Lai, Russell W.F.; Malavolta, Giulio

# Lattice-Based Timed Cryptography

*Published in:*
Advances in Cryptology – CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Proceedings

*DOI:*
10.1007/978-3-031-38554-4_25

Published: 01/01/2023

*Document Version*
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

*Please cite the original version:*
Lai, R. W. F., & Malavolta, G. (2023). Lattice-Based Timed Cryptography. In H. Handschuh, & A. Lysyanskaya (Eds.), Advances in Cryptology – CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Proceedings (pp. 782-804). (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Vol. 14085 LNCS). Springer.
https://doi.org/10.1007/978-3-031-38554-4_25

# Lattice-Based Timed Cryptography

Russell W. F. Lai[1] and Giulio Malavolta[2]

[1] Aalto University
[2] Max Planck Institute for Security and Privacy

**Abstract.** Timed cryptography studies primitives that retain their security only for a predetermined amount of time, such as proofs of sequential work and time-lock puzzles. This feature has proven to be useful in a large number of practical applications, e.g. randomness generation, sealed-bid auctions, and fair multi-party computation. However, the current state of affairs in timed cryptography is unsatisfactory: Virtually all efficient constructions rely on a single sequentiality assumption, namely that repeated squaring in unknown order groups cannot be parallelised. This is a single point of failure in the classical setting and is even false against quantum adversaries.

In this work we put forward a new sequentiality assumption, which essentially says that a repeated application of the standard lattice-based hash function cannot be parallelised. We provide concrete evidence of the validity of this assumption and, to substantiate its usefulness, we show how it enables a new proof of sequential work, with a stronger sequentiality guarantee than prior hash-based schemes.

## 1 Introduction

Timed cryptography studies a family of cryptographic primitives with diverse functionalities designed to meet their security goals only for a short (polynomial) amount of time. This includes, for example, time-lock puzzles [34], timed-commitments [11], proofs of sequential work [30], verifiable delay functions [10], and delay encryption [14]. This branch of cryptography has important theoretical implications in the context of non-malleable commitments [28] and in the average-case hardness of the class PPAD [8], which characterises the complexity of computing a Nash equilibrium. Furthermore, timed cryptography has attracted significant interest in the industry (e.g. [1]), in part due to their large number of practical applications (see [10,31] for a survey of applications).

**The Repeated Squaring Assumption.** The current state of affairs in timed cryptography is largely unsatisfactory: Virtually all efficient schemes are based on the hardness of a *single* problem (or variants thereof), namely the sequential squaring assumption. Loosely speaking, such an assumption postulates that the repeated application of the function

$$f_N(x) = x^2 \bmod N$$

where $N = pq$ is an RSA modulus, is the fastest algorithm to compute $x^{2^T} \bmod N$ given $x$. In other words, there is no better algorithm than $T$-sequential iterations of $f_N$, provided that the order of the group is unknown by the evaluator. Unfortunately, this assumption is clearly false if we allow the attacker to run in *quantum* polynomial time. At present, there is no valid alternative sequential function with conjectured post-quantum security. Besides post-quantum security, the lack of other candidates places the entirety of efficient timed cryptography on thin foundations, and only one cryptanalytic breakthrough away from being wiped out. The goal of our work is to make progress on this front, and to establish broader foundations for timed cryptographic primitives.

### 1.1 Our Contributions

The contributions of this work can be summarised as follows. A more detailed technical overview is in Section 1.3.

**A New Lattice-Based Sequential Function.** We put forward a new candidate family of sequential functions, whose design is closely connected with lattice-based cryptography. Concretely, we define our new sequential function to be the $T$-fold repeated application of the binary decomposition operation followed by the SIS-based collision-resistant hash function [2,25], with parameters set in such a way to make the domain and the range of the function coincide. In other words, our base function $f_{\mathbf{A}} : \mathbb{Z}_q^n \to \mathbb{Z}_q^n$ is defined as

$$f_{\mathbf{A}}(\mathbf{x}) := \mathbf{A} \cdot \left(-\mathbf{G}^{-1}(\mathbf{x})\right) \bmod q$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$, for $m \approx n \cdot \log q$, and $\mathbf{G}^{-1} : \mathbb{Z}_q^n \to \{0,1\}^m$ is the binary decomposition operator. Then we define $f_{\mathbf{A}}^T$ to be the $T$-fold repeated application of $f_{\mathbf{A}}$. Based on the observation that computing $\mathbf{y} = f_{\mathbf{A}}^T(\mathbf{x})$ is equivalent to establishing the satisfiability of a linear relation defined by $(\mathbf{A}, \mathbf{x}, \mathbf{y})$ by a binary vector $\mathbf{u}$, we conjecture that finding such $(\mathbf{u}, \mathbf{y})$ for random $(\mathbf{A}, \mathbf{x})$ is hard for (potentially quantum) circuits of depth less than $T$ by some super-constant function in $T$.

**Evidence of Sequentiality.** The design of our new sequential function is motivated by concrete properties that one can prove about the base function, balanced with enough algebraic structure to enable advanced cryptographic applications. More specifically, the choice of our sequential function is based on the following guiding principles:

- Recursive composition: In order to have a succinct description, the sequential function is defined as the recursive application of a *base function* with cryptographic properties. There is evidence that this is a robust design principle: If the base function is modelled as a random oracle, then one can show that sequentiality holds unconditionally [19].

– Collision resistance: The base function must be collision-resistant (and one way). This is a property that is trivially satisfied by a random oracle and something that we can prove using standard computational assumptions.
– Uniformity preserving: Similar to a random oracle, the base function must map uniform distributions to uniform distributions over the specified domains and co-domains. Once again, we are able to prove that this property holds assuming the intractability of standard problems over lattices.
– Post-quantum security: Contrary to the sequential squaring problem, we want to conjecture that the sequentiality of our function holds also against *quantum* algorithms.
– Algebraic structure: Unlike a random oracle, we want our base function to have enough algebraic structure to produce relations that are amenable to efficient proofs.

In particular, we justify our assumption by showing that $f_{\mathbf{A}}$ is collision-resistant and uniformity preserving (for some choice of parameters) based on the standard lattice assumptions, suggesting other heuristic evidence, and discussing (failed) attack strategies.

**Application: Proof of Sequential Work.** To substantiate the usefulness of our new family of sequential functions, we construct a simple and efficient proof of sequential work (PoSW), where a prover can convince a verifier that it has performed a $T$-steps sequential computation. The runtime of the verifier is logarithmic in $T$, and the protocol is statistically sound. Compared to prior hash-based constructions, our PoSW has a potentially stronger sequentiality guarantee against a dishonest prover, depending on the strength of the sequentiality assumption. More concretely, in our approach, soundness holds against any adversary running in parallel time $(1 - \omega(1)) \cdot T$, whereas prior hash-based proofs of sequential work [30,19,22] are only sound against cheating provers who run in parallel time $(1 - \alpha) \cdot T$, for any constant $0 \leq \alpha < 1$, where the verifier runtime is $\frac{1}{\log(1-\alpha)} \cdot O(\lambda)$.

**On the Necessity of New Assumptions.** We stress that we can only offer heuristic evidence for the sequentiality of our function family, and we are not able to reduce it to any "standard" computational problem. In fact, arguably the *only* "standard" computational assumption in timed cryptography is the repeated squaring assumption! Clearly, if we want to obtain a plausibly post-quantum candidate, new assumptions are necessary.

On the other hand, traditional computational assumptions in cryptography (such as LWE or DDH) do not make fine-grained distinctions on the parallelism of the attacker: The problem is assumed to be hard for all polynomial-size circuits, regardless of their depth/parallel runtime. In other words, such assumptions imply that $\mathsf{NP} \neq \mathsf{P}$ but do *not* imply that $\mathsf{NC} \neq \mathsf{P}$, which is a necessary condition for sequential functions to exist. Overall, this suggests that new assumptions may be necessary for timed cryptography, and we view our work as a promising first step towards a better understanding of this area.

### 1.2 Related Work

Besides works based on the repeated squaring assumption, there are various other approaches for constructing timed cryptographic primitives from different computational assumptions. In the following, we discuss the trade-offs when compared with our work.

**Hash-Based Schemes.** As alluded at earlier, random oracles are good candidates for constructing sequential functions, since the sequentiality of their repeated applications can be proven unconditionally. This approach has appealing properties: It offers a clean model to prove concrete statements, schemes are typically very efficient as they only involve symmetric-key operations, and one can conjecture (or even prove) post-quantum security. In fact, random oracles have been used to construct PoSW [30,19,22] with high concrete efficiency. However, when compared with our approach, the sequentiality guarantee that they offer is weaker: For any constant $0 \leq \alpha < 1$, the soundness of the scheme (parametrised by $\alpha$) is only guaranteed against cheating provers who run in parallel time $(1-\alpha) \cdot T$, where the verifier runtime is $\frac{1}{\log(1-\alpha)} \cdot O(\lambda)$. On the other hand, our approach allows us to catch any adversary running in parallel time $(1 - \omega(1)) \cdot T$, i.e. no adversary can speed up the computation by any additive factor super-constant in $T$, while the verifier runs in a fixed polynomial time.

**Isogeny-Based Schemes.** Recent works have explored constructions of timed cryptography from isogenies over elliptic curves [20,14]. This approach allows one to construct verifiable delay functions (VDF) [10] and even delay encryption [14]. However, such constructions are not post-quantum secure [20], or they rely on generic composition with succinct non-interactive arguments [15], making them impractical. Furthermore, the underlying assumptions have received substantially less scrutiny than sequential squaring.

