Post-Quantum Verifiable Oblivious Pseudorandom Functions

by

Helen Propson

S.B. Computer Science and Engineering, Massachusetts Institute of Technology 2024

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2024

Authored by: Helen Propson
Department of Electrical Engineering and Computer Science
May 10, 2024

Certified by: Vinod Vaikuntanathan
Professor of Computer Science, Thesis Supervisor

Accepted by: Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Post-Quantum Verifiable Oblivious Pseudorandom Functions

by

Helen Propson

Submitted to the Department of Electrical Engineering and Computer Science
on May 10, 2024 in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

ABSTRACT

This work presents the construction of a post-quantum verifiable oblivious pseudorandom function (VOPRF) with a focus on efficiency and practicality. Leveraging lattice-based cryptographic primitives, particularly the Learning With Errors (LWE) problem, our VOPRF construction aims to address the limitations of existing approaches by reducing proof sizes. The key component in our work is the integration of an efficient zero-knowledge proof of knowledge (ZKPoK) protocol. This ZKPoK is notably more efficient than the proof systems used in prior VOPRF constructions, ensuring the verifiability of PRF outputs while providing smaller proof sizes. Our construction relies on the hardness of the ring-LWE and short integer solution (SIS) problems, and we demonstrate its security in the random oracle model. Overall, our VOPRF construction represents a step towards the development of more practical post-quantum secure cryptographic protocols, highlighting the potential for further improvements in efficiency and real-world applicability.

Thesis supervisor: Vinod Vaikuntanathan
Title: Professor of Computer Science

Acknowledgments

I am deeply grateful to my advisor, Professor Vinod Vaikuntanathan, for introducing me to the field of cryptography and for his unwavering dedication to my development as a researcher over the past two years. His passion and enthusiasm for the field have constantly pushed me to delve deeper and think creatively, and I am immensely appreciative of his guidance and support. I would also like to thank Leo de Castro for his invaluable mentorship. His generous investment of time, support, and insight throughout the various stages of this research project has been instrumental to its success. I am thankful to all the members of my research group for their inspiration. I am extremely grateful for the opportunity to work alongside such dedicated individuals. I sincerely appreciate all my professors and educators, who have invested many hours in helping me strengthen my understanding of complex concepts. Their dedication has been fundamental to my ability to conduct meaningful research. To my friends, your encouragement and companionship have made enduring the challenges of undergraduate life at MIT an incredibly rewarding experience. I am profoundly grateful for the friendships I have made over these past four years. Most importantly, I would like to thank my family, including my mother, father, and brother, for their unwavering love and support. Without them, this thesis would not have been possible.

Contents

List of Figures

Chapter 1

Introduction

Verifiable Random Functions (VRFs) represent a category of pseudo-random functions, where the entity holding the secret key provides evidence of the function's evaluation validity. VRFs have found applications in a variety of domains, including the DNSSEC protocol [1], blockchain-based lottery schemes [2], and blockchain consensus mechanisms for establishing proof-of-stake [3]. However, widely used VRFs in these applications, such as ECVRF, are susceptible to known attacks by quantum computers. Consequently, there is a need to develop post-quantum secure VRFs, with a particular focus on creating a VRF that is both resilient against quantum attacks and efficient for practical use.

Extending the concept of VRFs, Verifiable Oblivious Pseudorandom Functions (VOPRFs) offer an additional layer of privacy by ensuring that the input to the PRF remains hidden from the entity performing the function's evaluation. This added obliviousness makes VOPRFs particularly useful in scenarios requiring both verifiability and privacy, such as secure multi-party computations, private set intersections, and privacy-preserving authentication protocols. Like VRFs, the development of post-quantum secure VOPRFs is crucial, as they need to withstand quantum attacks while remaining efficient and practical for real-world applications.

1.1 Technical Overview

In this work, we detail the construction of an oblivious VOPRF in the random oracle model. Our construction builds upon the VOPRF proposed by [4], which uses non-interactive zero-knowledge arguments of knowledge. To improve efficiency, we replace the use of the zero-knowledge proof system of [5] with a more efficient non-interactive zero-knowledge proof of knowledge (ZKPoK). As in [6], our ZKPoK involves parallel repetitions of the signing protocol from [7] and utilizes the Fiat-Shamir transform to convert the interactive protocol into a non-interactive one.

Although this proof system offers the advantage of significantly shorter proof sizes, it presents a challenge due to the soundness gap inherent in this type of ZKPoK. In the context of our VOPRF, the server responsible for computing the VOPRF evaluations must prove that the secret key it uses to generate the evaluations has an ℓ_∞ norm less than a specified bound. However, the ZKPoK can only be used to prove a bound that is larger than the bound on the ℓ_∞ norm of an honest server's key. This discrepancy is known as the soundness gap.

To address this, in our VOPRF protocol, the honest server draws their key from a distribution with a smaller ℓ_∞ norm bound than the bound they must prove. This approach contrasts with the original protocol by [7], where the actual bound on the ℓ_∞ norm of the server's key and the bound proven by the server are the same. By differentiating these two, we maintain the security guarantees of the protocol while enhancing its practical efficiency by providing smaller proof sizes.

While our work improves upon the existing VOPRF protocol, a significant bottleneck remains the non-interactive zero-knowledge argument of knowledge (NIZKAoK) required for the client to prove that its input to the VOPRF is well-formed. The relation of this proof is incompatible with our zero-knowledge proof of knowledge (ZKPoK) system and requires proof of quadratic relations. As a result, we are unable to apply our ZKPoK to the client's proof and must instead use the proof system proposed by [5] for this part of the VOPRF,

which hinders the practicality of the VOPRF construction. As such, our work highlights the importance of developing more practical VOPRF solutions, paving the way for future improvements, as discussed in the Chapters on efficiency 8 and future work 9.

1.2 Outline

The remainder of this paper is structured as follows. Chapter 2 reviews related work, while Chapter 3 introduces the necessary preliminaries. In Chapter 4, we present our ZKPoK protocol, followed by our VOPRF construction in Chapter 5, detailing how we instantiate the VOPRF using our ZKPoK in Chapter 6. Chapter 7 provides a discussion of our parameter choices. Chapter 8 analyzes the efficiency of our VOPRF, and Chapter 9 concludes with directions for future work.

Chapter 2

Related Work

The concept of verifiable random functions was introduced by Micali, Rabin, and Vadhan [8]. Since then, various constructions have been put forth. However, the security of many of these constructions relies on the hardness of discrete logarithms and factoring, foundational to numerous cryptosystems. Unfortunately, these assumptions become vulnerable in the quantum setting, as demonstrated by Shor's algorithm [9]. Consequently, our aim is to devise a construction grounded in a problem that remains resistant to quantum attacks.

2.1 Lattice-Based Cryptography

A work of Regev [10] used a quantum reduction to prove that if there exists an efficient algorithm that solves LWE, then there exists an efficient quantum algorithm that approximates the decision version of the shortest vector problem (GapSVP) and the shortest independent vectors problem (SIVP). This is of particular interest because there are no known efficient quantum algorithms for GapSVP or SIVP. Peikert later showed that the search version of LWE is at least as hard as approximating GapSVP in the worst case via a classical (probabilistic polynomial-time) reduction [11], further enhancing confidence in the hardness of LWE. Consequently, lattice-based cryptographic frameworks have garnered significant attention due to their resilience against quantum attacks.

2.2 Existing Approaches

The development of an efficient lattice-based VRF poses challenges, partly due to the absence of known efficient zero-knowledge proofs validating the computation of the VRF output.

One existing construction of a post-quantum secure VRF, relying on well-known lattice problems like Module-SIS and Module-LWE, yields a VRF value of only 84 bytes and a 5 KB proof but has k-time security constraints, necessitating frequent key updates for limited VRF outputs per key pair [12]. Another recent construction using symmetric primitives achieves a 3 KB proof size but requires users to maintain shared state and undergo key updates [13]. Addressing these issues, a practical lattice-based VRF construction, which is stateless and supports an almost unrestricted number (2^{128}) of VRF evaluations, utilizes relaxed proofs of knowledge, resulting in a 10.27 KB proof size [14]. In contrast, another lattice-based VRF construction in the random oracle model produces proofs in the order of megabytes [5]. Furthermore, there is a lattice-based VRF construction [15] in the standard model, but its efficiency is compromised by its use of general NIWI and constrained pseudorandom functions (PRFs), with no practical efficiency evaluation provided.

However, none of these constructions are oblivious. The first round-optimal VOPRF protocol that maintains security based on subexponential lattice hardness assumptions was presented by [4], but it suffers from poor efficiency, with proof sizes on the order of gigabytes. Consequently, this work explores options for enhancing the practicality of the construction.

Chapter 3

Preliminaries

3.1 Notation

In this work we consider a ring $\mathcal{R}$, which will be either $\mathbb{Z}$ (in our ZKPok) or the polynomial $\mathcal{R}_q = \mathbb{Z}_q[x]/(x^{\tilde{n}}+1)$ (in our VOPRF) where $\tilde{n}$ is a power of 2 and q is an integer modulus. We let the elements of $\mathbb{Z}_q$ be identified with the representatives $\{-(\frac{q}{2}), \ldots, \frac{q}{2}\}$. We will represent vectors by bold-face letters, and matrices by bold-face capital letters. The l_p norm of a ring element a is denoted by $\|a\|_p$, and we will sometimes omit the p for the l_2 norm. We extend the notation to vectors and matrices as follows: $\|\mathbf{a}\| = \sqrt{\sum \|a_i\|^2}$ and $\|\mathbf{A}\| = \sqrt{\sum \|\mathbf{a}_i\|^2}$. We will also consider the operator norm of matrices over $\mathcal{R}$ defined as $s_1(\mathbf{A}) = \max\left(\frac{\|\mathbf{Ax}\|_2}{\|\mathbf{x}\|_2}\right)$. For a distribution D, we use the notation $x \xleftarrow{\$} D$ to mean that x is chosen according to the distribution D. If S is a set, then $x \xleftarrow{\$} S$ means that x is chosen uniformly at random from S. We let $\mathsf{negl}(\kappa)$ denote a negligible function (i.e. a function that is $\kappa^{-\omega(1)}$) and write $r_1 \gg r_2$ as short-hand for $r_1 \geq \kappa^{\omega(1)} \cdot r_2$. Logarithm base 2 is denoted $\log_2$, and we sometimes omit the 2 for simplicity. For $x \in \mathbb{Z}_q$, the rounding operation is defined as $\lfloor x \rceil_p := \left\lfloor \frac{p}{q} \cdot x \right\rceil$ where $\lfloor \cdot \rceil$ denotes rounding to the nearest integer (rounding down in the case of a tie). We use tildes to distinguish symbols in our VOPRF protocol that also appear as parameters in our ZKPoK protocol but have different meanings (e.g., the modulus q does not have a tilde,

as it retains the same usage in both protocols).

