key $CK = \big([\mathbf{D}_0' \mid \mathbf{D}_0''],\ \mathbf{D}_1,\ \mathbf{D}_2,\ \mathbf{D}_3,\ \mathbf{D}_4\big)$ consists of random matrices $\mathbf{D}_0', \mathbf{D}_0'' \hookleftarrow U(\mathbb{Z}_{q_s}^{n \times m_0})$, $\mathbf{D}_1 \hookleftarrow U(\mathbb{Z}_{q_s}^{n \times m_1})$, $\mathbf{D}_2 \hookleftarrow U(\mathbb{Z}_{q_s}^{n \times m_2})$, $\mathbf{D}_3 \hookleftarrow U(\mathbb{Z}_{q_s}^{n \times m_3})$, $\mathbf{D}_4 \hookleftarrow U(\mathbb{Z}_{q_s}^{n \times m_4})$.
3. Select two binary matrices $\mathbf{P}_0, \mathbf{P}_1 \in \{0,1\}^{m \times m}$ uniformly among $\mathbb{Z}_q$-invertible matrices.
4. Finally, choose a full-rank difference function $H_{\mathrm{FRD}} : \mathbb{Z}_p^m \to \mathbb{Z}_p^{m \times m}$ such as [1], a collision-resistant hash function $H_0 : \{0,1\}^* \to \mathbb{Z}_p^m$ and another hash function $H : \{0,1\}^* \to \{1,2,3\}^\kappa$, for some $\kappa = \omega(\log \lambda)$, which will be modeled as a random oracle in the security analysis.

We define

$$\mathsf{par} := \Big(\mathbf{F},\ \{\mathbf{P}_0,\ \mathbf{P}_1\},\ H_{\mathrm{FRD}},\ H_0,\ H,\ CK\Big).$$

where $CK = (\mathbf{D}_0 = [\mathbf{D}_0' \mid \mathbf{D}_0''],\ \mathbf{D}_1, \mathbf{D}_2,\ \mathbf{D}_3,\ \mathbf{D}_4)$.

BKeygen$(1^\lambda, \mathsf{par})$**:** The bank $\mathcal{B}$ generates a key pair for the signature scheme with efficient protocols. This is done as follows.

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

The private key consists of $SK_{\mathcal{B}} := \mathbf{T_A}$ while the public key is

$$PK_{\mathcal{B}} := \big(\mathbf{A},\ \{\mathbf{A}_j\}_{j=0}^{\ell},\ \mathbf{D},\ \mathbf{u}\big).$$

UKeygen$(1^\lambda, \mathsf{par})$**:** As a secret key, the user picks $SK_{\mathcal{U}} := \mathbf{e}_u \hookleftarrow U(\{0,1\}^{m_f})$ at random and computes his public key $PK_{\mathcal{U}}$ as a syndrome $PK_{\mathcal{U}} = \mathbf{F} \cdot \mathbf{e}_u \in \mathbb{Z}_p^m$.

Withdraw$\big(\mathcal{U}(PK_{\mathcal{B}}, SK_{\mathcal{U}}, 2^L), \mathcal{B}(PK_{\mathcal{U}}, SK_{\mathcal{B}}, 2^L)\big)$**:** The bank $\mathcal{B}$, which has a key pair $(SK_{\mathcal{B}}, PK_{\mathcal{B}})$, interacts with $\mathcal{U}$, who has $SK_{\mathcal{U}} = \mathbf{e}_u$, as follows.

1. $\mathcal{U}$ picks $\mathbf{t}, \mathbf{k}_0 \hookleftarrow U(\mathbb{Z}_q^m)$ and computes $\tilde{\mathbf{t}} = \mathsf{vdec}_{m,q-1}(\mathbf{t}) \in \{0,1\}^{\bar{m}}$, $\tilde{\mathbf{k}}_0 = \mathsf{vdec}_{m,q-1}(\mathbf{k}_0) \in \{0,1\}^{\bar{m}}$. Then, he generates a commitment to the 3-block message $(\mathbf{e}_u, \tilde{\mathbf{t}}, \tilde{\mathbf{k}}_0) \in \{0,1\}^{m_f} \times \{0,1\}^{\bar{m}} \times \{0,1\}^{\bar{m}}$. To this end, $\mathcal{U}$ samples $\mathbf{r}_0 \hookleftarrow D_{\mathbb{Z}^{m_s},\sigma}$ and computes

$$\mathbf{c}_{\mathcal{U}} = \mathbf{D}_0' \cdot \mathbf{r}_0 + \mathbf{D}_1 \cdot \mathbf{e}_u + \mathbf{D}_2 \cdot \tilde{\mathbf{t}} + \mathbf{D}_3 \cdot \tilde{\mathbf{k}}_0 \in \mathbb{Z}_{q_s}^n, \tag{19}$$

   which is sent to $\mathcal{B}$. In addition, $\mathcal{U}$ generates an interactive zero-knowledge argument of knowledge of an opening

$$(\mathbf{r}_0, \mathbf{e}_u, \tilde{\mathbf{t}}, \tilde{\mathbf{k}}_0) \in D_{\mathbb{Z}^{m_s},\sigma} \times \{0,1\}^{m_f} \times \{0,1\}^{\bar{m}} \times \{0,1\}^{\bar{m}}$$

   of $\mathbf{c}_{\mathcal{U}} \in \mathbb{Z}_{q_s}^n$ satisfying (19) and such that $PK_{\mathcal{U}} = \mathbf{F} \cdot \mathbf{e}_u \in \mathbb{Z}_p^m$. We note that this argument system is obtained via a straightforward adaptation of the Stern-like protocol from [68].
2. If the argument of step 1 verifies, $\mathcal{B}$ samples $\mathbf{r}_1 \hookleftarrow D_{\mathbb{Z}^{m_s},\sigma}$, $\mathbf{k}_1 \hookleftarrow U(\mathbb{Z}_q^m)$ and computes $\tilde{\mathbf{k}}_1 = \mathsf{vdec}_{m,q-1}(\mathbf{k}_1) \in \{0,1\}^{\bar{m}}$ and a re-randomized version of $\mathbf{c}_{\mathcal{U}}$ which is obtained as $\mathbf{c}_{\mathcal{U}}' = \mathbf{c}_{\mathcal{U}} + \mathbf{D}_0'' \cdot \mathbf{r}_1 + \mathbf{D}_4 \cdot \tilde{\mathbf{k}}_1 \in \mathbb{Z}_{q_s}^n$. It defines $\mathbf{u}_{\mathcal{U}} = \mathbf{u} + \mathbf{D} \cdot \mathsf{vdec}_{n,q_s-1}\big(\mathbf{c}_{\mathcal{U}}'\big) \in \mathbb{Z}_{q_s}^n$. Next, $\mathcal{B}$ randomly picks $\tau \hookleftarrow \{0,1\}^\ell$ and uses $\mathbf{T_A}$ to compute a delegated basis $\mathbf{T}_\tau \in \mathbb{Z}^{2m_s \times 2m_s}$ for the matrix $\mathbf{A}_\tau \in \mathbb{Z}_{q_s}^{n \times 2m_s}$ defined as

$$\mathbf{A}_\tau = [\mathbf{A} \mid \mathbf{A}_0 + \sum_{j=1}^{\ell} \tau[j] \cdot \mathbf{A}_j] \in \mathbb{Z}_{q_s}^{n \times 2m_s}. \tag{20}$$