**Generic Approaches.** Finally, we mention that one can use general-purpose cryptographic primitives to build timed cryptographic schemes. Assuming only the existence (but not knowledge) of an (iterative) sequential function, it is possible to provably construct an (iterative) sequential function from fully homomorphic encryption [26]. Incremental verifiable computation [36] can be immediately used to construct PoSWs and VDFs given a sequential function [10,21], and indistinguishability obfuscation can be used to construct time-lock puzzles [9]. While theoretically elegant, such generic constructions use heavy cryptographic machinery and result in schemes that are (concretely) prohibitively inefficient.

### 1.3 Technical Overview

In the following, we elaborate more on the results summarised in Section 1.1. For simplicity, the exposition in this technical overview is done over the set of rational integers, i.e. $\mathbb{Z}$. In the technical sections, we will be working over a ring of integers $\mathcal{R}$ of some cyclotomic field, which captures $\mathbb{Z}$ as a special case.

**Lattice-based Sequential Function/Relation.** We propose a new candidate sequential function defined as the $T$-fold repeated application of the binary decomposition operation followed by the SIS-based collision-resistant hash function [2,25], with parameters set in such a way to make the domain and the codomain of the function coincide. Concretely, (a special case of) our base function $f_{\mathbf{A}} : \mathbb{Z}_q^n \to \mathbb{Z}_q^n$ is defined as

$$f_{\mathbf{A}}(\mathbf{x}) := \mathbf{A} \cdot \left(-\mathbf{G}^{-1}(\mathbf{x})\right) \bmod q$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$, for $m \approx n \cdot \log q$, and $\mathbf{G}^{-1} : \mathbb{Z}_q^n \to \{0,1\}^m$ is the binary decomposition operator. Below, we assume for simplicity that $m = n \cdot \log q$.

At first glance, it may seem that the function $f_{\mathbf{A}}$ is not proof-friendly, since $\mathbf{G}^{-1}$ is a highly non-linear operation. However, a few simple but crucial observations allow us to express the relations induced by $f_{\mathbf{A}}$ in a proof-friendly form. Specifically, we observe that a pair $(\mathbf{x}, \mathbf{y})$ satisfies $\mathbf{y} = f_{\mathbf{A}}(\mathbf{x})$ if and only if there exists a binary vector $\mathbf{u} \in \{0,1\}^m$ such that

$$\begin{pmatrix} \mathbf{G} \\ \mathbf{A} \end{pmatrix} \cdot \mathbf{u} = \begin{pmatrix} -\mathbf{x} \\ \mathbf{y} \end{pmatrix} \bmod q$$

where $\mathbf{G}$ is the binary reconstruction gadget matrix, which in particular is a linear operator.

Generalising, suppose $\mathbf{x}_T = f_{\mathbf{A}}^T(\mathbf{x}_0)$ is the $T$-fold repeated application of $f_{\mathbf{A}}$ on $\mathbf{x}_0$. Writing $\mathbf{x}_i = f_{\mathbf{A}}(\mathbf{x}_{i-1})$ and $\mathbf{u}_i = -\mathbf{G}^{-1}(\mathbf{x}_i)$, we observe the following equivalent relation:

$$\underbrace{\begin{pmatrix} \mathbf{G} & & & & \\ \mathbf{A} & \mathbf{G} & & & \\ & \mathbf{A} & \ddots & & \\ & & \ddots & \mathbf{G} & \\ & & & \mathbf{A} & \end{pmatrix}}_{\mathbf{A}_T =:} \cdot \underbrace{\begin{pmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_{T-1} \end{pmatrix}}_{\mathbf{u} =:} = \begin{pmatrix} -\mathbf{x}_0 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{x}_T \end{pmatrix} \bmod q \quad \text{and} \quad \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_{T-1} \end{pmatrix} \in \{0,1\}^{mT}. \tag{1}$$

Looking ahead, to make the relation more proof-friendly, we relax the binary constraint, i.e. $\mathbf{u} \in \{0,1\}^{mT}$, to a bounded-norm constraint, i.e. $\|\mathbf{u}\| \leq \beta$ for some $\beta \ll q$ where $\|\cdot\|$ denotes the infinity-norm.

We conjecture and give evidence that if the short integer solution problem $\mathsf{SIS}_{n,m,q,\beta}$ is hard, then for any $T \in \mathbb{N}$ and uniformly random $(\mathbf{A}, \mathbf{x}_0)$, it is infeasible for an adversary to find $(\mathbf{u}_0, \ldots, \mathbf{u}_{T-1}, \mathbf{x}_T)$ satisfying the above (relaxed) relation in parallel time $(1 - \omega(1)) \cdot T$. Reducing checking $\mathbf{y} = f_{\mathbf{A}}(\mathbf{x})$ to checking the satisfiability of a linear relation with a bounded-norm witness is the main leverage that will enable all applications in this work.

**Proof of Sequential Work.** In the sequential relation (Eq. (1)) proposed above, enforcing $\mathbf{u} \in \{0,1\}^{mT}$ ensures that for each instance $\mathbf{x}_0$ there exists a unique witness $(\mathbf{u}_0, \ldots, \mathbf{u}_{T-1}, \mathbf{x}_T)$. To construct a verifiable delay function (VDF), it suffices to prove the satisfiability of Eq. (1) with binary $(\mathbf{u}_0, \ldots, \mathbf{u}_{T-1})$ using a

(preprocessing) succinct non-interactive argument (SNARG) with a (quasi-)linear-time prover and a sublinear-time verifier (after preprocessing). Instantiating with a post-quantum-secure SNARG, which exists unconditionally in the quantum random oracle mode [17], we can obtain a candidate post-quantum VDF.

Although we believe that the above generic approach yields a somewhat efficient VDF, especially when instantiated with a SNARG optimised for proving the sequential relation, in this work we focus on constructing a tailor-made proof of sequential work (PoSW) which explicitly takes advantage of the block-bidiagonal structure of $\mathbf{A}_T$ in Eq. (1).

The main observation which underlies our PoSW construction is the following. When $T = 2t + 1$, the matrix $\mathbf{A}_T$ can be partitioned into

$$\mathbf{A}_T = \left(\begin{array}{c|c|c} \mathbf{A}_t & & \\ & \mathbf{G} & \\ \hline & \mathbf{A} & \\ & & \mathbf{A}_t \end{array}\right).$$

This structure allows us to construct a PoSW with a $O(\log T)$-time verifier in the random oracle model following the strategy in the (VDF) construction in [32].

In more detail, we sketch an interactive variant of the PoSW construction. Since the verifier is public-coin, the non-interactive variant follows from the Fiat-Shamir transform [23,6] in the random oracle model. An instance of our PoSW is set up by sampling a random matrix $\mathbf{A} \leftarrow\!\$\, \mathbb{Z}_q^{n\times m}$, which defines $\mathbf{A}_T$ for any $T$, and a random vector $\mathbf{x}_0 \leftarrow\!\$\, \mathbb{Z}_q^n$. To convince the verifier that Eq. (1) holds for some $T \in \mathbb{N}$, the prover and the verifier engage in the following interactive protocol: We focus on the more interesting where $T = 2t + 1$ is odd[3]. The prover sends $\mathbf{u}_t$ to the verifier, reducing the linear relation in Eq. (1) to

$$\mathbf{A}_t \cdot \begin{pmatrix} \mathbf{u}_0 & \mathbf{u}_{t+1} \\ \vdots & \vdots \\ \mathbf{u}_{t-1} & \mathbf{u}_{T-1} \end{pmatrix} = \begin{pmatrix} -\mathbf{x}_0 & -\mathbf{A}\cdot\mathbf{u}_t \\ \mathbf{0} & \mathbf{0} \\ \vdots & \vdots \\ \mathbf{0} & \mathbf{0} \\ -\mathbf{G}\cdot\mathbf{u}_t & \mathbf{x}_T \end{pmatrix} \bmod q.$$

The verifier checks that $\|\mathbf{u}_t\| \le \beta$. If the check passes, the verifier sends a random challenge $r \in S \subseteq \mathbb{Z}$ chosen from challenge set $S$ to the prover. The prover and verifier then engage in the same protocol but with parameter $t$ for proving

$$\mathbf{A}_t \begin{pmatrix} \mathbf{u}_0 + \mathbf{u}_{t+1}r \\ \vdots \\ \mathbf{u}_{t-1} + \mathbf{u}_{T-1}r \end{pmatrix} = \begin{pmatrix} -(\mathbf{x}_0 + \mathbf{A}\mathbf{u}_t r) \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{x}_T r - \mathbf{G}\mathbf{u}_t \end{pmatrix} \bmod q, \quad \left\| \begin{pmatrix} \mathbf{u}_0 + \mathbf{u}_{t+1}r \\ \vdots \\ \mathbf{u}_{t-1} + \mathbf{u}_{T-1}r \end{pmatrix} \right\| \le \beta'$$

[3] If $T$ is even, the prover can reveal the last step of the computation. It then suffices for the prover to prove Eq. (1) for $T - 1$, which is odd.

for an appropriately chosen $\beta' > \beta$. After recursing for $O(\log T)$ times, the prover and the verifier arrives at a statement of size independent of $T$ for which the prover can simply send the witness to the verifier. Using standard techniques for arguing about security of (lattice-based) $\Sigma$-protocols (e.g. [12,3,4]), one could argue that (a parallel repetition [5] of) the above protocol allows to convince the verifier that the prover has knowledge of a witness satisfying Eq. (1) with certain norm bound $\beta^* > \beta$.