3.2 Probability Distributions and Rejection Sampling

Definition 1 *The continuous Normal distribution over $\mathbb{R}^m$ centered at $\mathbf{v}$ with standard deviation σ is defined by the function $\rho^m_{\mathbf{v},\sigma}(\mathbf{x}) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^m e^{\frac{-\|\mathbf{x}-\mathbf{v}\|^2}{2\sigma^2}}$*

When $\mathbf{v} = 0$, we will just write $\rho^m_\sigma(\mathbf{x})$. We define the discrete Normal distribution over $\mathbb{Z}^m$ as follows:

Definition 2 *(Definition 4.2 [7]) The discrete Normal distribution over $\mathbb{Z}^m$ centered at some $\mathbf{v} \in \mathbb{Z}^m$ with standard deviation σ is defined as $D^m_{\mathbf{v},\sigma}(\mathbf{x}) = \frac{\rho^m_{\mathbf{v},\sigma}(\mathbf{x})}{\rho^m_\sigma(\mathbb{Z}^m)}$, where $\rho^m_\sigma(\mathbb{Z}^m) = \sum_{\mathbf{z}\in\mathbb{Z}^m} \rho^m_\sigma(\mathbf{z})$.*

The discrete Gaussian distribution over the ring $\mathcal{R}$, denoted as $\mathcal{R}(D_\sigma)$, is the distribution over $\mathcal{R}$ where each coefficient is distributed according to D_σ.

Lemma 3 *(Lemma 4.4 of [7])*

1. *For any $k > 0$, $\Pr[|z| > k\sigma; z \xleftarrow{\$} D^1_\sigma] \leq 2e^{\frac{-k^2}{2}}$,*
2. *For any $\mathbf{z} \in \mathbb{Z}^m$, and $\sigma \geq \frac{3}{\sqrt{2\pi}}$, $D^m_\sigma(\mathbf{z}) \leq 2^{-m}$*
3. *For any $k > 1$, $\Pr[\|\mathbf{z}\| > k\sigma\sqrt{m}; \mathbf{z} \xleftarrow{\$} D^m_\sigma] < k^m e^{\frac{m}{2}(1-k^2)}$.*

Lemma 4 *(Lemma 2 of [4]) Let $\tilde{\sigma} > 0$ and $y \in \mathbb{Z}$. The statistical distance between $D_{\tilde{\sigma}}$ and $D_{\tilde{\sigma}} + y$ is at most $|y|/\tilde{\sigma}$*

We employ the following rejection sampling algorithm and lemma in our ZKPoK construction.

Lemma 5 *(Lemma 1 [6]). Let $\mathbf{B} \in \mathcal{R}^{r\times n}$ be any matrix. Consider a procedure that samples a $\mathbf{Y} \leftarrow D^{r\times n}_\sigma$ and then returns the output of $\mathsf{Rej}(\mathbf{Z} := \mathbf{Y} + \mathbf{B}, \mathbf{B}, \sigma, \rho)$ where $\sigma \geq \frac{12}{\ln\rho} \cdot \|\mathbf{B}\|$. The probability that this procedure outputs 1 is within 2^{-100} of $1/\rho$. The distribution of $\mathbf{Z}$, conditioned on the output being 1, is within statistical distance of 2^{-100} of $D^{r\times n}_\sigma$.*

Algorithm 1 $\mathrm{Rej}(\mathbf{Z}, \mathbf{B}, \sigma, \rho)$

$u \leftarrow [1, 0)$
if $u > \frac{1}{\rho} \cdot e^{\frac{-2\langle \mathbf{Z}, \mathbf{B} \rangle + \|\mathbf{B}\|_2^2}{2\sigma^2}}$ **then return** 0
else return 1
end if

3.3 Computational Assumptions

In this Section, we will define the problems upon whose security our proof system will be based: ring-LWE [16] and SIS [17].

Definition 6 *(Definition 2 [4]) Let $q, m, \tilde{n}, \tilde{\sigma} > 0$ depend on κ ($q, m, \tilde{n}$ are integers). The decision-RLWE problem ($\mathsf{dRLWE}_{q,\tilde{n},m,\tilde{\sigma}}$) is to distinguish between:*

$$(a_i, a_i \cdot s + e_i)_{i \in [m]} \in (\mathcal{R}_q)^2 \quad \textit{and} \quad (a_i, u_i)_{i \in [m]} \in (\mathcal{R}_q)^2 \quad \textit{for} \quad a_i, u_i \leftarrow \mathcal{R}_q;\ s, e_i \leftarrow \mathcal{R}(D_{\tilde{\sigma}}).$$

We sometimes write $\mathsf{dRLWE}_{q,\tilde{n},\tilde{\sigma}}$, leaving the parameter m implicit.

Definition 7 *(Definition 3.4 [18]) The One-Dimensional Short Integer Solution problem, denoted $\mathsf{1D\text{-}SIS}_{q,m,t}$, is the following problem. Given a uniformly distributed vector $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_q^m$, find $z \in \mathbb{Z}^m$ such that $\|\mathbf{z}\| \leq t$ and also $\langle \mathbf{v}, \mathbf{z} \rangle \in [-t, t] + q\mathbb{Z}$.*

3.4 Zero-Knowledge Proof Systems

As stated in Chapter 1, we wish to give a proof system to be used as the non-interactive zero-knowledge argument of knowledge (NIZKAoK) used in our VOPRF protocol. The proof system must therefore meet the following requirements for an NIZKAoK. Note that we give a zero-knowledge *proof* of knowledge, which differs from a zero-knowledge *argument* of knowledge in that an argument of knowledge is computationally sound whereas a proof of knowledge is statistical sound.

Definition 8 *(NIZKoK [4]) Let $\mathbb{P}$ be a prover, let $\mathbb{V}$ be a verifier, let $\mathcal{L}$ be a language with accompanying relation predicate $\mathsf{P}_{\mathcal{L}}(\cdot,\cdot)$. Let $\mathcal{W}_{\mathcal{L}}(x)$ be a generic set of witnesses attesting to the fact that $x \in \mathcal{L}$, i.e. $\forall x \in \mathcal{L}$, and $w \in \mathcal{W}_{\mathcal{L}}(x)$ we have $\mathsf{P}_{\mathcal{L}}(x,w) = 1$. Let $\mathsf{nizk} = (\mathsf{Setup}, \mathbb{P}, \mathbb{V})$ be a tuple of algorithms defined as follows:*

- $\mathsf{crs} \leftarrow \mathsf{nizk.Setup}(1^\kappa)$*: outputs a common random string* crs*.*
- $\pi \leftarrow \mathsf{nizk}.\mathbb{P}(\mathsf{crs}, x, w)$*: on input* crs*, a word $x \in \mathcal{L}$ and a witness $w \in \mathcal{W}_{\mathcal{L}}(x)$; outputs a proof $\pi \in \{0,1\}^{\mathsf{poly}(\kappa)}$.*
- $b \leftarrow \mathsf{nizk}.\mathbb{V}(\mathsf{crs}, x, \pi)$*: on input* crs*, a word $x \in \mathcal{L}$ and a proof $\pi \in \{0,1\}^{\mathsf{poly}(\kappa)}$; outputs $b \in \{0,1\}$.*

where κ is the security parameter.

Definition 9 *(NIZKoK Security [4]) We say that* nizk *is a non-interactive zero-knowledge argument of knowledge (NIZKAoK) for $\mathcal{L}$ if the following holds.*

1. *(Completeness) Consider $x \in \mathcal{L}$ and $w \in \mathcal{W}_{\mathcal{L}}(x)$, where $\mathsf{P}_{\mathcal{L}}(x,w) = 1$. Then:*

$$\Pr\left[1 \leftarrow \mathsf{nizk}.\mathbb{V}(\mathsf{crs}, x, \pi) \middle| \mathsf{crs} \leftarrow \mathsf{nizk.Setup}(1^\kappa), \pi \leftarrow \mathsf{nizk}.\mathbb{P}(\mathsf{crs}, x, w)\right] \geq 1 - \mathsf{negl}(\kappa)$$

2. *(Computational knowledge Soundness): The proof system satisfies computational knowledge extraction with knowledge error $\bar{\kappa}$ if, for any PPT prover $\mathbb{P}^*$ with auxiliary information* aux*, the following holds. There exists a PPT algorithm* $\mathsf{nizk.Extract}$ *and a polynomial p such that, for any input x, then:*

$$\Pr[1 \leftarrow \mathsf{P}_{\mathcal{L}}(x, w') | w' \leftarrow \mathsf{nizk.Extract}(\mathbb{P}^*(\mathsf{crs}, x, \mathsf{aux}))] \geq \frac{v - \bar{\kappa}}{p(|x|)}$$

is satisfied, where v is the probability that $\mathsf{nizk}.\mathbb{V}(\mathsf{crs}, x, \mathbb{P}^*(\mathsf{crs}, x, \mathsf{aux}))$ *outputs 1.*

3. *(Computational zero-knowledge) There exists a simulated setup algorithm* $\mathsf{nizk.SimSetup}(1^\kappa)$ *outputting* $\mathsf{crs}_{\mathsf{Sim}}$ *and a trapdoor* $\mathcal{T}$ *along with a PPT algorithm* $\mathsf{nizk.Sim}(\mathsf{crs}_{\mathsf{Sim}}, \mathcal{T}, x)$ *satisfying*

$$\left\{ \begin{array}{c} \mathsf{crs} \leftarrow \mathsf{nikz.Setup}(1^\kappa) \\ \pi \leftarrow \mathsf{nizk}.\mathbb{P}(\mathsf{crs}, x, w) \end{array} \right\} \approx_c \left\{ \begin{array}{c} \mathsf{crs}_{\mathsf{Sim}} \\ \pi_{\mathsf{Sim}} \leftarrow \mathsf{nizk.Sim}(\mathsf{crs}_{\mathsf{Sim}}, \mathcal{T}, x) \end{array} \middle| (\mathsf{crs}_{\mathsf{Sim}}, \mathcal{T}) \leftarrow \mathsf{nizk.SimSetup}(1^\kappa) \right\}$$

$\forall x \in \mathcal{L}$ *and* $w \in \mathcal{W}_{\mathcal{L}}(x)$.

where $\mathsf{negl}(\kappa)$ *denotes a negligible function (i.e. a function that is* $\kappa^{-\omega(1)}$*).*

3.5 Pseudorandom Function

In this Section we introduce the PRF we use in our VOPRF and definitions and lemmas we make us of in the proof of security of our VOPRF. A pseudorandom function (PRF) is a function that, given an input and a secret key, produces an output that appears random.

