Lattice-Based Simulatable VRFs: Challenges and Future Directions

Carlo Brunetta*, Bei Liang, and Aikaterini Mitrokotsa
Chalmers University of Technology, Gothenburg, Sweden
{brunetta, lbei, aikmitr}@chalmers.se

Abstract

Lattice-based cryptography is evolving rapidly and is often employed to design cryptographic primitives that hold a great promise to be post-quantum resistant and can be employed in multiple application settings such as: e-cash, unique digital signatures, non-interactive lottery and others. In such application scenarios, a user is often required to prove non-interactively the correct computation of a pseudo-random function $F_k(x)$ without revealing the secret key k used. Commitment schemes are also useful in application settings requiring to commit to a chosen but secret value that could be revealed later. In this short paper, we provide our insights on constructing a *lattice-based simulatable verifiable random function (sVRF)* using non interactive zero knowledge arguments and dual-mode commitment schemes and we point out the main challenges that need to be addressed in order to achieve it.

Keywords: Dual-Mode Commitment Scheme, Lattice-based Cryptography, Non Interactive Zero Knowledge Arguments, Pseudo Random Functions, Verifiable Random Functions

1 Introduction

Zero-knowledge (ZK) proofs [15] are employed to prove the knowledge of secret information while preserving the provers' privacy with respect to an NP language. Depending on whether the zero-knowledge proof is performed interactively or not, we may have *interactive* or *non-interactive* protocols; while the latter are more efficient regarding communication costs.

Pseudo-random functions (PRFs) [11] are a very useful cryptographic primitive that is often employed in combination with *zero-knowledge* proofs in multiple application scenarios. Let us consider a general scenario: a prover $\mathscr{P}$ wants to prove to a verifier $\mathscr{V}$ the knowledge of a secret $\vec{w}$ and the correct computation of a PRF $F_{\vec{w}}$ on the input x, *i.e.,* $F_{\vec{w}}(x)$. A rather important question is:

How may $\mathscr{P}$ prove to $\mathscr{V}$ the correct evaluation of the PRF $F_{\vec{w}}(x)$ without leaking any information about $\vec{w}$, just by providing a proof π?

We consider the case where the communication between $\mathscr{P}$ and $\mathscr{V}$ should be **non**-interactive, *i.e.,* $\mathscr{P}$ needs to provide $\mathscr{V}$ all the necessary information to verify the correct computations in a single step.

The above stated question can be solved by employing a *verifiable random function* (VRF) [18]. A VRF is a PRF with two additional algorithms; one algorithm that is able to generate a proof π of the correct computation of the PRF $F_{\vec{w}}(x)$ as well as a *verification* algorithm.

Verifiable random functions have a broad range of applications where the verification of a pseudo-random value is required. For instance, VRFs are employed in non-interactive lottery systems used in micropayments [19], domain name security extensions (DNSSEC) [10, 22] as well as proof-of-stake blockchain protocols [13, 8]. For instance, recent papers [13, 8] use VRFs in *blockchain* consensus protocols *i.e.,* in order to either define a *fair and verifiable lottery* in which the winner will publish the next

Journal of Internet Services and Information Security (JISIS), volume: 8, number: 4 (November 2018), pp. 57-69
*Corresponding author: Department of Computer Science and Engineering, Chalmers University of Technology, 412 96, Gothenburg, Sweden, Tel: +46317721619

block, or as a way to generate a *"common and shared random string"* which can be seen as an equivalent of the CRS model.

Although algebraic pseudo-random functions and ZK proofs are well studied primitives, they have received limited attention in lattice settings; furthermore, to the best of our knowledge, *building lattice-based VRFs is an open problem*.

Lattice-based cryptographic primitives [1, 23], mainly rely on the *learning with errors* (LWE)[24] and the *short integer solution* (SIS)[20] problems; they are quite promising for providing post-quantum resistance guarantees, while also offering simpler arithmetic operations and thus, important efficiency guarantees.

Designing a lattice-based VRF is a challenging and currently open problem since it requires a non-interactive proof in the standard model. As a step towards this direction, in this short paper, we provide our insights on designing a lattice-based *simulatable VRF* (sVRF), originally introduced by Chase and Lysyanskaya [6]. Informally, an sVRF is a VRF defined in a public parameter model, such as the *common random string* (CRS) model, which implies the existence of honest common parameters on the top of the standard VRF system. More precisely, besides the usual algorithms in a VRF there is an additional parameter generation algorithm which takes as input the security parameters and output the public parametrs pp. Both the input domain and the output range of the sVRF depend on pp. Meanwhile, pp is included in the inputs for all the algorithms KeyGen, Eval, Prove and Verify. Moreover, except of the uniqueness and pseudorandomness properties, sVRFs should also satisfy *simulatability* which is a novel property making them different from VRFs. Simulatability states that there exists a simulator that is able to simulate the common parameters such that, corresponding to a verification key, for any x, y, it is possible to produce a proof π that $F(sk,x) = y$. The simulated transcription is required to be indistinguishable from the values computed from the parameters that are generated honestly. In this paper, we describe our insights on constructing an sVRF when relying on Libert *et al.*'s [15] method to prove zero-knowledge arguments for lattice-based PRFs. Furthermore, we describe the main challenges that need to be addressed in order to construct a lattice-based sVRF using this method.

1.1 A Roadmap to Lattice-based sVRFs

Chase and Lysyanskaya [6] provided a general construction of sVRFs from a perfectly binding computational hiding non-interactive commitment scheme and an unconditionally-sound multi-theorem NIZK for NP. Their main idea is to use a multi-theorem NIZK to generate the proof for a statement that the public verification key is a commitment of the secret key and the function value is the correct result on the input applied to the secret-keyed PRF, *i.e.,* $\mathsf{pk} = \mathsf{Com}(\mathsf{sk}; r) \wedge y = F(\mathsf{sk}, x)$. However, such solution is based on a general assumption; in order to propose a lattice-based sVRF, we should specify a lattice-based PRF scheme.

Fortunately, thanks to the very recent significant results of Boneh *et al.* [4] who proposed a LWE-based key homomorphic PRFs as well as Libert *et al.*'s [15] three round zero-knowledge arguments of correct evaluation for the LWE-based PRF Boneh *et al.* [4] w.r.t. committed keys and inputs, it is possible to obtain a non-interactive solution of $y = F(\mathsf{sk}, x)$ as the correct evaluation of a PRF for a secret input x and a committed key sk. These results could be potentially employed in order to construct a lattice-based sVRF as we explore in this paper.

