Papercraft: Lattice-based Verifiable Delay Function Implemented

Michał Osadnik
Aalto University, Finland

Darya Kaviani
UC Berkeley, USA

Valerio Cini
Bocconi University, Italy

Russell W. F. Lai
Aalto University, Finland

Giulio Malavolta
Bocconi University, Italy

***Abstract*—A verifiable delay function (VDF) requires a specified number of sequential steps to compute, yet the validity of its output can be verified efficiently, much faster than recomputing the function from scratch. VDFs are a versatile cryptographic tool, with many industrial applications, such as blockchain consensus protocols, lotteries and verifiable randomness. Unfortunately, without exceptions, all known practical VDF constructions are broken by quantum algorithms.**

In this work, we investigate the practicality of VDFs with plausible post-quantum security. We propose `Papercraft`, a working implementation of a VDF based entirely on lattice techniques and thus plausibly post-quantum secure. Our VDF is based on new observations on lattice-based succinct argument systems with many low-level optimisations, yielding the first lattice-based VDF that is implementable on today's hardware. As an example, our `Papercraft` implementation can verify a computation of over 6 minutes in just 7 seconds. Overall, our work demonstrates that lattice-based VDFs are not just a theoretical construct, paving the way for their practical deployment.

1. Introduction

The computation of a verifiable delay function (VDF) [1] is a slow sequential process which produces a small certificate that the computation was indeed performed. Crucially, verifying the certificate is much faster than performing the computation itself. They can be thought of as a cryptographic primitive that verifies the passage of time. VDFs are a cryptographic primitive of central importance, and they count a number of academic and industrial [2] applications, such as blockchain consensus protocols [3], randomness beacons [4, 5], and lotteries [1, 6], to mention a few.

Despite its applicability, we only know of a handful of VDF constructions that have practical enough to run on today's hardware [7, 8]. Even worse, to the best of our knowledge, with no exception all practical VDF constructions are either based on factoring-related [9] or other [10, 11, 12] assumptions that are insecure against quantum algorithms. Thus, in the hypothetical scenario where scalable quantum computers are capable of running (variants of) Shor's algorithm, we would be left with no viable VDF candidate.

The question of a *post-quantum* candidate for a VDF has attracted attention in the cryptographic community, and several works proposed theoretical solutions to this problem [13, 14, 15, 16]. Unfortunately, to the best of our knowledge, there is currently no VDF construction that is simultaneously: (i) plausibly post-quantum secure and (ii) feasible to implement with current hardware. In fact, we are not aware of *any attempt* to actually implement a (candidate) post-quantum VDF, calling into question whether this task is technologically feasible at all. A step towards this direction was taken in a recent work [17], which we discuss next.

1.1. A Lattice-Based Sequential Function

Before delving into the specifics of the approach, let us recall a well-known template to construct VDFs. We start from a *sequential function* f, whose recursive application $f(\dots f(x)\dots)$ requires time linear in the number of function calls. For instance, for VDFs based on number-theoretic assumptions [7, 8] the function f corresponds to squaring modulo an RSA integer $N = pq$. Intuitively the repeated application of f, serves as the "witness" for the passage of time. To verify this fact efficiently, one needs to then equip this sequential function with a *succinct verification procedure*, that certifies that $y = f(\dots f(x)\dots)$, without the need to recompute the function f many times.

The work of [17] suggests the following candidate sequential function: Let $\mathbf{A}$ be a matrix and $\mathbf{x}$ be an input vector, the function f is defined as[1]

$$f(\mathbf{x}) := \mathbf{G}^{-1}(\mathbf{A}\mathbf{x}) \mod q \tag{1}$$

where $\mathbf{G}^{-1}$ is the binary decomposition operator (a standard notation in lattice-based cryptography), and the parameters are set in such a way that the domain and range of the function coincide. As another way to put it, [17] conjectured that the repeated application Ajtai's hash function [18] is inherently sequential, and provided supporting evidence for this claim. The hardness of computational problems in lattice-based cryptography appears to be unaffected by quantum algorithms, allowing one to conjecture post-quantum security.

[1] Adjusted for the ease of presentation.

In order to make the function efficiently verifiable, [17] observed that verifying a single step of the function in Eq. (1) is equivalent to verifying that

$$\mathbf{G}\mathbf{y} \overset{?}{=} \mathbf{A}\mathbf{x} \quad \text{and} \quad \mathbf{y} \in \{0,1\}^n$$

where $\mathbf{G}$ is the gadget matrix, i.e. the binary reconstruction linear operator. The above is a linear relation, followed by a binariness check. This observation can be naturally extended to multiple invocation of the function to verify the full VDF computation. The advantage of this formulation is that lattice-based proof systems *natively* support this class of statements, and so there is hope to construct *concretely efficient* protocol. Towards this end, [17] proposed a further relaxation of the problem, where instead of proving the statement as defined above, one proves that

$$\mathbf{G}\mathbf{y} \overset{?}{=} \mathbf{A}\mathbf{x} \mod q \quad \text{and} \quad \mathbf{y} \in [-B, B]^n$$

for some small $1 < B < q$. We refer to the former relation as proof of *exact* sequentiality and the latter as a proof of *approximate* sequentiality. [17] then proposed succinct protocols to prove approximate sequentiality and derived some applications.

The approximate sequentiality relation is unfortunately vulnerable to sampling attacks. Indeed, the follow-up work [19] (see also a summary in [20]) has shown attacks against close variants of the protocols suggested in [17], casting doubts on the soundness of the approach. In more detail, a T-fold execution of the function in Eq. (1) induces a relation of the form $\mathbf{A}_T \cdot \mathbf{u}_T = \mathbf{x}_T \bmod q$, where $\mathbf{A}_T$ consists of the matrices $\mathbf{A}$ and $\mathbf{G}$ stacked in the shape of a staircase, $\mathbf{x}_T$ encodes the function input, and $\mathbf{u}_T$ is a binary vector encoding all intermediate steps of the function execution. For the *approximate* sequentiality relation, the prover is only required to prove that $\mathbf{u}_T$ has low-norm instead of being exactly binary. The attacks in [19] exploit this relaxation in a crucial manner. They work by first performing a low-depth computation to find a lattice trapdoor [21, 22] of the matrix $\mathbf{A}_T$, which then allows to perform another low-depth computation to sample a short preimage $\mathbf{u}_T$ satisfying $\mathbf{A}_T \cdot \mathbf{u}_T = \mathbf{x}_T \bmod q$. As discussed in [19, Section 1.3], this approach leads to an increase in the norm of the solution $\mathbf{u}_T$ and does not apply when the solution is required to be very short. In other words, the attacks do not seem to apply if one insists on *exact* or even *slightly approximate sequentiality*. Therefore, one way to prevent such attacks would be to design protocols that certify that $\mathbf{u}_T$ is *exactly binary*, as opposed to being low-norm. Unfortunately, these protocols tend to be much less efficient than approximate ones, leaving us without an efficient lattice-based VDF candidate.

1.2. Our Contributions

In this work, we revisit the [17] approach and propose a VDF protocol that (i) is entirely lattice based (and therefore plausibly post-quantum secure), (ii) relies on the conservative proof of *exact* sequentiality, and (iii) is concretely efficient for computations of reasonable length. Our technical contributions are summarised below.

We propose a *concretely efficient* lattice based VDF, `Papercraft`, based on the aforementioned lattice-based sequential function. Our main technical contribution is a new protocol to succinctly verify (structured) linear statements followed by a proof of exact binariness of the witness. In other words, we show an efficient proof system for the *exact* sequentiality relation. We propose a comprehensive analysis of the soundness of `Papercraft`, placing our candidate on firm theoretical foundations.

In slightly more detail, `Papercraft`[2] is obtained by constructing an efficient reduction of knowledge (RoK) protocol reducing the exact sequentiality relation to the principal relation supported by a recent lattice-based succinct argument system [23]. The principal relation of [23] is the bounded-norm/binary satisfiability of $\mathbf{A}\mathbf{x} = \mathbf{y} \bmod q$ where the matrix $\mathbf{A}$ has a row-tensor structure. Via the new RoK protocol, we extend the family of supported $\mathbf{A}$ to include those containing a "staircase" submatrix, thereby extending principal relation supported by [23].

An efficient VDF should strike a good balance between prover and verifier runtime, proof size, ease of implementation and strength of the underlying computational assumptions. By employing numerous low-level optimisation techniques and carefully picking parameters, we are able to optimise for these quantities and obtain a VDF that is concretely efficient from all angles.

To substantiate the practicality of our approach, we have developed a prototype implementation of our VDF. Furthermore, despite the slow-running nature of VDF provers by design, we have conducted an extensive experimental evaluation to establish various parameter trade-offs. For example, with a carefully selected set of parameters, we obtain a VDF that executes in over 6 minutes and can be verified in just 7 seconds. This configuration incurs a communication cost of 15.38MB and a prover runtime around 4 hours.

Overall, our work demonstrates that implementing lattice-based VDFs is technologically feasible, and opens the doors for the practical deployment of this primitive.

1.3. Other PQ-secure Candidates

Recent proposals such as MinRoot [24] and ZKBdf [25] have taken promising steps toward designing VDFs that remain secure in the face of quantum adversaries. Although their approaches differ considerably in terms of assumptions, design principles, and performance goals, both represent meaningful progress in understanding the design space of post-quantum VDFs.

MinRoot [24] introduces a simple algebraic sequential function designed for use in Ethereum's candidate VDF construction. The round function is based on low-degree

[2] Papercraft is the (time-taking) process of combining pieces of paper (resp. folding-based reductions of knowledge) for the creation of two or three-dimensional objects (resp. VDFs).

modular exponentiation over a finite field and is applied iteratively. Its sequentiality is conjectured to stem from the presumed difficulty of root extraction, which still requires more cryptanalysis effort to confirm. While the design is meant to be paired with recursive SNARK, e.g. based on Nova [26], a full implementation is currently not available.

ZKBdf [25] achieves sequentiality by modifying the existing ZKBoo proof system [27] to introduce inter-round dependencies. ZKBdf enforces a sequential structure by computing the randomness for each round as a hash function (specifically, HMAC-SHA256) of the previous round's transcript, effectively serializing the ZK proof generation. Despite its novel design, the implementation of ZKBdf remains limited to very small delay parameters ($T \approx 250$), and proof generation is rather inefficient: the authors report large memory and proof size overheads. While their construction supports a additional "prover-secret" feature, this property is orthogonal to the core VDF definition and use-cases, such as public randomness beacons or decentralized consensus.

By contrast, `Papercraft` is the first fully lattice-based VDF that (i) instantiates both the sequential function and the argument system using lattice-based techniques, and (ii) provides concrete performance results for high-delay parameters even up to hundreds of thousands of application of the sequential function.

1.4. Perspective

The objective of this work is to establish the technological feasibility of implementing a VDF construction that is entirely lattice-based, and therefore plausibly post-quantum secure. As mentioned before, we are not aware of any other attempt, except [25], of implementing a VDF that is not based on number-theoretic (and quantum-broken) assumptions. Making a VDF technologically feasible is not just an engineering challenge. Known post-quantum VDF candidates display poor scaling properties, as indicated by their prover runtimes. As a result, achieving practical VDFs requires both engineering effort and theoretical advances. Furthermore, our approach has favourable properties for real-world deployment, such as featuring a completely transparent setup.

Nevertheless, we also point out that the efficiency of our VDF does not match that of number-theory based constructions, especially in terms of the large gap between the VDF execution time and the proving time, and so their practical applicability is somewhat restricted.

To enhance the applicability of `Papercraft`, we identify several potential areas for improvement:

- Parameters could be selected more systematically, balancing various objectives. The current parameters strongly emphasise proof succinctness, often at the expense of prover runtime.
- Runtime (esp. prover runtime) could be further reduced by applying low-level and platform-specific optimisations related to modular reduction and ring arithmetic (see Section 8 for detailed information).
- The prover and verifier should be tested in more realistic non-interactive settings, that is, as separate programs with a single round of communication.

By focusing on these aspects, we can potentially significantly improve the practical applicability of `Papercraft`, making it more suitable for a broader range of applications. Overall, we optimistically view `Papercraft` as a first step towards widening the applicability of post-quantum VDFs: We expect that advances in lattice-based succinct arguments can further increase the practicality of our approach. We view such improvements as an exciting research direction.

2. Preliminaries

Let $\mathbb{N} = \{1, 2, \ldots\}$ denote natural numbers and $\lambda \in \mathbb{N}$ be the security parameter. For $m, n \in \mathbb{N}$, we write $[n] := \{0, \ldots, n-1\}$ counting from 0 and $[m : n] := [n] \setminus [m]$. For multidimensional ranges, we use the shorthand $(i, j, k) \in [n, m, \ell]$ for $i \in [n], j \in [m]$, and $k \in [\ell]$. The logarithm $\log$ is base-2. To represent $\mathbb{Z}_q$ we use the balanced representation, i.e. $\{-\lceil q/2 \rceil + 1, \ldots, \lfloor q/2 \rfloor\}$. We use bold lower-case letters to denote vector $\mathbf{v}$ and bold upper-case letters to denote matrix $\mathbf{M}$. For vectors, we use subscript witha a specified range to denote the subvector, e.g. $\mathbf{v}_{[1,|\mathbf{v}|]}$ denotes the vector without the element at the 0-th index. For matrices (or vectors) $\mathbf{M}_0, \ldots, \mathbf{M}_{k-1}$ of appropriate dimensions, we write $(\mathbf{M}_i)_{i \in [k]}$ and $(\mathbf{M}_0 || \ldots || \mathbf{M}_{k-1})$ for horizontal and vertical concatenation respectively. We say that $\mathbf{x} \in \bar{\mathcal{R}}^{1 \times \otimes_{i \in [\mu]} d_i}$ if $\mathbf{x}$ is of the form $\mathbf{x} = \mathbf{x}_0 \otimes \ldots \otimes \mathbf{x}_{\mu-1}$ with $\mathbf{x}_i \in \bar{\mathcal{R}}^{1 \times d_i}$. We say that matrix $\mathbf{F} \in \mathcal{R}^{n \times m}$ has a row-tensor structure if can be written as follows

$$\mathbf{F} = (\mathbf{f})^{\mathtt{T}}_{i \in [n]} = (\mathbf{r}_i \otimes \widetilde{\mathbf{f}}_i)^{\mathtt{T}}_{i \in [n]}$$

where $\mathbf{f}_i$ denotes the i-th row of $\mathbf{F}$, and $\widetilde{\mathbf{F}} = (\widetilde{\mathbf{f}})^{\mathtt{T}}_{i \in [n]} \in \mathcal{R}_q^{n \times \otimes \mathbf{d}_{[1,|\mathbf{d}|]}}$. To ease notation, we define $\mathbf{R} = (\mathbf{r}_i)^{\mathtt{T}}_{i \in [n]} \in \mathcal{R}_q^{n \times d_0}$. Then, we denote the row-tensor product as

$$\mathbf{R} \bullet \widetilde{\mathbf{F}} = \mathbf{F}.$$

An algorithm $\mathcal{A}$ is said to be in probabilistic polynomial time (PPT) if there exists polynomials $p(\cdot), q(\cdot)$ and a family of circuits $(C_i(\cdot))_{i \in \mathbb{N}}$ such that for all $i \in \mathbb{N}$, $|C_i| \leq p(i)$ and for every input $x \in \{0,1\}^i$ and random coins $r \in \{0,1\}^{q(i)}$, the computation of $\mathcal{A}(x; r)$ terminates with output $C_i(x, r)$. We define the depth of $\mathcal{A}$ on input $x \in \{0,1\}^i$ to be $\mathsf{Depth}(C_i)$.

We use the notation $(\mathcal{P}, \mathcal{V})$ to denote an interactive protocol and the notation $\langle \mathcal{P}(x, y), \mathcal{V}(x) \rangle$ to denote a run of the interactive protocol with inputs (x, y) and x respectively.

2.1. Algebraic Number Theory

Throughout this work, we let $\mathcal{K} = \mathbb{Q}(\zeta_{\mathfrak{f}})$ be a cyclotomic field with conductor $\mathfrak{f}$ of degree $\varphi = \varphi(\mathfrak{f})$, where $\zeta_{\mathfrak{f}}$ is a root of unity of order $\mathfrak{f}$ and φ is Euler's totient function, and $\mathcal{R}_{\mathfrak{f}} = \mathbb{Z}[\zeta_{\mathfrak{f}}]$ be its ring of integers. When it is clear from the context, we omit $\mathfrak{f}$ from the subscripts. We will also consider the maximal real subfield $\mathcal{K}^+ = \mathbb{Q}(\zeta + \zeta^{-1})$ of $\mathcal{K}$ and its ring of integers $\mathcal{R}^+ = \mathbb{Z}[\zeta + \zeta^{-1}]$. Fixing a $\mathbb{Z}$-basis $\mathbf{b} = (b_i)_{i \in [\varphi]} \subset \mathcal{R}_{\mathcal{K}}$, we can view $\mathcal{R}$ as a $\mathbb{Z}$-module of dimension φ. We will usually use $\bar{\mathcal{R}} \subseteq \mathcal{R}$ to denote a subring of dimension δ as a $\mathbb{Z}$-module.

For $q \in \mathbb{N}$, define the quotient ring $\mathcal{R}_q := \mathcal{R}/q\mathcal{R}$. We denote by $\mathcal{R}^\times$ and $\mathcal{R}_q^\times$ the sets of units in $\mathcal{R}$ and $\mathcal{R}_q$ respectively. We assume throughout that q is a rational prime unramified in $\mathcal{K}$ with multiplicative order e modulo $\mathfrak{f}$, so that $\mathcal{R}_q$ contains a subfield of size q^e. Denote the multiplicative group of this subfield by $\mathcal{C}_{\mathcal{R}_q}$ with $|\mathcal{C}_{\mathcal{R}_q}| = q^e - 1$.