Note that even if we start with $\beta = 1$, the above protocol can only convince the verifier about the satisfiability of Eq. (1) with some $\beta^* > \beta$, with respect to which witnesses are not unique. This is the why our construction of PoSW does not yield a VDF, even though our construction is analogous to the VDF construction of [32].

## 2 Preliminaries

We denote by $\lambda \in \mathbb{N}$ the security parameter. A function $\mathsf{negl}(\cdot)$ is negligible if it vanishes faster than any polynomial. The cryptographic definitions in the paper follow the convention of modeling security against non-uniform adversaries. An efficient adversary $\mathcal{A}$ is modeled as a sequence of circuits $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$, such that the circuit $\mathcal{A}_\lambda$ is of polynomial size in $\lambda$. We define the *parallel* runtime of a given algorithm as the depth of the corresponding circuit, whereas the *total* runtime is determined by the size of the circuit. For a finite set $S$, we write $U(S)$ for the uniform distribution over $S$.

**Lattice Background.** Let $\mathcal{R} = \mathbb{Z}[\zeta]$ be the ring of integers of a cyclotomic field $\mathbb{Q}(\zeta)$, where $\zeta \in \mathbb{C}$ is a fixed $\ell$-th primitive root of unity for some $\ell = \mathsf{poly}(\lambda)$.

An element $x \in \mathcal{R}$ is represented by its coefficients encoding $x = \sum_{i=0}^{\varphi(\ell)-1} x_i \cdot \zeta^i$, and its (infinity) norm is $\|x\| \coloneqq \max_{i=0}^{\varphi(\ell)-1} |x_i|$. The norm extends naturally to vectors $\mathbf{u} = (u_1, \ldots, u_m) \in \mathcal{R}^m$, where $\|\mathbf{u}\| = \max_{i\in[m]} \|u_i\|$. The expansion factor of $\mathcal{R}$ is defined as $\gamma_{\mathcal{R}} \coloneqq \max_{a,b\in\mathcal{R}} \frac{\|a\cdot b\|}{\|a\|\cdot\|b\|}$. We will always assume that $\ell$ is a prime-power, and in that case it is known that $\gamma_{\mathcal{R}} \leq 2\,\varphi(\ell)$ [3]. For $q \in \mathbb{N}$, define $\mathcal{R}_q \coloneqq \mathcal{R}/q\mathcal{R}$. By a slight abuse of notation, we identify $\mathcal{R}_q$ by $\left\{\sum_{i=0}^{\varphi(\ell)-1} x_i \cdot \zeta^i : x_i \in \{-\lceil q/2\rceil + 1, \ldots, \lfloor q/2\rfloor\}\right\}$, and thus $\|x\| \leq q/2$ for any $x \in \mathcal{R}_q$. The set of units in $\mathcal{R}$ is denoted by $\mathcal{R}^\times$. A set $S \subseteq \mathcal{R}$ is said to be subtractive if $(a - b) \in \mathcal{R}^\times$ for any distinct $a, b \in S$.

We recall the following useful fact.

**Lemma 1 (Adapted from [13, Lemma 7]).** *Let $n = \mathsf{poly}(\lambda)$, $p, q \in \mathbb{N}$, $q$ prime, and $m \geq n \log_p q + \omega(\log\lambda)$. The following distributions are statistically close in $\lambda$:*

$$\left\{\begin{matrix} \mathbf{A} \leftarrow\$\, \mathcal{R}_q^{n\times m} \\ (\mathbf{A}, \mathbf{v}) : \mathbf{u} \leftarrow\$\, \mathcal{R}_p^m \\ \mathbf{v} \coloneqq \mathbf{A}\cdot\mathbf{u} \bmod q \end{matrix}\right\} \quad \textit{and} \quad \left\{(\mathbf{A}, \mathbf{v}) : \begin{matrix} \mathbf{A} \leftarrow\$\, \mathcal{R}_q^{n\times m} \\ \mathbf{v} \leftarrow\$\, \mathcal{R}_q^n \end{matrix}\right\}.$$

**Gadget Matrices.** For any $n, p, q \in \mathbb{N}$, let $\ell = \lceil \log_p q \rceil$ and $m = n \cdot \ell$. If $q < p^\ell$, write $q = \sum_{i=0}^{\ell-1} q_i \cdot p^i$ in $p$-ary expansion. If $q = p^\ell$, let $q_0 = \ldots = q_{\ell-2} = 0$ and $q_{\ell-1} = p$. Define the generalised "gadget vector" $\mathbf{g}_{p,q}$, generalised "gadget matrix" $\mathbf{G}_{p,q}$, and "parity-check matrix" $\mathbf{H}_{p,q}$ by

$$\mathbf{g}_{p,q}^\mathsf{T} \coloneqq \left(1\ p \ldots p^{\ell-1}\right), \quad \mathbf{G}_{p,q} \coloneqq \mathbf{I}_n \otimes \mathbf{g}_{p,q}^\mathsf{T}, \quad \mathbf{H}_{p,q} \coloneqq \mathbf{I}_n \otimes \begin{pmatrix} p & & & q_0 \\ -1 & p & & q_1 \\ & -1 & \ddots & \vdots \\ & & \ddots & p \\ & & & -1\ q_{\ell-1} \end{pmatrix}$$

respectively. Define the operator $\mathbf{G}_{p,q}^{-1} : \mathcal{R}_q^n \to \mathcal{R}_p^m$ which maps $\mathbf{v} = (v_i)_{i=0}^{n-1} \in \mathcal{R}_q^n$ to the concatenation of its $p$-ary representation $\left((v_{0,j})_{j=0}^{\ell-1}, \ldots, (v_{n-1,j})_{j=0}^{\ell-1}\right) \in \mathcal{R}_p^m$, i.e. $v_i = \sum_{j=0}^{\ell-1} v_{i,j} \cdot p^j$. The operator $\mathbf{G}_{p,q}^{-1}$ is naturally extended to act on any matrix $\mathbf{V}$ over $\mathcal{R}_q$ with $n$ rows, with $\mathbf{G}_{p,q} \cdot \mathbf{G}_{p,q}^{-1}(\mathbf{V}) = \mathbf{V}$. Note that $\mathbf{G}_{p,q} \cdot \mathbf{H}_{p,q} = \mathbf{0} \bmod q$. Indeed, $\mathbf{H}_{p,q}$ is a basis of the right-kernel of $\mathbf{G}_{p,q}$ over $\mathcal{K}$. When the choices of $n, p, q$ are clear from the context, we omit the subscripts and write $\mathbf{G} \coloneqq \mathbf{G}_{p,q}$.

**Computational Assumptions.** In the following we define the ring variant of the well-known short integer solution (SIS) problem [2].

**Assumption 1 (Short Integer Solution).** *The* $\mathsf{SIS}_{\mathcal{R},n,m,q,\beta}$ *assumption states that for any* $\mathbf{v} \in \mathcal{R}_q^n$ *and any PPT adversary* $\mathcal{A}$ *it holds that*

$$\Pr\left[\mathbf{A} \cdot \mathbf{u} = \mathbf{v} \bmod q \ \wedge\ \|\mathbf{u}\| \leq \beta \;\middle|\; \begin{matrix} \mathbf{A} \leftarrow\!\$\, \mathcal{R}_q^{n \times m} \\ \mathbf{u} \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{v}) \end{matrix}\right] \leq \mathsf{negl}(\lambda).$$

We also recall the learning with errors (LWE) problem [33], and in particular the version over rings [29].

**Assumption 2 (Learning with Errors).** *The (normal form of the)* $\mathsf{LWE}_{\mathcal{R},n,m,q,\chi}$ *assumption states that for any PPT adversary* $\mathcal{A}$ *it holds that*

$$\left|\Pr\left[\mathcal{A}(\mathbf{A}, \mathbf{b}) = 1 \;\middle|\; \begin{matrix} \mathbf{A} \leftarrow\!\$\, \mathcal{R}_q^{n \times m} \\ \mathbf{s} \leftarrow\!\$\, \chi^n \\ \mathbf{e} \leftarrow\!\$\, \chi^m \\ \mathbf{b}^\mathsf{T} \coloneqq \mathbf{s}^\mathsf{T} \cdot \mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q \end{matrix}\right] - \Pr\left[\mathcal{A}(\mathbf{A}, \mathbf{b}) = 1 \;\middle|\; \begin{matrix} \mathbf{A} \leftarrow\!\$\, \mathcal{R}_q^{n \times m} \\ \mathbf{b} \leftarrow\!\$\, \mathcal{R}_q^m \end{matrix}\right]\right| \leq \mathsf{negl}(\lambda).$$

For convenience, we state here a decisional variant of the SIS problem, which is known to be as hard as LWE. For completeness, we recall also a proof of this fact.