   Using $\mathbf{T}_\tau \in \mathbb{Z}^{2m_s \times 2m_s}$, $\mathcal{B}$ samples a short vector $\mathbf{v} \in \mathbb{Z}^{2m_s}$ in $D_{\Lambda_{q_s}^{\mathbf{u}_{\mathcal{U}}}(\mathbf{A}_\tau),\sigma}$. It returns $\mathbf{k}_1 \in \mathbb{Z}_q^m$ and the vector $(\tau, \mathbf{v}, \mathbf{r}_1) \in \{0,1\}^\ell \times \mathbb{Z}^{2m_s} \times \mathbb{Z}^{m_s}$ to $\mathcal{U}$.

3. $\mathcal{U}$ computes $\mathbf{r} = \begin{bmatrix}\mathbf{r}_0 \\ \mathbf{r}_1\end{bmatrix} \in \mathbb{Z}^{2m_s}$ and verifies that

$$\mathbf{A}_\tau \cdot \mathbf{v} = \mathbf{u} + \mathbf{D} \cdot \mathsf{vdec}_{n,q_s-1}\Big(\mathbf{D}_0 \cdot \mathbf{r} + \mathbf{D}_1 \cdot \mathbf{e}_u + \mathbf{D}_2 \cdot \mathsf{vdec}_{m,q-1}(\mathbf{t}) \\ + \mathbf{D}_3 \cdot \mathsf{vdec}_{m,q-1}(\mathbf{k}_0) + \mathbf{D}_4 \cdot \mathsf{vdec}_{m,q-1}(\mathbf{k}_1)\Big) \bmod q_s$$

and $\|\mathbf{v}\| \leq \sigma\sqrt{2m_s}$, $\|\mathbf{r}_1\| \leq \sigma\sqrt{m_s}$. If so, $\mathcal{U}$ saves the wallet

$$\mathcal{W} := \Big(\mathbf{e}_u, \mathbf{t}, \mathbf{k}_0, \mathbf{k}_1, sig_{\mathcal{B}} = (\tau, \mathbf{v}, \mathbf{r}), J = 0\Big),$$

where $J \in \{0, \ldots, 2^L - 1\}$ is a counter initialized to 0 (otherwise, it outputs $\perp$). The bank $\mathcal{B}$ records a debit of 2^L for the account $PK_{\mathcal{U}}$.

Spend$\big(\mathcal{U}(\mathcal{W}, PK_{\mathcal{B}}, PK_{\mathcal{M}}, \texttt{info}), \mathcal{M}(SK_{\mathcal{M}}, PK_{\mathcal{B}}, 2^L)\big)$**:** The user $\mathcal{U}$, on input of a wallet $\mathcal{W} = \big(\mathbf{e}_u, \mathbf{t}, \mathbf{k}_0, \mathbf{k}_1, sig_{\mathcal{B}} = (\tau, \mathbf{v}, \mathbf{r}), J\big)$, outputs $\perp$ if $J > 2^L - 1$. Otherwise, it runs the following protocol with $\mathcal{M}$.

1. Hash $\mathsf{info} \in \{0,1\}^*$ and $PK_{\mathcal{M}}$ to obtain $R = H_0(PK_{\mathcal{M}}, \mathsf{info}) \in \mathbb{Z}_p^m$.
2. Compute $\mathbf{k} = \mathbf{k}_0 + \mathbf{k}_1 \bmod q$, which will serve a PRF seed $\mathbf{k} \in \mathbb{Z}_q^m$. Using $\mathbf{k}$, compute the serial number

$$\mathbf{y}_S = \lfloor \prod_{i=1}^{L} \mathbf{P}_{J[L+1-i]} \cdot \mathbf{k} \rfloor_p \;\in \mathbb{Z}_p^m, \qquad (21)$$

where $J[1] \ldots J[L] \in \{0,1\}^L$ is the representation of $J \in \{0, \ldots, 2^L - 1\}$.

3. Using the PRF seed $\mathbf{t} \in \mathbb{Z}_q^m$, compute the security tag

$$\mathbf{y}_T = PK_{\mathcal{U}} + H_{\mathrm{FRD}}(R) \cdot \lfloor \prod_{i=1}^{L} \mathbf{P}_{J[L+1-i]} \cdot \mathbf{t} \rfloor_p \;\in \mathbb{Z}_p^m. \qquad (22)$$

4. Generate a non-interactive argument of knowledge π_K to prove that:
   (i) The given serial number $\mathbf{y}_S$ is the correct output of the PRF with key $\mathbf{k} = \mathbf{k}_0 + \mathbf{k}_1 \bmod q$ and input $J[1] \ldots J[L]$ (Equation (21));
   (ii) The same input $J[1] \ldots J[L]$ and another key $\mathbf{t}$ involve in the generation of the security tag $\mathbf{y}_T$ of the form (22);
   (iii) The PRF keys $\mathbf{k}_0, \mathbf{k}_1, \mathbf{t}$ and the secret key $\mathbf{e}_u$ that corresponds to $PK_{\mathcal{U}}$ in (22) were certified by the bank.

   This is done by running the interactive zero-knowledge argument presented in Section 5.2, which can be seen as a combination of 2 instances of the protocol for the PRF layer from Section 4.3 and one instance of the protocol for the signature layer from [68]. The argument is repeated $\kappa = \omega(\log \lambda)$ times to achieve negligible soundness error, and then made non-interactive using the Fiat-Shamir heuristic [46] as a triple $\pi_K = (\{\mathsf{Comm}_{K,j}\}_{j=1}^{\kappa}, \mathsf{Chall}_K, \{\mathsf{Resp}_{K,j}\}_{j=1}^{\kappa})$ where

$$\mathsf{Chall}_K = H(R, \mathbf{y}_S, \mathbf{y}_T, \{\mathsf{Comm}_{K,j}\}_{j=1}^{t}) \in \{1,2,3\}^{\kappa}.$$

$\mathcal{U}$ sends $coin = \big(R, \mathbf{y}_S, \mathbf{y}_T, \pi_K\big)$ to $\mathcal{M}$ who outputs $coin$ if $\mathsf{VerifyCoin}$ accepts it and $\perp$ otherwise. $\mathcal{U}$ outputs an updated wallet $\mathcal{W}'$, where J is incremented. We note that $coin$ has bit-size $\widetilde{\mathcal{O}}(L \cdot \lambda + \lambda^2)$, which is inherited from that of the underlying zero-knowledge argument system of Section 5.2.