Definition 10 *(Psuedorandom Function [19]). An efficient, length-preserving, keyed function* $F : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^*$ *is a psuedorandom function if for all probabilistic polynomial-time adversaries* A*, there exists a negligible function* negl *such that*

$$\left| Pr[A^{F_k(\cdot)}(1^n) = 1] - Pr[A^{f(\cdot)}(1^n) = 1] \right| \leq \mathsf{negl}(n)$$

where $k \leftarrow \{0,1\}^n$ *is chosen uniformly at random and* f *is chosen uniformly at random from the set of functions mapping* n*-bit strings to* n*-bit strings.*

We next define the gadgets matrices G, G^{-1}, which are used in the PRF of our VOPRF. Define $G : \mathcal{R}_q^{\tilde{\ell} \times \tilde{\ell}} \rightarrow \mathcal{R}_q^{1 \times \tilde{\ell}}$ to be the linear operation corresponding to left multiplication by $(1, 2, \ldots, 2^{\tilde{\ell}-1})$. Further, define $G^{-1} : \mathcal{R}_q^{1 \times \tilde{\ell}} \rightarrow \mathcal{R}_q^{\tilde{\ell} \times \tilde{\ell}}$ to be the bit decomposition operation

that essentially inverts G i.e. the i^{th} column of $G^{-1}(\mathbf{a})$ is the bit decomposition of $a_i \in \mathcal{R}_q$ into binary polynomials.

Our VOPRF utilizes Banerjee and Peikert's construction [20] of a Ring-LWE-based psuedorandom function. The key of the PRF is an element k of the ring $\mathcal{R}_q = \mathbb{Z}_q[x]/(x^{\tilde{n}} + 1)$, and the input to the PRF is $x \in \{0,1\}^L$. The PRF is defined as $F_k(x) = \lfloor \mathbf{a}_x \cdot k \rceil_p$ where $\mathbf{a}_x \in \mathcal{R}_q^{1\times\tilde{\ell}}$ and $\tilde{\ell} = \lceil \log q \rceil$.

Definition 11 *(Definition 6 of [4]) Fix some $\mathbf{a}_0, \mathbf{a}_1 \in \mathcal{R}_q^{1\times\tilde{\ell}}$. For any $x = (x_1 \ldots, x_L) \in \{0,1\}^L$, define $\mathbf{a}_x \in \mathcal{R}_q^{1\times\tilde{\ell}}$*

$$\mathbf{a}_x := \mathbf{a}_{x_1} \cdot G^{-1}(\mathbf{a}_{x_2} \cdot G^{-1}(\mathbf{a}_{x_3} \cdot G^{-1}(\ldots(\mathbf{a}_{x_{L-1}} \cdot G^{-1}(\mathbf{a}_{x_L}))\ldots))) \in \mathcal{R}_q^{\tilde{\ell}\times\tilde{\ell}}$$

The PRF is secure by the $\mathsf{dRLWE}_{q,\tilde{n},\tilde{\sigma}}$ assumption.

Theorem 12 *([20]) Sample $k \leftarrow \mathcal{R}(D_{\tilde{\sigma}})$. If $q \gg p \cdot \tilde{\sigma} \cdot \sqrt{L} \cdot \tilde{n} \cdot \tilde{\ell}$, then the function $F_k(x) = \lfloor \mathbf{a}_x \cdot k \rceil_p$ is a PRF under the $\mathsf{dRLWE}_{q,\tilde{n},\tilde{\sigma}}$ assumption*

The following definitions and lemmas will be used in the security proof of our VOPRF.

Definition 13 *(Definition 7 of [4]) For $\mathbf{a}_0, \mathbf{a}_1 \in \mathcal{R}_q^{1\times\tilde{\ell}}$, define*

$$\mathbf{a}_{x\setminus i} := G^{-1}(\mathbf{a}_{x_{i+1}} \cdot G^{-1}(\mathbf{a}_{x_{i+2}} \cdot G^{-1}(\ldots(\mathbf{a}_{x_{L-1}} \cdot G^{-1}(\mathbf{a}_{x_L}))\ldots))) \in \mathcal{R}_q^{\tilde{\ell}\times\tilde{\ell}}$$

Furthermore, let $\mathcal{E}_{\mathbf{a}_0,\mathbf{a}_1,x,\tilde{\sigma}}$ be the distribution that is sampled by choosing $\mathbf{e}_i \leftarrow \mathcal{R}(D_{\tilde{\sigma}})^{1\times\tilde{\ell}}$ for $i = 1, \ldots, L$ and outputting

$$\mathbf{e} = \sum_{i=1}^{L-1} \mathbf{e}_i \cdot \mathbf{a}_{x\setminus i} + \mathbf{e}_L$$

Lemma 14 *(Lemma 3 of [4]) If $\mathbf{a}_0, \mathbf{a}_1 \leftarrow \mathcal{R}_q^{1\times\tilde{\ell}}, \mathbf{e} \leftarrow \mathcal{E}_{\mathbf{a}_0,\mathbf{a}_1,x,\tilde{\sigma}}$ and $s \leftarrow \mathcal{R}(D_{\tilde{\sigma}})$, then for any fixed $x \in \{0,1\}^L$,*

$$(\mathbf{a}_0, \mathbf{a}_1, \mathbf{a}_0, \mathbf{a}_x \cdot s + e)$$

is indistinguishable from uniform random by the $\mathsf{dRLWE}_{q,\tilde{n},\tilde{\sigma}}$ assumption.

Lemma 15 *(Lemma 4 of [4]) Let $x \in \{0,1\}^L$, $\tilde{\ell} = \lceil \log_2 q \rceil$ and $\tilde{n} = poly(\kappa)$. Samples from $\mathcal{E}_{\mathbf{a}_0,\mathbf{a}_1,x,\tilde{\sigma}}$ have infinity norm at most $L \cdot \tilde{\ell}\tilde{\sigma}\tilde{n}^{3/2}$ with all but negligible probability.*

Lemma 16 *(Lemma 5 of [4]) Fix any $x \in \{0,1\}^L$. Suppose there exists a PPT algorithm $\mathcal{D}_x(\mathbf{a}_0, \mathbf{a}_1)$ that outputs $r \in \mathcal{R}$ such that $\|r\|_\infty \leq B$ and at least one coefficient of $\mathbf{a}_x \cdot r$ is in the set $\left(\frac{q}{p}\right) \cdot \mathbb{Z} + [-T, T]$ with non-negligible probability (over a uniform choice of $\mathbf{a}_0, \mathbf{a}_1 \leftarrow \mathcal{R}_q^{\tilde{\ell}}$ and its random coins). Then there exists an efficient algorithm solving $\mathsf{1D\text{-}SIS}_{q/p,\tilde{n}\tilde{\ell},\max\{\tilde{n}\tilde{\ell}B,T\}}$ with non-negligible probability.*

3.6 Verifiable Oblivious Pseudorandom Function

Informally, a Verifiable Oblivious Pseudorandom Function (VOPRF) is a cryptographic protocol that allows a client $\mathbb{C}$ to securely obtain pseudorandom function evaluations from a server $\mathbb{S}$, with the ability to verify the correctness of the output, while keeping the client's input private from the server.

The formal Definition 17 of a VOPRF considers two scenarios: the "real" world and the "ideal" world. In the real world protocol Π, an adversary $\mathcal{A}(k)$ corrupting the server $\mathbb{S}(k)$ interacts directly with the client $\mathbb{C}(x)$ (or $\mathcal{A}(x)$ interacts directly with the server $\mathbb{S}(k)$ if the client $\mathbb{C}(x)$ is the corrupted party). We denote the joint output distribution of $\mathcal{A}(k)$ and $\mathbb{C}(x)$ as $\mathsf{real}_{\Pi,\mathcal{A},\mathbb{S}}(x, k, 1^\kappa)$ if the server is corrupted, and as $\mathsf{real}_{\Pi,\mathcal{A},\mathbb{C}}(x, \mathcal{K}, 1^\kappa)$ if the client is corrupted, where $\mathcal{K}$ is the distribution of keys under which the pseudorandom function F maintains its security.

In the "ideal" world, a simulator Sim acts as an intermediary, interacting with the adversary and the ideal functionality F_{VOPRF} (see Figure 3.1), producing the distribution $\mathsf{ideal}_{F_{\mathsf{VOPRF}},\mathsf{Sim},\mathcal{A},\mathbb{S}}(x, k, 1^\kappa)$ if the server is corrupted, and $\mathsf{ideal}_{F_{\mathsf{VOPRF}},\mathsf{Sim},\mathcal{A},\mathbb{C}}(x, \mathcal{K}, 1^\kappa)$ if the client is corrupted. Security is achieved if the adversary's influence in the real world can be emulated by Sim in the ideal world, making the two distributions indistinguishable to $\mathcal{A}$.

Let $\mathsf{output}(\Pi, x, k)$ denotes the output distribution of a client with input x running pro-

tocol Π with a server whose input key is k.