Libert *et al.* have significant contributions [15, 14, 16] in the area of designing efficient zero-knowledge proofs for lattice-related language. For instance, Libert *et al.* [14] considered how to construct zero-knowledge arguments of knowledge of a secret matrix X and vectors $\vec{s}, \vec{e}$ such that for a public vector $\vec{b}$ it holds $\vec{b} = \vec{X} \cdot \vec{s} + \vec{e} \bmod q$. Libert *et al.* [16] also investigated in the lattice setting how to design zero-knowledge arguments for the statement that c_x, c_y and c_z are the commitments to the polynomial-size bit-strings x, y and z which are the binary representations of large integers X, Y, Z satisfying certain

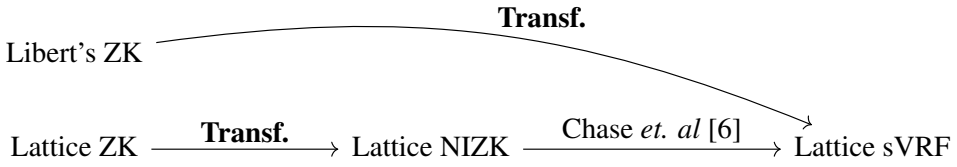

Figure 1: A roadmap to lattice-based sVRF. In **bold**, this paper's main research focus.

algebraic relations such as $Z = X + Y$ and $Z = X \cdot Y$.

In order to obtain zero-knowledge arguments for the correct evaluation of the key-homomorphic PRF [1] proposed by Boneh *et al.* [4] , Libert *et al.* [15] presented a useful abstraction of Stern's protocol [25] and they modified Boneh *et al.*'s lattice PRF [4] in order to efficiently prove the correct computation of the PRF value interactively, while providing zero-knowledge guarantees.

As stated in their paper [15], it is possible to obtain the first non-interactive lattice-based zero-knowledge protocol by directly applying the Fiat-Shamir transformation [9]. The main issue with this choice is that the Fiat-Shamir transformation is secure in the *Random Oracle Model* (ROM) which is against the original sVRF definition [6].

Thus, our main research objective is to find an appropriate transformation from ZK to NIZK, defined over lattices, not relying on the ROM. In Figure 1, we depict two different strategies in order to obtain a lattice-based sVRF: either by directly transforming Libert *et al.*'s ZK argument or by providing a different lattice-based ZK PRF proof system and applying a ZK to the NIZK transformation and then the Chase-Lysyanskaya's transformation from NIZK to sVRF.

2 Applying Lindell's Tranformation

In this section, we provide our findings on defining an sVRF based on Libert's ZK argument [15] and Lindell's transformation [17].

The latter [17] can be applied to any sigma-protocol and transform it into a corresponding NIZK protocol. In contrast to Fiat-Shamir's transformation [9], Lindell's transformation does not require the random oracle model. More precisely, in Lindell's transformation the *zero-knowledge* property holds in the *common reference string* (CRS) model, while in order to achieve *soundness*, the used hash function is modeled as a *non-programmable* random oracle [21]. In order to adopt Lindell's transformation an important requirement is that of a *dual-mode* commitment scheme.

The main concept of a commitment scheme is that it is possible to secretly fix some message m that is used in a protocol and in a second phase, open the commitment and therefore prove the correct knowledge or possession of the specific message m. Designing lattice-based commitment schemes has already received some attention in the literature [3, 2].

The *dual-mode* commitment represents the possibility to sample a statement in a language L via a bit b and use the commitment scheme in a *binding* way, *i.e.*, a commitment c can be decommitted in a *unique* message m, or in a *"trapdoor"* way, *i.e.,* that with some secret witness $\vec{w}$, it is possible to decommit c to any message m'.

Thus, the main property required for a dual-mode commitment scheme is that it is impossible to distinguish how the bit b is selected and therefore impossible to know if we are decommitting to the original message or we are using the trapdoor to decommit to an arbitrary message.

A *dual-mode commitment scheme* represents a specific type of commitment schemes that are equivalently defined by Catalano and Visconti as *hybrid commitment schemes* [5].

[1] Namely demonstrating knowledge of a committed secret key $\vec{k}$, a secret input $\vec{J}$ and an output $\vec{y}$ satisfying $\vec{y} = F_{\vec{k}}(\vec{J})$

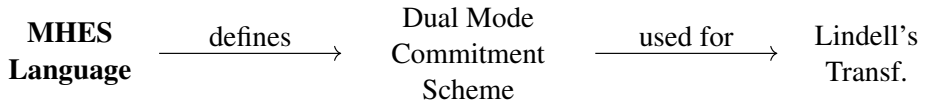

Figure 2: A roadmap to Lindell's transformation.

As described in [17], in order to define a dual-mode commitment scheme, Lindell requires a *membership-hard efficient-sampling language* defined as follows:

Definition 1 (Membership-hard with Efficient Sampling [17])**.** *Let L be a language. L is* membership-hard with efficient sampling *(MHES) if there exists a probabilistic polynomial-time sampler S_L such that for every probabilistic polynomial-time distinguisher D there exists a negligible function $\mu(\cdot)$ such that:*

$$|\mathsf{Pr}(D(S_L^x(0,1^n),1^n)=1)-\mathsf{Pr}(D(S_L(1,1^n),1^n)=1)|\leq\mu(n)$$

where $S_L(b,\cdot)$ is a sampler that returns an instance in the language L if $b=0$ and an instance not in the language L if $b=1$. S_L^x denotes only the instance without the witness.

In a nutshell, the MHES language L is a language in which it is hard to distinguish if an efficient sampling algorithm S_L sampled the statement x in the language L or not: it is hard to decide the membership of $x\in L$ but it is easy to sample x in the language (or not).

In summary, in order to build an sVRF while employing the Lindell's transformation, the main building blocks required are depicted in Figure 2.

By assuming the hardness of the *inhomogeneous short integer solution* (ISIS) problem, if we follow the idea and structure of the DDH language construction proposed by Lindell [17] in order to define the language L_{IS} of Eq. (1), the result is unfortunately not MHES for common lattice security parameters.

$$L_{\mathsf{IS}}:=\{(\vec{A},\vec{B},\vec{u},\vec{v})\mid\vec{A},\vec{B}\in{\mathbb{Z}_p}^{n\times m},\vec{w}\in\{0,1\}^m,\vec{u}=\vec{A}\vec{w},\vec{v}=\vec{B}\vec{w}\}. \tag{1}$$

This is the case since whenever we provide a statement not in the language $(\vec{A},\vec{B},\vec{u},\vec{v})\notin L_{\mathsf{IS}}$, it exists in fact a statement $(\vec{A},\vec{B},\vec{A}\vec{w}',\vec{A}\vec{w}')\in L_{\mathsf{IS}}$ in the language for some $\vec{w}'$. Thus it cannot be used to define a dual-mode commitment scheme mainly because the commitment scheme will not be perfectly binding, which is a necessary condition in order to use Lindell's transformation.