**Assumption 3 (Decisional Short Integer Solution).** *The* $\mathsf{dSIS}_{\mathcal{R},n,m,q,\chi}$ *assumption states that for any PPT adversary* $\mathcal{A}$ *it holds that*

$$\left|\Pr\left[\mathcal{A}(\mathbf{A},\mathbf{v})=1 \;\middle|\; \begin{array}{l}\mathbf{A}\leftarrow\$\,\mathcal{R}_q^{n\times m}\\ \mathbf{u}\leftarrow\$\,\chi^m\\ \mathbf{v}\coloneqq\mathbf{A}\cdot\mathbf{u} \bmod q\end{array}\right]-\Pr\left[\mathcal{A}(\mathbf{A},\mathbf{v})=1 \;\middle|\; \begin{array}{l}\mathbf{A}\leftarrow\$\,\mathcal{R}_q^{n\times m}\\ \mathbf{v}\leftarrow\$\,\mathcal{R}_q^{n}\end{array}\right]\right| \leq \mathsf{negl}(\lambda).$$

**Lemma 2.** *If* $m=n+\Omega(\lambda)$ *and the* $\mathsf{LWE}_{\mathcal{R},n,m,q,\chi}$ *assumption holds, then the* $\mathsf{dSIS}_{\mathcal{R},n,m,q,\chi}$ *assumption holds.*

*Proof.* Suppose there exists a PPT algorithm $\mathcal{A}$ which solves the $\mathsf{dSIS}_{\mathcal{R},n,m,q,\chi}$ problem. We construct a PPT algorithm $\mathcal{B}$ which solves the $\mathsf{LWE}_{\mathcal{R},n,m,q,\chi}$ problem. On input $(\bar{\mathbf{A}},\bar{\mathbf{b}})\in\mathcal{R}_q^{n\times m}\times\mathcal{R}_q^m$, $\mathcal{B}$ samples $\mathbf{A}\leftarrow\$\,\mathcal{R}_q^{n\times m}$ uniformly conditioned on $\bar{\mathbf{A}}\cdot\mathbf{A}^\intercal=\mathbf{0} \bmod q$. It then computes $\mathbf{v}\coloneqq\mathbf{A}\cdot\bar{\mathbf{b}} \bmod q$ and outputs $b\leftarrow\mathcal{A}(\mathbf{A},\mathbf{v})$.

We next analyse the distribution of $(\mathbf{A},\mathbf{v})$. First, since $\bar{\mathbf{A}}$ is uniformly random over $\mathcal{R}_q^{n\times m}$, so does $\mathbf{A}$. Furthermore, since $m=n+\Omega(\lambda)$, with overwhelming probability in $\lambda$ we have that the columns of $\mathbf{A}$ spans $\mathcal{R}_q^n$. Conditioning on this, we show that $\mathcal{B}$ is given an LWE sample if and only if $\mathcal{B}$ gives a SIS sample to $\mathcal{A}$. Observe that if $(\bar{\mathbf{A}},\bar{\mathbf{b}})$ is an LWE sample, then $\bar{\mathbf{b}}$ is of the form $\bar{\mathbf{b}}^\intercal=\mathbf{s}^\intercal\cdot\bar{\mathbf{A}}+\mathbf{e}^\intercal$ for some $\mathbf{e}\leftarrow\$\,\chi^m$. It follows that $\mathbf{v}$ is of the form $\mathbf{v}=\mathbf{A}\cdot\mathbf{e} \bmod q$. If $(\bar{\mathbf{A}},\bar{\mathbf{b}})$ is a random sample, then $\mathbf{v}=\mathbf{A}\cdot\bar{\mathbf{b}} \bmod q$ is uniformly random. □

## 3 A Lattice-Based Sequential Function/Relation

In what follows we formally define our family of sequential functions and state our conjecture regarding the sequentiality of the $T$-fold repetition of such functions.

**Our Sequential Function/Relation.** For any $\mathbf{A}\in\mathcal{R}_q^{n\times m}$, define the function $f_{\mathbf{A}}:\mathcal{R}_q^m\to\mathcal{R}_q^m$ as

$$f_{\mathbf{A}}(\mathbf{x})\coloneqq\mathbf{A}\cdot(-\mathbf{G}^{-1}(\mathbf{x})) \bmod q.$$

For $T\in\mathbb{N}$, denote by $f_{\mathbf{A}}^T$ the $T$-fold recursive evaluation of $f_{\mathbf{A}}$, i.e.

$$f_{\mathbf{A}}^T(\mathbf{x})\coloneqq\underbrace{f_{\mathbf{A}}(f_{\mathbf{A}}(\ldots(f_{\mathbf{A}}}_{T\text{ times}}(\mathbf{x})))\ldots).$$

The results in this work are based on the conjecture that, for a uniformly random $\mathbf{A}\leftarrow\$\,\mathcal{R}_q^{n\times m}$, the evaluations of the functions $f_{\mathbf{A}}^T$ take sequential time at least $\Omega(T)$. To formally state our conjecture, it is convenient to define the matrix

$$\mathbf{A}_T\coloneqq\underbrace{\begin{pmatrix}\mathbf{G}&&&&\\ \mathbf{A}&\mathbf{G}&&&\\ &\mathbf{A}&\ddots&&\\ &&\ddots&\mathbf{G}&\\ &&&\mathbf{A}&\end{pmatrix}}_{T\text{ columns}}$$

and observe that if the evaluation of $f_{\mathbf{A}}$ is split into two steps as $\mathbf{u}_{i-1} = -\mathbf{G}^{-1}(\mathbf{x}_{i-1}) \in \mathcal{R}_p^m$ and $\mathbf{x}_i = \mathbf{A} \cdot \mathbf{u}_{i-1} \bmod q$ for all $i \in [T]$ then

$$\mathbf{A}_T \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_{T-1} \end{pmatrix} = \begin{pmatrix} \mathbf{G} & & & \\ \mathbf{A} & \mathbf{G} & & \\ & \mathbf{A} & \ddots & \\ & & \ddots & \mathbf{G} \\ & & & \mathbf{A} \end{pmatrix} \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_{T-1} \end{pmatrix} = \begin{pmatrix} -\mathbf{x}_0 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{x}_T \end{pmatrix} \bmod q, \quad \left\| \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_{T-1} \end{pmatrix} \right\| \leq \beta$$

for any $\beta \geq p/2$. Furthermore, if $p$ is odd, $q = p^\ell$, and $\beta = p/2$, we observe that the preimage $\left(\mathbf{u}_0^\intercal \; \mathbf{u}_1^\intercal \; \dots \; \mathbf{u}_{T-1}^\intercal\right)^\intercal$, and hence the evaluation result $\mathbf{x}_T$, are unique.[4]

Formally, we state a family of conjectures parametrised by $n, p, q, \beta$ as follows.

**Assumption 4 ($\sigma(\lambda, T)$-SIS-Sequentiality).** *If the $\mathsf{SIS}_{\mathcal{R},n,m,q,\beta}$ assumption holds for $m = n\lceil \log_p q \rceil$, then for all polynomial-size adversary $\mathcal{A}$ it holds that*

$$\Pr\left[\begin{array}{l|l} \mathbf{A}_{T(\lambda)} \cdot \mathbf{u} = \left(-\mathbf{x}^\intercal \; \mathbf{0}^\intercal \; \dots \; \mathbf{0}^\intercal \; \mathbf{y}^\intercal\right)^\intercal \bmod q & \mathbf{A} \leftarrow\!\!\$\, \mathcal{R}_q^{n \times m} \\ \wedge \; \|\mathbf{u}\| \leq \beta & \mathbf{x} \leftarrow\!\!\$\, \mathcal{R}_q^n \\ \wedge \; \mathsf{Depth}(\mathcal{A}) < \sigma(\lambda, T) & (\mathbf{y}, \mathbf{u}) = \mathcal{A}(\mathbf{A}, \mathbf{x}) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

By the above discussion, $\left\{f_{\mathbf{A}}^T\right\}_{\mathbf{A}}$ induces a family of sequential relations. Although such a relation has potentially many solutions $(\mathbf{u}, \mathbf{y}) \in \mathcal{R}_p^{mT} \times \mathcal{R}_q^n$ to an input $\mathbf{x}$, each takes $\Omega(T)$ sequential steps to find under the SIS-sequentiality assumption.

### 3.1 Evidence of Sequentiality

To substantiate the plausibility of the SIS-sequentiality assumption, we shall offer some concrete evidence on the cryptographic properties satisfied by the function $f_{\mathbf{A}}$. First we show that the function $f_{\mathbf{A}}$ is collision resistant.

**Theorem 5 (Collision Resistance).** *If the $\mathsf{SIS}_{\mathcal{R},n,m,q,p}$ problem is hard for $m = n \cdot \lceil \log_p q \rceil$, then $f_{\mathbf{A}}$ is collision resistant.*

*Proof.* The proof is a trivial reduction from the $\mathsf{SIS}_{\mathcal{R},n,m,q,p}$ problem. Let $\mathbf{A}$ be an instance of $\mathsf{SIS}_{\mathcal{R},n,m,q,p}$. If $\mathbf{x}, \mathbf{x}' \in \mathcal{R}_q^n$ are distinct vectors such that $f_{\mathbf{A}}(\mathbf{x}) = f_{\mathbf{A}}(\mathbf{x}')$, write $\mathbf{u} = -\mathbf{G}^{-1}(\mathbf{x})$ and $\mathbf{u}' = -\mathbf{G}^{-1}(\mathbf{x}')$, we have $\mathbf{A} \cdot \mathbf{u} = \mathbf{A} \cdot \mathbf{u}' \bmod q$. In other words, we have $\mathbf{A} \cdot (\mathbf{u} - \mathbf{u}') = \mathbf{0} \bmod q$ and $\|\mathbf{u} - \mathbf{u}'\| \leq p$. □

Note that the same proof shows that $f_{\mathbf{A}}$ is one-way. Next, we show that the function $f_{\mathbf{A}}$ provably maps uniform distributions to distributions statistically or computationally close to uniform for certain parameter settings. It then follows from a standard hybrid argument that $f_{\mathbf{A}}^T$ also maps uniform distributions to near-uniform distributions for any polynomial $T$. First, we show that if $q$ is super-polynomial and is smaller than a sufficiently large power of $p$ by an additive polynomial factor, then the above claim holds statistically.