This is a two party functionality between a server $\mathbb{S}$ and a client $\mathbb{C}$. We assume there is a fixed PRF function defined by $F_k(x)$.
Init-S: On input of init from the server, the functionality waits for an input k from party $\mathbb{S}$. If $\mathbb{S}$ returns abort, then the functionality aborts. Otherwise, the functionality stores the value k if the key conforms to the pre-determined distribution and aborts if not.
Init-C: On input of init from a client, the functionality will return abort if the init procedure for the server has not successfully completed.
Query: On input of (query, x) from a client $\mathbb{C}$, if $x \neq \perp$ then the functionality waits for an input from party $\mathbb{S}$. If $\mathbb{S}$ returns deliver, then the functionality sends $y = F_k(x)$ to party $\mathbb{C}$. If $\mathbb{S}$ returns abort, then the functionality aborts.

Figure 3.1: The Ideal Functionality F_{VOPRF}

Definition 17 *(Definition 1 [4]) A protocol* Π *is a verifiable oblivious pseudorandom function if all of the following hold:*

1. ***Correctness:*** *For every pair of inputs* (x, k)*,*

$$\Pr[\mathsf{output}(\Pi, x, k) \neq F_k(x)] \leq \mathsf{negl}(\kappa).$$

2. ***Malicious server security:*** *For any PPT adversary* $\mathcal{A}$ *corrupting a server, there exists a PPT simulator* Sim *such that for every pair of inputs* (x, k)*:*

$$\mathsf{ideal}_{F_{\mathsf{VOPRF}},\mathsf{Sim},\mathcal{A},\mathbb{S}}(x, k, 1^\kappa) \approx_c \mathsf{real}_{\Pi,\mathcal{A},\mathbb{S}}(x, k, 1^\kappa).$$

3. ***Average case malicious client security:*** *For any PPT adversary* $\mathcal{A}$ *corrupting a client, there exists a PPT simulator* Sim *such that for all client inputs* x*:*

   - $\mathsf{ideal}_{F_{\mathsf{VOPRF}},\mathsf{Sim},\mathcal{A},\mathbb{C}}(x, \mathcal{K}, 1^\kappa) \approx_c \mathsf{real}_{\Pi,\mathcal{A},\mathbb{C}}(x, \mathcal{K}, 1^\kappa)$.
   - *If* $\mathcal{A}$ *correctly outputs* $F_k(x)$ *with all but negligible probability over the choice* $k \leftarrow \mathcal{K}$ *when interacting directly with* $\mathbb{S}(k)$ *using protocol* Π*, then* $\mathcal{A}$ *also outputs*

$F_k(x)$ *with all but negligible probability when interacting via* Sim.

Chapter 4

Zero-Knowledge Proof of Knowledge (ZKPoK)

In this chapter we adapt the proof system of [6] to give a zero-knowledge proof of knowledge for the following relation, where $B_{\mathsf{yes}} \leq B_{\mathsf{no}}$.

$$R_{\mathsf{yes}} = \{((\mathbf{A}, \mathbf{T}), \mathbf{S}) \in \mathcal{R}_q^{r \times v} \times \mathcal{R}^{v \times \ell} \times \mathcal{R}_q^{r \times \ell} \wedge \mathbf{AS} = \mathbf{T} \wedge [\|\mathbf{s}_i\|_\infty \leq B_{\mathsf{yes}}]_{i \in [\ell]}\}$$

and

$$R_{\mathsf{no}} = \{((\mathbf{A}, \mathbf{T}), \mathbf{S}) \in \mathcal{R}_q^{r \times v} \times \mathcal{R}^{v \times \ell} \times \mathcal{R}_q^{r \times \ell} \wedge \mathbf{AS} = \mathbf{T} \wedge [\|\mathbf{s}_i\|_\infty \leq B_{\mathsf{no}}]_{i \in [\ell]}\}$$

where $\mathbf{s}_i$ are the columns of $\mathbf{S}$.

In other words, if a prover holds $\mathbf{S}$ such that $((\mathbf{A}, \mathbf{T}), \mathbf{S}) \in R_{\mathsf{yes}}$, then completeness and zero-knowledge hold. On the other hand, there is an extractor that extracts an $\mathbf{S}$ such that $((\mathbf{A}, \mathbf{T}), \mathbf{S}) \in R_{\mathsf{no}}$ given any malicious convincing prover. We refer to the gap between B_{yes} and B_{no} as the soundness gap.

Theorem 18 *Let $\mathcal{R} = \mathbb{Z}$, $\mathcal{C} = \{0, 1\}$, and λ be a security parameter. Let $v, r = \mathsf{poly}(\lambda)$, $n \geq \lambda + 2$, $s > 0$ be an upper bound on $s_1(\mathbf{S})$, $\rho > 1$ be a constant, $\sigma \in \mathbb{R}$ be such that $\sigma \geq \frac{12}{\ln \rho} s\sqrt{\ell n}$, and $B = \sqrt{8v}\sigma$. Then the protocol described in Figure 4.1 is a zero-knowledge*

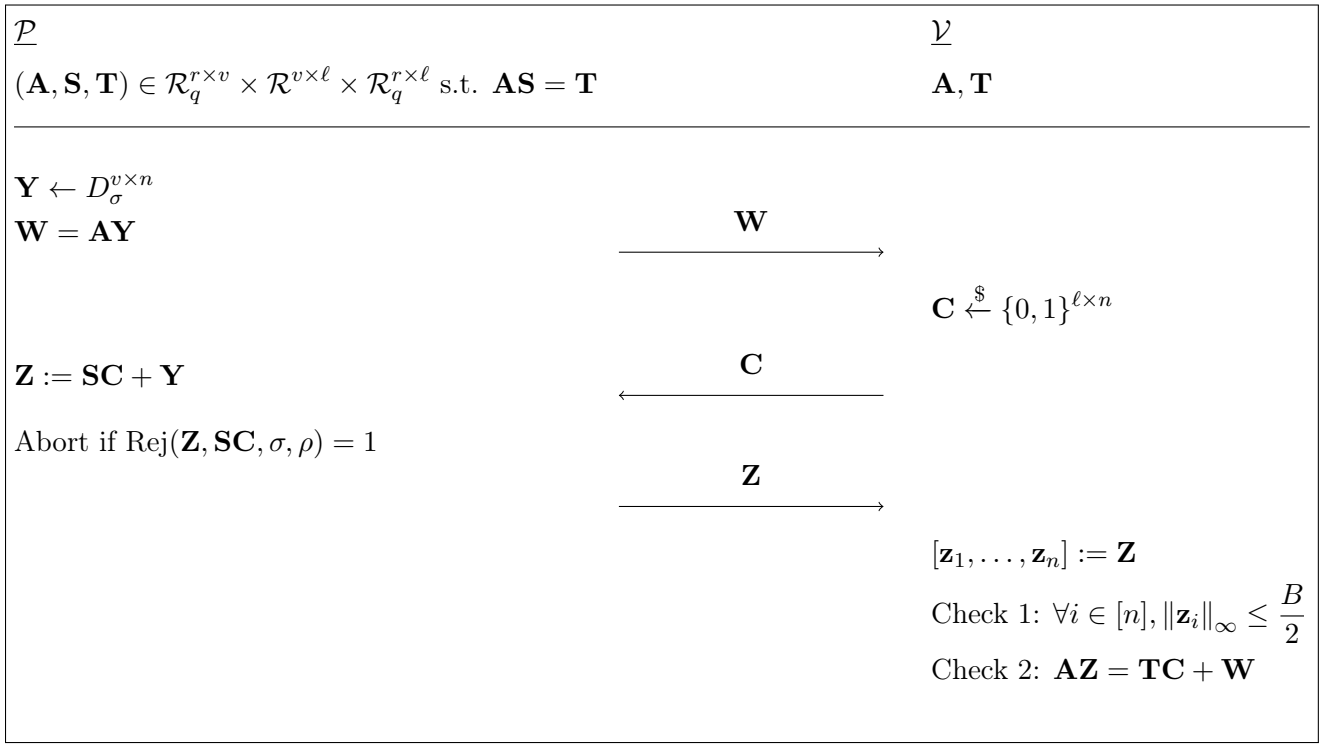

Figure 4.1: SHVZKPoK protocol

proof of knowledge for the gap relation $(R_{\mathsf{yes}}, R_{\mathsf{no}})$ *achieving* $B_{\mathsf{yes}} = \frac{s}{\sqrt{v\ell}}$ *and* $B_{\mathsf{no}} = B$*, i.e.* B_{yes} *and* B_{no} *that satisfy*

$$B_{\mathsf{no}} \geq B_{\mathsf{yes}} \frac{12}{\ln \rho} v\ell\sqrt{8n}$$

4.1 Completeness

If $\mathcal{P}$ and $\mathcal{V}$ are honest then $\|\mathbf{SC}\|_2 \leq s_1(\mathbf{S})\|\mathbf{C}\|_2 \leq s\sqrt{ln}$. Since $\sigma \geq \frac{12}{\ln \rho} s\sqrt{\ell n}$, by Lemma 5 the probability of abort is exponentially close to $1 - \frac{1}{\rho}$ and each coefficient of $\mathbf{Z}$ is statistically close to D_σ. Since $\|\mathbf{z}_i\|_\infty \leq \|\mathbf{z}_i\|_2$ it follows that

$$\Pr_{\mathbf{z} \xleftarrow{\$} D_\sigma^m}\left[\|\mathbf{z}_i\|_\infty > \frac{B}{2}\right] < \Pr_{\mathbf{z} \xleftarrow{\$} D_\sigma^m}\left[\|\mathbf{z}_i\|_2 > \frac{B}{2}\right]$$

so by Lemma 3 we have $\|\mathbf{z}_i\|_\infty \leq \frac{B}{2}$ with overwhelming probability. The verifier's second check holds the by construction of $\mathbf{Z}$.