Coefficient embedding and norms. We endow $\bar{\mathcal{R}}$ with two geometries via the coefficient embedding $\mathsf{coeff}_{\mathbf{b}} : \mathcal{R} \to \mathbb{Z}^\delta$ (for a given basis $\mathbf{b}$) and the canonical embedding $\sigma : \mathcal{K} \to \mathbb{C}^\varphi$ (of $\mathcal{K}$). Specifically, for a given $\mathbb{Z}$-basis $\mathbf{b} = (b_i)_{i \in [\delta]}$ of $\mathcal{R}$ and an element $x = \sum_{i \in [\delta]} x_i b_i \in \mathcal{R}$, we write

$$\mathsf{coeff}_{\mathbf{b}}(x) := (x_i)_{i \in [\delta]} \quad \text{and} \quad \sigma(x) := (\sigma_j(x))_{j \in [\varphi]}$$

where $\sigma_j \in \mathsf{Gal}(\mathcal{K}/\mathbb{Q})$. Note that we define $\sigma(x)$ by treating $x \in \mathcal{K}$ in order to avoid discussing the canonical embedding of subfields of $\mathcal{K}$. We extend the notation of $\mathsf{coeff}_{\mathbf{b}}$ and σ naturally to vectors, i.e. if $\mathbf{x} = (x_i)_{i \in [m]} \in \mathcal{R}^m$, then

$$\mathsf{coeff}_{\mathbf{b}}(\mathbf{x}) := (\mathsf{coeff}_{\mathbf{b}}(x_i))_{i \in [\delta]} \quad \text{and} \quad \sigma(\mathbf{x}) := (\sigma_j(x_i))_{j \in [\varphi]}$$

are defined as concatenations. The coefficient and canonical ℓ_p-norms of a vector $\mathbf{x} \in \mathcal{R}^m$ are denoted $\|\mathsf{coeff}(\mathbf{x})\|_p$ and $\|\sigma(\mathbf{x})\|_p$ respectively. Unless specified otherwise, we default $\|\cdot\| = \|\mathsf{coeff}\,(\cdot)\|_\infty$.

If $\mathcal{R} = \mathcal{O}_{\mathcal{K}}$ and $\mathbf{b} = \mathbf{b}^{\mathtt{pwf}}$ is the "powerful basis" [28], we may write $\mathsf{coeff}_{\mathtt{pwf}}$ or simply coeff instead of $\mathsf{coeff}_{\mathbf{b}^{\mathtt{pwf}}}$. More precisely, for a prime-power conductor $\mathfrak{f}$,

$$\mathbf{b}^{\mathtt{pwf}} := (1, \zeta, \ldots, \zeta^{\varphi - 1}).$$

For composite conductors $\mathfrak{f} = \prod_{i \in [k]} \mathfrak{f}_i$ where f_i are pairwise coprime prime-powers,

$$\mathbf{b}^{\mathtt{pwf}} := \bigotimes_{i \in [k]} \left(1, \zeta_{\mathfrak{f}_i}, \ldots, \zeta_{\mathfrak{f}_i}^{\varphi(\mathfrak{f}_i) - 1}\right).$$

Tensor ring $\bar{\mathcal{R}} \subseteq \mathcal{R}$. In this work, we put a particular emphasis on a subring $\bar{\mathcal{R}} \subseteq \mathcal{R}_{\mathfrak{f}}$. Let $\mathfrak{f} = \mathfrak{g} \cdot \prod_{i \in [k]} \mathfrak{f}_i$, where $\mathfrak{g} = 2^d$ for some $d \in \mathbb{N}$ and $\mathfrak{f}_0, \ldots, \mathfrak{f}_{k-1}$ are distinct odd primes, so that $\mathcal{R}_{\mathfrak{f}} \cong \mathcal{R}_{\mathfrak{g}} \otimes \mathcal{R}_{\mathfrak{f}_0} \otimes \ldots \otimes \mathcal{R}_{\mathfrak{f}_{k-1}}$. Then, let $\bar{\mathcal{R}} \subset \mathcal{R}_{\mathfrak{f}}$ be such that $\bar{\mathcal{R}} \cong \mathcal{R}_{\mathfrak{g}} \otimes \mathcal{R}^+_{\mathfrak{f}_0} \otimes \ldots \otimes \mathcal{R}^+_{\mathfrak{f}_{k-1}}$ with dimension $\delta = \varphi(\mathfrak{f})/2^{k+1}$ as a $\mathbb{Z}$-basis. For brevity, we refer to rings such as $\bar{\mathcal{R}}$ as "tensor rings".

For a tensor ring $\bar{\mathcal{R}}$, we consider the basis $\mathbf{b}^{\otimes}_{\mathfrak{f}}$ defined as follows. If $\mathfrak{f}$ is an odd prime, define

$$\mathbf{b}^{\otimes}_{\mathfrak{f}} := \left\{ \sum_{i = [j+1]} \left(\zeta_{\mathfrak{f}}^{\varphi/2 - i} + \zeta_{\mathfrak{f}}^{-(\varphi/2 - i)}\right) \right\}_{j \in [\varphi/2]}$$

where $\varphi = \varphi(\mathfrak{f})$. For the general case, define

$$\mathbf{b}^{\otimes}_{\mathfrak{f}} := \mathbf{b}^{\mathtt{pwf}}_{\mathfrak{g}} \otimes \mathbf{b}^{\otimes}_{\mathfrak{f}_0} \otimes \ldots \otimes \mathbf{b}^{\otimes}_{\mathfrak{f}_{k-1}}.$$

Simplifying notation, we define $\mathsf{coeff}_{\otimes}(\cdot) := \mathsf{coeff}_{\mathbf{b}^{\otimes}_{\mathfrak{f}}}(\cdot)$, where the conductor is implicit. Lastly, we define translation factor $\Phi_{\otimes/\mathtt{pwf}}$ and $\Phi_{\mathtt{pwf}/\otimes}$ between bases so that

(i) $\left\|\mathsf{coeff}_{\mathtt{pwf}}\,(x)\right\|_\infty \le \beta \implies \left\|\mathsf{coeff}_{\otimes}\,(x)\right\|_\infty \le \Phi_{\otimes/\mathtt{pwf}} \cdot \beta$,
(ii) $\left\|\mathsf{coeff}_{\otimes}\,(x)\right\|_\infty \le \beta \implies \left\|\mathsf{coeff}_{\mathtt{pwf}}\,(x)\right\|_\infty \le \Phi_{\mathtt{pwf}/\otimes} \cdot \beta$.

The translation factors are ring-specific. Instead of giving analytical upper bounds, we will specify the translation factors explicitly when instantiating $\bar{\mathcal{R}}$.

Twisted trace maps. For any Galois extension $\mathcal{K}/\mathcal{L}$, the field trace can be computed as $\mathsf{Trace}_{\mathcal{K}/\mathcal{L}} : \mathcal{K} \to \mathcal{L}$, $\mathsf{Trace}_{\mathcal{K}/\mathcal{L}}(x) := \sum_{\sigma_j \in \mathsf{Gal}(\mathcal{K}/\mathcal{L})} \sigma_j(x)$. When $\mathcal{L} = \mathbb{Q}$, we drop the subscript and write $\mathsf{Trace} = \mathsf{Trace}_{\mathcal{K}/\mathbb{Q}}$. For $a \in \mathcal{K}$, maps sending $(x, y) \in \mathcal{K}^2$ to $\mathsf{Trace}(a \cdot x \cdot \bar{y})$ are called twisted trace maps, where a is called the twist.

We consider specific twisted trace maps which compute the inner products of coefficients. We say that a subring $\bar{\mathcal{R}} \subseteq \mathcal{R} \subset \mathcal{K}$ is equipped with a (specific) twisted-trace map if there exists (efficiently computable) twist $a \in \bar{\mathcal{R}}$ and $\mathbb{Z}$-basis $\mathbf{b}$ of $\bar{\mathcal{R}}$ such that, for all $x, y \in \bar{\mathcal{R}}$,

$$\langle \mathsf{coeff}_{\mathbf{b}}(x), \mathsf{coeff}_{\mathbf{b}}(y) \rangle = \mathsf{Trace}(a \cdot x \cdot \bar{y}).$$

As a concrete instance, let $\bar{\mathcal{R}} = \mathcal{R}_{\mathfrak{g}} \otimes \mathcal{R}^+_{\mathfrak{f}_0} \otimes \ldots \otimes \mathcal{R}^+_{\mathfrak{f}_{k-1}}$ be any tensor ring. Such a ring is equipped with a twisted-trace map with the twist $a_{\otimes} = \prod_{i \in [k]} (2 - \zeta_{\mathfrak{f}_i} - \zeta_{\mathfrak{f}_i}^{-1})$ and the basis $\mathbf{b} = \mathbf{b}_{\otimes}$. In particular, for all $x, y \in \bar{\mathcal{R}}$

$$\langle \mathsf{coeff}_{\otimes}(x), \mathsf{coeff}_{\otimes}(y) \rangle = \mathsf{Trace}(a_{\otimes} \cdot x \cdot \bar{y}).$$

Sometimes, we may compute $z = x \cdot \bar{y}$ before evaluating the trace. For that, we write

$$\mathsf{TwistedTrace}(z) := \mathsf{Trace}(a_{\otimes} \cdot z).$$

$\mathcal{R}$ as an $\bar{\mathcal{R}}$-module. We adopt the following lemma from [23] which allows viewing $\mathcal{R}$ as an $\bar{\mathcal{R}}$-module.

Lemma 1. If $\mathfrak{f}$ be an odd prime, then $\mathcal{R}$ can be seen as an $\mathcal{R}^+$-module with the basis $\{1, \zeta\}$. More generally, let $\bar{\mathcal{R}} = \mathcal{R}_{\mathfrak{g}} \otimes \mathcal{R}^+_{\mathfrak{f}_0} \otimes \ldots \otimes \mathcal{R}^+_{\mathfrak{f}_{k-1}}$ where $\mathfrak{g} = 2^d$ for some $d \in \mathbb{N}$ and $\mathfrak{f}_0, \ldots, \mathfrak{f}_{k-1}$ are distinct odd primes. Let $\mathfrak{f} := \mathfrak{g} \prod_{i \in [k]} \mathfrak{f}_i$. Then $\mathcal{R}$ is an $\bar{\mathcal{R}}$-module with the basis $\bigotimes_{i \in [k]} (1, \zeta_{\mathfrak{f}_i})$.

Furthermore, write $\mathbf{x} \in \mathcal{R}^m$ as a $\bar{\mathcal{R}}^m$-combination of $\bigotimes_{i \in [k]} (1, \zeta_{\mathfrak{f}_i})$. Let $\left\|\mathsf{coeff}_{\mathtt{pwf}}\,(\mathbf{x})\right\|_\infty \le \beta$, then each $\mathcal{R}^+$-coefficient $\hat{\mathbf{x}}$ of $\mathbf{x}$ satisfies $\left\|\mathsf{coeff}_{\mathtt{pwf}}\,(\hat{\mathbf{x}})\right\|_\infty \le 2^k \varphi \beta$.

Subtractive sets. A set $\mathcal{C} = \{\mu_1, \ldots, \mu_k\} \subseteq \mathcal{R}$ is said to be subtractive [29] if $a - b \in \mathcal{R}^\times$ for all distinct $a, b \in \mathcal{C}$. The expansion and inverse-expansion factors of $\mathcal{C}$ are $\gamma_{\mathcal{C}} := \max_{c \in \mathcal{C}, t \in \mathcal{R}, t \ne 0} \|t \cdot c\| / \|t\|$ and $\theta_{\mathcal{C}} := \max_{c, c' \in \mathcal{C}, c \ne c', t \in \mathcal{R},, t \ne 0} \left\|t \frac{1}{c - c'}\right\| / \|t\|$ respectively.

2.2. Arguments and Reductions of Knowledge

We recall the definition of argument systems which allow a prover to convince a verifier that a relation is satisfiable. Formally, we define a (family of) relation(s) $\Xi := (\Xi_\lambda)_{\lambda \in \mathbb{N}}$ to be polynomial-time-decidable triples of the form $(\mathsf{pp}, \mathsf{stmt}, \mathsf{wit})$, corresponding to the public parameters of the argument system, the statement, and the witness respectively. For any fixed public parameters pp, we define the relation

$$\Xi_{\mathsf{pp}} := \{ (\mathsf{stmt}, \mathsf{wit}) : (\mathsf{pp}, \mathsf{stmt}, \mathsf{wit}) \in \Xi \}$$

and the corresponding language

$$\mathcal{L}_{\mathsf{pp}} := \{ \mathsf{stmt} : \exists \ \mathsf{wit}, \ (\mathsf{stmt}, \mathsf{wit}) \in \Xi_{\mathsf{pp}} \} .$$

Definition 2.1 (Reduction of Knowledge (modified)). Let Ξ_0, Ξ_1 be ternary relations. A *reduction of knowledge (RoK)* Π from Ξ_0 to Ξ_1, short $\Pi\colon \Xi_0 \to \Xi_1$, is defined by two PPT algorithms $\Pi = (\mathcal{P}, \mathcal{V})$, the prover $\mathcal{P}$, and the verifier $\mathcal{V}$, with the following interface:

- $\mathcal{P}(\mathsf{pp}, \mathsf{stmt}, \mathsf{wit}) \to (\widetilde{\mathsf{stmt}}, \widetilde{\mathsf{wit}})$: Interactively reduce the input statement $(\mathsf{pp}, \mathsf{stmt}, \mathsf{wit}) \in \Xi_0$ to a new statement $(\mathsf{pp}, \widetilde{\mathsf{stmt}}, \widetilde{\mathsf{wit}}) \in \Xi_1$ or $\perp$.
- $\mathcal{V}(\mathsf{pp}, \mathsf{stmt}) \to \widetilde{\mathsf{stmt}}$: Interactively reduce the task of checking the input statement $(\mathsf{pp}, \mathsf{stmt})$ w.r.t Ξ_0 to checking a new statement $(\mathsf{pp}, \widetilde{\mathsf{stmt}})$ w.r.t. Ξ_1.

Let $\Xi_\top = \{(\mathsf{true}, \perp)\}$ be the relation encoding true: a verifier reducing to $\Xi_\top$ can output true if it accepts and any other string otherwise. A reduction of knowledge from Ξ_0 to $\Xi_\top$ is called an argument system for Ξ_0.

Definition 2.2 (Completeness). Let $\Pi = (\mathcal{P}, \mathcal{V})$ be a reduction of knowledge from Ξ_0 to Ξ_1. We say Π is complete if for all $(\mathsf{pp}, \mathsf{stmt}, \mathsf{wit}) \in \Xi_0$

$$\Pr\left[(\mathsf{pp}, \widetilde{\mathsf{stmt}}, \widetilde{\mathsf{wit}}) \notin \Xi_1\right] \leq \mathsf{negl}(\lambda),$$

where $(\widetilde{\mathsf{stmt}}, \widetilde{\mathsf{wit}}) \leftarrow \langle \mathcal{P}, \mathcal{V} \rangle(\mathsf{pp}, \mathsf{stmt}, \mathsf{wit})$.

Let S be a finite set and $\ell \in \mathbb{N}$ denote the number of coordinates. First, take two vectors $\mathbf{x} := (x_1, \ldots, x_\ell), \mathbf{y} := (y_1, \ldots, y_\ell) \in S^\ell$. Then, we define the following relation "$\equiv_i$" for fixed $i \in [\ell]$ as:

$$\mathbf{x} \equiv_i \mathbf{y} \iff x_i \neq y_i \wedge \forall j \in [\ell] \backslash \{i\}, x_j = y_j .$$

In other words, vectors $\mathbf{x}$ and $\mathbf{y}$ have the same entries in all coordinates apart from the i-th one. For $\ell = 1$, the relations boil down to checking whether two elements are distinct. Next, we define the set

$$\mathsf{SS}(S, \ell, k) := \left\{ \begin{array}{l} \{\mathbf{x}_1, \ldots, \mathbf{x}_K\} \in (S^\ell)^K : \\ \quad \exists e \in [K], \forall i \in [\ell], \\ \quad \exists J = \{j_1, \ldots, j_{k-1}\} \subseteq [K] \setminus \{e\}, \\ \quad \forall j \in J, \mathbf{x}_e \equiv_i \mathbf{x}_j \end{array} \right\}$$

where $K := \ell(k-1) + 1$. Next, we define coordinate-wise special soundness (CWSS) [30].

Definition 2.3 (Coordinate-Wise Special Soundness). Let $\Pi = (\mathcal{P}, \mathcal{V})$ be a public-coin 2μ-round reduction of knowledge from Ξ_0 to Ξ_1, where in each round the verifier picks a uniformly random challenge from S^ℓ. A tree of transcripts is a set of $K = (\ell(k-1)+1)\mu$ transcripts arranged in the following tree structure. The nodes in the tree correspond to the prover's messages and the edges correspond to the verifier's challenges. Each node at depth i has exactly $\ell(k-1)+1$ children corresponding to $\ell(k-1)+1$ distinct challenges which, as a set of vectors, lie in $\mathsf{SS}(S, \ell, k)$. Each transcript corresponds to exactly one root-to-leaf path.
We say that Π is ℓ-coordinate-wise k-special sound if there is a polynomial time algorithm that given public parameters pp, statement stmt and the tree of transcripts, outputs a witness $(\mathsf{stmt}, \mathsf{wit}) \in \Xi_0$. We recover the standard k-special soundness notion if $\ell = 1$.

Next, we define depth-preserving knowledge-soundness.

Definition 2.4 ((Depth-Preserving) knowledge-soundness). Let $\Pi = (\mathcal{P}, \mathcal{V})$ be a reduction of knowledge from Ξ_0 to Ξ_1. We say that Π is knowledge-sound from Ξ_0 to Ξ_1 with knowledge error $\kappa = \kappa(\mathsf{pp}, \mathsf{stmt})$ if for every (unbounded) algorithms $\mathcal{P}^*$, there exists exists an extractor $\mathcal{E}_{\mathcal{P}^*}$ such that: for all pp

$$\Pr\left[(\mathsf{pp}, \mathsf{stmt}, \mathsf{wit}^*) \notin \Xi_0 \wedge (\mathsf{pp}, \widetilde{\mathsf{stmt}}, \widetilde{\mathsf{wit}}) \in \Xi_1\right] \leq \kappa,$$

where the probability is taken over $(\mathsf{stmt}, \mathsf{st}) \leftarrow \mathcal{P}^*(\mathsf{pp})$ and $(\widetilde{\mathsf{stmt}}, \widetilde{\mathsf{wit}} \mid \mathsf{wit}^*) \leftarrow \langle (\mathcal{P}^* | \mathcal{E}_{\mathcal{P}^*})(\mathsf{pp}, \mathsf{stmt}, \mathsf{st}), \mathcal{V}(\mathsf{pp}, \mathsf{stmt}) \rangle$.
The extractor runs in $\mathsf{poly}(\lambda)$ or $\mathsf{quasipoly}(\lambda) = \mathsf{poly}(\lambda)^{\mathsf{polylog}(\lambda)}$. In the latter case, we say that the reduction is knowledge-sound with a $\mathsf{quasipoly}(\lambda)$ extractor.
The reduction of knowledge Π is said to be knowledge-sound from Ξ_0 to Ξ_1 if it is κ-knowledge-sound from Ξ_0 to Ξ_1 for some $\kappa \in \mathsf{negl}(\lambda)$. If in addition there exists a polynomial $p(\lambda) \in \mathsf{poly}(\lambda)$ such that the knowledge extractor $\mathcal{E}_{\mathcal{P}^*}$ has depth $\mathsf{Depth}(\mathcal{E}_{\mathcal{P}^*}) \leq \mathsf{Depth}(\mathcal{P}^*) + p(\lambda)$, the reduction of knowledge Π is said to be depth-preserving knowledge-sound from Ξ_0 to Ξ_1 .