If we do assume that there exists a hard problem in the format that given $\vec{A}\in\mathsf{Dom}$ and $\vec{y}\in\mathsf{Rang}$, it is hard to find $\vec{z}\in\mathsf{PreD}$ such that $\vec{A}\vec{z}=\vec{y}$, then, on the ground of such an assumption, we can define our language as:

$$L:=\{(\vec{A},\vec{B},\vec{A}\vec{w},\vec{B}\vec{w})\mid\vec{A},\vec{B}\in\mathsf{Dom},\vec{w}\in\mathsf{PreD}\}. \tag{2}$$

The sampler S_L can be defined as: when $S_L(0,1^n)$ outputs a random tuple $(\vec{A},\vec{B},\vec{A}\vec{w},\vec{B}\vec{w})$, and $S_L(1,1^n)$ outputs a random tuple $(\vec{A},\vec{B},\vec{A}\vec{w},\vec{B}\vec{w}')$ of which $\vec{A}\vec{w}\neq\vec{A}\vec{w}'$ and $\vec{B}\vec{w}\neq\vec{B}\vec{w}'$. It is obvious to conclude that the language L defined as in Equation 2 is efficient sampling and membership-hard.

2.1 Dual-mode Commitment

In this section, we provide the formal definition of a dual-mode commitment scheme which is introduced by Lindell [17] and present a dual-mode commitment scheme based on the language L.

Definition 2 (Dual-mode commitment scheme [17])**.** *A dual-mode commitment scheme is a tuple of probabilistic polynomial-time algorithms (*GenCRS*,* Com*,* $\mathscr{S}_{\mathsf{com}}$*) such that:*

- $\mathsf{GenCRS}(1^\lambda)$: *outputs a common reference string, denoted* crs*,*

- $(\mathsf{GenCRS}, \mathsf{Com}, \mathsf{Decom}, \mathsf{ReceiverDecom})$: *When* $\mathsf{crs} \leftarrow \mathsf{GenCRS}(1^\lambda)$ *and* $m \in \{0,1\}^\lambda$*, the algorithm* $\mathsf{Com}_{\mathsf{crs}}(m;r)$ *with a random* r *is a non-interactive perfectly-binding commitment scheme with decommitment algorithm* Decom *and decommitment verification algorithm* ReceiverDecom*. (We require that* $\mathsf{ReceiverDecom}_{\mathsf{crs}}(\mathsf{Com}_{\mathsf{crs}}(m;r), \mathsf{Decom}_{\mathsf{crs}}(m;r)) = m$ *except with negligible probability.)*

- $(\mathsf{Com}, \mathscr{S}_{\mathsf{com}})$: *For every probabilistic polynomial-time adversary* $\mathscr{A}$ *and every polynomial* $p(\cdot)$*, the output of the following two experiments is computationally indistinguishable.*

$\mathsf{Real}_{\mathsf{Com},\mathscr{A}}(1^\lambda)$:	$\mathsf{Simulation}_{\mathscr{S}_{\mathsf{com}}}(1^\lambda)$:
1. $\mathsf{crs} \leftarrow \mathsf{GenCRS}(1^\lambda)$; $\vec{c}, \vec{d} \leftarrow \emptyset$	1. $\mathsf{crs} \leftarrow \mathscr{S}_{\mathsf{com}}(1^\lambda)$; $\vec{c}, \vec{d} \leftarrow \emptyset$
2. *For* $i = 1, \ldots, p(\lambda)$:	2. *For* $i = 1, \ldots, p(\lambda)$:
(a) $m_i \leftarrow \mathscr{A}(\mathsf{crs}, \vec{c}, \vec{d})$	(a) $c_i \leftarrow \mathscr{S}_{\mathsf{com}}$
(b) $c_i = \mathsf{Com}_{\mathsf{crs}}(m_i; r_i)$ *for* $r_i \in \{0,1\}^{\mathsf{poly}(\lambda)}$	(b) $m_i \leftarrow \mathscr{A}(\mathsf{crs}, \vec{c}, \vec{d})$
(c) $d_i = \mathsf{Decom}_{\mathsf{crs}}(m_i; r_i)$	(c) $d_i \leftarrow \mathscr{S}_{\mathsf{com}}(m_i)$
(d) *Set* $\vec{c} = c_1, \ldots, c_i$ *and* $\vec{d} = d_1, \ldots, d_i$	(d) *Set* $\vec{c} = c_1, \ldots, c_i$ *and* $\vec{d} = d_1, \ldots, d_i$
3. *Output* $\mathscr{A}(\mathsf{crs}, m_1, \ldots, m_{p(\lambda)}, \vec{c}, \vec{d})$	3. *Output* $\mathscr{A}(\mathsf{crs}, m_1, \ldots, m_{p(\lambda)}, \vec{c}, \vec{d})$

Below we describe an instantiation of a dual-mode commitment scheme.

Protocol 1 (Instantiation of Dual-Mode Commitment)**.**

- Regular CRS generation: $\vec{A}, \vec{B} \leftarrow$ *Dom*, $\vec{\vec{w}}_1, \vec{\vec{w}}_2 \leftarrow_R$ *PreD, and compute* $\vec{w}_1 = \vec{A}\vec{\vec{w}}_1$ *and* $\vec{w}_2 = \vec{B}\vec{\vec{w}}_2$*. The* CRS *is* $(\vec{A}, \vec{B}, \vec{w}_1, \vec{w}_2)$.

- Alternative CRS generation (equivocal): *As above, except of the fact that we also choose a single* $\vec{\vec{w}} \leftarrow_R$ *PreD and compute* $\vec{w}_1 = \vec{A}\vec{\vec{w}}$ *and* $\vec{w}_2 = \vec{B}\vec{\vec{w}}$.

- Commitment: *To commit to a bit* $e \in \{0,1\}$*, choose a random* $\vec{z} \leftarrow_R$ *PreD and compute* $\vec{y}_1 = \vec{A}\vec{z} - e\vec{w}_1$, $\vec{y}_2 = \vec{B}\vec{z} - e\vec{w}_2$*. The commitment is* $c = (\vec{y}_1, \vec{y}_2)$.

- Decommitment: *To decommit to* $c = (\vec{y}_1, \vec{y}_2)$*, provide* $(e, \vec{z})$.

- Receiver decommitment: *The receiver outputs* e *if* $\vec{A}\vec{z} = \vec{y}_1 + e\vec{w}_1$ *and* $\vec{B}\vec{z} = \vec{y}_2 + e\vec{w}_2$*. Otherwise, it outputs* $\perp$.

- Simulator $\mathscr{S}_{\mathsf{com}}$*:*

  1. *Run the sampler* S_L *for the language* L *as equation (2) with input* $(0, 1^\lambda)$*:* i.e.,

  $$(\vec{A}, \vec{B}, \vec{A}\vec{\vec{w}}, \vec{B}\vec{\vec{w}}, \vec{\vec{w}}) \leftarrow S_L(0, 1^\lambda)$$

  and set the CRS *as* $(\vec{A}, \vec{B}, \vec{A}\vec{\vec{w}}, \vec{B}\vec{\vec{w}})$*. Then,* $\mathscr{S}_{\mathsf{com}}$ *randomly samples* $\vec{\vec{y}} \leftarrow_R$ *PreD and computes* $\vec{y}_1 = \vec{A}\vec{\vec{y}}$ *and* $\vec{y}_2 = \vec{B}\vec{\vec{y}}$*. Set* $c = (\vec{y}_1, \vec{y}_2)$.
  2. *For a bit* $e \in \{0,1\}$*,* $\mathscr{S}_{\mathsf{com}}$ *computes* $\vec{z} = \vec{\vec{y}} + e\vec{\vec{w}}$*, and outputs the decommitment* $(e, \vec{z})$.