4.2 Soundness

For soundness, we recall the knowledge extractor given in [6], except we note that the extracted witness S' will have norm $\forall i \in [\ell]\ \|\mathbf{s}'_i\|_\infty \leq B$. Given a prover $\mathcal{P}^*$ who succeeds with probability $\epsilon > 2^{-\lambda}$, we may extract a witness S' such that $\mathbf{AS}' = \mathbf{T}$ and $\forall i \in [\ell]\ \|\mathbf{s}'_i\|_\infty \leq B$ as follows. Let $\mathbf{c}_i^T \in \mathcal{R}^{1\times n}$ denote the i^{th} row of a challenge matrix $\mathbf{C}$.

1. Run $\mathcal{P}^*$ on random challenges $\mathbf{C}'$ until it succeeds to obtain an accepting $\mathbf{Z}'$.
2. Run $\mathcal{P}^*$ on random challenges $\mathbf{C}''$ where $\forall j \neq i$, $\mathbf{c}''^T_j = \mathbf{c}'^T_j$ and $\mathbf{c}''^T_i$ is freshly sampled. Abort if $\mathcal{P}^*$ does not produce an valid $\mathbf{Z}''$ after $\frac{\lambda}{\epsilon}$ attempts.

By executing these steps $\mathcal{O}(\lambda)$ times, we obtain a valid $\mathbf{C}', \mathbf{Z}', \mathbf{C}'', \mathbf{Z}''$ such that $\forall j \neq i$, $\mathbf{c}''^T_j = \mathbf{c}'^T_j$ and $\mathbf{c}''^T_i \neq \mathbf{c}'^T_i$ with probability greater than $\frac{1}{2} + 2^{-\lambda}$ in expected $\frac{poly(\lambda)}{\epsilon}$ time by the heavy-row argument of [6]. From these valid pairs, we construct the equation

$$\mathbf{A}(\mathbf{Z}' - \mathbf{Z}'') = \mathbf{t}_i(\mathbf{c}'^T_i - \mathbf{c}''^T_i)$$

where $\mathbf{t}_i$ is the i^{th} column of $\mathbf{T}$ and $\mathbf{z}_i$ is the i^{th} column of $\mathbf{Z}' - \mathbf{Z}''$. Since $\mathbf{c}''^T_i \neq \mathbf{c}'^T_i$ for some index, we have a solution $\mathbf{A}(\mathbf{z}'_i - \mathbf{z}''_i) = \pm\mathbf{t}_i$ where $\|\mathbf{z}'_i - \mathbf{z}''_i\|_\infty \leq B$. We may then run the steps above for each column $i \in \ell$ to obtain the full witness $\mathbf{S}'$.

4.3 Zero-Knowledge

For the zero-knowledge property of the protocol, we provide a probabilistic polynomial time (PPT) algorithm $\mathcal{S}$ (a simulator). We define $\mathcal{S}$ as follows.

1. sample $\mathbf{C} \xleftarrow{\$} \{0,1\}^{\ell\times n}$
2. sample $\mathbf{Z} \xleftarrow{\$} D_\sigma^{v\times n}$
3. Set $\mathbf{W} = \mathbf{AZ} - \mathbf{TC}$ and output $(\mathbf{W}, \mathbf{Z}, \mathbf{C})$

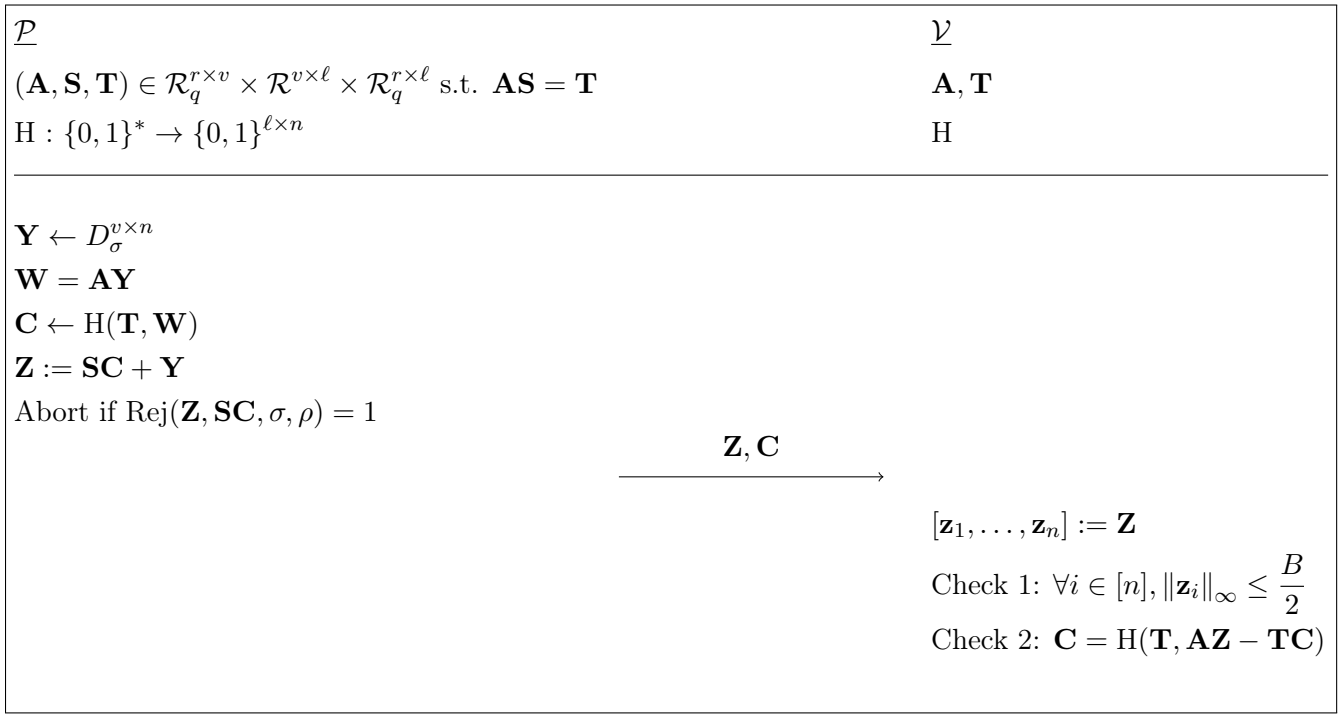

Figure 4.2: NIZKPoK protocol

It is clear that $\mathbf{Z}$ verifies with overwhelming probability. By Lemma 5 we know that in the real protocol when no abort occurs the distribution of $\mathbf{Z}$ is within statistical distance 2^{-100} of $D^{v\times n}$. Since $\mathbf{W}$ is completely determined by $\mathbf{A}$, $\mathbf{T}$, $\mathbf{Z}$ and $\mathbf{C}$, the distribution of $(\mathbf{W}, \mathbf{Z}, \mathbf{C})$ output by $\mathcal{S}$ is within 2^{-100} of the distribution of these variables in the actual non-aborting run of the protocol.

Since this is a 3-round, honest-verifier, public coin proof of knowledge with negligible soundness, we apply the apply the Fiat-Shamir transform [21] to get a non-interactive proof of knowledge that is zero-knowledge in the random oracle model. The transformed version of the protocol is given in Figure 4.2.

Chapter 5

Verifiable Oblivious Pseudorandom Function (VOPRF)

Here we present the construction and security proof of our VOPRF. As mentioned in Chapter 3, our VOPRF operates over the polynomial ring $\mathcal{R}_q = \mathbb{Z}_q[x]/(x^{\tilde{n}} + 1)$. We use tildes to distinguish parameters in the VOPRF protocol that are reused from our ZKPoK protocol to prevent confusion.

Our construction requires three NIZKAoKs to establish that all computations are performed honestly:

- **Server Proof in Initialization Phase ($\mathbb{P}_0$):** Proves knowledge of $k \in \mathcal{R}$ and $\mathbf{e} \in \mathcal{R}^{1\times\tilde{\ell}}$ with $\|k\|_\infty, \|\mathbf{e}\|_\infty \leq \tilde{\sigma} \cdot \sqrt{\tilde{n}}$, such that $\mathbf{c} = \mathbf{a} \cdot k + \mathbf{e} \mod q$, where crs_0 contains $\mathbf{a}$.
- **Client Proof ($\mathbb{P}_1$):** Proves knowledge of $x \in \{0,1\}^L$, $s \in \mathcal{R}$ with $\|s\|_\infty \leq \tilde{\sigma} \cdot \sqrt{\tilde{n}}$, and $\mathbf{e}_1 \in \mathcal{R}^{1\times\tilde{\ell}}$ with $\|\mathbf{e}_1\|_\infty \leq \tilde{\sigma} \cdot \sqrt{\tilde{n}}$, such that $\mathbf{c}_x = \mathbf{a} \cdot s + \mathbf{e}_1 + \mathbf{a}_x \mod q$.
- **Server Proof in Query Phase ($\mathbb{P}_2$):** Proves knowledge of $k \in \mathcal{R}$ with $\|k\|_\infty \leq \tilde{\sigma} \cdot \sqrt{\tilde{n}}$, $\mathbf{e} \in \mathcal{R}^{1\times\tilde{\ell}}$ with $\|\mathbf{e}\|_\infty \leq \tilde{\sigma} \cdot \sqrt{\tilde{n}}$, and $\mathbf{e}' \in \mathcal{R}^{1\times\tilde{\ell}}$ with $\|\mathbf{e}'\|_\infty \leq \tilde{\sigma}' \cdot \sqrt{\tilde{n}}$, such that $\mathbf{c} = \mathbf{a} \cdot k + \mathbf{e} \mod q$ and $\mathbf{d}_x = \mathbf{c}_x \cdot k + \mathbf{e}' \mod q$.

In Chapter 6, we explain how to instantiate proof system 0 $(\mathbb{P}_0, \mathbb{V}_0)$ of our VOPRF using

the ZKPoK given in Chapter 4. Proof systems 1 and 2 can be instantiated using the proof system of [5] as detailed in [4]. We note that proof system 2 can also be instantiated with a version of our ZKPoK, in which the verifier uses different bounds for different components of the prover's output $\mathbf{Z}$, allowing each component of the secret to have its own size bound.

Our construction is given in Figure 5.1 and we proceed with the security proof below.