$\mathsf{VerifyCoin}(\mathsf{par}, PK_{\mathcal{M}}, PK_{\mathcal{B}}, coin)$: Parse the coin as $coin = \big(R, \mathbf{y}_S, \mathbf{y}_T, \pi_K\big)$ and output 1 if and only if π_K properly verifies.

$\mathsf{Deposit}\big(\mathcal{M}(SK_{\mathcal{M}}, coin, PK_{\mathcal{B}})), \mathcal{B}(PK_{\mathcal{M}}, SK_{\mathcal{B}}, \mathsf{state}_{\mathcal{B}})\big)$: $coin = \big(R, \mathbf{y}_S, \mathbf{y}_T, \pi_K\big)$ is sent by $\mathcal{M}$ to the bank $\mathcal{B}$. If $\mathsf{VerifyCoin}(\mathsf{par}, PK_{\mathcal{M}}, PK_{\mathcal{B}}, coin) = 1$ and if serial number $\mathbf{y}_S$ does not already appear in any coin of the list $\mathsf{state}_{\mathcal{B}}$, $\mathcal{B}$ accepts $coin$, adds $(R, \mathbf{y}_S)$ in $\mathsf{state}_{\mathcal{B}}$ and credits $PK_{\mathcal{M}}$'s account. Otherwise, $\mathcal{B}$ returns "user" or "$\mathsf{merchant}$" depending on which party is declared faulty.

$\mathsf{Identify}\big(\mathsf{par}, PK_{\mathcal{B}}, coin_1, coin_2)\big)$: Given two coins $coin_1 = \big(R_1, \mathbf{y}_S, \mathbf{y}_{T,1}, \pi_{K,1}\big)$, $coin_2 = \big(R_2, \mathbf{y}_S, \mathbf{y}_{T,2}, \pi_{K,2}\big)$ with verifying proofs $\pi_{K,1}, \pi_{K,2}$ and the same serial number $\mathbf{y}_S \in \mathbb{Z}_p^m$ in distinct transactions $R_1 \neq R_2$, output $\perp$ if $\mathbf{y}_{T,1} = \mathbf{y}_{T,2}$. Otherwise, compute

$$\mathbf{y}'_T = \big(H_{\mathrm{FRD}}(R_1) - H_{\mathrm{FRD}}(R_2)\big)^{-1} \cdot (\mathbf{y}_{T,1} - \mathbf{y}_{T,2}) \in \mathbb{Z}_p^m$$

and then $PK_{\mathcal{U}} = \mathbf{y}_{T,1} - H_{\mathrm{FRD}}(R_1) \cdot \mathbf{y}'_T \in \mathbb{Z}_p^m$. The proof $\mathit{\Pi}_G$ that $\mathcal{U}$ is guilty simply consists of the two coins $coin_1$, $coin_2$ and the public key $PK_{\mathcal{U}}$.

In the full version of the paper, we show how to extend the scheme with a mechanism allowing to trace all the coins of an identified double-spender. Like Camenisch *et al.* [22], we can add this feature via a verifiable encryption step during the withdrawal phase. For this purpose, however, [22] crucially relies on properties of groups with a bilinear map that are not available here. To overcome this difficulty, we slightly depart from the design principle of [22] in that we rather use a secret-key verifiable encryption based on the hardness of LWE.

5.2 The Underlying Argument System of Our E-Cash Scheme

We now present the argument system employed by the Spend algorithm of the e-cash scheme in Section 5.1. This protocol is summarized as follows.

Let parameters λ, n, p, q, q_s, m, β, L, ℓ, $\bar{m} = m\delta_{q-1}$, $m_s = 2n\delta_{q_s-1}$ be as specified in Section 5.1. The public input consists of:

$$\begin{cases} \mathbf{D} \in \mathbb{Z}_{q_s}^{n\times(m_s/2)};\ \mathbf{D}_0 \in \mathbb{Z}_{q_s}^{n\times 2m_s};\ \{\mathbf{D}_k \in \mathbb{Z}_{q_s}^{n\times \bar{m}}\}_{k=1}^4;\ \mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell} \in \mathbb{Z}_{q_s}^{n\times m_s}; \\ \mathbf{F} \in \mathbb{Z}_p^{m\times\bar{m}};\ \mathbf{u} \in \mathbb{Z}_{q_s}^n;\ \mathbf{P}_0, \mathbf{P}_1 \in \{0,1\}^{m\times m};\ H_{\mathrm{FRD}}(R) \in \mathbb{Z}_p^{m\times m};\ \mathbf{y}_S, \mathbf{y}_T \in \mathbb{Z}_p^m. \end{cases}$$

The prover's goal is to prove in zero-knowledge the possession of

$$\begin{cases} \mathbf{v}_1, \mathbf{v}_2 \in [-\beta, \beta]^{m_s};\ \ \mathbf{r} \in [-\beta, \beta]^{2m_s};\ \ \tilde{\mathbf{w}} \in \{0,1\}^{m_s/2}; \\ \mathbf{e}_u,\ \tilde{\mathbf{k}}_0,\ \tilde{\mathbf{k}}_1,\ \tilde{\mathbf{t}} \in \{0,1\}^{\bar{m}};\ \ \mathbf{y}'_T \in \mathbb{Z}_p^m;\ \ \mathbf{k} \in \mathbb{Z}_q^m; \\ \mathbf{k}_0 = \mathbf{H}_{m,q-1} \cdot \tilde{\mathbf{k}}_0 \in \mathbb{Z}_q^m;\ \ \mathbf{k}_1 = \mathbf{H}_{m,q-1} \cdot \tilde{\mathbf{k}}_1 \in \mathbb{Z}_q^m;\ \ \mathbf{t} = \mathbf{H}_{m,q-1} \cdot \tilde{\mathbf{t}} \in \mathbb{Z}_q^m; \\ \tau[1] \ldots \tau[\ell] \in \{0,1\}^{\ell};\ \ J[1] \ldots J[L] \in \{0,1\}^{L}, \end{cases}$$

such that the following equations hold:

$$\mathbf{A}\cdot\mathbf{v}_1+\mathbf{A}_0\cdot\mathbf{v}_2+\sum_{j=1}^{\ell}\mathbf{A}_j\cdot(\tau[j]\cdot\mathbf{v}_2)-\mathbf{D}\cdot\tilde{\mathbf{w}}=\mathbf{u}\in\mathbb{Z}_{q_s}^n, \tag{23}$$