2.2 A Non-Interactive Zero-Knowledge Argument for a Lattice-Based PRF

In this subsection, we provide a non-interactive zero-knowledge argument for the correct evaluation of the lattice-based PRF proposed by Boneh *et al.* [4]. We construct non-interactive zero-knowledge arguments of knowledge of a committed seed $\vec{k}$, a secret input $\vec{J}$ and an output $\vec{y}$ satisfying $\vec{y} = F_{\vec{k}}(\vec{J})$. We describe such arguments for the key-homomorphic PRF of Boneh *et al.* [4] and the PRF obtained by applying the Goldreich-Goldwasser-Micali (GGM) [11] paradigm.

Recently, Libert *et al.* [15] have proposed zero-knowledge arguments for statements for which the given value $\vec{y} = \lfloor \prod_{i=1}^{L} \vec{P}_{\vec{J}[L+1-i]} \cdot \vec{k} \rfloor_p \in \mathbb{Z}_p^m$ is the correct evaluation for a committed seed $\vec{k} \in \mathbb{Z}_p^m$ and a secret input $\vec{J}[1] \ldots \vec{J}[L] \in \{0,1\}^L$, where $\vec{P}_0, \vec{P}_1 \in \{0,1\}^{m \times m}$ are public binary matrices, without revealing neither $\vec{k}$ nor $\vec{J}[1] \ldots \vec{J}[L]$. More precisely, they have used Stern's protocol [25], which is adapted to handle correlated witnesses across relations modulo distinct integers.

An added ingredient to our recipe is Lindell's transformation [17]: a Fiat-Shamir type transformation from sigma protocols to non-interactive zero knowledge argument; which employs a commitment scheme in the CRS model with the property that it is perfectly binding given the correctly constructed CRS, but it is equivocal to a simulator who generates the CRS in an alternative but indistinguishable way. In other words, the simulator can generate the CRS so that it looks like a real one, but a commitment can be decommitted to any value. In order to use Lindell's transformation in our context, a *lattice based dual-mode commitment scheme* is necessary.

In our Protocol 1, we show a concrete instantiation of a dual-mode commitment scheme. Following this, we can apply Lindell's transformation [17] on Libert's abstract sigma-protocol [15]. In Libert's paper [15], there is the precise translation from the Boneh's PRF [4] to the abstract protocol's hypothesis which are extracted and summarized in Section 3. To avoid heavy notation, we will use just the general and abstract construction since the specific instantiation for the PRF is proved to be correct and implementable by Libert *et al.* [15].

Definition 3 (The abstract statement [15])**.** *Let $n_i, d_i \geq n_i$ be positive integers. Let $d = \sum_{i=1}^{N} d_i$. Suppose* VALID $\subseteq \{-1,0,1\}^d$ *and $\mathscr{S}$ a finite set such that for any $\phi \in \mathscr{S}$ it is possible to associate to a permutation Γ_ϕ of d elements such that:*

$$\begin{cases} \vec{w} \in \text{VALID} \iff \Gamma_\phi(\vec{w}) \in \text{VALID} \\ \textit{If } \vec{w} \in \text{VALID} \wedge \phi \textit{ uniform in } \mathscr{S} \implies \Gamma_\phi(\vec{w}) \textit{ uniform in } \text{VALID} \end{cases}$$

Let us consider public matrices $\vec{M} := \{\vec{M}_i \in \mathbb{Z}_{q_i}{}^{n \times d_i}\}_{i \in [1..N]}$ and vectors $\vec{u} := \vec{u}_i \in \mathbb{Z}_{q_i}^{n_i}$, the prover argues in zero-knowledge the possession of integer vectors $\vec{w} := \{\vec{w} \in \{-1,0,1\}^{d_i}\}_{i \in [1..N]}$ such that:

$$\begin{cases} \vec{w} = (\vec{w}_1 \| \vec{w}_2 \| \ldots \| \vec{w}_N) \in \text{VALID} \\ \forall\, i \in [1..N] \,.\, \vec{M}_i \cdot \vec{w}_i = \vec{u}_i \pmod{q_i} \end{cases}$$

The described tuple $(\vec{M}, \vec{u})$ defines a statement of which $\vec{w}$ is the witness.

The protocol makes use of a statistically hiding and computationally binding string commitment scheme such as the SIS-based commitment of [12]. Libert *et al.* have also shown that by assuming the commitment scheme $(\text{Com}, \text{Decom})$ to be a statistically hiding and computationally binding string commitment, then their protocol is a zero-knowledge argument of knowledge for the given statement with perfect completeness and soundness error $2/3$. Based on the three round interaction protocol in [15] and Lindell's transformation [17], by employing our lattice-based dual commitment scheme instantiated in Protocol 1, a non-interactive zero-knowledge argument for the correct evaluation of Boneh *et al.*'s lattice-based pseudo-random function [4] is yielded as follows:

Protocol 2. *Let* Com *be a statistically hiding and computationally binding string commitment scheme, for example the SIS-based commitment defined in [12]. Let* $(\text{DRegularCRS}, \text{DCom}, \text{DReceiverDecom}, \text{DDecom})$ *be our lattice-based dual-mode commitment scheme of Protocol 1 and let* $H_k : \{0,1\}^* \to \{1,2,3\}$ *a keyed hash function.*

Our ***NIZK argument protocol*** $(\text{GenCRS}, \text{Prove}, \text{Verify})$ *for the correct evaluation of Boneh* et al.*'s lattice-based PRF [4] is defined as the following three algorithms:*

- *Inputs: Let* $\vec{M} = \{\vec{M}_i \in \mathbb{Z}_{q_i}{}^{n_i \times d_i}\}_{i \in [N]}$ *be a set of matrices, for all* $i \in [1, \ldots, N]$*, let* $\vec{w}_i \in \{-1, 0, 1\}^{d_i}$ *and* $\vec{w} = \|_{i=1}^{N} \vec{w}_i \in \text{VALID}$.
  Let $\vec{u}_i := \vec{M}_i \cdot \vec{w}_i \pmod{q_i}$ *and define* $\vec{u} = \|_{i=1}^{N} \vec{u}_i$.
  The common input consists of $\vec{M} = \{\vec{M}_i\}_{i \in [N]}$ *and* $\vec{u} = \{\vec{u}_i\}_{i \in [N]}$*, while the prover's secret input (witness) is* $\vec{w} = \vec{w_1} \| \cdots \| \vec{w_N}$.