[4] For even $p$ and $q = p^\ell$, we can replace the $\left\|\left(\mathbf{u}_0^\intercal \; \mathbf{u}_1^\intercal \; \dots \; \mathbf{u}_{T-1}^\intercal\right)\right\| \leq p/2$ check with the $\left(\mathbf{u}_0^\intercal \; \mathbf{u}_1^\intercal \; \dots \; \mathbf{u}_{T-1}^\intercal\right) \in \mathcal{R}_p^{mT}$ check to guarantee uniqueness.

**Theorem 6 (Uniformity Preserving for Large $q \lessapprox p^k$).** *Let $q$ be a prime of the form $q = p^k - r$ where $k > \log_p q + 2\lambda/n$, $r = \mathsf{poly}(\lambda)$, $0 < r < p^k - p^{k-1}$, and $1/q = \mathsf{negl}(\lambda)$. The following distributions are statistically close in $\lambda$:*

$$\left\{\mathbf{y} : \mathbf{y} \leftarrow\!\$\, \mathcal{R}_q^n\right\} \approx \left\{f_{\mathbf{A}}(\mathbf{x}) : \mathbf{A} \leftarrow\!\$\, \mathcal{R}_q^{n\times m}, \mathbf{x} \leftarrow\!\$\, \mathcal{R}_q^n\right\}.$$

*Proof.* We first show that the distributions

$$\left\{\mathbf{u} \leftarrow\!\$\, \mathcal{R}_p^m\right\} \approx \left\{\mathbf{G}^{-1}(\mathbf{x}) : \mathbf{x} \leftarrow\!\$\, \mathcal{R}_q^n\right\}$$

are statistically close in $\lambda$. Let $d = \mathsf{poly}(\lambda)$ be the degree of the ring $\mathcal{R}$. The statistical distance of the two distributions is given by

$$\begin{aligned}
\Delta &\coloneqq \frac{1}{2} \cdot \left( q^{dn} \cdot \left| \frac{1}{p^{dm}} - \frac{1}{q^{dn}} \right| + (p^{dm} - q^{dn}) \cdot \frac{1}{p^{dm}} \right) \\
&= \frac{1}{2} \cdot \left( 1 - \frac{q^{dn}}{p^{dm}} + 1 - \frac{q^{dn}}{p^{dm}} \right) \\
&= 1 - \left( \frac{q^n}{p^m} \right)^d.
\end{aligned}$$

Note that $m = n \cdot \lceil \log_p q \rceil = nk$. Since $p^k > q$ and $(1+x)^n \geq 1 + nx$ for all $n \in \mathbb{N}$ and $x \geq -1$, we have

$$\begin{aligned}
\left( \frac{q^n}{p^m} \right)^d = \left( \frac{q^n}{q^{nk}} \right)^d = \left( \frac{q}{p^k} \right)^{dn} = \left( \frac{p^k - r}{p^k} \right)^{dn} &= \left( 1 - \frac{r}{p^k} \right)^{dn} > \left( 1 - \frac{r}{q} \right)^{dn} \\
&\geq 1 - \frac{rdn}{q} \geq 1 - \mathsf{negl}(\lambda).
\end{aligned}$$

In other words, we have $\Delta \leq \mathsf{negl}(\lambda)$. Since $k > \log_p q + 2\lambda/n$, we have $m = nk > n \log_p q + 2\,\lambda$. The result then follows from the leftover hash lemma (Lemma 1). □

Next, we show that if $q$ is a power of $p$ and an LWE assumption with uniform noise holds, then the claim holds computationally.

**Theorem 7 (Uniformity Preserving for $q = p^k$).** *Let $q = p^k$ for some $k \in \mathbb{N}$. If the $\mathsf{LWE}_{\mathcal{R},n,m,q,U(\mathcal{R}_p)}$ assumption holds for $m = nk$, then the following distributions are computationally indistinguishable:*

$$\left\{\mathbf{y} : \mathbf{y} \leftarrow\!\$\, \mathcal{R}_q^n\right\} \approx \left\{f_{\mathbf{A}}(\mathbf{x}) : \mathbf{A} \leftarrow\!\$\, \mathcal{R}_q^{n\times m}, \mathbf{x} \leftarrow\!\$\, \mathcal{R}_q^n\right\}.$$

*Proof.* Since $q = p^k$ is a power of $p$, $\mathbf{G}^{-1}$ is a bijection and thus the following distributions are identical:

$$\left\{\mathbf{u} \leftarrow\!\$\, \mathcal{R}_p^m\right\} \equiv \left\{\mathbf{G}^{-1}(\mathbf{x}) : \mathbf{x} \leftarrow\!\$\, \mathcal{R}_q^n\right\}.$$

The result then follows from the $\mathsf{LWE}_{\mathcal{R},n,m,q,U(\mathcal{R}_p)}$ assumption. □

We remark that the above proof would still go through for other choices of $q$, by making an LWE assumption with skewed uniform noise, i.e. $U((\mathbf{g}^\intercal)^{-1}(\mathcal{R}_q))$.

**More Heuristic Evidence.** We offer more heuristic evidence that the function is indeed sequential. First, it was shown in [26] that a fully homomorphic encryption scheme (FHE) can be used to show the existence of a *universal* sequential function, i.e. a function that is sequential if and only if sequential functions exist at all. The evaluation algorithm of this construction consists of running an empty circuit homomorphically. Looking at a specific instantiation of an FHE scheme [24], the homomorphic evaluation algorithm consists exclusively of linear operations (over some ring), interleaved with binary decomposition. This bears strong resemblance with our candidate function, which also interleaves linear operations with $p$-ary decomposition, albeit with a fixed matrix $\mathbf{A}$. In this sense, our candidate can be seen as the *minimal* non-trivial operation that is performed in the FHE evaluation, which we conjecture to be already secure.

Another evidence for the cryptographic usefulness of binary decomposition is the recent work of Chen et al. [16] which shows that the binary decomposition operator can in some cases be used as a sound alternative to the Fiat-Shamir transformation, which is normally instantiated using a random oracle.[5] Here the heuristic argument that we propose is that binary decomposition bears similarities with random oracles (in the sense that they can be both used for Fiat-Shamir) and random oracles are known to be sequential. Thus, we can conjecture that binary decomposition also bears sequentiality properties.

### 3.2 Cryptanalysis

We discuss a few attack strategies and why they do not apply to our scheme.

**Finding Associative Structure.** One simple approach to attack the sequentiality of our scheme would be to find some associative structure in the computation. For example, say we omitted the decomposition operator $\mathbf{G}^{-1}$ from the definition of $f_{\mathbf{A}}$ (adjusting the parameters suitably), then one could use the associativity of matrix multiplication to parallelise the computation, since the function

$$g^T_{\mathbf{A}}(\mathbf{x}) = \underbrace{\mathbf{A} \cdot \mathbf{A} \cdots \mathbf{A}}_{T\text{-times}} \cdot \mathbf{x} \bmod q$$

can be computed in parallel time $O(\log(T))$ by computing the matrix products in a tree fashion. However, the same attack does not appear to be viable once we interleave each multiplication with the operator $\mathbf{G}^{-1}$, since the composition of these two operations is not associative.

**Gluing Parallel Threads.** Another (related) attack strategy is to *glue together* two parallel computation transcripts. For example, suppose $T = 2t + 1$, the adversary may sample a random $\mathbf{x}^* \leftarrow\!\!\$\, \mathcal{R}_q^n$ declare it to be output of the function at time $t + 1$. Then it would spawn two parallel threads computing $\mathbf{x}_t \leftarrow f^t_{\mathbf{A}}(\mathbf{x})$

[5] Although the signature scheme presented in [16] actually relies on random oracles, it is only used to upgrade random-message unforgeability to existential unforgeability, while the use of random oracles for the Fiat-Shamir transformation is eliminated.

and $\mathbf{y} \leftarrow f^t_{\mathbf{A}}(\mathbf{x}^*)$. To obtain a consistent transcript, the adversary must now find a vector $\mathbf{u}_t \in \mathcal{R}^m$ such that:

$$\begin{pmatrix}\mathbf{G}\\ \mathbf{A}\end{pmatrix} \cdot \mathbf{u}_t = \begin{pmatrix}-\mathbf{x}_t\\ \mathbf{x}^*\end{pmatrix} \bmod q \qquad \text{and} \qquad \|\mathbf{u}_t\| \leq \beta,$$

which is not easier than solving $\mathsf{SIS}_{\mathcal{R},n,m,q,\beta}$. Note that this attack is only plausible when the solution to the sequential relation is not unique, e.g. when $\beta > p/2$.