Notice that in the case of argument systems, i.e., when Ξ_1 is the relation $\Xi_\top$ encoding true, we recover the knowledge-soundness definition of arguments systems: the condition $(\mathsf{pp}, \widetilde{\mathsf{stmt}}, \widetilde{\mathsf{wit}}) \in \Xi_1$ translates to $b = 1$, with b being the output of $\mathcal{V}$ in the interaction with $\mathcal{P}^*$.

Lemma 2. Let $\Pi = (\mathcal{P}, \mathcal{V})$ be a 2-round coordinate-wise special sound reduction of knowledge from Ξ_0 to Ξ_1. Then Π is depth-preserving knowledge-sound.

Proof: As far as the knowledge-soundness error is concerned, the proof is almost identical to that of [30] for argument system. Therefore, we will only highlight the differences with [30, Lemma 7.1] and argue about the depth-preserving property which was not considered in [30].

To prove knowledge-soundness of a CWSS 2-round reduction of knowledge one proceed in two steps: (i) one exploits access to a PPT $\mathcal{P}^*$ to extract depth-1 tree of transcripts as in Definition 2.3, and (ii) one uses such a tree of transcripts to recover a valid witness. For us it will be important to keep track of the depth of the extractor, which is the sum of the depth of these two subroutines. Clearly, depth of subroutine (ii) is constant. Hence, we only need to argue about the depth of subroutine (i).

Subroutine (i) is generic and given in [30, Lemma 7.1]: let ET denote the expected (parallel) runtime of the extractor from [30] and κ its soundness error. We modify such subroutine in two ways: we truncate its computation to a sufficiently large polynomial number of steps (which may depend on the success probability of the cheating prover) bigger than its expected runtime ET, and we sample challenges in parallel, instead of sequentially. The first modification, which by Markov's inequality can be shown to still yield a truncated extractor with non-negligible success probability is used to obtain an extractor with a strict time bound. The second modification, which requires accounting for the additional (negligible) possibility of collision of challenges, is required to obtained an extractor with depth strictly linear in the depth of $\mathcal{P}^*$. In other words, the modified subroutine (i) yields an extractor with parallel runtime $p_1(\lambda) \cdot \mathsf{ET}$, soundness error $\kappa + \varepsilon(\lambda)$ and depth $\mathsf{Depth}(\mathcal{P}^*) + p_2(\lambda)$, for $p_1, p_2 \in \mathsf{poly}(\lambda)$ and $\varepsilon \in \mathsf{negl}(\lambda)$, as required. ■

Definition 2.5 (Succinctness). An argument system Π for Ξ is said to have succinct proofs (resp. succinct verifier) if for any $\mathsf{pp} \in \mathsf{Setup}(1^\lambda)$, $(\mathsf{stmt}, \mathsf{wit}) \in \Xi_{\mathsf{pp}}$, the communication complexity of $\langle \mathcal{P}(\mathsf{pp}, \mathsf{stmt}, \mathsf{wit}), \mathcal{V}(\mathsf{pp}, \mathsf{stmt}) \rangle$ (resp. computation complexity of $\mathsf{Verify}(\mathsf{pp}, \mathsf{stmt})$) is $\mathsf{polylog}(|\mathsf{stmt}| + |\mathsf{wit}|) \cdot \mathsf{poly}(\lambda)$ where the $\mathsf{poly}(\lambda)$ factor is independent of $|\mathsf{stmt}|$ and $|\mathsf{wit}|$.

2.3. Verifiable Delay Function

We recall the notion of verifiable delay functions (VDF). To avoid handling random oracles, in the below, we state a variant where the verification algorithm is an interactive algorithm run between a prover and a verifier, with the expectation that such a verification algorithm can be turned non-interactive using the Fiat-Shamir transform.

Definition 2.6. A VDF is a tuple of PPT (interactive) algorithms $\mathsf{VDF} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Eval}, \mathcal{P}, \mathcal{V})$ with the following syntax:

- $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$: On input a statistical security parameter 1^λ outputs public parameters pp.
- $\mathsf{inst} \leftarrow \mathsf{Gen}(\mathsf{pp})$: On input public parameters pp, generates a problem instance $\mathsf{inst} \in \mathcal{X}$.
- $(\mathsf{val}, \mathsf{aux}) \leftarrow \mathsf{Eval}(\mathsf{pp}, \mathsf{inst}, 1^T)$: On input an instance inst and a time difficulty parameter T (in unary), deterministically output a value val and some auxiliary information aux.
- $b \leftarrow \langle \mathcal{P}(\mathsf{pp}, \mathsf{inst}, T, \mathsf{aux}), \mathcal{V}(\mathsf{pp}, \mathsf{inst}, T, \mathsf{val}) \rangle$: Both parties input the public parameters pp, an instance inst and a time parameter T (in binary). The prover further inputs some auxiliary information aux while the verifier inputs a value val. The interactive algorithm concludes with the verifier outputting $b \in \{0, 1\}$.

By the definition of PPT, Setup and Gen are required to run in fixed $\mathsf{poly}(\lambda)$ time independent of T. Moreover, for any inst, Eval and Verify run in time $\mathsf{poly}(\lambda, T)$ and $\mathsf{poly}(\lambda, \log T)$ respectively.

Correctness states that every output of Eval must be accepted by Verify with a high enough probability.

Definition 2.7 (ϵ-Correctness). VDF is $\epsilon = \epsilon(\lambda, T)$-correct if for all λ, T it holds that

$$\Pr[\neg \langle \mathcal{P}(\mathsf{pp}, \mathsf{inst}, T, \mathsf{aux}), \mathcal{V}(\mathsf{pp}, \mathsf{inst}, T, \mathsf{val}) \rangle] \leq \epsilon$$

with probability taken over the randomness of $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$, $\mathsf{inst} \leftarrow \mathsf{Gen}(\mathsf{pp})$, $(\mathsf{val}, \mathsf{aux}) \leftarrow \mathsf{Eval}(\mathsf{pp}, \mathsf{inst}, 1^T)$ and $\langle \mathcal{P}, \mathcal{V} \rangle$.

The output val for an instance inst at any time T is unique because Eval is deterministic. Furthermore, soundness guarantees that no efficient adversary could convince an honest verifier to accept an incorrect output. In the below, we state a soundness definition which excludes cases where the adversary specifies an instance which cannot be verified, i.e. correctness fails.

Definition 2.8 (Soundness). VDF is sound if for all PPT algorithm $\mathcal{P}^*$

$$\Pr\left[\begin{array}{c} \langle \mathcal{P}^*(\mathsf{aux}'), \mathcal{V}(\mathsf{pp}, \mathsf{inst}, T, \mathsf{val}') \rangle = 1, \\ \mathsf{val} \neq \mathsf{val}' \end{array}\right] \leq \mathsf{negl}(\lambda)$$

where the probability is taken over $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$, $(\mathsf{inst}, T, \mathsf{val}', \mathsf{aux}') \leftarrow \mathcal{P}^*(1^\lambda)$, and $(\mathsf{val}, \mathsf{aux}) \leftarrow \mathsf{Eval}(\mathsf{pp}, \mathsf{inst}, 1^T)$.

Computing val should be an inherently sequential process. This is modelled by the σ-sequentiality property which requires that no adversary is able to compute the value val at time T of an honestly generated instance inst in parallel time $\sigma(T) < T$.

Definition 2.9 (Sequentiality). Let $\sigma = \sigma(\lambda, T)$. VDF is said to be σ-sequential if, for any PPT algorithm $\mathcal{P}^*$, it holds that

$$\Pr\left[\begin{array}{c} \langle \mathcal{P}^*(\mathsf{aux}), \mathcal{V}(\mathsf{pp}, \mathsf{inst}, T, \mathsf{val}) \rangle = 1 \\ \mathsf{Depth}(\mathcal{P}^*) < \sigma(\lambda, T) \end{array}\right] \leq \mathsf{negl}(\lambda)$$

where the probability is taken over $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$, $\mathsf{inst} \leftarrow \mathsf{Gen}(\mathsf{pp})$, and $(T, \mathsf{val}, \mathsf{aux}) \leftarrow \mathcal{P}^*(\mathsf{pp}, \mathsf{inst})$.

2.4. Computational Assumptions

We state some variants of the short integer solution (SIS) assumption [18].[3] First, we state the standard (module) SIS assumptions with canonical ℓ_2- and coefficient ℓ_∞-norms respectively.

Definition 2.10. The $\mathsf{SIS}_{\bar{\mathcal{R}},q,m,n,\beta^{\mathsf{sis}},\mathbf{b}}$ problem is, given $\mathbf{A} \leftarrow\$ \bar{\mathcal{R}}_q^{n \times m}$, to find $\mathbf{x} \in \bar{\mathcal{R}}^m$ such that $\mathbf{A}\mathbf{x} = \mathbf{0} \bmod q$ and $0 < \|\mathsf{coeff}_{\mathbf{b}}(\mathbf{x})\| \leq \beta^{\mathsf{sis}}$. For such witness $\mathbf{w}$ and $\mathbf{A}$ we say that $(\mathbf{A}, \mathbf{w}) \in \Xi^{\mathsf{sis}}_{\bar{\mathcal{R}},q,n,m,\beta^{\mathsf{sis}},\mathbf{b}}$.

Next, we recall the vanishing SIS assumption [16], or specifically the specialisation defined in [23].

[3] More precisely, we state the problems, and the assumptions are that these problems are hard.

Definition 2.11. The $\mathsf{vSIS}_{\mathcal{R},q,n,\beta^{\mathsf{sis}},\mathbf{d},\mathbf{b}}$ problem is, given $\mathbf{F} \hookleftarrow\$ \mathcal{R}_q^{n\times\otimes\mathbf{d}}$, to find $\mathbf{x} \in \mathcal{R}^{\otimes\mathbf{d}}$ such that $\mathbf{F}\mathbf{x} = \mathbf{0} \bmod q$ and $0 < \|\mathsf{coeff}_{\mathbf{b}}(\mathbf{x})\| \leq \beta^{\mathsf{vsis}}$. Suppose $\mathbf{W} \in \mathcal{R}^{\otimes\mathbf{d}\times r}$ and $\mathbf{Y} \in \mathcal{R}_q^{n\times r}$ are such that $\mathbf{F}\mathbf{W} = \mathbf{Y} \bmod q$ and $0 < \|\mathsf{coeff}_{\mathbf{b}}(\mathbf{W})\| \leq \beta^{\mathsf{vsis}}$. We write $((\mathbf{F},\mathbf{Y}),\mathbf{W}) \in \Xi^{\mathsf{vsis}}_{\mathcal{R},q,n,r,\beta^{\mathsf{vsis}},\mathbf{d},\mathbf{b}}$. We omit $\mathbf{Y}$ if $\mathbf{Y} = \mathbf{0}$ and r if $r = 1$.

For convenience, we state the decision SIS assum (e.g. [31]) which is the computational analogue of the leftover hash lemma (LHL). For appropriate parameters $\bar{\mathcal{R}}, q, n, m, \chi$, decision SIS is implied by the LHL (unconditionally) or by the standard learning with errors (LWE) assumption (see e.g. [17]).

Definition 2.12. The $\mathsf{dSIS}_{\bar{\mathcal{R}},q,m,n,\chi}$ problem is, given $\mathbf{A} \hookleftarrow\$ \bar{\mathcal{R}}_q^{n\times m}$, and $\mathbf{t} \in \bar{\mathcal{R}}_q^{n}$, to distinguish whether $\mathbf{t} \hookleftarrow\$ \bar{\mathcal{R}}_q^{n}$ or $\mathbf{t} = \mathbf{A}\mathbf{x} \bmod q$ where $\mathbf{x} \hookleftarrow\$ \chi$. The distinguishing advantage of a PPT adversary $\mathcal{A}$, denoted $\epsilon_{\mathtt{dsis}}$, is

$$\left|\begin{array}{l} \Pr\left[\mathcal{A}(\mathbf{A},\mathbf{t}) = 1 \;\middle|\; \begin{array}{l} \mathbf{A} \hookleftarrow\$ \bar{\mathcal{R}}_q^{n\times m} \\ \mathbf{x} \hookleftarrow\$ \chi^m \\ \mathbf{t} := \mathbf{A}\cdot\mathbf{x} \end{array}\right] \\ - \Pr\left[\mathcal{A}(\mathbf{A},\mathbf{t}) = 1 \;\middle|\; \begin{array}{l} \mathbf{A} \hookleftarrow\$ \bar{\mathcal{R}}_q^{n\times m} \\ \mathbf{t} \hookleftarrow\$ \bar{\mathcal{R}}_q^{n} \end{array}\right] \end{array}\right|.$$

Finally, we recall the SIS-sequentiality assumption [17].[4]

Definition 2.13. The $\mathsf{SIS\text{-}Seq}_{\bar{\mathcal{R}},q,n,m,\mathbf{b}^{\otimes},\sigma}$ assumption states that, for any $T \in \mathbb{N}$ and any all polynomial-size adversary $\mathcal{A}$, it holds that

$$\Pr\left[\begin{array}{l} \begin{bmatrix}\mathbf{G}\\ \mathbf{A}\end{bmatrix}_{\searrow T} \cdot \mathbf{u} = \begin{pmatrix} -\mathbf{y}_0 \\ \mathbf{0} \\ \mathbf{y}_T \end{pmatrix} \bmod q \\ \wedge\ \mathsf{coeff}_{\otimes}(\mathbf{u}) \in \{0,1\}^{m\cdot T\cdot\delta} \\ \wedge\ \mathsf{Depth}(\mathcal{A}) < \sigma(\lambda, T) \end{array}\right]$$

is negligible in λ, where $\mathbf{A} \hookleftarrow\$ \bar{\mathcal{R}}_q^{n\times m}$, $\mathbf{y}_0 \hookleftarrow\$ \bar{\mathcal{R}}_q^{n}$, $\mathbf{y}_T \in \bar{\mathcal{R}}_q^{n}$ and $(\mathbf{y}_T, \mathbf{u}) \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{y}_0)$.

3. VDF Construction

At a high level, our VDF is essentially the one suggested in [16], i.e. equipping the sequential function of [17] with a compatible succinct argument system Π_T^{vdf}. In this section we formally state an abstract VDF construction based on a succinct argument system Π_T^{vdf}. Then, throughout Sections 4 to 6, we show how to efficiently instantiate Π_T^{vdf} based on the machinery developed in [23].

Let subring $\bar{\mathcal{R}} \subseteq \mathcal{R}$, modulus q and dimensions $\overline{n}', m' \in \mathbb{N}$ be parametrised by λ, where $\bar{\mathcal{R}}$ is a tensor ring equipped with the basis $\mathbf{b}_{\mathfrak{f}}^{\otimes}$, q is an odd prime and $m' := \overline{n}' \cdot \lfloor \log q \rfloor$. For $T \in \mathbb{N}$, let Π_T^{vdf} be an argument system for the relation $\Xi^{\mathsf{vdf}}_{\bar{\mathcal{R}},q,\overline{n}',T}$:

$$\boxed{\begin{array}{ll} \multicolumn{2}{l}{(\mathbb{x},\mathbb{w}) \in \Xi^{\mathsf{vdf}}_{\bar{\mathcal{R}},q,\overline{n}',t,r}, \text{ where } r = 1 \text{ when omitted}} \\ \hline \mathbb{x}\text{:} & \mathbf{A} \in \bar{\mathcal{R}}_q^{\overline{n}'\times m'},\ (\overline{\mathbf{y}}_{j\cdot t})_{j=0}^{r} \in (\bar{\mathcal{R}}_q^{\overline{n}'})^{r+1} \\ \mathbb{w}\text{:} & \mathbf{W} \in \bar{\mathcal{R}}^{m\times r} \\ & \text{where: } m = m' \cdot t,\ \overline{n} = \overline{n}' \cdot (t+1),\ T = t \cdot r \\ \text{s.t.} & \mathsf{coeff}_{\otimes}(\mathbf{W}) \in \{0,1\}^{m\times r\times\delta}\ \wedge \\ & \begin{bmatrix}\mathbf{G}\\ \mathbf{A}\end{bmatrix}_{\searrow t} \mathbf{W} = \begin{pmatrix} -\overline{\mathbf{y}}_0 & \cdots & -\overline{\mathbf{y}}_{(r-1)\cdot t} \\ \mathbf{0} & \cdots & \mathbf{0} \\ \overline{\mathbf{y}}_t & \cdots & \overline{\mathbf{y}}_T \end{pmatrix} \in \bar{\mathcal{R}}_q^{\overline{n}\times r} \end{array}}$$

We construct a VDF in Fig. 1 which makes use of the "gadget matrix" $\mathbf{G}$ and the binary decomposition operation $\mathbf{G}^{-1}$ recalled below.