Theorem 19 *Assume $p|q$. The protocol in Figure 5.1 is a secure VOPRF protocol (according to Definition 17) if the following conditions hold:*

- $\forall i \in \{0,1,2\}$, $(\mathbb{P}_i, \mathbb{V}_i)$ *is a NIZKAoK*
- $\mathsf{dRLWE}_{q,\tilde{n},\tilde{\sigma}_{\mathsf{server}}}$ *is hard*
- $\frac{q}{2p} \gg \tilde{\sigma}' \gg \max\{L \cdot \tilde{\ell} \cdot \tilde{\sigma}_{\mathsf{server}} \cdot \tilde{n}^{3/2}, \tilde{\sigma} \cdot \tilde{\sigma}_{\mathsf{server}} \tilde{n}^2\}$
- $\mathsf{1D\text{-}SIS}_{\frac{q}{2p},\tilde{n}\tilde{\ell},\max\{\tilde{n}^{3/2}\tilde{\ell}\tilde{\sigma},2\tilde{\sigma}^2\cdot\tilde{n}^2+\tilde{\sigma}'\sqrt{\tilde{n}}\}}$ *is hard*

5.1 Correctness

Lemma 20 *Assume an honest client and server. Define $T := 2\tilde{\sigma}_{\mathsf{server}} \cdot \tilde{\sigma} \cdot \tilde{n}^2 + \tilde{\sigma}'\sqrt{\tilde{n}}$. For any $x \in \{0,1\}^L$, $k \in \mathcal{R}_q$ such that $\|k\|_\infty \leq \tilde{\sigma}_{\mathsf{server}} \cdot \sqrt{\tilde{n}}$, we have that*

$$\Pr[\mathbf{y}_x \neq F_k(x)] \leq \mathsf{negl}(\kappa)$$

over the choice of PRF parameters $\mathbf{a}_0, \mathbf{a}_1 \leftarrow \mathcal{R}_q^{1\times\tilde{\ell}}$ assuming the hardness of $\mathsf{1D\text{-}SIS}_{q/p,\tilde{n}\tilde{\ell},\max\{\tilde{n}^{3/2}\tilde{\ell}\tilde{\sigma}_{\mathsf{server}},T\}}$.

Assume there exists a k' such that $\|k'\|_\infty \leq \tilde{\sigma}_{\mathsf{server}} \cdot \sqrt{\tilde{n}}$ where $\Pr[\mathbf{y}_x \neq F_{k'}(x)]$ is non-negligible over the choice of $\mathbf{a}_0, \mathbf{a}_1 \leftarrow \mathcal{R}_q^{1\times\tilde{\ell}}$. Expanding $\mathbf{c}$ and $\mathbf{d}_x$ from the protocol, we have that

$$\mathbf{y}_x = \lfloor \mathbf{a}_x \cdot k' + \mathbf{e}_1 \cdot k' + \mathbf{e}' - \mathbf{e} \cdot s \rceil_p.$$

CRS SetUp: To set up the CRS execute the following steps:

– Pick $\mathbf{a}_0, \mathbf{a}_1 \leftarrow \mathcal{R}_q^{1\times\tilde{\ell}}$

– Sample $\mathbf{a} \leftarrow \mathcal{R}_q^{1\times\tilde{\ell}}$, sample $\overline{\mathsf{crs}}_0$ for proof system $\mathbb{P}_0$ and set $\overline{\mathsf{crs}}_0 := (\mathsf{crs}_0, \mathbf{a})$

– Sample crs_1 and crs_2 for proof systems $\mathbb{P}_1$ and $\mathbb{P}_2$ respectively

Init: The initialisation procedure is executed by the server $\mathbb{S}$ and a client $\mathbb{C}$ both with initial input crs_0 .

– Init-S: The server $\mathbb{S}$ executes the following steps

$k \leftarrow \mathcal{R}(D_{\tilde{\sigma}_{\mathsf{server}}})$, $\mathbf{e} \leftarrow \mathcal{R}(D_{\tilde{\sigma}_{\mathsf{server}}})^{1\times\tilde{\ell}}$, $a \leftarrow \mathcal{R}_q^{1\times\tilde{\ell}}$.

$\mathbf{c} \leftarrow \mathbf{a} \cdot k + \mathbf{e} \mod q$

$\pi_0 \leftarrow \mathbb{P}_0(k, e : \mathsf{crs}_0, \mathbf{c})$

and sends $(\mathbf{c}, \pi_0)$ to a client $\mathbb{C}$.

– Init-C: On receipt of $(\mathbf{c}, \pi_0)$ a client executes

$b \leftarrow \mathbb{V}_0(\mathsf{crs}_0, \mathbf{c}, \pi_0)$.

Output abort if $b = 0$, otherwise store $\mathbf{c}$.

Query: This is a two round protocol between a client and the server, with a client going first.

1. On input of $(x \in \{0,1\}^L, \mathsf{crs}_1, \mathsf{crs}_2)$ a client $\mathbb{C}$ executes the following steps

$s \leftarrow \mathcal{R}(D_{\tilde{\sigma}})$, $\mathbf{e}_1 \leftarrow \mathcal{R}(D_{\tilde{\sigma}})^{1\times\tilde{\ell}}$

$\mathbf{a}_x = \mathbf{a}_{x_1} \cdot G^{-1}(\ldots(\mathbf{a}_{x_{L-1}} \cdot G^{-1}(\mathbf{a}_{x_L}))\ldots) \mod q$

$\mathbf{c}_x \leftarrow \mathbf{a} \cdot s + \mathbf{e}_1 + \mathbf{a}_x \mod q$

$\pi_1 \leftarrow \mathbb{P}_1(x, s, \mathbf{e}_1 : \mathsf{crs}_1, \mathbf{c}_x, \mathbf{a}, \mathbf{a}_0, \mathbf{a}_1)$

and sends $(\mathbf{c}_x, \pi_1)$ to the server $\mathbb{S}$.

2. On receipt of $(\mathbf{c}_x, \pi_1)$ the server $\mathbb{S}$ executes the following steps

$b \leftarrow \mathbb{V}_1(\mathsf{crs}_1, \mathbf{c}_x, \mathbf{a}_0, \mathbf{a}_1, \pi_1)$

Output abort if $b = 0$

$\mathbf{e}' \leftarrow \mathcal{R}(D_{\tilde{\sigma}'})^{1\times\tilde{\ell}}$

$\mathbf{d}_x = \mathbf{c}_x \cdot k + \mathbf{e}' \mod q$

$\pi_2 \leftarrow \mathbb{P}_2(k, \mathbf{e}', \mathbf{e} : \mathsf{crs}_2, \mathbf{c}, \mathbf{d}_x, \mathbf{c}_x, \mathbf{a})$

and sends $(\mathbf{d}_x, \pi_2)$ to a client $\mathbb{C}$ while outputting $\perp$.

3. On receipt of $(\mathbf{d}_x, \pi_2)$ a client $\mathbb{C}$ executes

$b \leftarrow \mathbb{V}_2(\mathsf{crs}_0, \mathsf{crs}_2, \mathbf{c}, \mathbf{d}_x, \mathbf{c}_x, \pi_2)$

Output abort if $b = 0$

$\mathbf{y}_x = \lfloor \mathbf{d}_x - \mathbf{c} \cdot s \rceil_p$

Output $\mathbf{y}_x$

Figure 5.1: VOPRF protocol

It follows that there must be at least one coefficient of $\mathbf{a}_x \cdot k'$ in the set $(q/p) \cdot \mathbb{Z} + [T, T]$ with non-negligible probability, otherwise $\mathbf{y}_x = \lfloor \mathbf{a}_x \cdot k' \rceil_p =: F_{k'}(x)$. Note that $\mathbf{e}'' := \mathbf{e}_1 \cdot k' - \mathbf{e} \cdot s + \mathbf{e}'$ has infinity norm less than T as defined in the lemma statement with all but negligible probability. Applying Lemma 16 to the algorithm $\mathcal{D}_x(\mathbf{a}_0, \mathbf{a}_1)$ that ignores $\mathbf{a}_0, \mathbf{a}_1$ and simply outputs k' implies an efficient algorithm solving $\mathsf{1D\text{-}SIS}_{q/p, \tilde{n}\tilde{\ell}, \max\{\tilde{n}^{3/2}\tilde{\ell}\tilde{\sigma}_{\mathsf{server}}, T\}}$.

Next we prove the correctness of non-aborting malicious protocol runs, which is utilized in the malicious client proof.

Lemma 21 *Assume that* $\mathsf{dRLWE}_{q,\tilde{n},\tilde{\sigma}_{\mathsf{server}}}$ *is hard,* $\tilde{\sigma}$ *and* $\tilde{n}$ *are* $\mathsf{poly}(\kappa)$*, and* $\frac{q}{2p} \gg \tilde{\sigma}' \gg \max\{L \cdot \tilde{\ell} \cdot \tilde{\sigma}_{\mathsf{server}} \cdot \tilde{n}^{3/2}, \tilde{\sigma} \cdot \tilde{\sigma}_{\mathsf{server}} \tilde{n}^2\}$*. For any* $x \in \{0,1\}^L$*, consider a non-aborting run of the protocol in Figure 5.1 between a (potentially malicious) efficient client* $\mathbb{C}^*$ *and honest server* $\mathbb{S}$*. Further, let* s *be the value that is extractable from the client's proof in the query phase. Then, the value of* $\lfloor \mathbf{d}_x - \mathbf{c} \cdot s \rceil_p$ *is equal to* $\lfloor \mathbf{a}_{\mathbf{x}} \cdot k \rceil_p$ *with all but negligible probability.*