**Preprocessing Attack.** If we were to *fix* a matrix $\mathbf{A}$, instead of sampling it as part of the instance, then it turns out that our sequentiality assumption does *not hold* unless SIS is easy for $\beta > (\gamma_{\mathcal{R}} \cdot p \cdot n \cdot \lceil \log q \rceil)/2$. The idea of the attack is to precompute a witness for $(\mathbf{A}, 2^i \cdot \mathbf{e}_j)$, for $i = 0, \ldots, \lceil \log q \rceil - 1$ and $j = 1, \ldots, n$ and where $\mathbf{e}_j \in \{0,1\}^n$ denotes the $j$-th unit vector. Let us denote the $(i,j)$-th precomputed witness by $(\mathbf{u}_0^{(i,j)}, \ldots, \mathbf{u}_{T-1}^{(i,j)}, \mathbf{y}^{(i,j)})$. Note that $\left\|\mathbf{u}_k^{(i,j)}\right\| \leq p/2$. Upon receiving $\mathbf{x} \in \mathcal{R}_q^n$, decompose $\mathbf{x}$ into

$$\mathbf{x} = \sum_{i=0}^{\lceil \log q \rceil - 1} \sum_{j=1}^{n} x^{(i,j)} \cdot 2^i \cdot \mathbf{e}_j$$

where $x^{(i,j)} \in \mathcal{R}_2$. Set the witness to

$$(\mathbf{u}_0, \ldots, \mathbf{u}_{T-1}, \mathbf{y}) := \sum_{i=0}^{\lceil \log q \rceil - 1} \sum_{j=1}^{n} x^{(i,j)} \cdot (\mathbf{u}_0^{(i,j)}, \ldots, \mathbf{u}_{T-1}^{(i,j)}, \mathbf{y}^{(i,j)}).$$

Note that $\|\mathbf{u}_k\| \leq (\gamma_{\mathcal{R}} \cdot p \cdot n \cdot \lceil \log q \rceil)/2$, and furthermore the parallel runtime of the attack (after the preprocessing phase) is $O(\log T)$. This would break a hypothetical version of our sequentiality assumption, where we fix $\mathbf{A}$ for all instances. Although we shall mention explicitly that, in the restrictive settings where we require the witness to be binary (i.e. we do not allow any slack), then the above attack does not seem to apply, even with a fixed $\mathbf{A}$.

In contrast, if we sample $\mathbf{A}$ as part of the instance (which is what we do in our actual assumption), then simple linearity attacks as above do not appear to be working. In particular, while we can decompose

$$\mathbf{A} = \sum_{i,j,k} a_{i,j,k} \cdot 2^k \cdot \mathbf{E}_{i,j}$$

where $\mathbf{E}_{i,j}$ is the matrix with the $(i,j)$-th entry being 1 and everywhere else being 0, we don't know how to decompose $\mathbf{G}$ consistently (while being able to evaluate the decomposition of $\mathbf{G}^{-1}$). Moreover, even if we can decompose the $\mathbf{G}$ matrix, the attack would need to perform a *bi-linear* combination on both the matrix and the preimage vector, which breaks the linear structure of the relation.

### 3.3 Verifiable Delay Functions

We shall remark that our sequential function can be combined with a succinct non-interactive argument with a (quasi-)linear-time prover to obtain a verifiable delay function. While this is a known implication [21], we explicitly mention this here since the statement to be proven has a particularly simple form. Specifically, for an input instance $\mathbf{x}$ and an output $\mathbf{y}$ the prover only needs to show the existence of a vector $\mathbf{u}$ such that:

$$\begin{pmatrix} \mathbf{G} & & & & \\ \mathbf{A} & \mathbf{G} & & & \\ & \mathbf{A} & \ddots & & \\ & & \ddots & \mathbf{G} & \\ & & & \mathbf{A} & \end{pmatrix} \cdot \mathbf{u} = \begin{pmatrix} -\mathbf{x} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{y} \end{pmatrix} \bmod q \qquad \text{and} \qquad \mathbf{u} \in \mathcal{R}_p^{mT}.$$

In other words, the statement to be proven consists of a highly structured linear relation and a bounded-norm or set-membership constraint. For small $p$, e.g. $p = 2$, and ring $\mathcal{R}$ of low degree, e.g. $\mathcal{R} = \mathbb{Z}$, the latter can be viewed as a simple low-degree relation $(\prod_{a \in \mathcal{R}_p}(u_i - a) = 0)_{i=1}^{mT}$ which reduces to $(u_i \cdot (u_i - 1) = 0)_{i=1}^{mT}$ for $(p, \mathcal{R}) = (2, \mathbb{Z})$. We expect that recent constructions of efficient succinct arguments for structured relations, e.g. [7,18], can efficiently prove statements of this form without too much overhead needed to manipulate the statement. We leave exploring the concrete efficiency of this approach as future work.

## 4 Proof of Sequential Work

We recall the definition of proofs of sequential work (PoSW) and present a construction based on the new SIS-sequentiality assumption.

### 4.1 Definitions

We recall the definition of a proof of sequential work (PoSW).

**Definition 1 ((Interactive) Proof of Sequential Work (PoSW)).** *An (interactive) proof of sequential work (PoSW) is a tuple of PPT algorithms/protocols* $(\mathsf{Gen}, \langle \mathsf{Eval}, \mathsf{Verify} \rangle)$ *with the following syntax:*

- $\mathsf{x} \leftarrow \mathsf{Gen}(1^\lambda)$*: The instance generation algorithm inputs the security parameter* $\lambda \in \mathbb{N}$ *and generates a problem instance* $\mathsf{x}$*.*
- $b \leftarrow \langle \mathsf{Eval}(\mathsf{x}, 1^T), \mathsf{Verify}(\mathsf{x}, T) \rangle$*: The evaluation-verification protocol is run between the the interactive evaluation and verification algorithms. Both algorithms input an instance* $\mathsf{x}$*. The evaluation algorithm further inputs a time parameter* $1^T$ *in unary while the verification algorithm inputs* $T$ *in binary. The protocol terminates when the verification algorithm returns a bit* $b \in \{0, 1\}$*.*

*A PoSW is required to satisfy the following properties:*

**Efficiency.** *For any* $\mathsf{x} \in \mathsf{Gen}(1^\lambda)$*, the circuit-depth of* $\mathsf{Eval}$ *(as a function of* $(\lambda, T)$*) satisfies*

$$\mathsf{Depth}(\mathsf{Eval}(\cdot, 1^T)) = T \cdot \mathsf{poly}(\lambda).$$

**Completeness.** *For any* $\lambda \in \mathbb{N}$*,* $\mathsf{x} \in \mathsf{Gen}(1^\lambda)$*, and* $T \in \mathbb{N}$*, it holds that*

$$\langle \mathsf{Eval}(\mathsf{x}, 1^T), \mathsf{Verify}(\mathsf{x}, T) \rangle = 1.$$

**$\sigma(\lambda, T)$-Sequential Soundness.** *For any pair of PPT adversaries* $(\mathcal{A}_0, \mathcal{A}_1)$ *it holds that*

$$\Pr\left[\begin{array}{l|l} \langle \mathcal{A}_1(\mathsf{x}), \mathsf{Verify}(\mathsf{x}, T) \rangle = 1 & \mathsf{x} \leftarrow \mathsf{Gen}(1^\lambda) \\ \wedge\ \mathsf{Depth}(\mathcal{A}_1) < \sigma(\lambda, T)) & (\mathsf{st}, T) \leftarrow \mathcal{A}_0(1^\lambda) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

*The function* $\sigma$ *is called the sequentiality of the PoSW.*

| $\mathsf{Gen}(\mathsf{pp})$ | $\langle \mathsf{Eval}(\mathsf{x}, 1^T), \mathsf{Verify}(\mathsf{x}, T) \rangle$ |
|---|---|
| $\mathbf{A} \leftarrow\!\$\ \mathcal{R}_q^{n \times m}$ | $\mathsf{Eval}:$ |
| $\mathbf{x}_0 \leftarrow\!\$\ \mathcal{R}_q^n$ | **for** $i \in \{0, \ldots, T-1\}$ **do** |
| $\mathsf{x} \coloneqq (\mathbf{A}, \mathbf{x}_0)$ | $\mathbf{u}_i := -\mathbf{G}^{-1}(\mathbf{x}_i)$ |
| **return** $\mathsf{x}$ | $\mathbf{x}_{i+1} := \mathbf{A} \cdot \mathbf{u}_i \bmod q$ |
| | **send** $\mathbf{x}_T$ |
| | $\langle \mathsf{Eval}, \mathsf{Verify} \rangle:$ |
| | **for** $j \in [\lambda / \log \lambda]$ **do** |
| | $b_j \leftarrow \langle \mathcal{P}(\mathbf{A}, (\mathbf{u}_i)_{i=0}^{T-1}, (\mathbf{x}_i)_{i=0}^{T}, p/2), \mathcal{V}(\mathbf{A}, \mathbf{x}_0, \mathbf{x}_T, p/2, T) \rangle$ |
| | $\mathsf{Verify}:$ **return** $(b_1 \wedge \ldots \wedge b_{\lambda/\log\lambda})$ |

**Fig. 1.** Construction of proof of sequential work.

### 4.2 Construction

Let $S \subseteq \mathcal{R}$ be a subtractive set where $\|r\| = 1$ for all $r \in S$. We first construct a core protocol $\langle \mathcal{P}(\mathbf{A}, (\mathbf{u}_i)_{i=0}^{T-1}, (\mathbf{x}_i)_{i=0}^{T}, \beta), \mathcal{V}(\mathbf{A}, \mathbf{x}_0, \mathbf{x}_T, \beta, T) \rangle$ recursively as follows:

- If $T = 1$, $\mathcal{P}$ sends $\mathbf{u}_0$ and $\mathcal{V}$ returns 1 if $\mathbf{A}_1 \cdot \mathbf{u}_0 = (-\mathbf{x}_0, \mathbf{x}_1) \bmod q$ and $\|\mathbf{u}_0\| \leq \beta$.[6]
- If $T > 1$ and $T$ is even:
  - $\mathcal{P}$ sends $\mathbf{u}_{T-1}$.

[6] This step serves to make the security proof a bit simpler. Alternatively, $\mathcal{P}$ does not need to send anything and $\mathcal{V}$ could simply return $(\mathbf{x}_1 \stackrel{?}{=} -\mathbf{A} \cdot \mathbf{G}^{-1}(\mathbf{x}_0))$.