$$\mathbf{D}_0\cdot\mathbf{r}+\mathbf{D}_1\cdot\mathbf{e}_u+\mathbf{D}_2\cdot\tilde{\mathbf{t}}+\mathbf{D}_3\cdot\tilde{\mathbf{k}}_0+\mathbf{D}_4\cdot\tilde{\mathbf{k}}_1-\mathbf{H}_{n,q_s-1}\cdot\tilde{\mathbf{w}}=\mathbf{0}\in\mathbb{Z}_{q_s}^n, \tag{24}$$

$$\mathbf{k}=\mathbf{k}_0+\mathbf{k}_1\in\mathbb{Z}_q^m;\quad \mathbf{y}_S=\Big\lfloor\prod_{i=1}^{L}\mathbf{P}_{J[L+1-i]}\cdot\mathbf{k}\Big\rfloor_p\in\mathbb{Z}_p^m, \tag{25}$$

$$\mathbf{y}'_T=\Big\lfloor\prod_{i=1}^{L}\mathbf{P}_{J[L+1-i]}\cdot\mathbf{t}\Big\rfloor_p\in\mathbb{Z}_p^m,\quad \mathbf{y}_T=\mathbf{F}\cdot\mathbf{e}_u+H_{\mathrm{FRD}}(R)\cdot\mathbf{y}'_T\ \in\mathbb{Z}_p^m. \tag{26}$$

Our strategy is to reduce the above statement to an instance of the abstraction in Section 4.1. To this end, we will combine the zero-knowledge proofs of signatures from the Stern-like techniques of [68] and our techniques for the PRF layer from Section 4.3. Specifically, we let $q_1=q_s$, $q_2=q$, $q_3=pq$, $q_4=p$, and perform the following transformations.

Regarding the two equations of the signature relation in (23)-(24), we apply the following decompositions and/or extensions to the underlying secret vectors:

$$\begin{cases}\{\check{\mathbf{v}}_i=\mathsf{ThreeExt}\big(\mathsf{vdec}'_{m_s,\beta}(\mathbf{v}_i)\big)\in\mathsf{B}^3_{m_s\delta_\beta}\}_{i=1}^2;\quad\{\mathbf{c}_j=\mathsf{expand}(\tau[j],\check{\mathbf{v}}_2)\}_{j=1}^{\ell};\\ \check{\mathbf{r}}=\mathsf{ThreeExt}\big(\mathsf{vdec}'_{2m_s,\beta}(\mathbf{r})\big)\in\mathsf{B}^3_{2m_s\delta_\beta};\quad\widehat{\mathbf{w}}=\mathsf{TwoExt}\big(\tilde{\mathbf{w}}\big)\in\mathsf{B}^2_{m_s/2};\\ \widehat{\mathbf{e}}=\mathsf{TwoExt}(\mathbf{e}_u)\in\mathsf{B}^2_{\bar{m}};\quad\forall\alpha\in\{\mathbf{t},\mathbf{k}_0,\mathbf{k}_1\}:\widehat{\alpha}=\mathsf{TwoExt}\big(\tilde{\alpha}\big)\in\mathsf{B}^2_{\bar{m}}.\end{cases} \tag{27}$$

At the same time, we also transform the associated public matrices $\mathbf{A}$, $\{\mathbf{A}_j\}_{j=0}^{\ell}$, $\mathbf{D}$, $\{\mathbf{D}_j\}_{j=0}^{4}$, $\mathbf{H}_{n,q_s-1}$ accordingly, so that the equations are preserved. Next, we combine the vectors obtained in (27) into:

$$\mathbf{w}_1=\big(\check{\mathbf{v}}_1\,\|\,\check{\mathbf{v}}_2\,\|\,\mathbf{c}_1\,\|\ldots\|\,\mathbf{c}_\ell\,\|\,\check{\mathbf{r}}\,\|\,\widehat{\mathbf{w}}\,\|\,\widehat{\mathbf{e}}\,\|\,\widehat{\mathbf{t}}\,\|\,\widehat{\mathbf{k}}_0\,\|\,\widehat{\mathbf{k}}_1\big)\in\{-1,0,1\}^{d_1}, \tag{28}$$

where $d_1=6(\ell+2)m_s\delta_\beta+m_s+8\bar{m}$. We observe that the two equations can be unified into just one equation of the form $\mathbf{M}_1\cdot\mathbf{w}_1=\mathbf{u}_1 \bmod q_1$, where $\mathbf{M}_1\in\mathbb{Z}_{q_1}^{2n\times d_1}$ is built from public matrices, and $\mathbf{u}_1=(\mathbf{u}\,\|\,\mathbf{0})\in\mathbb{Z}_{q_1}^{2n}$.

We now consider equations in (25) and (26), which involve PRF evaluations. We note that, for all $\alpha\in\{\mathbf{t},\mathbf{k}_0,\mathbf{k}_1\}$ appearing in this layer, we have the connection

$$\alpha=\mathbf{H}_{m,q-1}\cdot\tilde{\alpha}=\widehat{\mathbf{H}}_{m,q-1}\cdot\widehat{\alpha},$$

where $\widehat{\alpha}$ is constructed in 27. To transform the equation $\mathbf{k}=\mathbf{k}_0+\mathbf{k}_1\in\mathbb{Z}_q^m$ in (25), we let $\widehat{\mathbf{k}}=\mathsf{TwoExt}\big(\mathsf{vdec}_{m,q-1}(\mathbf{k})\big)\in\mathsf{B}^2_{\bar{m}}$, and rewrite the equation as

$$\widehat{\mathbf{H}}_{m,q-1}\cdot\widehat{\mathbf{k}}_0+\widehat{\mathbf{H}}_{m,q-1}\cdot\widehat{\mathbf{k}}_1-\widehat{\mathbf{H}}_{m,q-1}\cdot\widehat{\mathbf{k}}=\mathbf{0}\bmod q.$$

Letting $\mathbf{M}_{k,2} = [\widehat{\mathbf{H}}_{m,q-1} \mid \widehat{\mathbf{H}}_{m,q-1} \mid -\widehat{\mathbf{H}}_{m,q-1}]$ and $\mathbf{u}_{k,2} = \mathbf{0}$, we have the equation $\mathbf{M}_{k,2} \cdot \mathbf{w}_{k,2} = \mathbf{u}_{k,2} \bmod q_2$, where:

$$\mathbf{w}_{k,2} = (\widehat{\mathbf{k}}_0 \parallel \widehat{\mathbf{k}}_1 \parallel \widehat{\mathbf{k}}). \tag{29}$$