By the sercurity of the NIZKAoK, note that for a non-aborting protocol run, any efficient client $\mathbb{C}^*$ must have produced $\mathbf{c}_x$ correctly using some $x \in \{0,1\}^L$, s, $\mathbf{e}_1$ where $\|s\|_\infty, \|\mathbf{e}_1\|_\infty \leq \tilde{\sigma} \cdot \sqrt{\tilde{n}}$. Suppose that $\mathbf{e}_x \leftarrow \mathcal{E}_{\mathbf{a}_0, \mathbf{a}_1, x, \tilde{\sigma}_{\mathsf{server}}}$. If $\tilde{\sigma}' \gg \max\{L \cdot \tilde{\ell} \cdot \tilde{\sigma}_{\mathsf{server}} \cdot \tilde{n}^{3/2}, \tilde{\sigma}\tilde{\sigma}_{\mathsf{server}} \tilde{n}^2\}$, then $\mathbf{e}' \leftarrow \mathcal{R}(D_{\tilde{\sigma}'})^{1 \times \tilde{\ell}}$ and $(\mathbf{e}_x - \mathbf{e}_1 \cdot k - \mathbf{e} \cdot s) + \mathbf{e}'$ are statistically close by Lemma 15 and Lemma 4. Therefore, replacing $\mathbf{e}'$ by $(\mathbf{e}_x - \mathbf{e}_1 \cdot k - \mathbf{e} \cdot s) + \mathbf{e}'$ the client output equation in Figure 5.1 can be written as

$$\lfloor \frac{p}{q}(\mathbf{d}_x - \mathbf{c} \cdot s) \rceil = \lfloor \frac{p}{q}(\mathbf{a}_x \cdot k + \mathbf{e}_x) + \frac{p}{q}\mathbf{e}' \rceil$$

By Lemma 14, $\frac{p}{q}(\mathbf{a}_x \cdot k + \mathbf{e}_x)$ is computationally indistinguishable from uniform random over $\frac{p}{q}\mathcal{R}_q^{1 \times \tilde{\ell}}$ assuming the hardness of $\mathsf{dRLWE}_{q,\tilde{n},\tilde{\sigma}_{\mathsf{server}}}$. Thus every coefficient in $\frac{p}{q}(\mathbf{a}_x \cdot k + \mathbf{e}_x)$ is at least T away from $\mathbb{Z} + \frac{1}{2}$ with all but negligible probability for any $T \ll 1$. Setting $T = \frac{p}{q}(\tilde{\sigma}' \cdot \sqrt{\tilde{n}} + L \cdot \tilde{\ell} \cdot \tilde{\sigma}_{\mathsf{server}} \cdot \tilde{n}^{3/2}) \ll 1$ ensures that $T \leq \frac{p}{q} \cdot \|\mathbf{e}_x + \mathbf{e}'\|_\infty$ with all but negligible probability. It then follows that

$$\lfloor \frac{p}{q}(\mathbf{a}_x \cdot k + \mathbf{e}_x) + \frac{p}{q}\mathbf{e}' \rceil = \lfloor \frac{p}{q}(\mathbf{a}_x \cdot k) \rceil$$

5.2 Malicious Client and Server Proofs

For the malicious client, we note that the proofs are almost entirely unchanged from those presented in Sections 5.1 and 5.2 of [4], as the proof follows the same argument. The primary difference in our construction is that the honest server draws its key (k, e) from the distribution $D_{\tilde{\sigma}_{\mathsf{server}}}$, requiring a different setting of parameters in the assumptions needed for the proof to hold:

The client proof requires:

- $\mathsf{dRLWE}_{q,\tilde{n},\tilde{\sigma}_{\mathsf{server}}}$
- $\tilde{\sigma}' \gg L \cdot \tilde{\ell} \cdot \tilde{\sigma}_{\mathsf{server}} \cdot \tilde{n}^{3/2}$
- $\tilde{\sigma}' \gg \tilde{\sigma} \cdot \tilde{\sigma}_{\mathsf{server}} \cdot \tilde{n}^{2}$

The malicious server proof requires:

- $\mathsf{dRLWE}_{q,\tilde{n},\tilde{\sigma}}$
- $\mathsf{1D\text{-}SIS}_{q/2p,\tilde{n}\tilde{\ell},\max\{\tilde{n}^{3/2}\tilde{\ell}\tilde{\sigma},2\tilde{\sigma}^2\cdot\tilde{n}^2+\tilde{\sigma}'\sqrt{\tilde{n}}\}}$ is hard

These requirements are encapsulated in our Theorem 19.

Chapter 6

Instantiating Proof System 0

In this Chapter we explain how to instantiate proof system 0 in our VOPRF using the ZKPoK in Chapter 4. In proof system 0, a server proves knowledge of a

- $k \in \mathcal{R}$ where $\|k\|_\infty \leq \tilde{\sigma}_{\mathsf{server}}\sqrt{\tilde{n}}$
- $\mathbf{e} \in \mathcal{R}^{1\times\tilde{l}}$ where $\|\mathbf{e}\|_\infty \leq \tilde{\sigma}_{\mathsf{server}}\sqrt{\tilde{n}}$

such that

$$\mathbf{c} = \mathbf{a} \cdot k + \mathbf{e} \mod q$$

where $\mathbf{c} \in \mathcal{R}_q^{1\times\tilde{\ell}}$ and $\mathbf{a} \in \mathcal{R}_q^{1\times\tilde{\ell}}$ are public.

We can write this as an instance of the relation in Chapter 4 with $B_{\mathsf{yes}} = \tilde{\sigma}_{\mathsf{server}}\sqrt{\tilde{n}}$, $B_{\mathsf{no}} = \tilde{\sigma}\sqrt{\tilde{n}}$, and

$$A = [a^T || I_{\tilde{n}\tilde{\ell}}] \in \mathbb{Z}_q^{\tilde{n}\tilde{\ell}\times\tilde{n}(\tilde{\ell}+1)}$$
$$S = [k||e]^T \in \mathbb{Z}^{\tilde{n}(\tilde{\ell}+1)\times 1}$$
$$T = [c^T] \in \mathbb{Z}_q^{\tilde{n}\tilde{\ell}\times 1}$$

where a^T is the vertical concatenation of the negacyclic matrices associated to multiplication

by the ring elements of $a \in \mathcal{R}_q^{1\times\tilde{\ell}}$, c^T is the vertical concatenation of coefficient vectors of ring elements in $\mathbf{c}$, and $I_{\tilde{n}\tilde{\ell}}$ is the $\tilde{n}\tilde{\ell} \times \tilde{n}\tilde{\ell}$ identity matrix. We note that setting $B_{\mathsf{yes}} = \tilde{\sigma}_{\mathsf{server}}\sqrt{\tilde{n}}$ and $B_{\mathsf{no}} = \tilde{\sigma}\sqrt{\tilde{n}}$ imposes an additional constraint upon the parameters of our VOPRF as discussed in Chapter 7.

Chapter 7

Parameter Setting

From Theorem 19 we have the constraints

- $\mathsf{dRLWE}_{q,\tilde{n},\tilde{\sigma}_{\mathsf{server}}}$ is hard
- $\frac{q}{2p} \gg \tilde{\sigma}' \gg \max\{L \cdot \tilde{\ell} \cdot \tilde{\sigma}_{\mathsf{server}} \cdot \tilde{n}^{3/2}, \tilde{\sigma} \cdot \tilde{\sigma}_{\mathsf{server}} \tilde{n}^2\}$
- $\mathsf{1D\text{-}SIS}_{\frac{q}{2p},\tilde{n}\tilde{\ell},\max\{\tilde{n}^{3/2}\tilde{\ell}\tilde{\sigma},2\tilde{\sigma}^2\cdot\tilde{n}^2+\tilde{\sigma}'\sqrt{\tilde{n}}\}}$ is hard

which requires that

$$\tilde{\sigma}' = \tilde{\sigma}^2 \tilde{n}^2 \cdot \kappa^{\omega(1)} \tag{7.1}$$

and

$$q = p \cdot \tilde{\sigma}' \cdot \kappa^{\omega(1)} \tag{7.2}$$

(for a full list of parameter requirements, we refer the reader to Table 1 of [4]). We make a note here that these requirements are slightly stricter than necessary for the security of our VOPRF protocol (i.e. we only require the drowning distribution $\tilde{\sigma}'$ to be $\tilde{\sigma}' = \tilde{\sigma} \cdot \tilde{\sigma}_{\mathsf{server}} \tilde{n}^2 \cdot \kappa^{\omega(1)}$), but we adopt these stricter requirements for simplicity when giving our parameter estimation below.

Additionally, the use of the ZKPoK from Chapter 4 for proof system 0 imposes an additional constraint: the bound on the ℓ_∞ norm that the server proves of their key $(k, \mathbf{e})$ (i.e.

$B_{\mathsf{no}} = \tilde{\sigma}\sqrt{\tilde{n}}$), must be larger than the bound on the distribution from which an honest server draws their key, $B_{\mathsf{yes}} = \tilde{\sigma}_{\mathsf{server}}\sqrt{\tilde{n}}$. The relationship between these bounds is established by Theorem 18, where we set $n = \lambda + 2$ and the rejection sampling parameter $\rho = 2.72$:

$$B_{\mathsf{no}} \geq B_{\mathsf{yes}} \frac{12}{\ln(2.72)} v\ell\sqrt{8(\lambda + 2)}$$

In accordance with our reduction in Chapter 6, we substitute $\lambda = \kappa$, $B_{\mathsf{yes}} = \tilde{\sigma}_{\mathsf{server}}\sqrt{\tilde{n}}$, $B_{\mathsf{no}} = \tilde{\sigma}\sqrt{\tilde{n}}$, $v = \tilde{n}(\tilde{\ell} + 1)$, and $\ell = 1$, yielding:

$$\tilde{\sigma}\sqrt{\tilde{n}} \geq \tilde{\sigma}_{\mathsf{server}}\sqrt{\tilde{n}} \frac{12}{\ln(2.72)} \tilde{n}(\tilde{\ell} + 1)\sqrt{8(\kappa + 2)}$$

[4] sets the distribution the server's key is drawn from $\tilde{\sigma}_{\mathsf{server}} = 3.2$, and gives $\kappa = 128$, $\tilde{\ell} = \log q$, and $\tilde{n} = 16,384$ as rough parameter estimates. We pick $q = 2^{318}$. Setting these values yields

$$\tilde{\sigma} \geq \tilde{\sigma}_{\mathsf{server}} \cdot 2.02 \cdot 10^9 \tag{7.3}$$

$$= 6.46 \cdot 10^9 \tag{7.4}$$

Note that we grow the modulus from the original setting of $\log q = 256$ in [4] to $\log q = 318$ in order to maintain the relationships in Equations 7.1 and 7.2 imposed by the constraints. We note that by Table 1 of the Homomorphic Encryption Standard [22], setting $\tilde{n} = 16,384$ and $\log q = 318$ for $\tilde{\sigma}_{\mathsf{server}} = 3.2$ maintains a security of $\kappa = 128$.