- $\mathcal{V}$ checks that $(\mathbf{x}_T \stackrel{?}{=} \mathbf{A} \cdot \mathbf{u}_{T-1})$, returns 0 if not.
- $\mathcal{P}$ and $\mathcal{V}$ compute the following:
  - $\mathbf{x}_{T-1} = -\mathbf{G} \cdot \mathbf{u}_{T-1}$
- Run $\langle \mathcal{P}(\mathbf{A}, (\mathbf{u}_i)_{i=0}^{T-2}, (\mathbf{x}_i)_{i=0}^{T-1}, \beta), \mathcal{V}(\mathbf{A}, \mathbf{x}_0, \mathbf{x}_{T-1}, \beta, T-1) \rangle$.

– If $T > 1$ and $T$ is odd:
- Write $T = 2t + 1$.
- $\mathcal{P}$ sends $\mathbf{u}_t$.
- $\mathcal{V}$ checks that $\|\mathbf{u}_t\| \leq \beta$, returns 0 if not.
- $\mathcal{V}$ samples $r \leftarrow\!\!\$\, S$ and sends $r$ to $\mathcal{P}$.
- $\mathcal{P}$ and $\mathcal{V}$ compute the following:
  - $\mathbf{x}'_0 \coloneqq -\mathbf{x}_0 - \mathbf{A} \cdot \mathbf{u}_t \cdot r \bmod q$
  - $\mathbf{x}'_t \coloneqq \mathbf{x}_T \cdot r - \mathbf{G} \cdot \mathbf{u}_t \bmod q$
  - $\beta' \coloneqq 2\,\gamma_{\mathcal{R}}\,\beta$
- $\mathcal{P}$ computes $\mathbf{u}'_i \coloneqq \mathbf{u}_i + \mathbf{u}_{t+i+1} \cdot r$ for all $i \in \{0, \ldots, t-1\}$.
- Run $\langle \mathcal{P}(\mathbf{A}, (\mathbf{u}'_i)_{i=0}^{t-1}, (\mathbf{x}'_i)_{i=0}^{t}, \beta'), \mathcal{V}(\mathbf{A}, \mathbf{x}'_0, \mathbf{x}'_t, \beta', t) \rangle$.

The PoSW protocol is then specified in Fig. 1.

**Analysis.** In the following we show that the above protocol is $(2, 2, \ldots, 2)$-special sound. First, we recall the definition of special soundness and a useful fact from [3].

**Definition 2 ($(k_1, \ldots, k_\mu)$-out-of-$(N_1, \ldots, N_\mu)$-Special Soundness [5]).** *Let $k_1, \ldots, k_\mu, N_1, \ldots, N_\mu \in \mathbb{N}$. A $(2\mu+1)$-round public-coin protocol $(\mathcal{P}, \mathcal{V})$ for relation $\Phi$, where $\mathcal{V}$ samples the $i$-th challenge from a set of cardinality $N_i \geq k_i$ for $i \in [\mu]$, is $(k_1, \ldots, k_\mu)$-out-of-$(N_1, \ldots, N_\mu)$-special-sound if there exists a polynomial-time algorithm that, on input a statement $\mathsf{stmt}$ and a $(k_1, \ldots, k_\mu)$-tree of accepting transcripts, outputs a witness $\mathsf{wit}$ such that $(\mathsf{stmt}; \mathsf{wit}) \in \Phi$. We also say $(\mathcal{P}, \mathcal{V})$ is $(k_1, \ldots, k_\mu)$-special-sound.*

For the (standard) definitions of public-coin protocol and trees of accepting transcripts we refer to [5].

**Lemma 3 ([3]).** *Let $\mathcal{R} = \mathbb{Z}[\zeta_{d+1}]$ be the $(d+1)$-th cyclotomic ring where $(d+1)$ is prime. The set $S = \{\mu_1, \ldots, \mu_d\} \subseteq \mathcal{R}^\times$ where $\mu_i = (\zeta^i - 1)/(\zeta - 1)$ is subtractive, i.e. for any $r_0, r_1 \in S$ it holds that $(r_1 - r_0)^{-1} \in \mathcal{R}$. Furthermore, for any $r_0, r_1 \in S$, we have $\left\| \frac{r_0}{r_1 - r_0} \right\| \leq 1$, $\left\| \frac{r_1}{r_1 - r_0} \right\| \leq 1$, and $\left\| \frac{1}{r_1 - r_0} \right\| \leq 1$.*

**Lemma 4 (Special Soundness).** *The above folding argument is $(2, 2, \ldots, 2)$-special sound for the relation*

$$\left\{ ((\mathbf{A}, \mathbf{x}_0, \mathbf{x}_T); \mathbf{u}) : \ \mathbf{A}_T \mathbf{u} = (-\mathbf{x}_0, \mathbf{0}, \ldots, \mathbf{0}, \mathbf{x}_T) \bmod q \ \wedge \ \|\mathbf{u}\| \leq (2\,\gamma_{\mathcal{R}})^{2 \log T}\, \beta \right\}.$$

*Proof.* In this proof, we focus on the (more interesting) special case where $T = 2^{\mu+1} - 1$ for some $\mu \in \mathbb{N}$, so that $\frac{T-1}{2} = 2^\mu - 1$ is also an odd integer. It is clear that for such $T$ the above folding argument is $(2\mu+1)$-round. The general case can be dealt with analogously.

In the following, we construct an extractor $\mathcal{E}$ which extracts a witness $\mathbf{u}$ given a $(2,\ldots,2)$-tree of accepting transcripts recursively from depth $i=\mu$ to depth $i=1$. Let $T_i \coloneqq 2^{\mu-i+1}-1$ for $i \in \{0,\ldots,\mu\}$ so that $T_0 = T$ and $T_\mu = 1$. Note that $T_{i-1} = 2\,T_i + 1$ for all $i \in [\mu]$. Let $\mathsf{node}_0$ and $\mathsf{node}_1$ be siblings at depth-$i$ associated with the challenges $r_0$ and $r_1$ respectively, and let $\mathsf{node}_\epsilon$ be the parent node of $\mathsf{node}_0$ and $\mathsf{node}_1$. From the tree of accepting transcripts, $\mathcal{E}$ fetches the vectors $\mathbf{x}_0^{(\mathsf{node})}, \mathbf{x}_{T_{\mathsf{depth}(\mathsf{node})}}^{(\mathsf{node})}, \mathbf{u}_t^{(\mathsf{node})}$ associated to each node $\mathsf{node} \in \{\mathsf{node}_\epsilon, \mathsf{node}_0, \mathsf{node}_1\}$ recursively defined such that $\mathbf{x}_0^{\mathsf{root}} = \mathbf{x}_0$, $\mathbf{x}_{T_0}^{\mathsf{root}} = \mathbf{x}_T$, and

$$\begin{pmatrix} \mathbf{x}_0^{(\mathsf{node}_b)} \\ \mathbf{x}_{T_i}^{(\mathsf{node}_b)} \end{pmatrix} = \begin{pmatrix} \mathbf{x}_0^{(\mathsf{node}_\epsilon)} \\ -\mathbf{G}\cdot\mathbf{u}_{T_i}^{(\mathsf{node}_\epsilon)} \end{pmatrix} + \begin{pmatrix} \mathbf{A}\cdot\mathbf{u}_{T_i}^{(\mathsf{node}_\epsilon)} \\ \mathbf{x}_{T_{i-1}}^{(\mathsf{node}_\epsilon)} \end{pmatrix} \cdot r_b \bmod q.$$

Suppose the vector $(\mathbf{u}_0^{\mathsf{node}_b}, \ldots, \mathbf{u}_{T_i-1}^{\mathsf{node}_b})$ extracted at $\mathsf{node}_b$ for $b \in \{0,1\}$ satisfies

$$\mathbf{A}_{T_i} \cdot \begin{pmatrix} \mathbf{u}_0^{\mathsf{node}_b} \\ \vdots \\ \mathbf{u}_{T_i-1}^{\mathsf{node}_b} \end{pmatrix} = \begin{pmatrix} -\mathbf{x}_0^{(\mathsf{node}_b)} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{x}_{T_i}^{(\mathsf{node}_b)} \end{pmatrix} \bmod q \quad \text{and} \quad \left\| \begin{pmatrix} \mathbf{u}_0^{\mathsf{node}_b} \\ \vdots \\ \mathbf{u}_{T_i-1}^{\mathsf{node}_b} \end{pmatrix} \right\| \leq (2\,\gamma_{\mathcal{R}})^{2\mu-i} \cdot \beta.$$

Expanding the expressions, the L.H.S. becomes

$$\mathbf{A}_{T_i} \cdot \begin{pmatrix} \mathbf{u}_0^{\mathsf{node}_b} \\ \vdots \\ \mathbf{u}_{T_i-1}^{\mathsf{node}_b} \end{pmatrix} = \begin{pmatrix} -\mathbf{x}_0^{(\mathsf{node}_\epsilon)} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ -\mathbf{G}\cdot\mathbf{u}_{T_i}^{(\mathsf{node}_\epsilon)} \end{pmatrix} + \begin{pmatrix} -\mathbf{A}\cdot\mathbf{u}_{T_i}^{(\mathsf{node}_\epsilon)} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{x}_T^{(\mathsf{node}_\epsilon)} \end{pmatrix} \cdot r_b \bmod q.$$

Let

$$\begin{pmatrix} \mathbf{u}_0 & \mathbf{u}_{t+1} \\ \vdots & \vdots \\ \mathbf{u}_{T_i-1} & \mathbf{u}_{T_{i-1}-1} \end{pmatrix} = \begin{pmatrix} \mathbf{u}_0^{\mathsf{node}_0} & \mathbf{u}_0^{\mathsf{node}_1} \\ \vdots & \vdots \\ \mathbf{u}_{T_i-1}^{\mathsf{node}_0} & \mathbf{u}_{T_i-1}^{\mathsf{node}_1} \end{pmatrix} \cdot \begin{pmatrix} \frac{r_1}{r_1-r_0} & \frac{-1}{r_1-r_0} \\ \frac{-r_0}{r_1-r_0} & \frac{1}{r_1-r_0} \end{pmatrix}.$$