- GenCRS*: The* CRS *consists of the regular CRS* ρ *of our dual-mode commitment scheme, Protocol 1, and a key s for the hash function H.*

- Prove*: Takes as input the statement* $(\vec{M}, \vec{u})$*, the witness* $\vec{w}$ *and the CRS* ρ*. The algorithm computes three different commitments* C_1, C_2, C_3 *using the standard commitment scheme.*
  Afterwards, these commitments are then committed using the dual-mode commitment scheme and the commitment c is computed and the decommit (a, τ) *used for the decommitment. As a final step, depending on the hash of the statement and the commit c, we provide b.*

  The algorithm outputs the statement $(\vec{M}, \vec{u})$*, the commit c, the decommit information* (a, τ) *and b. We can see* (c, a, τ, b) *as the proof for* $(\vec{M}, \vec{u})$.

  Algorithm 1 describe it in details.

- Verify*: Takes as input the statement* $(\vec{M}, \vec{u})$*, the commit c, the decommit information* a, τ *and b. Initially, the algorithm decommits c of the dual-mode commitment using* (a, τ) *in order to obtain* $a = (C_1, C_2, C_3)$*. Depending on the digest of the hash of the statement and c, a different check is made on* C_i *and b.*

  The output is 1 *if all the checks hold,* 0 *otherwise.*

  Algorithm 2 describes the Verify *algorithm in details.*

3 Translation of Boneh's PRF

For completeness of the paper, we provide the specific instantiation of Boneh's lattice-based PRF [4] used in our Protocol 2, which is described and explained in details in Libert *et al.* [15].

For any t positive integer, define the following:

- $\mathbf{S}_t$: the set of all t-elements permutations.
- $\mathbf{B}_t^2$: the set of vectors in $\{0,1\}^{2t}$ with Hamming weight t.
- $\mathbf{B}_t^3$: the set of vectors in $\{-1,0,1\}^{3t}$ with exactly t elements equal to -1, t elements equal to 0 and t elements equal to 1.

Let Expand be the function that for every bit c and for all vectors $\vec{v} \in \mathbb{Z}^t$, it is defined as:

$$\text{Expand}(c, \vec{v}) := \begin{pmatrix} (1-c) \cdot \vec{v} \\ c \cdot \vec{v} \end{pmatrix} \in \mathbb{Z}^{2t}$$

Algorithm 1 Prover $\mathscr{P}$: Prove$((\vec{M},\vec{u}),\vec{w},\rho)$

1. – Sample $\vec{\phi} \leftarrow_R S$, for $i \in 1,\ldots,N$ sample $\vec{r}_i \leftarrow_R \mathbb{Z}_{q_i}^{d_i}$ and define $\vec{r} = \|_{i=1}^{N} \vec{r}_i$ as the concatenation of $\vec{r}_i$s and $\vec{v} = \vec{w} \boxplus \vec{r}$ as $v_i = w_i + r_i \pmod{q_i}$ for all $i \in \{1,\ldots,N\}$.
   – Sample ρ_1,ρ_2,ρ_3 and compute

$$C_1 = \mathrm{Com}(\vec{\phi}, \left\{\vec{M}\cdot\vec{r}_i \pmod{q_i}\right\}_{i=1}^{N}; \rho_1) \quad C_2 = \mathrm{Com}(\vec{\Gamma}_\phi(\vec{r}); \rho_2)$$

$$C_3 = \mathrm{Com}(\vec{\Gamma}_\phi(\vec{v}); \rho_3)$$

2. Define $a = (C_1, C_2, C_3)$
3. Compute $c = \mathrm{DCom}_\rho(a;\tau)$ and $\mathrm{DDecom}_\rho(a;\tau) = (a,\tau)$, where $\mathrm{DCom}_\rho(a;\tau)$ is our dual-mode commitment to a using randomness τ and CRS ρ, and (a,τ) is its corresponding decommitment;
4. Compute $e = H_s((\vec{M},\vec{u}),c)$
5. Define b to be

$$b = \begin{cases} (\vec{\Gamma}_\phi(\vec{w}), \vec{\Gamma}_\phi(\vec{r}), \rho_2, \rho_3) & \text{when } e = 1 \\ (\vec{\phi}, \vec{v}, \rho_1, \rho_3) & \text{when } e = 2 \\ (\vec{\phi}, \vec{r}, \rho_1, \rho_2) & \text{when } e = 3 \end{cases}$$

Output: $\pi = ((\vec{M},\vec{u}),c,a,\tau,b)$

Algorithm 2 Verifier $\mathscr{V}$: Verify$((\vec{M},\vec{u}),c,d,b)$

1. Compute $a = \mathrm{DReceiverDecom}(c,(a,\tau))$. If $a = \perp$, output 0.
2. Compute $e = H_s((\vec{M},\vec{u}),c)$
3. Compute and verify
   (a) If $e = 1$, let $b = (\vec{t},\vec{s},\rho_2,\rho_3)$. Check that $t \in \mathrm{VALID}$, $C_2 = \mathrm{Com}(s;\rho_2)$ and $C_3 = \mathrm{Com}(t \boxplus s;\rho_3)$.
   (b) If $e = 2$, let $b = (\vec{\pi},\vec{x},\rho_1,\rho_3)$, parse $\vec{x} = (\vec{x_1}\|\cdots\|\vec{x_N}), \vec{x}_i \in \mathbb{Z}_{q_i}^{d_i}$, check that $C_2 = \mathrm{Com}(\vec{\pi}, \{\vec{M}_i\cdot\vec{x}_i - \vec{u}_i \pmod{q_i}\}_{i=1}^{N}; \rho_1)$ and $C_3 = \mathrm{Com}(\vec{\Gamma}_\pi(\vec{x}); \rho_3)$.
   (c) If $e = 3$, let $b = (\vec{\psi},\vec{y},\rho_1,\rho_2)$. parse $\vec{y} = (\vec{y_1}\|\cdots\|\vec{y_N}), \vec{y}_i \in \mathbb{Z}_{q_i}^{d_i}$ and check that $C_1 = \mathrm{Com}(\vec{\psi}, \{\vec{M}_i\cdot\vec{y}_i \pmod{q_i}\}_{i=1}^{N}; \rho_1)$ and $C_2 = \mathrm{Com}(\vec{\Gamma}_\psi(\vec{y}); \rho_2)$.
4. If the verification fails, output 0. Otherwise output 1.

Let $T_{c,\pi}$ be defined for every bit c, for all vectors $\vec{v} := \begin{pmatrix}\vec{v_0}\\ \vec{v_1}\end{pmatrix} \in \mathbb{Z}^{2t}$ where $\vec{v_0},\vec{v_1} \in \mathbb{Z}^t$ and for all permutation $\pi \in \mathbf{S}_t$, as

$$T_{c,\pi}(\vec{v}) := \begin{pmatrix}\pi(\vec{v}_c)\\ \pi(\vec{v}_{1-c})\end{pmatrix}$$

For any $B \in \mathbb{Z}, B > 0$, let us consider a specific way to represent integers, similar to the *binary representation of B*: define $\delta_B := \lfloor \log_2 B \rfloor + 1$ and the sequence $\{B_j\}_{j\in[1..\delta_B]}$ with $B_j := \left\lfloor \frac{B+2^{j-1}}{2^j} \right\rfloor$ for

every $j \in [1..\delta_B]$. For every integer $v \in [0..B]$, define $\mathrm{idec}_B(v) := (v_{(1)}, v_{(2)}, \ldots, v_{(\delta_B)}) \in \{0,1\}^{\delta_B}$ such that $\sum_{j \in [1..\delta_B]} B_j v_{(j)} = v$.