Define the "gadget matrix" $\mathbf{G} := \mathbf{I}_{\overline{n}'} \otimes \mathbf{g}^{\mathtt{T}}$ where $\mathbf{I}_{\overline{n}'}$ is the $\overline{n}'$-dimensional identity matrix and $\mathbf{g}^{\mathtt{T}} := (1, 2, 4, \ldots, 2^{\lfloor \log q \rfloor - 1})$. Define the binary decomposition operator $\mathbf{G}^{-1} : \bar{\mathcal{R}}_q^{\overline{n}'} \to \bar{\mathcal{R}}^{m'} \cup \{\perp\}$ which maps an $\bar{\mathcal{R}}_q$ vector to its component-wise binary decomposition whenever possible or a failure symbol $\perp$ otherwise. That is, suppose $\mathbf{v} = \sum_{i=0}^{\lceil \log q \rceil} \mathbf{v}_i \cdot 2^i$ where $\mathsf{coeff}_{\otimes}(\mathbf{v}_i) \in \{0,1\}^{\overline{n}'\cdot\delta}$ for each i. Then $\mathbf{G}^{-1}(\mathbf{v}) = (\mathbf{v}_0 || \ldots || \mathbf{v}_{\lfloor \log q \rfloor - 1})$ if $\mathbf{v}_{\lceil \log q \rceil} = \mathbf{0}$ or $\mathbf{G}^{-1}(\mathbf{v}) = \perp$ otherwise. For a matrix $\mathbf{W} \in \bar{\mathcal{R}}_q^{\overline{n}'\times r}$, the notation $\mathbf{G}^{-1}$ is extended column-wise and we define $\mathbf{G}^{-1}(\mathbf{W}) = \perp$ if $\mathbf{G}^{-1}(\mathbf{w}_j) = \perp$ for any column $\mathbf{w}_j$. Note that $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{W}) = \mathbf{V} \bmod q$ unless $\mathbf{G}^{-1}(\mathbf{W}) = \perp$. We assume that any operation performed on $\perp$ results in $\perp$.

We next analyse the correctness error of the VDF.

Lemma 3. For a uniform random $\overline{\mathbf{y}} \hookleftarrow\$ \bar{\mathcal{R}}_q^{\overline{n}'}$,

$$\Pr\left[\mathbf{G}^{-1}(\overline{\mathbf{y}}) = \perp\right] \leq \overline{n}' \cdot \delta \cdot 2^{\log q - \lfloor \log q \rfloor}/q.$$

Furthermore, conditioning on $\mathbf{G}^{-1}(\overline{\mathbf{y}}) \neq \perp$, the distribution of $\mathbf{x} = \mathbf{G}^{-1}(\overline{\mathbf{y}}) = \perp$ is uniform over $\bar{\mathcal{R}}^{m'}$ subject to $\mathsf{coeff}_{\otimes}(\mathbf{x}) \in \{0,1\}^{m'\cdot\delta}$.

Proof: The first claim follows by viewing $\bar{\mathcal{R}}_q^{\overline{n}'} \cong \mathbb{Z}_q^{\overline{n}'\cdot\delta}$ as a $\mathbb{Z}_q$-module and by a union bound. The second claim is due to $\mathbf{G}^{-1}$ being bijective whenever well-defined. ■

Theorem 3.1. Suppose Π_T^{vdf} is perfectly complete for the relation $\Xi^{\mathsf{vdf}}_{\bar{\mathcal{R}},q,\overline{n}',T}$. Let $\epsilon_{\mathtt{dsis}}$ denote an upper of the advantage of any PPT adversary against $\mathsf{dSIS}_{\bar{\mathcal{R}},q,\overline{n}',m',\chi}$ where $\mathbf{x} \hookleftarrow\$ \chi$ is the uniform distribution over $\bar{\mathcal{R}}^{m'}$ subject to $\mathsf{coeff}_{\otimes}(\mathbf{x}) \in \{0,1\}^{m'\cdot\delta}$. The VDF in Fig. 1 is ϵ-correct for

$$\epsilon(\lambda, T) \leq T \cdot (\epsilon_{\mathtt{dsis}} + \overline{n}' \cdot \delta \cdot 2^{\log q - \lfloor \log q \rfloor}/q).$$

Proof: Since Π_T^{vdf} is perfectly complete for $\Xi^{\mathsf{vdf}}_{\bar{\mathcal{R}},q,\overline{n}',T}$, it is clear that whenever Eval successfully outputs $(\mathsf{val}, \mathsf{aux})$ with $\mathsf{val} \neq \perp$ then the verification of $(\mathsf{val}, \mathsf{aux})$ will pass. It therefore suffices to analyse the probability of val being $\perp$.

Define the following distributions of $(\mathbf{A}, \overline{\mathbf{y}}_T)$:

- $\mathcal{D}_0$: Sample uniformly random $\mathbf{A} \hookleftarrow\$ \bar{\mathcal{R}}_q^{\overline{n}'\times m'}$ and $\overline{\mathbf{y}}_0 \hookleftarrow\$ \bar{\mathcal{R}}_q^{\overline{n}'}$ and let $\overline{\mathbf{y}}_1, \ldots, \overline{\mathbf{y}}_T$ be as defined in Eval of Fig. 1.
- $\mathcal{D}_{t+1}$ for $t \in [T]$: Identical to $\mathcal{D}_t$ except that if $\overline{\mathbf{y}}_t \neq \perp$ then $\overline{\mathbf{y}}_{t+1}$ is replaced by a uniformly random sample from $\bar{\mathcal{R}}_q^{\overline{n}'}$.

[4] In [17], the assumption is stated in the "if ... then ..." style. We directly state the "then" part of the exact variant.

```
$\mathsf{Setup}(1^\lambda)$
$\mathbf{A} \leftarrow\!\!\$\ \bar{\mathcal{R}}_q^{\bar{n}' \times m'}$
return $\mathsf{pp} := \mathbf{A}$

$\mathsf{inst} \leftarrow \mathsf{Gen}(\mathsf{pp})$
$\bar{\mathbf{y}}_0 \leftarrow\!\!\$\ \bar{\mathcal{R}}_q^{\bar{n}'}$
return $\mathsf{inst} := \bar{\mathbf{y}}_0$
```

```
$(\mathsf{val}, \mathsf{aux}) \leftarrow \mathsf{Eval}(\mathsf{pp}, \mathsf{inst}, 1^T)$
for $i \in [T]$ do
    $\bar{\mathbf{x}}_i := \mathbf{G}^{-1}(-\bar{\mathbf{y}}_i)$
    $\bar{\mathbf{y}}_{i+1} := \mathbf{A}\bar{\mathbf{x}}_i \bmod q$
$\mathsf{val} := \bar{\mathbf{y}}_T$
$\mathsf{aux} := (\bar{\mathbf{x}}_i)_{i \in [T]}$
return $(\mathsf{val}, \mathsf{aux})$
```

```
$b \leftarrow \langle \mathcal{P}(\mathsf{pp}, \mathsf{inst}, T, \mathsf{aux}), \mathcal{V}(\mathsf{pp}, \mathsf{inst}, T, \mathsf{val}) \rangle$
$\mathcal{P}, \mathcal{V}$ :
    $\mathsf{params} := (\bar{\mathcal{R}}, q, \bar{n}', m', T, 1)$
    $\mathsf{stmt} := (\mathbf{A}, T, (\bar{\mathbf{y}}_0, \bar{\mathbf{y}}_T))$
    $b \leftarrow \Pi_T^{\mathsf{vdf}}.\langle \mathcal{P}(\mathsf{params}, \mathsf{stmt}, \mathsf{aux}), \mathcal{V}(\mathsf{params}, \mathsf{stmt}) \rangle$
$\mathcal{V}$ : return $b$
```

Figure 1: VDF Construction.

By the $\mathsf{dSIS}_{\bar{\mathcal{R}},q,\bar{n}',m',\chi}$ assumption, for any PPT $\mathcal{A}$,

$$|\Pr[\mathcal{A}(\mathcal{D}_t) = 1] - \Pr[\mathcal{A}(\mathcal{D}_{t+1}) = 1]| \leq \epsilon_{\mathtt{dsis}}.$$

In particular, the difference in the probability of $\bar{\mathbf{y}}_T = \bot$ between $(\mathbf{A}, \bar{\mathbf{y}}_T)$ sampled from $\mathcal{D}_0$ and $\mathcal{D}_T$ respectively is at most $T \cdot \epsilon_{\mathtt{dsis}}$. We next analyse the probability of $\bar{\mathbf{y}}_T = \bot$ for $(\mathbf{A}, \bar{\mathbf{y}}_T) \leftarrow\!\!\$\ \mathcal{D}_T$. In $\mathcal{D}_T$, note that $\bar{\mathbf{y}}_T \neq \bot$ if and only if $\bar{\mathbf{y}}_t \neq \bot$ for all $t \in [T]$ where $\bar{\mathbf{y}}_{t+1} \leftarrow\!\!\$\ \bar{\mathcal{R}}_q^{\bar{n}'}$ is independently uniformly random conditioned on $\bar{\mathbf{y}}_t \neq \bot$. By a union bound and applying Lemma 3 T times, the probability is at most $T \cdot \bar{n}' \cdot \delta \cdot 2^{\log q - \lfloor \log q \rfloor}/q$. ■

Note that in practical instantiations the modulus q is polynomial in λ (concretely e.g. 64-bit) and thus the correctness error ϵ in Theorem 3.1 is non-negligible. In practice, if the computation fails at some step t, i.e. $\mathbf{G}^{-1}(\bar{\mathbf{y}}_t) = \bot$, then one could rerandomise $\bar{\mathbf{y}}_t$ by hashing, i.e. $\bar{\mathbf{y}}'_t = \mathsf{H}(\bar{\mathbf{y}}_t)$, and continue with the computation with $\bar{\mathbf{y}}'_t$. In that case, the prover would prove about the validity of each successful segment separately. As long as (the true value of) ϵ is decently small (e.g. say below $1/1000$), the number of segments would not be too large and thus succinctness is retained.

Theorem 3.2. Suppose Π_T^{vdf} is a depth-preserving knowledge-sound argument system for the relation $\Xi^{\mathsf{vdf}}_{\bar{\mathcal{R}},q,\bar{n}',T}$ for all $T \in \mathbb{N}$. Then:

(i) the VDF in Fig. 1 is sound (Definition 2.8);
(ii) if the $\mathsf{SIS\text{-}Seq}_{\bar{\mathcal{R}},q,\bar{n}',m',\mathbf{b}^{\otimes},\sigma'}$ assumptions holds then the VDF in Fig. 1 is σ-sequential for some $\sigma = \sigma'/\mathsf{poly}(\lambda)$ (Definition 2.9).

Proof: Let κ denote the knowledge-soundness error of argument system Π_T^{vdf}.

(i). If there exists an adversary $\mathcal{P}^*$ that breaks the soundness of VDF with probability $\epsilon_{\mathtt{vdf\text{-}snd}}$, then $\kappa \geq \epsilon_{\mathtt{vdf\text{-}snd}}$.

This is quite straightforward: indeed a valid adversary $\mathcal{P}^*$ against the soundness of the VDF is able to convince $\mathcal{V}$ of an incorrect output $\mathsf{val}' = \mathbf{y}'_T$ for some time difficulty parameter $T \in \mathbb{N}$, public parameters $\mathsf{pp} = \mathbf{A}$ and instance $\mathsf{inst} = \mathbf{y}_0$. By definition of the relation $\Xi^{\mathsf{vdf}}_{\bar{\mathcal{R}},q,\bar{n}',T}$ given $(\mathbf{A}, \mathbf{y}_0)$ then statement $\mathsf{stmt} = (\mathbf{A}, \mathbf{y}_0, \mathbf{y}_T)$ and corresponding withness $\mathsf{wit} = \mathbf{w}$ are uniquely determined. In particular, this implies that $\mathsf{stmt}' := (\mathbf{A}, \mathbf{y}_0, \mathbf{y}'_T)$ is not in the language defined by the relation $\Xi^{\mathsf{vdf}}_{\bar{\mathcal{R}},q,\bar{n}',T}$, i.e. no valid possible witness exists.

(ii). We show that if there exists an adversary $\mathcal{P}^*$ that breaks the σ-sequentiality of the VDF VDF with probability $\epsilon_{\mathtt{vdf\text{-}seq}}$, then we can construct a reduction R against the $\mathsf{SIS\text{-}Seq}_{\bar{\mathcal{R}},q,\bar{n}',m',\mathbf{b}^{\otimes},2,\sigma'}$ problem that succeeds with probability $\epsilon_{\mathtt{sis\text{-}seq}} + \kappa \geq \epsilon_{\mathtt{vdf\text{-}seq}}$.

Consider a reduction R, which on input a $\mathsf{SIS\text{-}Seq}_{\bar{\mathcal{R}},q,\bar{n}',m',\mathbf{b}^{\otimes},2,\sigma'}$ instance outputs $T \in \mathbb{N}$ and $\mathbf{u} \in \bar{\mathcal{R}}^{n(T+1)}$:

- it parses the $\mathsf{SIS\text{-}Seq}_{\bar{\mathcal{R}},q,\bar{n}',m',\mathbf{b}^{\otimes},2,\sigma}$ instance $(\mathbf{A}, \mathbf{y}_0)$ and sets $\mathsf{pp} := \mathbf{A}$ and $\mathsf{inst} := \mathbf{y}_0$.
- it runs $\mathcal{P}^*$ on input $(\mathsf{pp}, \mathsf{inst})$ and obtains $(T, \mathsf{val}, \mathsf{aux}) \leftarrow \mathcal{P}^*(\mathsf{pp}, \mathsf{inst})$.
- it sets $\mathsf{stmt} := (\mathbf{A}, T, \mathbf{y}_0, \mathsf{val})$, and $\mathsf{st} = \mathsf{aux}$.
- let $\mathcal{E}_{\mathcal{P}^*}$ the extractor whose existence is guaranteed by the knowledge-soundness of Π_T^{vdf}, it runs

$$(b, \mathsf{wit}^*) \leftarrow \langle (\mathcal{P}^* | \mathcal{E}_{\mathcal{P}^*})(\mathsf{pp}, \mathsf{stmt}, \mathsf{st}), \mathcal{V}(\mathsf{pp}, \mathsf{stmt}) \rangle\,.$$

- it sets $\mathbf{u} := \mathsf{wit}^*$ and outputs $(T, \mathbf{u})$.

Notice that, by construction, $\mathsf{Depth}(\mathsf{R}) \leq \mathsf{Depth}(\mathcal{E}_{\mathcal{P}^*}) + q(\lambda)$ for some $q \in \mathsf{poly}(\lambda)$. By the depth-preserving (knowledge-soundness) of Π_T^{vdf}, $\mathsf{Depth}(\mathcal{E}_{\mathcal{P}^*}) \leq \mathsf{Depth}(\mathcal{P}^*) + p(\lambda)$ for some $p \in \mathsf{poly}(\lambda)$. Combining the two we obtain that

$$\mathsf{Depth}(\mathsf{R}) \leq \mathsf{Depth}(\mathcal{P}^*) + r(\lambda), \tag{2}$$

for some $r \in \mathsf{poly}(\lambda)$. If $(\mathcal{A}, \mathcal{P}^*)$ is a valid adversary against the σ-sequentiality of the VDF VDF then $\mathsf{Depth}(\mathcal{P}^*) < \sigma(\lambda, T)$, which, together with the Eq. (2), implies that

$$\mathsf{Depth}(\mathsf{R}) < r(\lambda) \cdot \sigma(\lambda, T) = \underbrace{\mathsf{poly}(\lambda) \cdot T}_{\sigma'(\lambda, T)}\,. \tag{3}$$

Let us now consider the properties of the output of the reduction R. By the (depth-preserving) knowledge-soundness of Π_T^{vdf}, except with probability κ, it must be the case that whenever $b = 1$ it holds that $(\mathsf{stmt}, \mathsf{wit}^*) \in \Xi^{\mathsf{vdf}}_{\bar{\mathcal{R}},q,\bar{n}',T}$, i.e.,

$$\begin{bmatrix}\mathbf{G}\\ \mathbf{A}\end{bmatrix}_{\searrow T} \mathbf{u} = \mathbf{y} \bmod q \quad \wedge \quad \mathsf{coeff}_{\otimes}(\mathbf{u}) \in \{0,1\}^{m \times \delta} \tag{4}$$

where $\mathbf{u} = \mathsf{wit}^*$ and $\mathbf{y}^{\mathsf{T}} = \begin{pmatrix}\mathbf{y}_0^{\mathsf{T}} & \mathbf{0}^{\mathsf{T}} & \cdots & \mathbf{0}^{\mathsf{T}} & \mathbf{y}_T^{\mathsf{T}}\end{pmatrix}^{\mathsf{T}}$, i.e., $\mathsf{wit}^* = \mathbf{u}$ is a valid solution to the SIS-sequentiality problem. Putting together Eqs. (3) and (4), we obtain that if $(\mathcal{A}, \mathcal{P}^*)$ breaks the σ-sequentiality of the VDF VDF then R breaks the $\mathsf{SIS\text{-}Seq}_{\bar{\mathcal{R}},q,\bar{n}',m',\mathbf{b}^{\otimes},2,\sigma'}$ assumption as required. ■

4. Folding-based and Norm-control RoKs

The work of in [23] has established a comprehensive toolkit for succinct lattice-based arguments. The toolkit comprises a set of self-RoK (Reduction of Knowledge) protocols for Ξ^{lin}, the bounded-norm satisfiability for a specific set of structured linear relations, and also RoKs from other extended relations to Ξ^{lin}. A refined formulation of Ξ^{lin} is the following.

$$\boxed{\begin{array}{l} \underline{(\mathbb{x}, \mathbb{w}) \in \Xi^{\mathsf{lin}}_{\mathcal{R},q,m,n,r,\mathbf{d},\beta}} \\ \mathbb{x}\text{:}\ \mathbf{F} \in \mathcal{R}_q^{n \times \otimes \mathbf{d}},\ \mathbf{Y} \in \mathcal{R}_q^{n \times r} \\ \mathbb{w}\text{:}\ \mathbf{W} \in \mathcal{R}^{m \times r},\ \text{where } m = \prod_i d_i \\ \text{s.t. } \|\mathbf{W}\| \leq \beta,\ \mathbf{F}\mathbf{W} = \mathbf{Y} \bmod q \end{array}}$$

The main purpose of these RoKs is twofold:

(i) To reduce the witness dimension and hence size.
(ii) To reduce the norm of the witness during the RoK execution and knowledge extraction.

Below, we state lemmas summarising the RoKs which will be used throughout this work.

4.1. Folding-based RoKs

Π^{split} and Π^{fold} are RoKs aimed at reducing the dimension of the witness. Π^{split} splits the witness matrix $\mathbf{W}^{(0)}$ into d_0 parts vertically and then concatenates these chunks horizontally, forming a new matrix $\mathbf{W}^{(1)}$. Additionally, the statement $\mathbf{F} = \mathbf{R} \bullet \widetilde{\mathbf{F}}$ is transformed so that the new statement becomes $\widetilde{\mathbf{F}}$. For the newly computed image $\widetilde{\mathbf{Y}}$, which is sent to the verifier, the following holds:

$$\widetilde{\mathbf{F}}\mathbf{W}^{(1)} = \widetilde{\mathbf{Y}} \bmod q.$$

Π^{split} is typically followed by Π^{fold}, which randomly combines the r_{in} columns of $\mathbf{W}^{(0)}$ into $\mathbf{W}^{(1)}$ with r_{out} columns, at the expense of a slight increase in the witness norm.