The evaluation process of $\mathbf{y}_S$ in (25) is handled as in Section 4.3, resulting in equations $\mathbf{M}_{S,2} \cdot \mathbf{w}_{S,2} = \mathbf{u}_{S,2} \bmod q_2$, and $\mathbf{M}_{S,3} \cdot \mathbf{w}_{S,3} = \mathbf{u}_{S,3} \bmod q_3$, where

$$\mathbf{w}_{S,2} = \big(\mathbf{s}_{S,0} \parallel \widehat{\mathbf{x}}_{S,1} \parallel \cdots \parallel \mathbf{s}_{S,L-1} \parallel \widehat{\mathbf{x}}_{S,L}\big); \quad \mathbf{w}_{S,3} = \big(\widehat{\mathbf{x}}_{S,L} \parallel \widehat{\mathbf{z}}_S\big), \tag{30}$$

satisfy $\{\widehat{\mathbf{x}}_{S,i}\}_{i=1}^{L}$, $\widehat{\mathbf{z}}_S \in \mathsf{B}^2_{\bar{m}}$ and

$$\mathbf{s}_{S,0} = \mathsf{expand}(J[1], \widehat{\mathbf{k}}); \qquad \{\mathbf{s}_{S,i-1} = \mathsf{expand}(J[i], \widehat{\mathbf{x}}_{S,i-1})\}_{i=2}^{L}.$$

Regarding the evaluation of $\mathbf{y}'_T$ in (26), equations appearing in the iteration step can also be unified into one of the form $\mathbf{M}_{T,2} \cdot \mathbf{w}_{T,2} = \mathbf{u}_{T,2} \bmod q_2$, where

$$\mathbf{w}_{T,2} = \big(\mathbf{s}_{T,0} \| \widehat{\mathbf{x}}_{T,1} \| \cdots \| \mathbf{s}_{T,L-1} \| \widehat{\mathbf{x}}_{T,L}\big), \tag{31}$$

satisfy

$$\{\widehat{\mathbf{x}}_{T,i} \in \mathsf{B}^2_{\bar{m}}\}_{i=1}^{L}; \quad \mathbf{s}_{T,0} = \mathsf{expand}(J[1], \widehat{\mathbf{t}}); \quad \{\mathbf{s}_{T,i-1} = \mathsf{expand}(J[i], \widehat{\mathbf{x}}_{T,i-1})\}_{i=2}^{L}.$$

Meanwhile, the rounding step is handled in a slightly different manner as the output $\mathbf{y}'_T \in \mathbb{Z}_p^m$ is hidden. Letting $\widehat{\mathbf{y}}'_T = \mathsf{TwoExt}\big(\mathsf{vdec}_{m,p-1}(\mathbf{y}'_T)\big) \in \mathsf{B}^2_{m\delta_{p-1}}$, we are presented with the equation

$$(p \cdot \widehat{\mathbf{H}}_{m,q-1}) \cdot \widehat{\mathbf{x}}_{T,L} - \widehat{\mathbf{H}}_{m,q-1} \cdot \widehat{\mathbf{z}}_T - (q \cdot \widehat{\mathbf{H}}_{m,p-1}) \cdot \widehat{\mathbf{y}}'_T = \mathbf{0} \bmod pq,$$

where $\widehat{\mathbf{z}}_T \in \mathsf{B}^2_{\bar{m}}$. This equation can be written as $\mathbf{M}_{T,3} \cdot \mathbf{w}_{T,3} = \mathbf{u}_{T,3} \bmod q_3$, where $\mathbf{M}_{T,3} = \big[p \cdot \widehat{\mathbf{H}}_{m,q-1} \mid -\widehat{\mathbf{H}}_{m,q-1} \mid -q \cdot \widehat{\mathbf{H}}_{m,p-1}\big]$, $\mathbf{u}_{T,3} = \mathbf{0}$, and

$$\mathbf{w}_{T,3} = (\widehat{\mathbf{x}}_{T,L} \parallel \widehat{\mathbf{z}}_T \parallel \widehat{\mathbf{y}}'_T). \tag{32}$$

Furthermore, we observe that, the three equations modulo q_2, as well as the two equations modulo q_3 we have obtained above can be unified as follows. Let

$$\mathbf{M}_2 = \begin{bmatrix} \mathbf{M}_{k,2} & & \\ & \mathbf{M}_{S,2} & \\ & & \mathbf{M}_{T,2} \end{bmatrix}; \ \mathbf{u}_2 = \begin{pmatrix} \mathbf{u}_{k,2} \\ \mathbf{u}_{S,2} \\ \mathbf{u}_{T,2} \end{pmatrix}; \ \mathbf{M}_3 = \begin{bmatrix} \mathbf{M}_{S,3} & \\ & \mathbf{M}_{T,3} \end{bmatrix}; \ \mathbf{u}_3 = \begin{pmatrix} \mathbf{u}_{S,3} \\ \mathbf{u}_{T,3} \end{pmatrix},$$

then we have $\mathbf{M}_2 \cdot \mathbf{w}_2 = \mathbf{u}_2 \bmod q_2$ and $\mathbf{M}_3 \cdot \mathbf{w}_3 = \mathbf{u}_3 \bmod q_3$, where

$$\mathbf{w}_2 = (\mathbf{w}_{k,2} \parallel \mathbf{w}_{S,2} \parallel \mathbf{w}_{T,2}) \in \{-1,0,1\}^{d_2}; \tag{33}$$

$$\mathbf{w}_3 = (\mathbf{w}_{S,3} \parallel \mathbf{w}_{T,3}) \in \{-1,0,1\}^{d_3}, \tag{34}$$

for $\mathbf{w}_{k,2}, \mathbf{w}_{S,2}, \mathbf{w}_{T,2}, \mathbf{w}_{S,3}, \mathbf{w}_{T,3}$ defined by (29)-(32), and for $d_2 = 6\bar{m}(2L+1)$, $d_3 = 8\bar{m} + 2m\delta_{p-1}$.

Now, the remaining equation in (26) can be written as:

$$\big[\mathbf{F} \mid \mathbf{0}^{m\times\bar{m}}\big]\cdot\widehat{\mathbf{e}} + \big(H_{\mathrm{FRD}}(R)\cdot\widehat{\mathbf{H}}_{m,p-1}\big)\cdot\widehat{\mathbf{y}}'_T = \mathbf{y}_T \bmod p,$$

where $\widehat{\mathbf{e}}$ and $\widehat{\mathbf{y}}'_T$ are as constructed earlier. We therefore obtain the equation $\mathbf{M}_4\cdot\mathbf{w}_4 = \mathbf{u}_4 \bmod q_4$, where $\mathbf{M}_4 = \big[H_{\mathrm{FRD}}(R)\cdot\widehat{\mathbf{H}}_{m,p-1} \mid \mathbf{F} \mid \mathbf{0}^{m\times\bar{m}}\big]$, $\mathbf{u}_4 = \mathbf{y}_T$ and, for $d_4 = 2\bar{m} + 2m\delta_{p-1}$,

$$\mathbf{w}_4 = (\widehat{\mathbf{y}}'_T \parallel \widehat{\mathbf{e}}) \in \{-1,0,1\}^{d_4}. \tag{35}$$

At this point, we have transformed all the considered equations into four equations $\{\mathbf{M}_i\cdot\mathbf{w}_i = \mathbf{u}_i \bmod q_i\}_{i=1}^{4}$. We then let $d = \sum_{i=1}^{4} d_i$ and, for $\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \mathbf{w}_4$ defined by (28), (33), (34), (35), respectively, let

$$\mathbf{w} = (\mathbf{w}_1 \| \mathbf{w}_2 \| \mathbf{w}_3 \| \mathbf{w}_4) \in \{-1,0,1\}^{d}. \tag{36}$$

Let us now specify the set VALID containing $\mathbf{w}$, the set $\mathcal{S}$ and the associated permutation $\varGamma_\phi$, satisfying conditions in 9.