Let $\sigma(x)$ be the standard sign function and define

$$\begin{gathered}\mathrm{vdec}'_{t,B} : [-B,B]^t \to \{-1,0,1\}^{t\delta_B} \\ (w_1, \ldots, w_t) \mapsto (\sigma(w_1) \cdot \mathrm{idec}_B(w_1), \ldots, \sigma(w_t) \cdot \mathrm{idec}_B(w_t))\end{gathered}$$

and $\mathrm{vdec}_{t,B} : [0,B]^t \to \{0,1\}^{t\delta_B}$ as vdec' on the smaller domain $[0,B]^t$.

In order to close the change-representation function, let us define the matrix:

$$H_{t,B} = \begin{pmatrix} B_1 & \ldots & B_{\delta_B} & & & & & & \\ & & & B_1 & \ldots & B_{\delta_B} & & & \\ & & & & & & \ddots & & \\ & & & & & & & B_1 & \ldots & B_{\delta_B} \end{pmatrix} \in \mathbb{Z}^{t \times t\delta_B}$$

while, for all $\vec{v} \in [-B,B]^t$, it holds $H_{t,B} \cdot \mathrm{vdec}'_{t,B}(\vec{v}) = \vec{v}$

Next, let us consider specific maps that map a vector inside either $\mathbf{B}_t^2$ or $\mathbf{B}_t^3$: for all $\vec{v} \in \{0,1\}^t$, define $\mathrm{TwoExt}(\vec{v}) := (\vec{v} \| \vec{0}^{t-n_0} \| \vec{1}^{t-n_1}) \in \mathbf{B}_t^2$ where n_j is the number of coordinates in $\vec{v}$ equal to j. For all $\vec{v} \in \{-1,0,1\}^t$, define $\mathrm{ThreeExt}(\vec{v}) := (\vec{v} \| \vec{0}^{t-n_0} \| \vec{1}^{t-n_1} \| (\vec{-1})^{t-n_{-1}}) \in \mathbf{B}_t^3$ where n_j is the number of coordinates in $\vec{v}$ equal to j.

Similarly, for any $B \in \mathbb{Z}, B > 0$, let us consider the matrices:

$$\widehat{H}_{t,B} := \left[\ H_{t,B} \ \middle|\ 0^{t \times t\delta_B}\ \right] \qquad \widetilde{H}_{t,B} := \left[\ H_{t,B} \ \middle|\ 0^{t \times 2t\delta_B}\ \right]$$

that just extend the specific identity property of $H_{t,B}$ with respect to the image of TwoExt and ThreeExt respectively.

The next step is to transform Boneh's PRF [4]. Let us consider public binary matrices $P_0, P_1 \in \{0,1\}^{m \times m}$, a committed seed $\vec{k} \in \mathbb{Z}_q^m$ and a secret bitstring $(J_1, \ldots, J_L) \in \{0,1\}^L$ and define the matrix

$$P := \left[\ P_0 \cdot \widehat{H}_{m,q-1} \ \middle|\ P_1 \cdot \widehat{H}_{m,q-1}\ \right] \in \mathbb{Z}_q^{m \times 4\overline{m}}$$

Let us consider public matrices $D_0 \in \mathbb{Z}_{q_1}^{n \times m_0}, D_1 \in \mathbb{Z}_{q_1}^{n \times \overline{m}}$ for some modulus q_1, for some integer m_0 and $\overline{m} = m\delta_{q-1}$. Let $\vec{r} \in [-\beta,\beta]^{m_0}$ be a discrete Gaussian vector with small β and define the vector

$$\vec{\widetilde{r}} := \mathrm{ThreeExt}(\mathrm{vdec}'_{m_0,\beta}(r)) \in \mathbf{B}^3_{m_0\delta_\beta}$$

Define $\vec{x_0} = \vec{k}$, for each $i \in [1,L]$, compute $\vec{x_i} = P_{J_i} \vec{x_{i-1}} \pmod q$ and $\vec{y} = \lfloor \vec{x_L} \rfloor_p$. For all $i \in [1,L]$, define the vectors $\vec{\widehat{x}_i} := \mathrm{TwoExt}(\mathrm{vdec}_{m,q-1}(\vec{x_i})) \in \mathbf{B}^2_{\overline{m}}$ and $\vec{s_{i-1}} := \mathrm{Expand}(J_i, \vec{\widehat{x}_{i-1}}) \in \mathbf{B}^2_{2\overline{m}}$.

Let $\vec{\widehat{z}} \in \mathbf{B}^2_{\overline{m}}$. Let $\vec{w_1} := (\vec{\widetilde{r}} \| \vec{\widehat{x}_0})$, $\vec{M_1} = \left[\ D_0 \cdot \widetilde{H}_{m_0,\beta} \ \middle|\ D_1 \ \middle|\ \vec{0}^{n \times \overline{m}}\ \right]$ and $\vec{u_1} = \vec{M_1} \cdot \vec{w_1} \pmod{q_1}$.

Let $\vec{w_2} := (\vec{s_0} \| \vec{\widehat{x}_1} \| \vec{s_1} \| \vec{\widehat{x}_2} \| \ldots \| \vec{s_{L-1}} \| \vec{\widehat{x}_L})$, $\vec{u_2} = \vec{0}$ and

$$\vec{M_2} = \begin{bmatrix} P & -\widehat{H}_{m,q-1} & & \\ & \ddots & \ddots & \\ & & P & -\widehat{H}_{m,q-1} \end{bmatrix}$$

Let $\vec{w_3} := (\vec{\widehat{x}_L} \| \vec{z})$, $\vec{M_3} := \left[\ (p \cdot \widehat{H}_{m,q-1}) \ \middle|\ \widehat{H}_{m,q-1}\ \right]$ and $\vec{u_3} := q \cdot \vec{y}$.

Let us define $d_1 = 3m_0\delta_\beta + 2\overline{m}$, $d_2 = 6L\overline{m}$ and $d_3 = 4\overline{m}$ as the dimensions of $\vec{w}_1, \vec{w}_2$ and $\vec{w}_3$. Let $q_2 = q$, $q_3 = pq$ and $d = d_1 + d_2 + d_3$.