Lemma 4 (Split). There exists a self-reduction of knowledge Π^{split} which is perfectly complete for Ξ^{lin} with parameters

$$(m, n, r, \mathbf{d}, \beta) \mapsto (m/d_0, n, r, \mathbf{d}_{[1,|\mathbf{d}|]}, \beta)\ .$$

and d_0-special sound with challenge set $\mathcal{C}_{\mathcal{R}_q}$ for $\Xi^{\mathsf{lin}} \cup \Xi^{\mathsf{vsis}}$ with parameters

$$\begin{matrix} (m, n, r, \mathbf{d}, \beta') \\ (\mathcal{R}, q, n, \mathbf{d}, \beta^{\mathsf{vsis}}) \end{matrix} \hookleftarrow \begin{matrix} (m/d_0, n, r, \mathbf{d}_{[1,|\mathbf{d}|]}, \beta') \\ (\mathcal{R}, q, n, \mathbf{d}_{[1,|\mathbf{d}|]}, \beta^{\mathsf{vsis}}) \end{matrix}$$

where $2\beta' \leq \beta^{\mathsf{vsis}}$. The prover communicates $nrd_0^2 \log |\mathcal{R}_q|$ bits.

Lemma 5 (Fold). There exists a self-reduction of knowledge Π^{fold} which is perfectly complete for Ξ^{lin} with parameters

$$(m, n, r_{\mathsf{in}}, \mathbf{d}, \beta) \mapsto (m, n, r_{\mathsf{out}}, \mathbf{d}, r_{\mathsf{in}}\gamma_{\mathcal{C}}\beta)$$

and r_{in}-coordinate-wise special-sound with challenge set $\mathcal{C}^{r_{\mathsf{out}}}$ for $\Xi^{\mathsf{lin}} \cup \Xi^{\mathsf{vsis}}$ with parameters

$$\begin{matrix} (m, n, r_{\mathsf{in}}, \mathbf{d}, 2\theta_\infty\beta') \\ (\mathcal{R}, q, n, \mathbf{d}, \beta^{\mathsf{vsis}}) \end{matrix} \hookleftarrow \begin{matrix} (m, n, r_{\mathsf{out}}, \mathbf{d}, r_{\mathsf{in}}\beta') \\ (\mathcal{R}, q, n, \mathbf{d}_{[1,|\mathbf{d}|]}, \beta^{\mathsf{vsis}}) \end{matrix}$$

where $0 \leq \beta' \leq \beta^{\mathsf{vsis}} \leq q$ and $\mathcal{C}$ is a subtractive set. There is no prover communication cost.

4.2. Norm-control RoKs

The RoKs recalled in Section 4.1 do not control the norm growth of the witness in the RoK executions nor in knowledge extraction. Below, we recall another category of RoKs for managing the norm of the witness.

$\Pi^{b\text{-}\mathsf{decomp}}$ performs a radix-b decomposition of the witness into some ℓ parts. This ensures that the coefficient ℓ_∞-norm of the resulting witness is effectively reduced to $b/2$ (assuming balanced representation). Π^{norm} upgrades a relaxed norm claim to an exact one. $\Pi^{\mathsf{bin\text{-}norm}}$ strengthens Π^{norm}. It upgrades a relaxed norm claim to a binariness claim, i.e. that the coefficients of the witness are binary, but only works if the witness resides in a tensor ring $\bar{\mathcal{R}} \subseteq \mathcal{R}$.

Lemma 6 (Decomp). There exists a self-reduction of knowledge $\Pi^{b\text{-}\mathsf{decomp}}$ which is perfectly complete for Ξ^{lin} with parameters

$$(m, n, r, \mathbf{d}, \beta) \mapsto (m, n, r, \mathbf{d}, \tfrac{b}{2})$$

and 1-special-sound for $\Xi^{\mathsf{lin}} \cup \Xi^{\mathsf{vsis}}$ with parameters

$$\begin{matrix} (m, n, r_{\mathsf{in}}, \mathbf{d}, \beta_0') \\ (\mathcal{R}, q, n, \mathbf{d}, \beta^{\mathsf{vsis}}) \end{matrix} \hookleftarrow \begin{matrix} (m, n, r_{\mathsf{out}}, \mathbf{d}, \beta_1') \\ (\mathcal{R}, q, n, \mathbf{d}, \beta^{\mathsf{vsis}}) \end{matrix}$$

where $\beta_0' = \frac{b^\ell - 1}{b-1} \cdot \beta_1'$ and $\ell = \lceil \log_b(2\beta + 1) \rceil$. The prover communicates $nr\ell \log |\mathcal{R}_q|$ bits.

So far, we have stated the soundness guarantees of all RoKs in terms of (coordinate-wise) special-soundness, expecting that they will be upgraded to depth-preserving knowledge-soundness via Lemma 2. For the following RoKs Π^{norm} and $\Pi^{\mathsf{bin\text{-}norm}}$, we directly state that they are depth-preserving knowledge-sound instead because their soundness relies on a non-black-box combination of special-soundness and the Schwartz-Zippel lemma. The formal analyses follow along the lines of Lemma 2 and are omitted.

Lemma 7 (Norm Check). There exists a self-reduction of knowledge Π^{norm} which is perfectly complete for Ξ^{lin}

$$(m, n, r, \mathbf{d}, \beta) \mapsto (m, n', r', \mathbf{d}, \max(\beta, b_{\mathsf{ip}}))$$

and depth-preserving knowledge-sound for $\Xi^{\mathsf{lin}} \cup \Xi^{\mathsf{vsis}}$ with parameters

$$\begin{matrix} (m, n, r_{\mathsf{in}}, \mathbf{d}, \beta_0') \\ (\mathcal{R}, q, n, \mathbf{d}, \beta^{\mathsf{vsis}}) \end{matrix} \hookleftarrow \begin{matrix} (m, n', r', \mathbf{d}, \beta_1') \\ (\mathcal{R}, q, n, \mathbf{d}, \beta^{\mathsf{vsis}}) \end{matrix}$$

and knowledge error $(2m-1)/|\mathcal{C}_{\mathcal{R}_q}|$ where $\ell \leq r \cdot \log_{b_{\mathsf{ip}}}(mr\gamma_{\mathcal{R}}\beta^2)$, $n' = n + 3$, $r' = r + \ell + 1$, $0 \leq \beta \leq \beta^{\mathsf{sis}} \leq q$, $2\beta_1' \leq \beta^{\mathsf{sis}}$, $\beta_0' = \sqrt{mr\hat{\mathfrak{f}}\varphi\beta}$. The prover communicates

$$(n \log_{b_{\mathsf{ip}}} q + 3 + 3 \log b_{\mathsf{ip}}) r \log |\mathcal{R}_q| \text{bits}.$$

We define a supportive relation $\Xi^{\mathsf{bin}\otimes}$ which certifies the binariness of the witness.

$(\mathbb{x}, \mathbb{w}) \in \Xi^{\mathsf{bin}\otimes}_{\mathcal{R},\bar{\mathcal{R}},m,r}$

$\mathbb{x}$: $*$
$\mathbb{w}$: $\mathbf{W} \in \mathcal{R}^{m \times r}$
s.t. $\mathsf{coeff}_{\otimes}(\mathbf{W}) \in \{0,1\}^{mr\delta}$ or $\mathbf{W} \notin \bar{\mathcal{R}}^{m \times r}$

Lemma 8 (Bin Check). Let $\bar{\mathcal{R}} \in \mathcal{R}$ be a ring equipped with a twisted-trace map. There exists a reduction of knowledge $\Pi^{\mathsf{bin\text{-}norm}}$ which is perfectly complete from $\Xi^{\mathsf{bin}\otimes}_{\mathcal{R},\bar{\mathcal{R}}} \cap \Xi^{\mathsf{lin}}_{\mathcal{R}}$ to $\Xi^{\mathsf{lin}}_{\mathcal{R}}$ with parameters

$$(m, n, r, \mathbf{d}, \beta) \mapsto (m, n', r', \mathbf{d}, \max(\beta, b_{\mathsf{ip}}))$$

and depth-preserving knowledge-sound from $(\Xi^{\mathsf{bin}\otimes}_{\mathcal{R},\bar{\mathcal{R}}} \cap \Xi^{\mathsf{lin}}_{\mathcal{R}}) \cup \Xi^{\mathsf{vsis}}$ to $\Xi^{\mathsf{lin}}_{\mathcal{R}} \cup \Xi^{\mathsf{vsis}}$ with parameters

$$\begin{matrix}(m, n, r_{\mathsf{in}}, \mathbf{d}, \beta'_0) \\ (\mathcal{R}, q, n, \mathbf{d}, \beta^{\mathsf{vsis}})\end{matrix} \hookleftarrow \begin{matrix}(m, n', r', \mathbf{d}, \beta'_1) \\ (\mathcal{R}, q, n, \mathbf{d}, \beta^{\mathsf{vsis}})\end{matrix}$$

and knowledge error $(2m-1)/\left|\mathcal{C}_{\mathcal{R}_q}\right|$, where $\ell \leq r \cdot \log_{b_{\mathsf{ip}}}(mr\gamma_{\mathcal{R}}\beta^2)$, $n' = n + 4$, $r' = r + \ell + 1$, $\beta = 1$, $2\beta'_1 \leq \beta^{\mathsf{sis}}$, $\beta'_0 = \sqrt{mr\hat{\mathfrak{f}}\varphi\beta}$. The prover communicates

$$(n \log b_{\mathsf{ip}}(q) + 4 + 4 \log b_{\mathsf{ip}}) r \log |\mathcal{R}_q| \text{bits}.$$

Proof: The $\Pi^{\mathsf{bin\text{-}norm}}$ protocol is a combination of two protocols from [23]: Π^{norm} and Π^{bin}. Therefore, the proof of completeness follows almost verbatim. For the extraction, we rely on the proofs from [23] with slight modifications. In brief, the binariness check operates on the subring $\bar{\mathcal{R}} \subseteq \mathcal{R}$ equipped with a twisted-trace map. In [23], it was implicitly assumed that $\mathcal{R} = \bar{\mathcal{R}}$. However, in this work we consider $\bar{\mathcal{R}}$ to be a proper subring. Suppose $\widetilde{\mathbf{W}}$ is an extracted candidate witness. There are two cases: (i) $\widetilde{\mathbf{W}} \in \bar{\mathcal{R}}$ and $\mathsf{coeff}_{\otimes}(\mathbf{W}) \in \{0,1\}^{mr\delta}$, i.e. the norm bound of $\widetilde{\mathbf{W}}$ is tightened to a binariness constraint. (ii) $\widetilde{\mathbf{W}} \notin \bar{\mathcal{R}}$, in this case Π^{bin} does not improve the norm bound. ■

5. Reduction of Sequential Relation

This section aims to construct a sequence of RoKs which will be used in Section 6 to reduce the relation Ξ^{vdf} defined in Section 3 to the relation Ξ^{lin} defined in Section 4.

Lemma 9. There exists a perfectly complete and depth-preserving knowledge-sound self-reduction of knowledge Π^{pack} for Ξ^{vdf} with parameters

$$(\bar{\mathcal{R}}, q, \overline{n}', T) \mapsto (\bar{\mathcal{R}}, q, \overline{n}', t, r),$$

where $t = T/r$ and $r \mid T$. The prover communicates $(r-1) \cdot \overline{n}' \log |\bar{\mathcal{R}}_q|$ bits.

Proof: Consider a $\Xi^{\mathsf{vdf}}_{\bar{\mathcal{R}},q,\overline{n}',T}$ instance $((\mathbf{A}, \overline{\mathbf{y}}_0, \overline{\mathbf{y}}_T), \mathbf{w})$. We construct Π^{vdf}_T as follows. The prover computes $\overline{\mathbf{y}}_t, \overline{\mathbf{y}}_{2t}, \ldots, \overline{\mathbf{y}}_{(r-1)t} \in \bar{\mathcal{R}}_q^{\overline{n}'}$ using $\mathbf{w}$, and send them to the verifier. It also rearranges $\mathbf{w} \in \bar{\mathcal{R}}_q^{m' \cdot T}$ into a r-column matrix $\mathbf{W} \in \bar{\mathcal{R}}_q^{(m' \cdot t) \times r}$. The resulting $\Xi^{\mathsf{vdf}}_{\bar{\mathcal{R}},q,\overline{n}',t,r}$ instance is then $((\mathbf{A}, (\overline{\mathbf{y}}_{i \cdot t})_{i=0}^{r}), \mathbf{W})$. Clearly, the prover communicates $(r-1) \cdot \overline{n}' \log |\bar{\mathcal{R}}_q|$ bits to the verifier, and the RoK is perfectly complete. Since the above transformation is efficiently invertible, perfect depth-preserving knowledge-soundness follows. ■

We enrich the relation Ξ^{vdf} with a “bottom part”, capturing the commitment relation of a vSIS-based commitment.

$(\mathbb{x}, \mathbb{w}) \in \Xi^{\mathsf{bind}}_{\mathcal{R},\bar{\mathcal{R}},q,n,t,r,\mathbf{d}}$

$\mathbb{x}$: $\mathbf{A} \in \bar{\mathcal{R}}_q^{\overline{n}' \times m'}$, $\underline{\mathbf{F}} \in \mathcal{R}_q^{\underline{n} \times \otimes \mathbf{d}}$,
$(\overline{\mathbf{y}}_{j \cdot t})_{j=0}^{r} \in (\bar{\mathcal{R}}_q^{\overline{n}'})^{r+1}$, $\underline{\mathbf{Y}} \in \mathcal{R}_q^{\underline{n} \times r}$
where:

$$n = \underline{n} + \overline{n}, \qquad \overline{n} = \overline{n}' \cdot (t+1), \qquad m' = d_0,$$
$$m = m' \cdot t, \qquad m' = \overline{n}' \cdot \lfloor \log q \rfloor$$

$\mathbb{w}$: $\mathbf{W} \in \bar{\mathcal{R}}^{m \times r}$, where $m = \prod_i d_i$
s.t. $((\underline{\mathbf{F}}, \underline{\mathbf{Y}}), \mathbf{W}) \in \Xi^{\mathsf{vsis}}_{\mathcal{R},q,\underline{n},\mathbf{d},\beta^{\mathsf{vsis}}} \wedge$
$((\mathbf{A}, (\overline{\mathbf{y}}_{j \cdot t})_{j=0}^{r}), \mathbf{W}) \in \Xi^{\mathsf{vdf}}_{\bar{\mathcal{R}},q,\overline{n}',t,r}$

The following lemma is immediate.

Lemma 10. There exists a reduction of knowledge Π^{bind} which is perfectly complete from $\Xi^{\mathsf{lin}} \cap \Xi^{\mathsf{vdf}}$ to Ξ^{bind} with parameters

$$\begin{matrix}(\mathcal{R}, q, m, \underline{n}, r, \mathbf{d}, \beta) \\ (\bar{\mathcal{R}}, q, \overline{n}', t, r)\end{matrix} \mapsto (\mathcal{R}, \bar{\mathcal{R}}, q, n, t, r, \mathbf{d})$$

and perfectly depth-preserving knowledge-sound from $(\Xi^{\mathsf{lin}} \cap \Xi^{\mathsf{vdf}}) \cup \Xi^{\mathsf{vsis}} \cup \Xi^{\mathsf{sis}}$ to $\Xi^{\mathsf{bind}} \cup \Xi^{\mathsf{vsis}} \cup \Xi^{\mathsf{sis}}$ with parameters

$$\begin{matrix}(\mathcal{R}, q, m, \underline{n}, r, \mathbf{d}, \beta) \\ (\bar{\mathcal{R}}, q, \overline{n}', t, r) \\ (\mathcal{R}, q, n, r, \mathbf{d}, \beta^{\mathsf{vsis}}) \\ (\bar{\mathcal{R}}, q, \overline{n}', m', \beta^{\mathsf{sis}})\end{matrix} \hookleftarrow \begin{matrix}(\mathcal{R}, \bar{\mathcal{R}}, q, n, t, r, \mathbf{d}) \\ (\mathcal{R}, q, n, r, \mathbf{d}, \beta^{\mathsf{vsis}}) \\ (\bar{\mathcal{R}}, q, \overline{n}', m', \beta^{\mathsf{sis}})\end{matrix}$$

where $n = \underline{n} + \overline{n}$ and $\beta = \Phi_{\otimes/_{\mathtt{pwf}}}$. There is no communication cost.