Chapter 8

Efficiency Analysis

In this Chapter we analyze the contribution of proof system 0 to the overall communication cost of the VOPRF protocol. The size of the proof π in Figure 4.1 is $|\pi| = |\mathbf{Z}| + |\mathbf{C}|$. $\mathbf{C}$ can be represented by ℓn bits. By the verifier's first check, each entry of $\mathbf{Z}$ is of length at most $\frac{B}{2}$, so $\mathbf{Z}$ can be represented by $vn \log \frac{B}{2}$ bits. Setting $\sigma = \frac{12}{\ln \rho} s\sqrt{\ell n}$, $n = \lambda + 2$ as in Theorem 18, we obtain

$$|\pi| = v(\lambda + 2) \log \frac{B}{2} + \ell(\lambda + 2)$$

Substituting $B = \tilde{\sigma}\sqrt{\tilde{n}}$, $v = \tilde{n}(\tilde{\ell} + 1)$, $\ell = 1$, and $\lambda = \kappa$, we can derive an expression for the proof size $|\pi|$ in terms of the parameters of our VOPRF

$$\begin{aligned} |\pi| &= \tilde{n}(\tilde{\ell} + 1)(\kappa + 2) \log \frac{\tilde{\sigma}\sqrt{\tilde{n}}}{2} + 1 \cdot (\kappa + 2) \\ &\approx 2.65 \cdot 10^{10} \end{aligned}$$

bits or 3.08 GB. While large, the main bottleneck of the VOPRF protocol as a whole remains the size of the proof required for proof system 1. For a modulus of $\log q = 256$, [4] provides a rough lower bound of is 20^{40} bits = 128 GB of communication per repetition of the protocol from [5] when used for proof system 1 (the protocol from [5] requires $\frac{\kappa}{\log \bar{p}}$ repetitions to reach a soundness error of $2^{-\kappa}$). Excluding the proofs, the communication cost of the protocol

involves the server sending $2\tilde{\ell}$ RWLE samples and the client sending $\tilde{\ell}$ RWLE samples. With our choice of $\log q = 318$ and $\tilde{n} = 16,384$, one RLWE sample is about 0.65MB, so the total communication cost excluding the proofs is around 620MB. Thus, the proof systems are the main source of inefficiency in our VOPRF construction.

Chapter 9

Future Work

Our ZKPoK from Chapter 4 can be adapted to operate over the polynomial ring instead of the integers, further reducing the proof size of proof system 0. However, as stated in Chapter 8, the primary bottleneck of the VOPRF construction is the client proof $\mathbb{P}_1$.

Future work will investigate applying the recent advancements from [23] in proving the quadratic relations involved in proof system 1. This protocol presents the most efficient method known for proving quadratic relations between committed polynomials in $\mathcal{R}_q$. By leveraging this approach, it is anticipated that we can achieve a shorter and more efficient proof than the system described in [5] currently used for the client proof $\mathbb{P}_1$.

Another potential direction is to explore making efficient hybridized VRFs [14] oblivious. These hybridized proofs currently produce the shortest VRF outputs among standard (i.e., long-term and stateless) VRFs based on quantum-safe assumptions, but they lack obliviousness.

These enhancements could lead to significantly reduced proof sizes and improved efficiency, contributing to the development of a more practical and quantum-safe VOPRF.

References

[1] S. Goldberg, M. Naor, D. Papadopoulos, L. Reyzin, S. Vasant, and A. Ziv, "Nsec5: Provably preventing dnssec zone enumeration," Jan. 2015. DOI: 10.14722/ndss.2015.23211.

[2] Y. Pan, Y. Zhao, X. Liu, G. Wang, and M. Su, "Fplotto: A fair blockchain-based lottery scheme for privacy protection," in *2022 IEEE International Conference on Blockchain (Blockchain)*, 2022, pp. 21–28. DOI: 10.1109/Blockchain55522.2022.00014.

[3] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, "Algorand: Scaling byzantine agreements for cryptocurrencies," in *Proceedings of the 26th Symposium on Operating Systems Principles*, ser. SOSP '17, Shanghai, China: Association for Computing Machinery, 2017, pp. 51–68, ISBN: 9781450350853. DOI: 10.1145/3132747.3132757. URL: https://doi.org/10.1145/3132747.3132757.

[4] M. R. Albrecht, A. Davidson, A. Deo, and N. P. Smart, *Round-optimal verifiable oblivious pseudorandom functions from ideal lattices*, Cryptology ePrint Archive, Paper 2019/1271, https://eprint.iacr.org/2019/1271, 2019. URL: https://eprint.iacr.org/2019/1271.

[5] R. Yang, M. H. Au, Z. Zhang, Q. Xu, Z. Yu, and W. Whyte, "Efficient lattice-based zero-knowledge arguments with standard soundness: Construction and applications," in *Advances in Cryptology – CRYPTO 2019*, A. Boldyreva and D. Micciancio, Eds., Cham: Springer International Publishing, 2019, pp. 147–175, ISBN: 978-3-030-26948-7.

[6] C. Baum, J. Bootle, A. Cerulli, R. del Pino, J. Groth, and V. Lyubashevsky, "Sub-linear lattice-based zero-knowledge arguments for arithmetic circuits," in *Advances in Cryptology – CRYPTO 2018*, H. Shacham and A. Boldyreva, Eds., Cham: Springer International Publishing, 2018, pp. 669–699, ISBN: 978-3-319-96881-0.

[7] V. Lyubashevsky, "Lattice signatures without trapdoors," in *Advances in Cryptology – EUROCRYPT 2012*, D. Pointcheval and T. Johansson, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 738–755, ISBN: 978-3-642-29011-4.

[8] S. Micali, M. Rabin, and S. Vadhan, "Verifiable random functions," in *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)*, 1999, pp. 120–130. DOI: 10.1109/SFFCS.1999.814584.

[9] P. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proceedings 35th Annual Symposium on Foundations of Computer Science*, 1994, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.

[10] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, H. N. Gabow and R. Fagin, Eds., ACM, 2005, pp. 84–93. DOI: 10.1145/1060590.1060603. URL: https://doi.org/10.1145/1060590.1060603.

[11] C. Peikert, *Public-key cryptosystems from the worst-case shortest vector problem*, Cryptology ePrint Archive, Paper 2008/481, https://eprint.iacr.org/2008/481, 2008. URL: https://eprint.iacr.org/2008/481.

[12] M. F. Esgin, V. Kuchta, A. Sakzad, R. Steinfeld, Z. Zhang, S. Sun, and S. Chu, "Practical post-quantum few-time verifiable random function with applications to algorand," in *Financial Cryptography and Data Security: 25th International Conference, FC 2021, Virtual Event, March 1–5, 2021, Revised Selected Papers, Part II*, Berlin, Heidelberg: Springer-Verlag, 2021, pp. 560–578, ISBN: 978-3-662-64330-3. DOI: 10.1007/978-3-662-64331-0_29. URL: https://doi.org/10.1007/978-3-662-64331-0_29.

[13] M. Buser, R. Dowsley, M. F. Esgin, S. K. Kermanshahi, V. Kuchta, J. K. Liu, R. Phan, and Z. Zhang, *Post-quantum verifiable random function from symmetric primitives in pos blockchain*, Cryptology ePrint Archive, Paper 2021/302, https://eprint.iacr.org/2021/302, 2021. URL: https://eprint.iacr.org/2021/302.

[14] M. F. Esgin, R. Steinfeld, D. Liu, and S. Ruj, *Efficient hybrid exact/relaxed lattice proofs and applications to rounding and vrfs*, Cryptology ePrint Archive, Paper 2022/141, https://eprint.iacr.org/2022/141, 2022. URL: https://eprint.iacr.org/2022/141.

[15] R. Goyal, S. Hohenberger, V. Koppula, and B. Waters, "A generic approach to constructing and proving verifiable random functions," Nov. 2017, pp. 537–566, ISBN: 978-3-319-70502-6. DOI: 10.1007/978-3-319-70503-3_18.

[16] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," vol. 60, May 2010, pp. 1–23, ISBN: 978-3-642-13189-9. DOI: 10.1007/978-3-642-13190-5_1.

[17] M. Ajtai, "Generating hard instances of lattice problems," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, 1996, pp. 99–108.

[18] Z. Brakerski and V. Vaikuntanathan, "Constrained key-homomorphic prfs from standard lattice assumptions," in *Theory of Cryptography*, Y. Dodis and J. B. Nielsen, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 1–30, ISBN: 978-3-662-46497-7.

[19] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. Chapman and Hall/CRC Press, 2007, ISBN: 978-1-58488-551-1. URL: http://www.cs.umd.edu/%5C%7Ejkatz/imc.html.

[20] A. Banerjee and C. Peikert, *New and improved key-homomorphic pseudorandom functions*, Cryptology ePrint Archive, Paper 2014/074, https://eprint.iacr.org/2014/074, 2014. URL: https://eprint.iacr.org/2014/074.

[21] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Advances in Cryptology — CRYPTO' 86*, A. M. Odlyzko, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 1987, pp. 186–194, ISBN: 978-3-540-47721-1.

[22] M. Albrecht, M. Chase, H. Chen, J. Ding, S. Goldwasser, S. Gorbunov, S. Halevi, J. Hoffstein, K. Laine, K. Lauter, *et al.*, "Homomorphic encryption standard," *Protecting privacy through homomorphic encryption*, pp. 31–62, 2021.

[23] T. Attema, V. Lyubashevsky, and G. Seiler, *Practical product proofs for lattice commitments*, Cryptology ePrint Archive, Paper 2020/517, https://eprint.iacr.org/2020/517, 2020. URL: https://eprint.iacr.org/2020/517.