It holds $\vec{M}_i \cdot \vec{w}_i = \vec{u}_i \pmod{q_i}$ for $i \in \{1,2,3\}$ and $\vec{w} = (\vec{w}_1 \| \vec{w}_2 \| \vec{w}_3) \in \{-1,0,1\}^d$ of the form

$$\vec{w} = (\vec{\tilde{r}} \| \vec{\widehat{x}}_0 \| \vec{s}_0 \| \vec{\widehat{x}}_1 \| \vec{s}_1 \| \vec{\widehat{x}}_2 \| \ldots \| \vec{s_{L-1}} \| \vec{\widehat{x}}_L \| \vec{\widehat{z}})$$

Let us define the set VALID as the set of $\vec{w}$ such that:

- $\vec{\tilde{r}} \in \mathbf{B}^3_{m_0\delta_\beta}$ and $\vec{\widehat{x}}_0, \ldots, \vec{\widehat{x}}_L, \vec{z} \in \mathbf{B}^2_{\overline{m}}$
- for all $i \in [1,L]$, $s_{i-1}^{\rightarrow} = \text{Expand}(J_i, \widehat{x}_{i-1}^{\rightarrow})$ for some $J_i \in \{0,1\}$

Let us define $\mathbf{S} := \mathbf{S}_{3m_0\delta_\beta} \times (\mathbf{S}_{2\overline{m}})^{L+2} \times \{0,1\}^L$.
For every element $\pi = (\pi_r, \pi_0, \pi_1, \ldots, \pi_L, \pi_z, b_1, \ldots, b_L) \in \mathbf{S}$, let Γ_π be the permutation of the vector $\vec{w} \in \mathbb{Z}^d$ defined as

$$\begin{aligned}\Gamma_\pi(\vec{w}) := (\pi_r(\vec{\tilde{r}}) \| \pi_0(\vec{\widehat{x}}_0) \| T_{b_1,\pi_0}(\vec{s}_0) \| \pi_1(\vec{\widehat{x}}_1) \| T_{b_2,\pi_1}(\vec{s}_1) \| \\ \| \pi_2(\vec{\widehat{x}}_2) \| \ldots \| T_{b_L,\pi_{L-1}}(s_{L-1}^{\rightarrow}) \| \pi_L(\vec{\widehat{x}}_L) \| \pi_L(\vec{\widehat{x}}_L) \| \pi_L(\vec{z}))\end{aligned}$$

4 Challenges and Future Directions

In this section we will briefly discuss and collect our conjectures and/or our future research directions by dividing them into into two major classes: a first class of questions related to *transformations* from ZK to NIZK and a second class of challenges regarding *lattice languages*.

4.1 ZK Transformations

Choosing Lindell's transformation is not optimal for the final goal of constructing an sVRF since the transformation is defined in the non-programmable ROM.

Ciampi *et al.* [7] modified and improved Lindell's transformation: the transformation does not require the non-programmable random oracle *nor* a perfectly binding commitment scheme at the cost of a more specific language. By using Ciampi *et al.*'s transformation, it might be possible to obtain a ZK to NIZK transformation not based on the ROM.

Challenge 1. *Is it possible to use Ciampi* et al. *transformation in our sVRF construction-idea? The main challenge of this approach is to check if any lattice-based language can be defined in order to fulfil the transformation hypothesis and requirement.*

With the same spirit, we find an additional challenge of more general interest: a ZK to NIZK transformation that is not defined in the random oracle model (or any similar ones). Therefore, we state as a general challenge for future investigation:

Challenge 2. *Are there any other transformations in the literature that can be used for our construction-idea? Are they efficient? How do they compare among themselves or with respect to the Fiat-Shamir's transformation?*

4.2 Lattice Languages

When considering the Lindell's transformation, the language L_{IS} is ill-defined and therefore cannot be used in order to build a dual-mode commitment scheme. Furthermore, the language challenge of defining a membership-hard language can be seen as of perpendicular interest.

Challenge 3. *Is there a way to define a lattice-based membership-hard efficient sampling language L that can be used to define a dual-mode commitment scheme?*

Generally speaking and quite informally, the main obstacle is finding *"good"*-languages that have a *"unique-witness"*. This means that it would be incredibly useful to find a lattice-language L in which the witness of a statement $x \in L$ is unique. Solving this problem will open new directions in lattice-based cryptography.

Challenge 4. *Find a lattice-based language L in which every statement $x \in L$ has a unique witness w.*

As a different but related problem, if we consider a different ZK PRF proof system, the ZK language used for our construction-idea requires an additional property in order to be used by the Chase-Lysyanskaya's transformation. The ZK system has to be able to prove the correct computation of the PRF **and** the correctness of an additional commitment. It has to be defined over lattices **and**, after transforming it with the best ZK to NIZK transformation possible, the obtained NIZK has to be multi-theorem.

Challenge 5. *Given the best ZK transformation, find a ZK PRF argument/proof system that can be used for the Chase-Lysyanskaya's transformation.*

Acknowledgement.

We are grateful to the anonymous reviewers for their insightful comments, suggestions, discussions and the new literature-directions provided. This work was partially supported by the Swedish Research Council (Vetenskapsrådet) through the grant PRECIS (621-2014-4845).

References

[1] M. Ajtai. Generating Hard Instances of Lattice Problems (Extended Abstract). In *Proc. of the 28th Annual ACM Symposium on Theory of Computing (STOC'96), Philadelphia, Pennsylvania, USA*, pages 99–108. ACM, May 1996.

[2] C. Baum, I. Damgård, V. Lyubashevsky, S. Oechsner, and C. Peikert. More Efficient Commitments from Structured Lattice Assumptions. In *Proc. of the 11th International Conference on Security and Cryptography for Networks (SCN'18), Amalfi, Italy*, volume 11035 of *Lecture Notes in Computer Science*, pages 368–385. Springer, Cham, September 2016.

[3] F. Benhamouda, S. Krenn, V. Lyubashevsky, and K. Pietrzak. Efficient Zero-Knowledge Proofs for Commitments from Learning with Errors over Rings. In *Proc. of the 20th European Symposium on Research in Computer Security (ESORICS'15), Vienna, Austria*, volume 9326 of *Lecture Notes in Computer Science*, pages 305–325, New York, NY, USA, September 2015. Springer, Cham.

[4] D. Boneh, K. Lewi, H. Montgomery, and A. Raghunathan. Key Homomorphic PRFs and Their Applications. In *Proc. of the 33rd Annual Cryptology Conference (CRYPTO'13), Santa Barbara, CA, USA*, volume 8042 of *Lecture Notes in Computer Science*, pages 410–428. Springer, Berlin, Heidelberg, August 2013.

[5] D. Catalano and I. Visconti. Hybrid Commitments and Their Applications to Zero-knowledge Proof Systems. *Theoretical Computer Science*, 374(1-3):229–260, April 2007.

[6] M. Chase and A. Lysyanskaya. Simulatable VRFs with Applications to Multi-theorem NIZK. In *Proc. of the 27th Annual International Cryptology Conference (CRYPTO'07), Santa Barbara, CA, USA*, volume 4622 of *Lecture Notes in Computer Science*, pages 303–322. Springer, Berlin, Heidelberg, August 2007.