Finally, we define the following staircase structure induced by a matrix $\mathbf{A} \in \mathcal{R}_q^{n' \times m'}$

$$\mathsf{diag}_{s,n,m} : \mathcal{R}_q^{n' \times m'} \to \mathcal{R}_q^{n \times m},$$

where $\nu = m/m'$:

$$\mathsf{diag}_{s,n,m}(\mathbf{A}) = \begin{pmatrix} \mathbf{0} \in \mathcal{R}^{s \times m} \\ \mathbf{I}_{\nu} \otimes \mathbf{A} \\ \mathbf{0} \in \mathcal{R}^{(n - n' \cdot \nu - s) \times m} \end{pmatrix}.$$

We define a new relation $\Xi^{\mathsf{staircase}}$ as follows:

$(\mathbb{x}, \mathbb{w}) \in \Xi^{\mathsf{staircase}}_{\mathcal{R},q,m,\underline{n},\overline{n},\overline{n}',r,\mathbf{d},n^{\mathsf{sc}},\beta}$

$\mathbb{x}$: $\mathbf{F} \in \mathcal{R}_q^{n \times \otimes \mathbf{d}}$, $\mathbf{Y} \in \mathcal{R}_q^{n \times r}$

$\mathbb{w}$: $\mathbf{W} \in \mathcal{R}^{m \times r}$, where $m = \prod_i d_i$
s.t. $\|\mathbf{W}\| \leq \beta \ \wedge \ \mathbf{F}\mathbf{W} = \mathbf{Y} \bmod q$ where
$\mathbf{F} = \begin{pmatrix} \overline{\mathbf{F}} \\ \underline{\mathbf{F}} \end{pmatrix}$ can be parsed as $\underline{\mathbf{F}} \in \mathcal{R}_q^{\underline{n} \times \otimes \mathbf{d}}$ and
$\overline{\mathbf{F}} = \sum_{i \in [n^{\mathsf{sc}}]} \mathsf{diag}_{s_i,\overline{n},m}(\mathbf{A}_i) \in \mathcal{R}_q^{\overline{n} \times m}$
for some $\left((\mathbf{A}_i, s_i) \in (\mathcal{R}_q^{\overline{n}' \times d_0}, \mathbb{N})\right)_{i \in [n^{\mathsf{sc}}]}$

Lemma 11. Let $\bar{\mathcal{R}} \subseteq \mathcal{R}$ be a tensor ring. There exists a reduction of knowledge $\Pi^{\mathsf{staircase}}$ which is perfectly complete from Ξ^{bind} to $\Xi^{\mathsf{staircase}} \cap \Xi^{\mathsf{bin}\otimes}$ with parameters

$$(\mathcal{R}, \bar{\mathcal{R}}, q, n, t, r, \mathbf{d}) \mapsto \begin{matrix} (\mathcal{R}, q, m, \underline{n}, \overline{n}, d_0, r, \mathbf{d}, n^{\mathsf{sc}}, \beta) \\ (\bar{\mathcal{R}}, m, r) \end{matrix},$$

and perfectly depth-preserving knowledge-sound from $\Xi^{\mathsf{bind}} \cup \Xi^{\mathsf{vsis}} \cup \Xi^{\mathsf{sis}}$ to $(\Xi^{\mathsf{staircase}} \cap \Xi^{\mathsf{bin}\otimes}) \cup \Xi^{\mathsf{vsis}}$ with parameters

$$\begin{matrix} (\mathcal{R}, \bar{\mathcal{R}}, q, n, t, r, \mathbf{d}) \\ (\mathcal{R}, q, n, \mathbf{d}, \beta^{\mathsf{vsis}}) \\ (\bar{\mathcal{R}}, q, \overline{n}', m', \beta^{\mathsf{sis}}) \end{matrix} \hookleftarrow \begin{matrix} (\mathcal{R}, q, m, \underline{n}, \overline{n}, \overline{n}', r, \mathbf{d}, n^{\mathsf{sc}}, \beta') \\ (\mathcal{R}, q, n, \mathbf{d}, \beta^{\mathsf{vsis}}) \end{matrix}$$

where $n^{\mathsf{sc}} = 2$, $m' = \overline{n}' \cdot \lfloor \log q \rfloor$, $m = m' \cdot t$,$n = \underline{n} + \overline{n}$, $m' = d_0$, $\beta = \Phi_{\mathtt{pwf}/\otimes}$, $\beta^{\mathsf{sis}} \geq 2^k \varphi \beta'$ and k is defined as in Lemma 1. There is no communication cost.

Proof: We construct $\Pi^{\mathsf{staircase}}$ as follows. Consider a Ξ^{bind} instance $((\mathbf{A}, \underline{\mathbf{F}}, (\overline{\mathbf{y}}_{j \cdot t})_{j=0}^{r}, \underline{\mathbf{Y}}), \mathbf{W})$. Both the prover and the verifier constructs $\overline{\mathbf{F}} := \begin{bmatrix} \mathbf{G} \\ \mathbf{A} \end{bmatrix}_{\searrow t}$, $\overline{\mathbf{Y}} := \begin{pmatrix} -\overline{\mathbf{y}}_0 & \cdots & -\overline{\mathbf{y}}_{(r-1) \cdot t} \\ \mathbf{0} & \cdots & \mathbf{0} \\ \overline{\mathbf{y}}_t & \cdots & \overline{\mathbf{y}}_T \end{pmatrix}$, $\mathbf{F} := \begin{pmatrix} \overline{\mathbf{F}} \\ \underline{\mathbf{F}} \end{pmatrix}$ and $\mathbf{Y} := \begin{pmatrix} \overline{\mathbf{Y}} \\ \underline{\mathbf{Y}} \end{pmatrix}$. Note that $\overline{\mathbf{F}}$ has a staircase structure. Indeed, we have $((\mathbf{F}, \mathbf{Y}), \mathbf{W}) \in \Xi^{\mathsf{staircase}}$, hence perfect completeness follows.

For perfect knowledge-soundness, if the knowledge extractor is given an instance of $\Xi^{\mathsf{vsis}}_{\mathcal{R},q,\underline{n},\mathbf{d},\beta^{\mathsf{vsis}}}$, it simply outputs the instance. It remains to consider two cases. First, if $\mathbf{W} \in \bar{\mathcal{R}}^{m \times r}$, then $\mathsf{coeff}(\mathbf{W}) \in \{0,1\}^{mr\delta}$ and $((\mathbf{A}, \underline{\mathbf{F}}, \mathbf{Y}), \mathbf{W}) \in \Xi^{\mathsf{bind}}$. Second, suppose $\mathbf{W} \notin \bar{\mathcal{R}}^{m \times r}$. We write

$$\begin{bmatrix} \mathbf{G} \\ \mathbf{A} \end{bmatrix}_{\searrow t} \mathbf{W} = \overline{\mathbf{Y}}$$

and split the witness $\mathbf{W}$ into $t \times r$ blocks $(\mathbf{w}_{i,j})_{(i,j) \in [t,r]}$. We observe that $\mathbf{A} \cdot \mathbf{w}_{t_i, r-1} = \overline{\mathbf{y}}_t$ where $\mathbf{A} \in \bar{\mathcal{R}}_q^{\overline{n}' \times m'}$ and $\overline{\mathbf{y}}_t \in \bar{\mathcal{R}}_q^{\overline{n}'}$. Viewing $\mathcal{R}$ as a $\bar{\mathcal{R}}$-module with basis $\mathbf{b} := \{b_0, \ldots, b_{\eta-1}\}$, we can split the witness as $\mathbf{w}_{t,r-1} = \sum_{i \in [\eta]} b_i \mathbf{w}^{(i)}_{t,r-1}$, where $\mathbf{w}^{(i)}_{t,r-1} \in \bar{\mathcal{R}}^{m'}$. By Lemma 1, $\left\|\mathsf{coeff}\left(\mathbf{w}^{(i)}_{t,r-1}\right)\right\|_{\infty} \leq 2^k \varphi \beta'$. Therefore, for $i \neq 0$, $\mathbf{A} \cdot \mathbf{w}^{(i)}_{t,r-1} = \mathbf{0}$. In other words, $(\mathbf{A}, \mathbf{w}^{(i)}_{t,r-1}) \in \Xi^{\mathsf{sis}}_{\bar{\mathcal{R}},q,\overline{n}',m',\beta^{\mathsf{sis}}}$. ■

Lemma 12. Consider a reduction of knowledge Π^{flatten} where, on input a $\Xi^{\mathsf{staircase}}_{\mathcal{R},q,m,\underline{n},\overline{n},\overline{n}',r,\mathbf{d},n^{\mathsf{sc}},\beta}$ instance $(\mathbf{F}, \mathbf{Y})$, the verifier sends $c \leftarrow\$ \ \mathcal{C}_{\mathcal{R}_q}$ and both the prover and the verifier compress $\mathbf{F} = \begin{pmatrix} \overline{\mathbf{F}} \\ \underline{\mathbf{F}} \end{pmatrix}$ and $\mathbf{Y} = \begin{pmatrix} \overline{\mathbf{Y}} \\ \underline{\mathbf{Y}} \end{pmatrix}$ into $\widetilde{\mathbf{F}} = \begin{pmatrix} \mathbf{c}^{\mathsf{T}} \overline{\mathbf{F}} \\ \underline{\mathbf{F}} \end{pmatrix}$ and $\widetilde{\mathbf{Y}} = \begin{pmatrix} \mathbf{c}^{\mathsf{T}} \overline{\mathbf{Y}} \\ \underline{\mathbf{Y}} \end{pmatrix}$ respectively, where $\mathbf{c} := (1, c, \ldots, c^{\overline{n}-1})^{\mathsf{T}}$. The witness remains to be $\mathbf{W}$. The reduction of knowledge Π^{flatten} in is a perfectly complete from $\Xi^{\mathsf{staircase}}$ to Ξ^{lin} with parameters

$$\left(m, \underline{n}, \overline{n}, \overline{n}', r, \mathbf{d}, n^{\mathsf{sc}}, \beta\right) \mapsto \left(m, n = \underline{n} + 1, r, \mathbf{d}, \beta\right).$$

and depth-preserving knowledge-sound from $\Xi^{\mathsf{staircase}} \cup \Xi^{\mathsf{vsis}}$ to $\Xi^{\mathsf{lin}} \cup \Xi^{\mathsf{vsis}}$ with parameters

$$\begin{matrix} \left(m, \underline{n}, \overline{n}, \overline{n}', r, \mathbf{d}, n^{\mathsf{sc}}, \beta'\right) \\ \left(\mathcal{R}, q, n, \mathbf{d}, \beta^{\mathsf{vsis}}\right) \end{matrix} \hookleftarrow \begin{matrix} \left(m, n = \underline{n} + 1, r, \mathbf{d}, \beta'\right) \\ \left(\mathcal{R}, q, n, \mathbf{d}, \beta^{\mathsf{vsis}}\right) \end{matrix}$$

and knowledge error $\overline{n}/\left|\mathcal{C}_{\mathcal{R}_q}\right|$ where $2\beta' \leq \beta^{\mathsf{sis}}$. There is no prover communication cost.

Proof: We analyse the perfect completeness and depth-preserving knowledge-soundness of Π^{flatten}.

Completeness. The perfect completeness of Π^{flatten} is almost immediate, except that we need to show that the compressed relation still maintains a "row-tensor" structure. Clearly, the bottom part $\underline{\mathbf{F}}$ remains structured. For the top part $\overline{\mathbf{F}}$, we observe the flatten row

$$\overline{\mathbf{f}}^{\mathsf{T}} := \mathbf{c}^{\mathsf{T}} \overline{\mathbf{F}},$$

which we split into $d' = \prod_{i=1}^{|\mathbf{d}|} d_i$ vectors of size d, i.e.

$$\overline{\mathbf{f}}^{\mathsf{T}} = (\widehat{\mathbf{f}}_i^{\mathsf{T}})_{i \in [d']}.$$

Let $\overline{\mathbf{F}} = (\widehat{\mathbf{F}}_i)_{i \in [d']}$ and then $\widehat{\mathbf{f}}_i^{\mathsf{T}} = \mathbf{c}^{\mathsf{T}} \overline{\mathbf{F}}_i$. Due to the construction, $\widehat{\mathbf{F}}_i$ is as matrix $\widehat{\mathbf{F}}_0$ with non-zero elements moved down by $i \cdot \overline{n}'$ positions. We denote $\widetilde{\mathbf{f}} := \widehat{\mathbf{f}}_0$, $r := c^{\overline{n}'}$ and observe that $\widehat{\mathbf{f}}_i = r^i \cdot \widetilde{\mathbf{f}}$. We write:

$$\overline{\mathbf{f}} = \mathbf{r} \otimes \widetilde{\mathbf{f}},$$

where $\mathbf{r}^{\mathsf{T}} = (1, r, \ldots, r^{d'})$. Such power structure conforms to any tensor structure and we conclude that

$$\widetilde{\mathbf{F}} = \begin{pmatrix} \overline{\mathbf{f}}^{\mathsf{T}} \\ \underline{\mathbf{F}} \end{pmatrix} \in \mathcal{R}_q^{n \times \otimes \mathbf{d}}.$$

Soundness. First, we call the prover $\mathcal{P}^*$ (producing an accepting transcript with probability ϵ) in parallel to obtain two transcripts.

Consider the first transcript. Assume that $((\widetilde{\mathbf{F}}_1, \widetilde{\mathbf{Y}}_1), \widetilde{\mathbf{W}}_1) \in \Xi^{\mathsf{lin}}$. Otherwise, abort. Then if $((\mathbf{F}, \mathbf{Y}), \mathbf{W}_1) \in \Xi^{\mathsf{staircase}}$, the extraction is immediate. If not, we follow with the reasoning.

First, recall that the Schwartz-Zippel lemma asserts that for any non-zero polynomial of degree d over $\mathbb{F}$, the probability that the polynomial evaluates to zero at a randomly selected point from $S \subseteq \mathbb{F}$ is no greater than $d/|S|$. Also recall that $\mathcal{C}_{\mathcal{R}_q}$ is almost a subfield $\mathbb{F}$ of $\mathcal{R}_q$, except that it does not contain the element 0.

We translate the upper bound into a *knowledge* error and observe the following: As assumed, $((\widetilde{\mathbf{F}}_1, \widetilde{\mathbf{Y}}_1), \widetilde{\mathbf{W}}_1) \in \Xi^{\mathsf{lin}}$, but $((\mathbf{F}, \mathbf{Y}), \mathbf{W}_1) \notin \Xi^{\mathsf{staircase}}$. By the Schwartz-Zippel lemma (setting $\mathcal{C}_{\mathcal{R}_q} \equiv S$), only a small fraction of $\kappa = \overline{n}/\left|\mathcal{C}_{\mathcal{R}_q}\right|$ challenges can be accepted and result in $\mathbf{W}_1$ as a witness by the Schwartz-Zippel lemma.

In other words, with probability $\epsilon - \kappa$ over the challenge space, we have $\mathbf{W}_1 \neq \mathbf{W}_2$. Now, $\mathbf{V} = \mathbf{W}_1 - \mathbf{W}_2$ is a non-zero preimage with a non-zero column $\mathbf{v}$, such that $\underline{\mathbf{F}}\mathbf{v} = \mathbf{0}$ of norm at most $2\beta' \leq \beta^{\mathsf{vsis}}$, i.e., a witness to Ξ^{vsis}.

■

6. Succinct Argument for Sequential Relation

We compose the RoKs in Sections 4 and 5 to obtain an instantiation of Π^{vdf}_T, the succinct argument system for Ξ^{vdf} required in Section 3 by the VDF construction. Although keeping track of all the parameter mappings is tedious, it is technically straightforward and does not convey much insight about the composition. Instead, we summarise the composition result informally in Theorem 6.1 and provide intuitive explanations in subsequent subsections.

Theorem 6.1 (Informal). Assume the hardness of $\mathsf{SIS}_{\bar{\mathcal{R}},q,\overline{n}',m',\beta^{\mathsf{sis}}}$ and $\mathsf{vSIS}_{\mathcal{R},q,n,r,\mathbf{d},\beta^{\mathsf{vsis}}}$ where $m' = \overline{n}' \cdot \lfloor \log q \rfloor$ and $\mathbf{d} = (2, \ldots, 2, m')$. For any $T \in \mathbb{N}$, there exists a perfectly complete and depth-preserving knowledge-sound succinct argument system Π^{vdf}_T for the relation $\Xi^{\mathsf{vdf}}_{\bar{\mathcal{R}},q,\overline{n}',T}$. The knowledge extractor has circuit size $\lambda^{O(\mu)}$ for some $\mu \leq O(\log(T + \lambda))$.

6.1. RoKs Composition

We describe a composition of RoKs in Sections 4 and 5 to obtain an argument system Π^{vdf}_T for Ξ^{vdf}_T in Theorem 6.1. Pictorially, the composition can be summarised as

$$\begin{aligned} &\Pi^{\mathsf{pack}} \to \Pi^{\mathsf{bind}} \to \Pi^{\mathsf{staircase}} \to \Pi^{\mathsf{flatten}} \to \Pi^{\mathsf{bin\text{-}norm}} \\ &\qquad \to \left(\Pi^{b\text{-}\mathsf{decomp}}_{\mathsf{if}\ i \in \{\mu-3, \mu-1\}} \to \Pi^{\mathsf{norm}}_{\mathsf{if}\ i \geq 1} \to \Pi^{\mathsf{split}} \to \Pi^{\mathsf{fold}}\right)_{i \in [\mu]}. \end{aligned}$$

After the execution of the final Π^{fold}, the witness is simply sent in plain. Since all RoKs in the chain are perfectly complete, the perfect completeness of Π^{vdf}_T is clear. A more detailed visualisation is given in Fig. 2.

The chain of reductions achieves several objectives:

(i) Maintain a low operating modulus such that q fits within a single 64-bit register.
(ii) Minimise the verifier's runtime, even if it results in a longer prover runtime.
(iii) Keep the proof size relatively small.
(iv) Ensure the prover's runtime remains executable within a reasonable timeframe.

The initial steps (from Π^{pack} to $\Pi^{\mathsf{bin\text{-}norm}}$) of the chain aims to reduce a Ξ^{vdf} instance to a Ξ^{lin} instance. In more detail, Π^{pack} is first applied to pack the single-column witness vector into a multi-column matrix. Then, the prover would commit to the witness matrix with a vSIS-based commitment scheme, resulting in a Ξ^{bind} instance. The $\Pi^{\mathsf{bind}}, \Pi^{\mathsf{staircase}}, \Pi^{\mathsf{flatten}}$ and $\Pi^{\mathsf{bin\text{-}norm}}$ reductions are then applied back-to-backto reduce the Ξ^{bind} instance into a Ξ^{lin} instance, which is natively supported by the machinery developed in [23]. Note that the last initial step $\Pi^{\mathsf{bin\text{-}norm}}$ ensures that if the witness norm is demonstrated to be small enough then it must be binary.

The main loop, composing $\Pi^{b\text{-}\mathsf{decomp}}$, Π^{norm}, Π^{split} and Π^{fold} from [23], aims to shrink the Ξ^{lin} instance into one with witness that is small enough to be sent in plain. To recall, $\Pi^{b\text{-}\mathsf{decomp}}$ decomposes the witness with a small radix, significantly lowering the witness norm. The RoK Π^{norm} creates a "checkpoint" in the sense that if the witness norm is demonstrated to be small enough then it must actually be further bounded by a value known to the verifier. The Π^{split} and Π^{fold} RoKs shrink the dimension of the witness while increasing its norm.

Together, $\Pi^{b\text{-}\mathsf{decomp}}$ and Π^{norm} control the norm of the witness throughout the RoK executions and in knowledge extraction, hence keeping the modulus q small. It has been observed empirically that the norm does not grow substantially during folding, and thus the norm-controlling RoKs $\Pi^{b\text{-}\mathsf{decomp}}$ and Π^{norm} need not be run in every iteration of the loop. This optimisation is significant since $\Pi^{b\text{-}\mathsf{decomp}}$ has quite high communication cost and runtime costs for both the prover and the verifier.

The number of rounds μ is optimised for achieving the smallest proof size with the given composition strategy. By proof size, we refer to the overall prover-to-verifier communication including the final witness. Concretely, the number of rounds is selected so that the accumulated communication cost at the penultimate round exceeds the (folded) witness size. In our concrete parameter selection, the number of rounds turns out to be $\mu \in \{5, 6, 7\}$. Concrete parameters will be further discussed in Section 7.