Let VALID be the set of all vectors in $\{-1,0,1\}^d$ having the form (36) (which follows from (28)-(35)), whose block-vectors satisfy the following conditions:

$$\begin{cases} \check{\mathbf{v}}_1, \check{\mathbf{v}}_2 \in \mathsf{B}^3_{m_s\delta_\beta}; \quad \check{\mathbf{r}} \in \mathsf{B}^3_{2m_s\delta_\beta}; \quad \widehat{\mathbf{w}} \in \mathsf{B}^2_{m_s/2}; \quad \mathbf{y}'_T \in \mathsf{B}^2_{m\delta_{p-1}}; \\ \widehat{\mathbf{e}},\ \widehat{\mathbf{t}},\ \widehat{\mathbf{k}}_0,\ \widehat{\mathbf{k}}_1,\ \widehat{\mathbf{k}},\ \widehat{\mathbf{x}}_{S,1},\ldots,\widehat{\mathbf{x}}_{S,L},\ \widehat{\mathbf{x}}_{T,1},\ldots,\widehat{\mathbf{x}}_{T,L},\ \widehat{\mathbf{z}}_S,\ \widehat{\mathbf{z}}_T \in \mathsf{B}^2_{\bar{m}}; \\ \{\mathbf{c}_j = \mathsf{expand}(\tau[j], \check{\mathbf{v}}_2)\}_{j=1}^{\ell}; \quad \mathbf{s}_{S,0} = \mathsf{expand}(J[1], \widehat{\mathbf{k}}); \quad \mathbf{s}_{T,0} = \mathsf{expand}(J[1], \widehat{\mathbf{t}}); \\ \{\mathbf{s}_{S,i-1} = \mathsf{expand}(J[i], \widehat{\mathbf{x}}_{S,i-1}),\ \mathbf{s}_{T,i-1} = \mathsf{expand}(J[i], \widehat{\mathbf{x}}_{T,i-1})\}_{i=2}^{L}, \end{cases}$$

for some $\tau[1]\ldots\tau[\ell] \in \{0,1\}^\ell$ and some $J[1]\ldots J[L] \in \{0,1\}^L$. By construction, our vector $\mathbf{w}$ belongs to this tailored set VALID.

Now, we define

$$\mathcal{S} := (\mathcal{S}_{3m_s\delta_\beta})^2 \times \mathcal{S}_{6m_s\delta_\beta} \times \mathcal{S}_{m_s} \times \mathcal{S}_{2m\delta_{p-1}} \times (\mathcal{S}_{2\bar{m}})^{2L+7} \times \{0,1\}^\ell \times \{0,1\}^L.$$

Then, for any element $\phi \in \mathcal{S}$ of the form

$$\begin{aligned}\phi = \ &\big(\phi_{\check{\mathbf{v}}_1},\ \phi_{\check{\mathbf{v}}_2},\ \phi_{\check{\mathbf{r}}},\ \phi_{\widehat{\mathbf{w}}},\ \phi_{\mathbf{y}'_T},\ \phi_{\widehat{\mathbf{e}}},\ \phi_{\widehat{\mathbf{t}}},\ \phi_{\widehat{\mathbf{k}}_0},\ \phi_{\widehat{\mathbf{k}}_1},\ \phi_{\widehat{\mathbf{k}}},\ \phi_{\widehat{\mathbf{x}}_{S,1}},\ \ldots,\ \phi_{\widehat{\mathbf{x}}_{S,L}},\\ &\quad \phi_{\widehat{\mathbf{x}}_{T,1}},\ \ldots,\ \phi_{\widehat{\mathbf{x}}_{T,L}},\ \phi_{\widehat{\mathbf{z}}_S},\ \phi_{\widehat{\mathbf{z}}_T},\ a[1]\ldots a[\ell],\ b[1]\ldots b[L]\big),\end{aligned}$$

let $\varGamma_\phi$ be the permutation that, on input vector $\mathbf{w} \in \mathbb{Z}^d$ of the form (18) (which is implied by (28)-(35)), it transforms the block-vectors of $\mathbf{w}$ as follows:

- Apply permutation ϕ_α to block α, for all

  $$\alpha \in \big\{\check{\mathbf{v}}_1,\ \check{\mathbf{v}}_2,\ \check{\mathbf{r}},\ \widehat{\mathbf{w}},\ \mathbf{y}'_T,\ \widehat{\mathbf{e}},\ \widehat{\mathbf{t}},\ \widehat{\mathbf{k}}_0,\ \widehat{\mathbf{k}}_1,\ \widehat{\mathbf{k}},\ \widehat{\mathbf{x}}_{S,1},\ldots,\widehat{\mathbf{x}}_{S,L},\ \widehat{\mathbf{x}}_{T,1},\ldots,\widehat{\mathbf{x}}_{T,L},\ \widehat{\mathbf{z}}_S,\ \widehat{\mathbf{z}}_T\big\}.$$

- For $j \in [\ell]$, apply permutation $T_{a[j],\phi_{\check{\mathbf{v}}_2}}$ to block $\mathbf{c}_j$.

- Apply permutation $T_{b[1],\phi_{\widehat{\mathbf{k}}}}$ to block $\mathbf{s}_{S,0}$, and $T_{b[1],\phi_{\widehat{\mathbf{t}}}}$ to block $\mathbf{s}_{T,0}$.
- For $i \in [2, L]$, apply permutation $T_{b[i],\phi_{\widehat{\mathbf{x}}_{S,i-1}}}$ to block $\mathbf{s}_{S,i-1}$, and permutation $T_{b[i],\phi_{\widehat{\mathbf{x}}_{T,i-1}}}$ to block $\mathbf{s}_{T,i-1}$.

It can be checked that, we have $\mathbf{w} \in \mathsf{VALID}$ if and only if $\mathit{\Gamma}_\phi(\mathbf{w}) \in \mathsf{VALID}$, thanks to the equivalences (3), (4), (5) from Section 3. Furthermore, if $\phi \leftarrow U(\mathcal{S})$, then $\mathit{\Gamma}_\phi(\mathbf{w})$ is uniform in VALID. In other words, the conditions in 9 are satisfied.