[7] M. Ciampi, G. Persiano, L. Siniscalchi, and I. Visconti. A Transform for NIZK Almost as Efficient and General as the Fiat-Shamir Transform Without Programmable Random Oracles. In *Proc. of the 13th International Conference on Theory of Cryptography (TCC'16), Tel Aviv, Israel, Part 2*, volume 9563 of *Lecture Notes in Computer Science*, pages 83–111. Springer, Berlin, Heidelberg, January 2016.

[8] B. David, P. Gaži, A. Kiayias, and A. Russell. Ouroboros Praos: An Adaptively-Secure, Semi-synchronous Proof-of-Stake Blockchain. In *Proc. of the 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'18), Tel Aviv, Israel*, volume 10821 of *Lecture Notes in Computer Science*, pages 66–98. Springer, Cham, April 2018.

[9] A. Fiat and A. Shamir. How To Prove Yourself: Practical Solutions to Identification and Signature Problems. In *Proc. of the Conference on the Theory and Application of Cryptographic Techniques (CRYPTO'86), Santa Barbara, CA, USA*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, Berlin, Heidelberg, August 2006.

[10] S. Goldberg, M. Naor, D. Papadopoulos, L. Reyzin, S. Vasant, and A. Ziv. NSEC5: provably preventing DNSSEC zone enumeration. In *Proc. of the 22nd Annual Network and Distributed System Security Symposium (NDSS'15), San Diego, CA, USA*. Internet Society, February 2015.

[11] O. Goldreich, S. Goldwasser, and S. Micali. How to Construct Random Functions. *Journal of the ACM*, 33(4):792–807, October 1986.

[12] A. Kawachi, K. Tanaka, and K. Xagawa. Concurrently Secure Identification Schemes Based on the Worst-Case Hardness of Lattice Problems. In *Proc. of the 14th International Conference on the Theory and Application of Cryptology and Information Security (Asiacrypt'08), Melbourne, Australia*, volume 5350 of *Lecture Notes in Computer Science*, pages 372–389. Springer, Berlin, Heidelberg, December 2008.

[13] W. Li, S. Andreina, J.-M. Bohli, and G. Karame. Securing Proof-of-Stake Blockchain Protocols. In *Proc. of the ESORICS 2017 International Workshops on Data Privacy Management (DPM'17), and Cryptocurrencies and Blockchain Technology (CBT'17), Oslo, Norway*, volume 10436 of *Lecture Notes in Computer Science*, pages 297–315. Springer, Cham, September 2017.

[14] B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang. Zero-knowledge arguments for matrix-vector relations and lattice-based group encryption. In *Proc. of the 22nd International Conference on the Theory and Application of Cryptology and Information Security (Asiacrypt'16), Hanoi, Vietnam*, volume 10032 of *Lecture Notes in Computer Science*, pages 101–131. Springer, Berlin, Heidelberg, December 2016.

[15] B. Libert, S. Ling, K. Nguyen, and H. Wang. Zero-Knowledge Arguments for Lattice-Based PRFs and Applications to E-Cash. In *Proc. of the 23rd International Conference on the Theory and Applications of Cryptology and Information Security (Asiacrypt'17), Hong Kong, China, Part 3*, volume 10626 of *Lecture Notes in Computer Science*, pages 304–335, Hong Kong, China, December 2017. Springer, Cham.

[16] B. Libert, S. Ling, K. Nguyen, and H. Wang. Lattice-Based Zero-Knowledge Arguments for Integer Relations. In *Proc. of the 38th Annual International Cryptology Conference (CRYPTO'18), Santa Barbara, CA, USA, Part 2*, volume 10992 of *Lecture Notes in Computer Science*, pages 700–732. Springer, Cham, August 2018.

[17] Y. Lindell. An Efficient Transform from Sigma Protocols to NIZK with a CRS and Non-programmable Random Oracle. In *Proc. of the 12th Theory of Cryptography Conference on Theory of Cryptography (TCC'15), Warsaw, Poland*, volume 9014 of *Lecture Notes in Computer Science*, pages 93–109. Springer, Berlin, Heidelberg, March 2015.

[18] S. Micali, M. Rabin, and S. Vadhan. Verifiable random functions. In *Proc. of the 40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039), New York City, NY, USA*, pages 120–130. IEEE, October 1999.

[19] S. Micali and R. L. Rivest. Micropayments revisited. In *Proc. of the 2002 Cryptographers' Track at the RSA Conference (CT-RSA'02), San Jose, CA, USA*, volume 2271 of *Lecture Notes in Computer Science*, pages 149–163. Springer, Berlin, Heidelberg, February 2002.

[20] D. Micciancio and C. Peikert. Hardness of SIS and LWE with small parameters. In *Proc. of the 33rd Annual Cryptology Conference (CRYPTO'13), Santa Barbara, CA, USA*, volume 8042 of *Lecture Notes in Computer Science*, pages 21–39. Springer, Berlin, Heidelberg, August 2013.

[21] J. B. Nielsen. Separating Random Oracle Proofs from Complexity Theoretic Proofs: The Non-committing

Encryption Case. In *Proc. of the 22nd Annual International Cryptology Conference (CRYPTO'02), Santa Barbara, CA, USA*, volume 2442 of *Lecture Notes in Computer Science*, pages 111–126. Springer, Berlin, Heidelberg, August 2002.

[22] D. Papadopoulos, D. Wessels, S. Huque, M. Naor, J. Včelák, L. Reyzin, and S. Goldberg. Making nsec5 practical for dnssec. Cryptology ePrint Archive, Report 2017/099, 2017. `https://eprint.iacr.org/2017/099` [Online; Accessed on November 30, 2018].

[23] C. Peikert. A Decade of Lattice Cryptography. *Foundations and Trends® in Theoretical Computer Science*, 10(4):283–424, 2016.

[24] O. Regev. The Learning with Errors Problem (Invited Survey). In *Proc. of the IEEE 25th Annual Conference on Computational Complexity (CCC'10), Cambridge, MA, USA*, pages 191–204. IEEE, June 2010.

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

Author Biography

Carlo Brunetta received the B.S. and M.S. degrees in Mathematics from the University of Trento in 2014 and 2016 respectively. Currently, he is a PhD student at Chalmers University of Technology. His research interests include blockciphers security, privacy-preserving cryptographic protocols and blockchains.

Bei Liang received the Ph.D. degree in Information Security from the Institute of Information Engineering of Chinese Academy of Sciences, Beijing, China, in 2016. Currently, she continues to pursue further studies as a post-doctoral researcher at the Department of Computer Science and Engineering at Chalmers University of Technology, Sweden. She has worked in the area of theoretical cryptography especially with a focus on the study of multilinear maps, program obfuscation, verifiable random functions and functional encryption.

Aikaterini Mitrokotsa is an Associate Professor at the Department of Computer Science and Engineering, at Chalmers University of Technology. Formerly, she held positions as a visitor professor at ETHZ and the Tokyo Institute of Technology. Her main research interests include information and network security, cryptographic protocols, privacy-preservation, and provable security. She has been awarded the Young Researcher Grant from the Swedish Research Council, the Rubicon Research Grant by NWO, and a Marie Curie Intra European Fellowship.