6.2. Succinctness

We discuss the succinctness of Π^{vdf}_T, i.e. that the prover communication is sublinear in the statement size and the witness size. The size of the initial Ξ^{vdf}_T witness is $T \cdot \overline{n}' \cdot \lfloor \log q \rfloor \cdot \delta = T \cdot \mathsf{poly}(\lambda)$, which dominates the statement size. Each RoK has a fixed $\mathsf{poly}(\lambda)$ prover communication cost, independent of T. Since $\mu \leq O(\log(T + \lambda))$, we end up with a total prover communication cost of $\log T \cdot \mathsf{poly}(\lambda)$.

6.3. Depth-Preserving Knowledge-soundness

We next elaborate on the depth-preserving knowledge-soundness of Π^{vdf}_T. Note that all RoKs involved are either coordinate-wise special-sound, which then by Lemma 2 are also depth-preserving knowledge-sound, or directly stated as being depth-preserving knowledge-sound. As we discussed

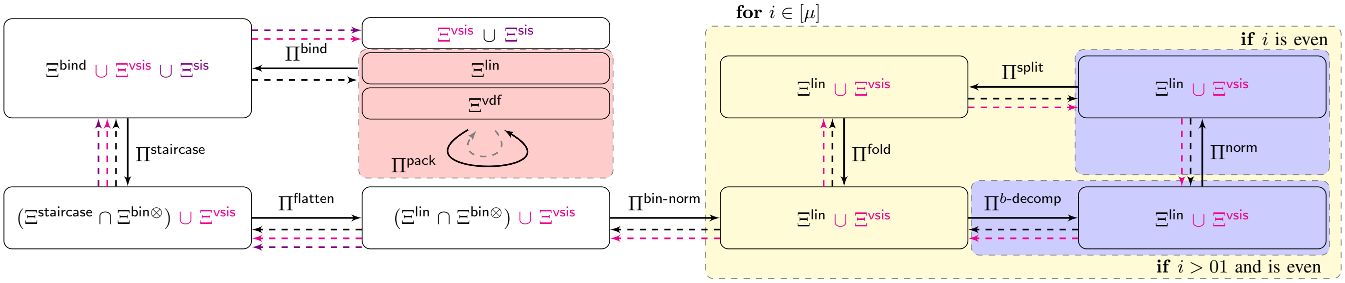

for $i \in [\mu]$
if i is even
$\Xi^{\mathsf{bind}} \cup \Xi^{\mathsf{vsis}} \cup \Xi^{\mathsf{sis}}$
$\Xi^{\mathsf{vsis}} \cup \Xi^{\mathsf{sis}}$
Π^{bind}
Ξ^{lin}
Ξ^{vdf}
Π^{pack}
$\Pi^{\mathsf{staircase}}$
$\Xi^{\mathsf{lin}} \cup \Xi^{\mathsf{vsis}}$
Π^{split}
$\Xi^{\mathsf{lin}} \cup \Xi^{\mathsf{vsis}}$
Π^{fold}
Π^{norm}
$(\Xi^{\mathsf{staircase}} \cap \Xi^{\mathsf{bin}\otimes}) \cup \Xi^{\mathsf{vsis}}$
Π^{flatten}
$(\Xi^{\mathsf{lin}} \cap \Xi^{\mathsf{bin}\otimes}) \cup \Xi^{\mathsf{vsis}}$
$\Pi^{\mathsf{bin\text{-}norm}}$
$\Xi^{\mathsf{lin}} \cup \Xi^{\mathsf{vsis}}$
$\Pi^{b\text{-}\mathsf{decomp}}$
$\Xi^{\mathsf{lin}} \cup \Xi^{\mathsf{vsis}}$
if $i > 01$ and is even

Figure 2: Composition of RoKs. (i) Black, solid lines depict the "reduction" direction in the reductions of knowledge. (ii) Dashed lines depict the "extraction" direction in the reductions of knowledge, distinct for different branches of extraction. (iii) The red box includes the initial relations. The details of the Π^{bind} reductions are summarised in Lemma 10. (iv) The yellow box encloses the reductions from [23]. (v) Blue boxes enclose steps which are optional and might be omitted in some iterations.

in Section 4.2, the soundness analyses of the latter RoKs (except for perfectly sound ones) involve non-black-box combinations of special-soundness and the Schwartz-Zippel lemma. However, their depth-preserving properties can be argued along the same lines as in the proof of Lemma 2.

We next argue that a sequential composition of depth-preserving RoKs is still depth-preserving.[5] Let $\bar{\mu} = \Theta(\mu) \leq O(\log(T + \lambda))$ denote the total number of RoKs involved in the sequential composition. Roughly, for the i-th RoK, we construct an extractor $\mathcal{E}^{(i)}$ with oracle access to $\mathcal{E}^{(i+1)}$ viewing $\mathcal{E}_{i+1}$ as a prover for $(i-1)$-th RoK. The extractor $\mathcal{E}_{\bar{\mu}-1}$ of the last RoK has oracle access to prover $\mathcal{P}^*$. Due to the depth-preserving knowledge-soundness of individual RoKs, we write

$$\begin{aligned}\mathsf{Depth}(\mathcal{E}^{(0)}) &\leq \mathsf{Depth}(\mathcal{E}^{(1)}_{\mathcal{E}^{(2)}}) + \mathsf{poly}(\lambda)\\ &\leq \dots\\ &\leq \mathsf{Depth}(\mathcal{P}^*) + \log T \cdot \mathsf{poly}(\lambda)\\ &= \mathsf{Depth}(\mathcal{P}^*) + \mathsf{poly}(\lambda)\end{aligned}$$

and deduce the depth-preserving knowledge-soundness of Π^{vdf}_T given that $\log T \leq \mathsf{poly}(\lambda)$.

In terms of total work, we note the work of $\mathcal{E}^{(i)}$ is at most $\mathsf{poly}(\lambda)$ times that of $\mathcal{E}^{(i+1)}$, and thus the total work of $\mathcal{E}^{(0)}$ is at most $\mathsf{poly}(\lambda)^{\bar{\mu}}$ times that of $\mathcal{P}^*$. In the case where $\mu > \omega(1)$, total work of $\mathcal{E}^{(0)}$ is a quasi-polynomial factor greater than the work of $\mathcal{P}^*$. This limitation is common to lattice-based succinct arguments which natively achieve negligible soundness error, such as [33, 23].

Regarding the knowledge error, note that all involved RoKs are either perfectly sound or has knowledge error inversely proportional to $|\mathcal{C}_{\mathcal{R}_q}| = q^e - 1$ or $|\mathcal{C}^{\ell}|$. Thus, picking large enough e and ℓ ensures that the individual RoKs and hence the resulting argument to have negligible knowledge error.

7. Concrete Parameter Selection

We discuss considerations which are taken when selecting parameters. We begin in Section 7.1 by explaining requirements on the modulus q for ring switching reasons. We then explain in Section 7.2 the choices of global parameters, which are common across all experiment instances, and in Section 7.3 instance-specific parameters.

7.1. Ring Switching

From a computational perspective, the Π^{norm} protocol includes a step that involves computing the "convolution" over elements $\mathbf{w} \in \mathcal{R}_q^m$. This step entails computing the coefficients of the polynomial $L_{\mathbf{w}}(X) \cdot L_{\overline{\mathbf{w}}}(X^{-1})$ for $\mathbf{w}$ representing the columns of the witness. Instead of performing the convolution over $\mathcal{R}_q$, we lift the computation to a larger ring $\tilde{\mathcal{R}}_q$, where $\tilde{\mathcal{R}} = \mathbb{Z}[\zeta_{\tilde{\mathfrak{f}}}]$, $\tilde{\mathfrak{f}} = 2^{2+\lceil \log_2 m \rceil}$ and $\tilde{\varphi} = \varphi(\tilde{\mathfrak{f}})$, and map the result back to $\mathcal{R}_q$.

For optimal efficiency, it is desirable to pick the modulus q such that the ring $\tilde{\mathcal{R}} = \mathbb{Z}[\zeta_{\tilde{\mathfrak{f}}}] \cong (\mathbb{F}_{q^{\tilde{e}}})^{\tilde{\varphi}/\tilde{e}}$ has a low residue degree, specifically $\tilde{e} = 1$. Consequently, q should be a prime and satisfy $q \equiv 1 \bmod \tilde{\mathfrak{f}}$. Concretely, we pick $\tilde{\mathfrak{f}} = 2^{\ell}$ and $q \equiv 1 \pmod{2^{\ell}}$, where q is prime and $2^{\ell} \geq 2m$. Typically, $\ell = 28$ is sufficient for most scenarios.

7.2. Global Parameters

The global parameters, detailed in Table 2, are carefully chosen to take advantage of certain mathematical properties. We highlight the choices of $\mathcal{R}$, $\bar{\mathcal{R}}$ and q as they require the most care. The remaining parameters are then natural.

We pick $\mathcal{R} = \mathbb{Z}[\zeta_{24}] = \mathbb{Z}[\zeta_3] \otimes \mathbb{Z}[\zeta_8]$ and $\bar{\mathcal{R}} = \mathbb{Z}[\zeta_3 + \zeta_3^{-1}] \otimes \mathbb{Z}[\zeta_8] = \mathbb{Z}[\zeta_8]$, which offer several benefits:

(i) Fast multiplication of ring elements leveraging the Karatsuba algorithm (without requiring CRT representation).
(ii) Simple structure of the cyclotomic polynomial,
(iii) Larger than binary subtractive set over $\mathcal{R}$, specifically $\mathcal{C} := \{1, \zeta, \zeta^2\}$,
(iv) Small norm expansion factor for ring multiplication.

[5] Note that [32, Theorem 2.1] analyses the sequential composition of RoKs but does not discuss the depth of the extractor.

	$\lambda = 128$, $\kappa = 80$			$\lambda = 96$, $\kappa = 65$			$\lambda = 64$, $\kappa = 50$			
	C.L	C.M	C.S	B.L	B.M	B.S	A.XL	A.L	A.M	A.S
$\lvert\mathbf{W}\rvert$ witness size	72.45 MB	36.24 MB	18.12 MB	81.40 MB	40.69 MB	20.35 MB	161.05 MB	80.55 MB	40.27 MB	20.13 MB
|comm| communication cost	29.79 MB	25.88 MB	18.94 MB	20.56 MB	18.39 MB	15.38 MB	15.41 MB	13.22 MB	11.36 MB	9.82 MB
$\mathcal{P}$	01h56m26s	00h51m47s	00h26m33s	01h51m23s	00h47m53s	00h23m16s	04h00m40s	01h51m40s	00h51m50s	00h22m44s
$\mathcal{V}$	6.00s	5.43s	5.10s	6.17s	6.19s	4.87s	7.19s	6.09s	5.52s	5.24s
Eval	162.40s	75.83s	38.54s	185.15s	112.95s	46.00s	364.66s	182.66s	85.84s	41.84s
Eval step	0.93ms	0.87ms	0.88ms	0.94ms	1.15ms	0.94ms	0.94ms	0.94ms	0.88ms	0.86ms
Eval rounds	175104	87552	43776	196608	98304	49152	389120	194560	97280	48640
r	57	57	57	48	48	48	38	38	38	38
$\underline{n}$	220	216	210	180	177	162	118	115	113	103
$\overline{n}'$	14	14	14	14	14	14	14	14	14	14
δ	0.00170	0.00090	0.00040	0.00190	0.00100	0.00050	0.28410[a]	0.14210[a]	0.00100	0.00050
$\log_2 \beta^{\mathsf{vsis}}$	52.90	52.40	51.60	54.80	54.30	51.80	54.40	53.60	53.10	51.30
Prover rounds	10	9	8	10	9	8	11	10	9	8

TABLE 1: Experimental results and instance-specific parameters for various λ and κ values.[6]

[a] In A.XL and A.L batches we used a different prime $q = 2^{62} + 2^{35} + 2^{34} + 2^{33} + 1$ so that $2^{33} \mid q - 1$ (cf. Section 7.1) resulting in larger δ.

Parameter	Value
$\mathcal{R}$	$\mathbb{Z}[\zeta_{24}]$
$\bar{\mathcal{R}}$	$\mathbb{Z}[\zeta_8]$
q	$2^{62} + 2^{29} + 2^{28} + 1$
$\mathfrak{f}$	24
φ	8
δ	4
$\Phi_{\otimes/\mathtt{pwf}}$, $\Phi_{\mathtt{pwf}/\otimes}$	1
$\lfloor \log_2 q \rfloor$	62
$\lvert\mathcal{C}\rvert$	3
e, where $\mathcal{R}_q \cong (\mathbb{F}_{q^e})^{\varphi/e}$	2
$\lvert\mathcal{C}_{\mathcal{R}_q}\rvert = q^e - 1$	$\approx 2^{124}$

TABLE 2: Global parameters.

The modulus q is selected to satisfy multiple criteria:

(i) Requirements for ring switching (Section 7.1) are met.
(ii) q is less than 63 bits to allow arithmetic operations within 64-bit constraints, approximately $q \approx 2^{62}$.
(iii) q is "slightly" larger than a power of two (i.e., 2^{62}) to keep the VDF correctness error δ small.
(iv) The challenge set $\mathcal{C}_{\mathcal{R}_q}$, which is a subfield of $\mathcal{R}_q$, is of large enough cardinality $\left|\mathcal{C}_{\mathcal{R}_q}\right| = q^e - 1$, where e is the multiplicative order of $q \bmod \mathfrak{f}$.

7.3. Instance-Specific Parameters

We consider 3 different security levels $(\lambda, \kappa) = (64, 50), (96, 65), (128, 80)$, denoted (A), (B) and (C) respectively. To capture different levels of sequentiality, we consider 3 choices of the number of rounds $T \approx 20\ 000,\ 40\ 000,\ 80\ 000$, denoted (S), (M) and (L) respectively. Combinations of the above yield 9 different parameter sets, which we call e.g. A.S, B.M and C.L. For security level (A), we further consider $T \approx 160\ 000$, yielding the parameter set A.XL.

It may seem unconventional to pick a security parameter as low as $\lambda = 64$. However, for most applications, timed cryptographic primitives such as VDFs only need to be "moderately secure" by design (this is the case, for example when the proof and the VDF value are delivered together shortly after time T). Attacking security level (A) requires $\lambda = 2^{64}$ CPU cycles or almost 19 days on a 10GHz machine with 1 000 cores corresponds, which already significantly exceeds the considered VDF runtime. Nonetheless, such a security parameter should be considered aggressive. More conservative choices are (B), which requires $\lambda = 2^{96}$ CPU cycles or over 23 397 years for a 1THz machine with 100 000 cores, and (C) which is the defacto standard nowadays for more majority protocols.

In a hypothetical scenario where `Papercraft` is deployed, we expect that the Fiat-Shamir transform is applied to make the $\langle \mathcal{P}, \mathcal{V} \rangle$ protocol non-interactive. There is currently no formal theorem stating the knowledge soundness of the Fiat-Shamir transform of the protocols in [23] and hence also for `Papercraft` which heavily relies on [23]. However, as discussed in [34, Section 9], we can heuristically assume that the knowledge soundness analysis in [35] applies.

As summarised in Section 6, the security of our VDF depends on the hardness of vSIS and SIS, and the sequentiality assumption SIS-Seq. As in [23], we use the heuristic that vSIS is no easier than SIS and that SIS over modules is no easier than the integer case. Furthermore, as in [17], we heuristically assume that the sequentiality assumption SIS-Seq holds as long as the underlying SIS assumption holds. Under these heuristics, we use the Lattice Estimator [36][7] to estimate concrete security levels. The remaining parameters are optimised to yield the smallest possible proof sizes (that we could find) using the RoK, Paper, SISsors Estimator

[7] Commit `848cc1e` of https://github.com/malb/lattice-estimator/.

[23][8]. The above process is partly automated using a script available in the repository[9] which includes both estimators mentioned above as submodules. All considered parameters are listed in Table 1.

8. Implementation and Evaluation

We implement and evaluate the performance of `Papercraft` with parameters choices in Section 7 measured by the runtime of $\mathcal{P}$, $\mathcal{V}$ and the proof size.

8.1. Implementation

We have implemented `Papercraft` in $\approx 7\ 000$ lines of Rust[10], supplemented with SageMath[11] scripts. The choice of Rust was guided by its efficiency, modern syntax, clean API, easy embedding of Assembly code and a rich collection of community libraries. Parallelisation is extensively utilised throughout our implementation, facilitated by the Rayon[12] library, which provides data parallelism with a high-level API, abstracting the complexities of thread management.

SageMath is employed for two main purposes:

(i) To generate Rust code for complex ring-based arithmetic and basis transitions, simplifying the implementation of intricate operations.

(ii) To compute the inverse over $\mathcal{R}_q$, which is a rare operation performed once per each round of the RoKs loop. Offloading this step to SageMath reduces the complexity of the codebase, as it is relatively simple to implement in SageMath.

The protocol is developed in an pseudo-interactive setting where it simulates the interaction between both parties is simulated. Transitioning to the non-interactive setting is straightforward by deriving verifier challengers using a hash function, following the Fiat-Shamir transformation.

Optimisations. For efficient multiplication of small cyclotomic ring elements, we utilise the Karatsuba algorithm which outperforms the theoretically faster Chinese Remainder Theorem (CRT)-based multiplication due to the low polynomial degree. However, for computing convolutions over vectors in quasi-polynomial time, we employ the CRT transformation. Specifically, the technique elaborated in Section 7.1 allows to compute multiplication of high-degree (over 2^{22}) polynomials in quasi-linear time. Detailed explanations of these techniques are well-documented in the literature [37, 28, 38].

While our implementation prioritises extensive parallelisation and enhancements in algebraic computation, we have deliberately avoided additional optimisations to maintain codebase portability across different platforms. However, we acknowledge that low-level, platform-specific optimisations, such as improving modular reduction using Montgomery multiplication or Barrett reduction techniques [39], could yield further performance gains, especially in constrained environments or specialised hardware scenarios.

Node type	Dell PowerEdge XE8640
Year	2024
Architecture	saphr avx2 h100 hopper
CPU	2x48 core Xeon Platinum 8468 2.1GHz
Virtual cores (total/used)	192/75
Memory	1024GB DDR5-4800
Infiniband	HDR
OS	CentOS 7

TABLE 3: Specifications of the node used for experiments.