Given the above transformations and specifications, we can run the abstract protocol of Figure 1 to prove knowledge of $\mathbf{w} = (\mathbf{w}_1 \| \mathbf{w}_2 \| \mathbf{w}_3 \mathbf{w}_4) \in \mathsf{VALID}$ satisfying $\{\mathbf{M}_i \cdot \mathbf{w}_i = \mathbf{u}_i \bmod q_i\}_{i\in[4]}$, where public matrices/vectors $\{\mathbf{M}_i, \mathbf{u}_i\}_{i\in[4]}$ are as constructed above. As a result, we obtain a statistical zero-knowledge argument of knowledge for the considered statement.

Each round of the protocol has communication cost $\mathcal{O}(\sum_{i=1}^{4} d_i \cdot \log q_i)$. For the parameters in Section 5.1, this cost is of order $\widetilde{\mathcal{O}}(L \cdot \lambda + \lambda^2)$. In the Spend algorithm, the protocol is repeated $\kappa = \omega(\log \lambda)$ to achieve negligible soundness error. The global communication cost is $\widetilde{\mathcal{O}}(L \cdot \lambda + \lambda^2) \cdot \omega(\log \lambda) = \widetilde{\mathcal{O}}(L \cdot \lambda + \lambda^2)$.

6 Security

We now state the security results for which proofs are available in the full version of the paper.

Theorem 2. *The scheme guarantees balance under the* SIS *assumption in the random oracle model.*

Theorem 3 shows that, under the SIS assumption and assuming the collision-resistance of H_0, double-spenders can always be identified by the bank. Analogously to the security proof of Camenisch *et al.* [22] which relies on a similar feature of the Dodis-Yampolskiy PRF [44], the proof uses some range-disjointness property of the underlying small-domain PRF: namely, two functions keyed by independent keys should have disjoint ranges with high probability. In the full paper, we prove this property unconditionally for the BLMR PRF [17].

Theorem 3. *If H_0 is a collision-resistant hash function and H is modeled as a random oracle, the scheme guarantees the identification of double spenders under the* SIS *assumption.*

Theorem 4. *The scheme provides strong exculpability under the* SIS *assumption in the random oracle model.*

Theorem 5. *The scheme provides anonymity under the* LWE *assumption in the random oracle model.*

Our scheme can be modified so as to use the more efficient LWR-based PRF based on the GGM technique. This allows significantly improving the choice of parameters at the expense of a longer description and a more complex proof for the identification property. The reason is that, in the GGM-based PRF, the range disjointness property (for small domains) does not appear to be provable in the statistical sense. This can be addressed by relying on the pseudo-randomness of the function, as in the security proof of Belenkiy *et al.* [10]. Relying on the pseudo-randomness is perhaps counter-intuitive since the adversary knows the PRF seed in the proof of the identification property. Nevertheless, the reduction still works as in [10, Appendix F] when the domain has polynomial size.

Acknowledgements

Part of this research was funded by Singapore Ministry of Education under Research Grant MOE2016-T2-2-014(S), by the French ANR ALAMBIC project (ANR-16-CE39-0006) and by BPI-France in the context of the national project RISQ (P141580).

References

1. S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *Eurocrypt*, 2010.
2. M. Ajtai. Generating hard instances of the short basis problem. In *ICALP*, volume 1644, pages 1–9, 1999.
3. J. Alwen, S. Krenn, K. Pietrzak, and D. Wichs. Learning with rounding revisited – new reduction, properties and applications. In *Crypto 2013*.
4. J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. In *STACS 2009*.
5. M. H. Au, Q. Wu, W. Susilo, and Y. Mu. Compact e-cash from bounded accumulator. In *CT-RSA*, 2007.
6. W. Banaszczyk. New bounds in some transference theorems in the geometry of number. 296, 1993.
7. A. Banerjee and C. Peikert. New and improved key-homomorphic pseudo-random functions. In *Crypto*, 2014.
8. A. Banerjee, C. Peikert, and A. Rosen. Pseudo-random functions and lattices. In *Eurocrypt*, 2012.
9. C. Baum, I. Damgård, K.-G. Larsen, and M. Nielsen. How to prove knowledge of small secrets. In *Crypto 2016*.
10. M. Belenkiy, M. Chase, M. Kohlweiss, and A. Lysyanskaya. Compact e-cash and simulatable vrfs revisited. In *Pairing*, 2009.
11. M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *ACM-CCS'93*, 1993.
12. E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *IEEE S&P*, 2014.
13. F. Benhamouda, J. Camenisch, S. Krenn, V. Lyubashevsky, and G. Neven. Better zero-knowledge proofs for lattice encryption and their application to group signatures. In *Asiacrypt*, 2014.
14. F. Benhamouda, S. Krenn, V. Lyubashevsky, and K. Pietrzak. Efficient Zero-Knowledge Proofs for Commitments from Learning With Errors over Rings. In *ESORICS*, 2015.

15. A. Bogdanov, S. Guo, D. Masny, S. Richelson, and A. Rosen. On the hardness of leanring with rounding over small modulus. In *TCC 2016*.
16. F. Böhl, D. Hofheinz, T. Jager, J. Koch, and C. Striecks. Confined guessing: New signatures from standard assumptions. *Journal of Cryptology*, 28(1), 2015.
17. D. Boneh, K. Lewi, H. Montgomery, and A. Raghunathan. Key-homomorphic prfs and their applications. In *Crypto*, 2013.
18. X. Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In *PKC 2010*, 2010.
19. Z. Brakerski and Y. T. Kalai. A Framework for Efficient Signatures, Ring Signatures and Identity Based Encryption in the Standard Model. *IACR Cryptology ePrint Archive*, 2010:86, 2010.
20. Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. On the classical hardness of learning with errors. In *STOC*, 2013.
21. J. Camenisch, S. Hohenberger, M. Kohlweiss, A. Lysyanskaya, and M. Meyerovich. How to win the clone wars: Efficient periodic n-times anonymous authentication. In *ACM-CCS*, 2006.
22. J. Camenisch, S. Hohenberger, and A. Lysyanskaya. Compact e-cash. In *Eurocrypt 2005*.
23. J. Camenisch, S. Hohenberger, and A. Lysyanskaya. Balancing accountability and privacy using e-cash. In *SCN*, 2005.
24. J. Camenisch and A. Lehmann. (un)linkable pseudonyms for governmental databases. In *ACM-CCS*, 2015.
25. J. Camenisch and A. Lysyanskaya. A signature scheme with efficient protocols. In *SCN*, 2002.
26. J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *Crypto*, 2004.
27. J. Camenisch, A. Lysyanskaya, and M. Meyerovich. Endorsed e-cash. In *IEEE S&P*, 2007.
28. S. Canard and A. Gouget. Divisible e-cash systems can be truly anonymous. In *Eurocrypt*, 2007.
29. S. Canard, D. Pointcheval, O. Sanders, and J. Traoré. Divisible e-cash made practical. In *PKC*, 2015.
30. S. Canard, D. Pointcheval, O. Sanders, and J. Traoré. Scalable divisible e-cash. In *ACNS*, 2015.
31. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *Eurocrypt*, 2010.
32. M. Chase, C. Ganesh, and P. Mohassel. Efficient zero-knowledge proof of algebraic and non-algebraic statements with applications to privacy preserving credentials. In *Crypto*, 2016.
33. D. Chaum. Blind signatures for untraceable payments. In *Crypto*, 1982.
34. D. Chaum. Blind signature system. In *Crypto*, 1983.
35. D. Chaum. Security without identification: Transactions ssystem to make big brother obsolete. *Comm. of the ACM*, 28(10), 1985.
36. D. Chaum. On-line cash checks. In *Eurocrypt*, 1989.
37. D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash. In *Crypto*, 1988.
38. D. Chaum and T. Pedersen. Transferred cash grows in size. In *Eurocrypt*, 1992.
39. A. Chiesa, M. Green, J. Liu, P. Miao, I. Miers, and P. Mishra. Decentralized anonymous micropayments. In *Eurocrypt*, 2017.
40. S. Coull, M. Green, and S. Hohenberger. Controlling access to an oblivious database using stateful anonymous credentials. In *PKC*, 2009.