Quality Assurance. Our codebase has undergone rigorous testing to ensure its reliability and correctness. Extensive unit tests cover all relevant parts of the implementation, including low-level arithmetic helpers and complete subroutine executions for small instances. In total, 75 unit tests are incorporated into the codebase.

8.2. Evaluation

The performance evaluation in all settings was conducted on a computing node with specifications provided in Table 3. We observe that this setup, utilising only a limited fraction of the machine's capacity (e.g. number of cores), facilitated consistent performance metrics. The results are summarised in Table 1, based on which Figs. 3 to 5, 6a, 6b, 7a, 7b, 8a and 8b are generated. Detailed reports with precise measurements of each runtime, including breakdowns into individual steps, are available in the repository[13]. We use the runtime of Eval as a common reference to compare the runtime of $\mathcal{P}$ and $\mathcal{V}$ and the proof size. We also report the witness size as a reference for comparing with the proof size.

For more realistic estimations of the witness and proof sizes, we report the sizes estimated by the provided script [14] instead of measuring them directly in the implementation. This is because data serialisation is not implemented in our codebase, and thus measuring sizes directly would result in significant overestimations, e.g. due to the suboptimal representation of short $\mathcal{R}$ values as large $\mathcal{R}_q$ values.

As demonstrated in Fig. 3, the runtime of $\mathcal{P}$ exhibits a quasi-linear scaling with the runtime of Eval. Similarly, as depicted in Fig. 4, the runtime of $\mathcal{V}$ scales sublinearly inandignificantly faster than that of Eval. By construction the witness size is linearly proportional to the runtime of Eval, while Fig. 5 indicates that the proof size scales sublinearly in the runtime of Eval. The runtime and size benchmarks per security level are summarised in Figs. 6a, 6b, 7a, 7b, 8a and 8b.

[8] Commit `6f45149` of https://github.com/russell-lai/rok-paper-sissors-estimator/.

[9] https://github.com/osdnk/papercraft/tree/main/estimates.ipynb

[10] https://www.rust-lang.org

[11] https://github.com/SageMath

[12] https://github.com/rayon-rs/rayon

[13] https://github.com/osdnk/papercraft/tree/main/reports

[14] https://github.com/osdnk/papercraft/tree/main/estimates.ipynb

9. Acknowledgments

The work of M.O. and R.L. was supported by the Research Council of Finland project No. 358951.

V.C. and G.M. are supported by the European Research Council through an ERC Starting Grant (Grant agreement No. 101077455, ObfusQation). G.M. is also funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA – 390781972.

The calculations presented above were performed using computer resources within the Aalto University School of Science "Science-IT" project.

References

[1] D. Boneh, J. Bonneau, B. Bünz, and B. Fisch, "Verifiable delay functions," in *CRYPTO 2018, Part I*, ser. LNCS, H. Shacham and A. Boldyreva, Eds., vol. 10991. Springer, Cham, Aug. 2018, pp. 757–788.

[2] "Vdfalliance," https://www.vdfalliance.org.

[3] B. Cohen and K. Pietrzak, "The chia network blockchain," *White Paper, Chia. net*, vol. 9, 2019.

[4] K. Choi, A. Manoj, and J. Bonneau, "Sok: Distributed randomness beacons," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 75–92.

[5] A. Kavousi, Z. Wang, and P. Jovanovic, "Sok: Public randomness," in *2024 IEEE 9th European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2024, pp. 216–234.

[6] "Fairdrop," https://ethglobal.com/showcase/fairdrop-qcayi.

[7] B. Wesolowski, "Efficient verifiable delay functions," in *EUROCRYPT 2019, Part III*, ser. LNCS, Y. Ishai and V. Rijmen, Eds., vol. 11478. Springer, Cham, May 2019, pp. 379–407.

[8] K. Pietrzak, "Simple verifiable delay functions," in *ITCS 2019*, A. Blum, Ed., vol. 124. LIPIcs, Jan. 2019, pp. 60:1–60:15.

[9] R. L. Rivest, A. Shamir, and D. A. Wagner, "Time-lock puzzles and timed-release crypto," Tech. Rep., 1996.

[10] L. De Feo, S. Masson, C. Petit, and A. Sanso, "Verifiable delay functions from supersingular isogenies and pairings," in *ASIACRYPT 2019, Part I*, ser. LNCS, S. D. Galbraith and S. Moriai, Eds., vol. 11921. Springer, Cham, Dec. 2019, pp. 248–277.

[11] C. Hoffmann, P. Hubácek, C. Kamath, and T. Krnák, "(Verifiable) delay functions from lucas sequences," in *TCC 2023, Part IV*, ser. LNCS, G. N. Rothblum and H. Wee, Eds., vol. 14372. Springer, Cham, Nov. / Dec. 2023, pp. 336–362.

[12] T. Decru, L. Maino, and A. Sanso, "Towards a quantum-resistant weak verifiable delay function," in *International Conference on Cryptology and Information Security in Latin America*. Springer, 2023, pp. 149–168.

[13] N. Döttling, S. Garg, G. Malavolta, and P. N. Vasudevan, "Tight verifiable delay functions," in *International Conference on Security and Cryptography for Networks*. Springer, 2020, pp. 65–84.

[14] J. Chavez-Saab, F. Rodríguez-Henríquez, and M. Tibouchi, "Verifiable isogeny walks: towards an isogeny-based postquantum vdf," in *International Conference on Selected Areas in Cryptography*. Springer, 2021, pp. 441–460.

[15] C. Freitag, R. Pass, and N. Sirkin, "Parallelizable delegation from LWE," in *TCC 2022, Part II*, ser. LNCS, E. Kiltz and V. Vaikuntanathan, Eds., vol. 13748. Springer, Cham, Nov. 2022, pp. 623–652.

[16] V. Cini, R. W. F. Lai, and G. Malavolta, "Lattice-based succinct arguments from vanishing polynomials - (extended abstract)," in *CRYPTO 2023, Part II*, ser. LNCS, H. Handschuh and A. Lysyanskaya, Eds., vol. 14082. Springer, Cham, Aug. 2023, pp. 72–105.

[17] R. W. F. Lai and G. Malavolta, "Lattice-based timed cryptography," in *CRYPTO 2023, Part V*, ser. LNCS, H. Handschuh and A. Lysyanskaya, Eds., vol. 14085. Springer, Cham, Aug. 2023, pp. 782–804.

[18] M. Ajtai, "Generating hard instances of lattice problems (extended abstract)," in *28th ACM STOC*. ACM Press, May 1996, pp. 99–108.

[19] C. Peikert and Y. Tang, "Cryptanalysis of lattice-based sequentiality assumptions and proofs of sequential work," in *CRYPTO 2024, Part V*, ser. LNCS, L. Reyzin and D. Stebila, Eds., vol. 14924. Springer, Cham, Aug. 2024, pp. 129–157.

[20] R. W. F. Lai and G. Malavolta, "Lattice-based timed cryptography," Cryptology ePrint Archive, Report 2024/540, 2024. [Online]. Available: https://eprint.iacr.org/2024/540

[21] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *40th ACM STOC*, R. E. Ladner and C. Dwork, Eds. ACM Press, May 2008, pp. 197–206.

[22] D. Micciancio and C. Peikert, "Trapdoors for lattices: Simpler, tighter, faster, smaller," in *EUROCRYPT 2012*, ser. LNCS, D. Pointcheval and T. Johansson, Eds., vol. 7237. Springer, Berlin, Heidelberg, Apr. 2012, pp. 700–718.

[23] M. Klooß, R. W. F. Lai, N. K. Nguyen, and M. Osadnik, "RoK, paper, SISsors toolkit for lattice-based succinct arguments - (extended abstract)," in *ASIACRYPT 2024, Part V*, ser. LNCS, K.-M. Chung and Y. Sasaki, Eds., vol. 15488. Springer, Singapore, Dec. 2024, pp. 203–235.

[24] D. Khovratovich, M. Maller, and P. R. Tiwari, "MinRoot: Candidate sequential function for ethereum VDF," Cryptology ePrint Archive, Report 2022/1626, 2022. [Online]. Available: https://eprint.iacr.org/2022/1626

[25] T. G. Tan, V. Sharma, Z. Li, P. Szalachowski, and J. Zhou, "ZKBdf: A ZKBoo-based quantum-secure verifiable delay function with prover-secret," Cryptology ePrint Archive, Report 2022/1373, 2022. [Online]. Available: https://eprint.iacr.org/2022/1373

[26] A. Kothapalli, S. Setty, and I. Tzialla, "Nova: Recursive zero-knowledge arguments from folding schemes," in *CRYPTO 2022, Part IV*, ser. LNCS, Y. Dodis and T. Shrimpton, Eds., vol. 13510. Springer, Cham, Aug. 2022, pp. 359–388.

[27] I. Giacomelli, J. Madsen, and C. Orlandi, "ZKBoo: Faster zero-knowledge for Boolean circuits," in *USENIX Security 2016*, T. Holz and S. Savage, Eds. USENIX Association, Aug. 2016, pp. 1069–1083.

[28] V. Lyubashevsky, C. Peikert, and O. Regev, "A toolkit for ring-LWE cryptography," in *EUROCRYPT 2013*, ser. LNCS, T. Johansson and P. Q. Nguyen, Eds., vol. 7881. Springer, Berlin, Heidelberg, May 2013, pp. 35–54.

[29] M. R. Albrecht and R. W. F. Lai, "Subtractive sets over cyclotomic rings - limits of Schnorr-like arguments over lattices," in *CRYPTO 2021, Part II*, ser. LNCS, T. Malkin and C. Peikert, Eds., vol. 12826. Virtual Event: Springer, Cham, Aug. 2021, pp. 519–548.

[30] G. Fenzi, H. Moghaddas, and N. K. Nguyen, "Lattice-based polynomial commitments: Towards asymptotic and concrete efficiency," *Journal of Cryptology*, vol. 37, no. 3, p. 31, Jul. 2024.

[31] V. Lyubashevsky, "Lattice signatures without trapdoors," in *EUROCRYPT 2012*, ser. LNCS, D. Pointcheval and T. Johansson, Eds., vol. 7237. Springer, Berlin, Heidelberg, Apr. 2012, pp. 738–755.

[32] A. Kothapalli, "A theory of composition for proofs of knowledge," Ph.D. dissertation, Microsoft Research, 2024.

[33] V. Cini, G. Malavolta, N. K. Nguyen, and H. Wee, "Polynomial commitments from lattices: Post-quantum security, fast verification and transparent setup," in *CRYPTO 2024, Part X*, ser. LNCS, L. Reyzin and D. Stebila, Eds., vol. 14929. Springer, Cham, Aug. 2024, pp. 207–242.

[34] M. Klooß, R. W. F. Lai, N. K. Nguyen, and M. Osadnik, "RoK, paper, SISsors – toolkit for lattice-based succinct arguments," Cryptology ePrint Archive, Paper 2024/1972, 2024. [Online]. Available: https://eprint.iacr.org/2024/1972

[35] T. Attema, S. Fehr, and M. Klooß, "Fiat-shamir transformation of multi-round interactive proofs," in *TCC 2022, Part I*, ser. LNCS, E. Kiltz and V. Vaikuntanathan, Eds., vol. 13747. Springer, Cham, Nov. 2022, pp. 113–142.

[36] M. R. Albrecht, R. Player, and S. Scott, "On the concrete hardness of Learning with Errors," *Journal of Mathematical Cryptology*, vol. 9, no. 3, pp. 169–203, October 2015.

[37] D. E. Knuth, "The art of computer programming vol. 2: Seminumerical methods," 1981.

[38] J. Johnston, "Notes on multiplying cyclotomic polynomials on a GPU," Cryptology ePrint Archive, Paper 2024/1094, 2024. [Online]. Available: https://eprint.iacr.org/2024/1094

[39] Z. Cao, R. Wei, and X. Lin, "A fast modular reduction method," Cryptology ePrint Archive, Paper 2014/040, 2014. [Online]. Available: https://eprint.iacr.org/2014/040

Appendix A. Supportive Figures

We provide the visualisations in Figs. 3 to 5, 6a, 6b, 7a, 7b, 8a and 8b of the data reported in Table 1.

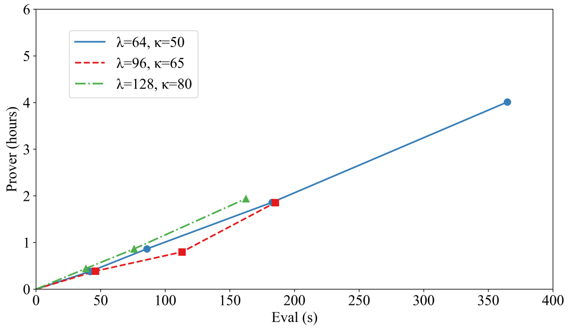


Figure 3: Prover runtime.

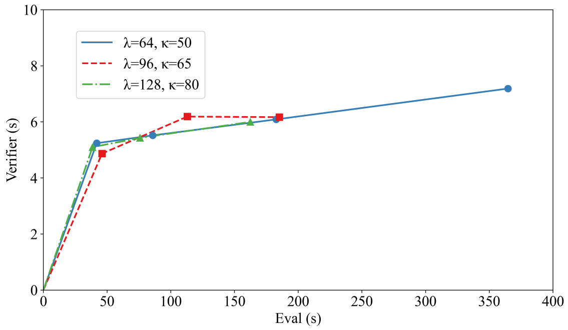


Figure 4: Verifier runtime.

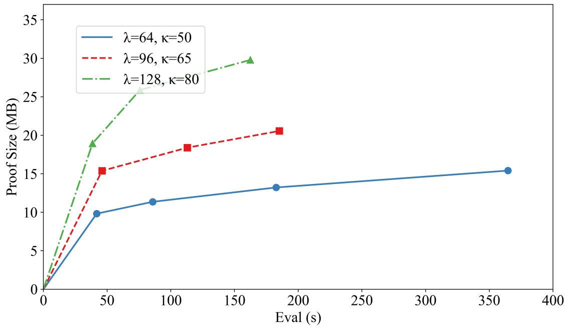
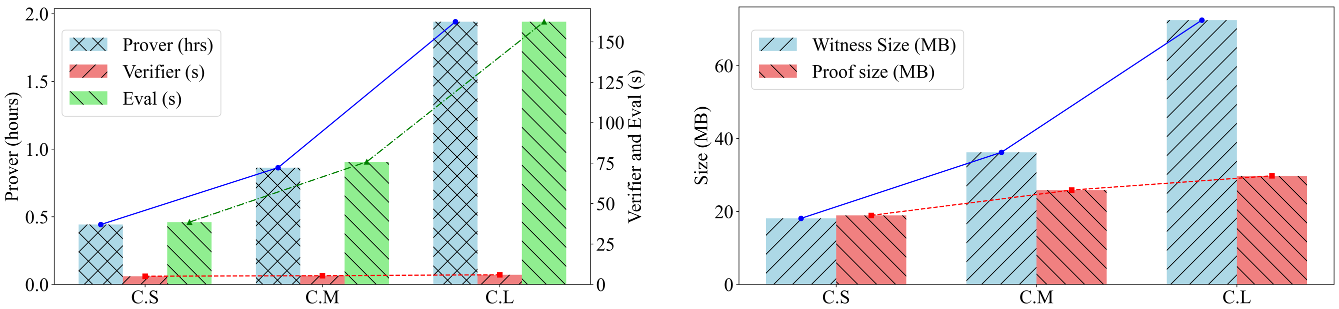


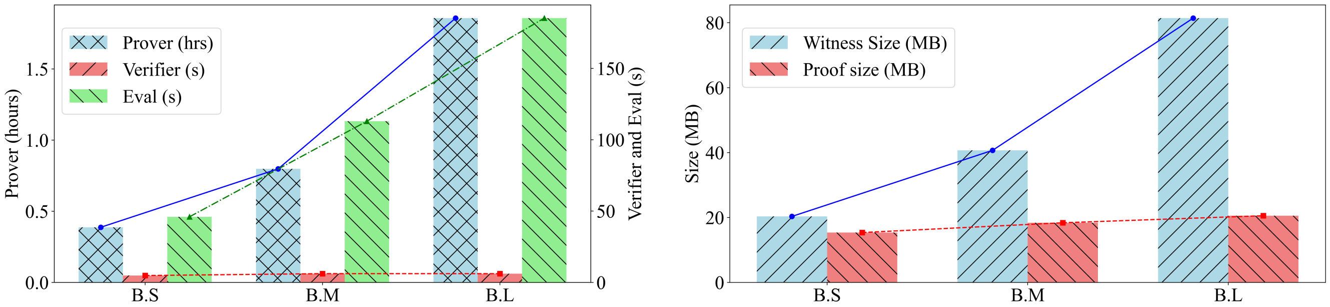
Figure 5: Proof size.



(a) Runtime.

(b) Size.

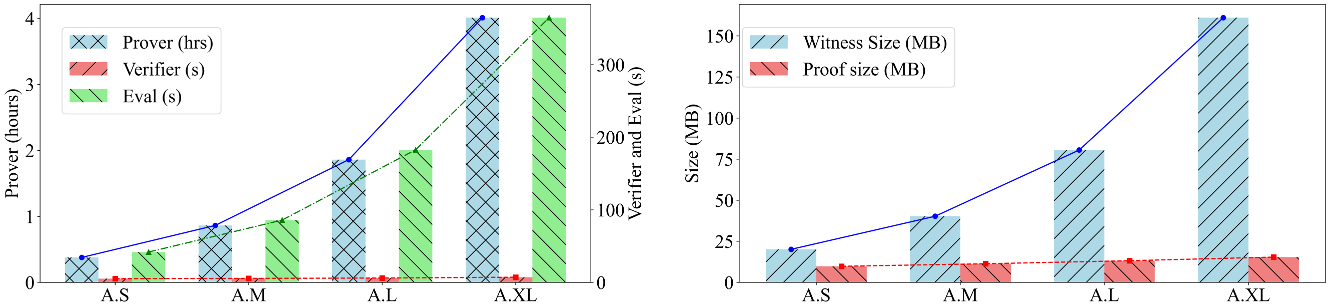
Figure 6: Comparisons for $\lambda = 128$ and $\kappa = 80$ (C).



(a) Runtime.

(b) Size.

Figure 7: Comparisons for $\lambda = 96$ and $\kappa = 65$ (B).



(a) Runtime.

(b) Size.

Figure 8: Comparisons for $\lambda = 64$ and $\kappa = 50$ (A).