41. R. Cramer and I. Damgård. On the amortized complexity of zero-knowledge protocols. In *Crypto*, 2009.
42. R. Cramer, I. Damgård, C. Xing, and C. Yuan. Amortized complexity of zero-knowledge proofs revisited: Achieving linear soundness slack. In *Eurocrypt 2017*.
43. R. del Pino and V. Lyubashevsky. Amortization with fewer equations for proving knowledge of small secrets. Cryptology ePrint Archive: Report 2017/280, 2017.
44. Y. Dodis and A. Yampolskiy. A verifiable random function with short proofs and keys. In *PKC*, 2005.
45. N. Döttling and D. Schröder. Efficient pseudorandom functions via on-the-fly adaptation. In *Crypto*, 2015.
46. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Crypto*, 1987.
47. M. Franklin and M. Yung. Secure and efficient off-line digital money. In *ICALP*, 1993.
48. T. Frederiksen, J.-B. Nielsen, and C. Orlandi. Privacy-free garbled circuits with applications to efficient zero-knowledge. In *Eurocrypt*, 2015.
49. M. Freedman, Y. Ishai, B. Pinkas, and O. Reingold. Keyword search and oblivious pseudorandom functions. In *TCC*, 2005.
50. E. Fujisaki and K. Suzuki. Traceable ring signature. In *PKC*, 2007.
51. C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, 2009.
52. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, 2008.
53. I. Giacomelli, J. Madsen, and C. Orlandi. Zkboo: Faster zero-knowledge for boolean circuits. In *USENIX Security Symposium*, 2016.
54. O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. In *J. of ACM*, volume 33, 1986.
55. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. In *STOC*, 1985.
56. S. D. Gordon, J. Katz, and V. Vaikuntanathan. A group signature scheme from lattice assumptions. In *Asiacrypt*, 2010.
57. M. Green and I. Miers. Bolt: Anonymous payment channels for decentralized currencies. Cryptology ePrint Archive: Report 2016/701, 2016.
58. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Eurocrypt 2008*.
59. S. Hohenberger, S. Myers, R. Pass, and a. shelat. ANONIZE: A large-scale anonymous survey system. In *IEEE S&P 2014*, 2014.
60. Y. Ishai, E. Kushilevitz, R. Ostrovksy, and A. Sahai. Zero-knowledge from secure multiparty computation. In *STOC*, 2007.
61. S. Jarecki, A. Kiayias, and H. Krawczyk. Round-optimal password-protected secret sharing and T-PAKE in the password-only model. In *Asiacrypt*, 2014.
62. S. Jarecki and X. Liu. Efficient oblivious pseudorandom function with applications to adaptive OT and secure computation of set intersection. In *TCC*, 2009.
63. M. Jawurek, F. Kerschbaum, and C. Orlandi. Zero-knowledge using garbled circuits: how to prove non-algebraic statements efficiently. In *ACM-CCS*, 2013.
64. A. Kawachi, K. Tanaka, and K. Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *Asiacrypt*, 2008.
65. F. Laguillaumie, A. Langlois, B. Libert, and D. Stehlé. Lattice-based group signatures with logarithmic signature size. In *Asiacrypt*, 2013.
66. A. Langlois, S. Ling, K. Nguyen, and H. Wang. Lattice-based group signature scheme with verifier-local revocation. In *PKC 2014*.

67. 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*.
68. 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.
69. 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.
70. S. Ling, K. Nguyen, D. Stehlé, and H. Wang. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In *PKC 2013*, 2013.
71. S. Ling, K. Nguyen, and H. Wang. Group signatures from lattices: Simpler, tighter, shorter, ring-based. In *PKC*. 2015.
72. H. Lipmaa, N. Asokan, and V. Niemi. Secure vickrey auctions without threshold trust. In *Financial Cryptography*, 2002.
73. V. Lyubashevsky. Lattice-Based Identification Schemes Secure Under Active Attacks. In *PKC*, 2008.
74. S. Micali, M. Rabin, and S. Vadhan. Verifiable random functions. In *FOCS*, 1999.
75. S. Micali and R. Sidney. A simple method for generating and sharing pseudo-random functions. In *Crypto*, 1995.
76. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Eurocrypt*, 2012.
77. D. Micciancio and S. Vadhan. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In *Crypto*, 2003.
78. I. Miers, C. Garman, M. Green, and A. Rubin. Zerocoin: Anonymous distributed e-cash from bitcoin. In *IEEE S&P*, 2013.
79. S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. `www.bitcoin.org`.
80. M. Naor, B. Pinkas, and O. Reingold. Distributed pseudo-random functions and KDCs. In *Eurocrypt*, 1999.
81. P. Nguyen, J. Zhang, and Z. Zhang. Simpler efficient group signatures from lattices. In *PKC*, 2015.
82. K. Ohta and T. Okamoto. Universal electronic cash. In *Crypto*, 1991.
83. T. Okamoto. An efficient divisible electronic cash scheme. In *Crypto*, 1995.
84. C. Peikert and V. Vaikuntanathan. Non-interactive statistical zero-knowledge proofs for lattice problems. In *Crypto*, 2008.
85. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, 2005.
86. J. Stern. A new paradigm for public key identification. *Information Theory, IEEE Transactions on*, 42(6), 1996.
87. Y. Tsiounis. Efficient electronic cash: New notions and techniques. PhD thesis, Northeastern University, 1997.
88. S. von Solms and D. Naccache. On blind signatures and perfect crimes. *Computers & Security*, 11, 1992.
89. X. Xie, R. Xue, and M. Wang. Zero knowledge proofs from ring-LWE. In *CANS*, 2013.