It follows that

$$\mathbf{A}_{T_i} \cdot \begin{pmatrix} \mathbf{u}_0 & \mathbf{u}_{t+1} \\ \vdots & \vdots \\ \mathbf{u}_{T_i-1} & \mathbf{u}_{T_{i-1}-1} \end{pmatrix} = \begin{pmatrix} -\mathbf{x}_0^{(\mathsf{node}_\epsilon)} & -\mathbf{A}\cdot\mathbf{u}_t^{(\mathsf{node}_\epsilon)} \\ \mathbf{0} & \mathbf{0} \\ \vdots & \vdots \\ \mathbf{0} & \mathbf{0} \\ -\mathbf{G}\cdot\mathbf{u}_{T_i}^{(\mathsf{node}_\epsilon)} & \mathbf{x}_{T_{i-1}}^{(\mathsf{node}_\epsilon)} \end{pmatrix} \bmod q,$$

or equivalently

$$\mathbf{A}_{T_{i-1}} \cdot \begin{pmatrix} \mathbf{u}_0 \\ \vdots \\ \mathbf{u}_{T_{i-1}-1} \end{pmatrix} = \begin{pmatrix} -\mathbf{x}_0^{(\mathsf{node}_\epsilon)} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{x}_{T_{i-1}}^{(\mathsf{node}_\epsilon)} \end{pmatrix} \bmod q,$$

and $\|(\mathbf{u}_0, \ldots, \mathbf{u}_{T-1})\| \leq (2\,\gamma_{\mathcal{R}})^{2\mu-i+1} \cdot \beta$, where the inequality is due to Lemma 3.

By recursion, $\mathcal{E}$ extracts at the root node a vector $(\mathbf{u}_0, \ldots, \mathbf{u}_{T-1})$ satisfying

$$\mathbf{A}_T \begin{pmatrix} \mathbf{u}_0 \\ \vdots \\ \mathbf{u}_{T-1} \end{pmatrix} = \begin{pmatrix} -\mathbf{x}_0 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{x}_T \end{pmatrix} \bmod q, \quad \left\| \begin{pmatrix} \mathbf{u}_0 \\ \vdots \\ \mathbf{u}_{T-1} \end{pmatrix} \right\| \leq (2\,\gamma_{\mathcal{R}})^{2\mu}\beta < (2\,\gamma_{\mathcal{R}})^{2\log T}\beta$$

as desired. □

Finally, we are ready to show that the construction is sound, which follows by invoking the extractor of the above protocol, which returns a valid computation transcript. Since the extractor runs in time sublinear in $T$, this contradicts the sequentiality of our function.

**Theorem 8 (Soundness).** *There exists $p(\lambda) \in \mathsf{poly}(\lambda)$ such that if $\mathsf{SIS}_{\mathcal{R},n,m,q,\beta}$ and the $\sigma'(\lambda, T)$-SIS-sequentiality assumption hold, then the PoSW constructed in Fig. 1 is $\sigma(\lambda, T)$-sequentially sound, where $\sigma'(\lambda, T) = \sigma(\lambda, T) \cdot p(\lambda)$.*

*Proof (Sketch).* The theorem follows from Lemma 4 and standard techniques. We provide a proof sketch. Since the above folding argument is $(2, 2, \ldots, 2)$-special-sound with a challenge set size of $\Omega(\lambda)$, it follows from [5] that the $(\lambda/\log\lambda)$-fold parallel repetition of it is knowledge-sound with negligible knowledge error. Furthermore, we observe that the extractor constructed in [5] is depth-preserving, i.e. there exists a polynomial $p(\lambda) \in \mathsf{poly}(\lambda)$ such that the knowledge extractor $\mathcal{E}_{\mathcal{A}}$ has depth $\mathsf{Depth}(\mathcal{E}_{\mathcal{A}}) \leq p(\lambda) \cdot \mathsf{Depth}(\mathcal{A})$. Suppose there exists a polynomial-size adversary $\mathcal{A}$ which breaks the $\sigma(\lambda, T)$-sequentially-soundness of the PoSW, then the above shows that $\mathcal{E}_{\mathcal{A}}$ is a polynomial-size adversary which breaks the $\sigma'(\lambda, T)$-SIS-sequentiality assumption. □

In Appendix A, we discuss challenges of formally proving the security of our PoSW against quantum adversaries.

## Acknowledgments

The authors wish to thank Andrej Bogdanov and Alon Rosen for insightful discussions and for comments on an earlier draft of this work. The authors are

also grateful to the anonymous reviewers for suggesting the preprocessing attack described in Section 3.2.

G.M. is partially funded by the German Federal Ministry of Education and Research (BMBF) in the course of the 6GEM research hub under grant number 16KISK038 and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA – 390781972.

## References

1. Vdf alliance (2019), https://www.vdfalliance.org, accessed in June 2023.
2. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: 28th ACM STOC. pp. 99–108. ACM Press (May 1996).
3. Albrecht, M.R., Lai, R.W.F.: Subtractive sets over cyclotomic rings - limits of Schnorr-like arguments over lattices. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021, Part II. LNCS, vol. 12826, pp. 519–548. Springer, Heidelberg, Virtual Event (Aug 2021).
4. Attema, T., Cramer, R., Kohl, L.: A compressed $\Sigma$-protocol theory for lattices. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021, Part II. LNCS, vol. 12826, pp. 549–579. Springer, Heidelberg, Virtual Event (Aug 2021).
5. Attema, T., Fehr, S.: Parallel repetition of $(k_1, \ldots, k_\mu)$-special-sound multi-round interactive proofs. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022, Part I. LNCS, vol. 13507, pp. 415–443. Springer, Heidelberg (Aug 2022).
6. Attema, T., Fehr, S., Klooß, M.: Fiat-shamir transformation of multi-round interactive proofs. In: Kiltz, E., Vaikuntanathan, V. (eds.) TCC 2022, Part I. LNCS, vol. 13747, pp. 113–142. Springer, Heidelberg (Nov 2022).
7. Ben-Sasson, E., Chiesa, A., Goldberg, L., Gur, T., Riabzev, M., Spooner, N.: Linear-size constant-query iops for delegating computation. In: Hofheinz, D., Rosen, A. (eds.) Theory of Cryptography (TCC) (2019)
8. Bitansky, N., Choudhuri, A.R., Holmgren, J., Kamath, C., Lombardi, A., Paneth, O., Rothblum, R.D.: PPAD is as hard as LWE and iterated squaring. In: Kiltz, E., Vaikuntanathan, V. (eds.) TCC 2022, Part II. LNCS, vol. 13748, pp. 593–622. Springer, Heidelberg (Nov 2022).
9. Bitansky, N., Goldwasser, S., Jain, A., Paneth, O., Vaikuntanathan, V., Waters, B.: Time-lock puzzles from randomized encodings. In: Sudan, M. (ed.) ITCS 2016. pp. 345–356. ACM (Jan 2016).
10. Boneh, D., Bonneau, J., Bünz, B., Fisch, B.: Verifiable delay functions. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part I. LNCS, vol. 10991, pp. 757–788. Springer, Heidelberg (Aug 2018).
11. Boneh, D., Naor, M.: Timed commitments. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 236–254. Springer, Heidelberg (Aug 2000).
12. Bootle, J., Lyubashevsky, V., Nguyen, N.K., Seiler, G.: A non-PCP approach to succinct quantum-safe zero-knowledge. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part II. LNCS, vol. 12171, pp. 441–469. Springer, Heidelberg (Aug 2020).
13. Boudgoust, K., Jeudy, C., Roux-Langlois, A., Wen, W.: Towards classical hardness of module-LWE: The linear rank case. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part II. LNCS, vol. 12492, pp. 289–317. Springer, Heidelberg (Dec 2020).

14. Burdges, J., De Feo, L.: Delay encryption. In: Canteaut, A., Standaert, F.X. (eds.) EUROCRYPT 2021, Part I. LNCS, vol. 12696, pp. 302–326. Springer, Heidelberg (Oct 2021).
15. Chávez-Saab, J., Rodríguez-Henríquez, F., Tibouchi, M.: Verifiable isogeny walks: Towards an isogeny-based postquantum VDF. In: AlTawy, R., Hülsing, A. (eds.) SAC 2021. LNCS, vol. 13203, pp. 441–460. Springer, Heidelberg (Sep / Oct 2022).
16. Chen, Y., Lombardi, A., Ma, F., Quach, W.: Does fiat-shamir require a cryptographic hash function? In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021, Part IV. LNCS, vol. 12828, pp. 334–363. Springer, Heidelberg, Virtual Event (Aug 2021).
17. Chiesa, A., Manohar, P., Spooner, N.: Succinct arguments in the quantum random oracle model. In: Hofheinz, D., Rosen, A. (eds.) TCC 2019, Part II. LNCS, vol. 11892, pp. 1–29. Springer, Heidelberg (Dec 2019).
18. Cini, V., Lai, R.W.F., Malavolta, G.: Lattice-based succinct arguments from vanishing polynomials. In: CRYPTO 2023 (2023)
19. Cohen, B., Pietrzak, K.: Simple proofs of sequential work. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part II. LNCS, vol. 10821, pp. 451–467. Springer, Heidelberg (Apr / May 2018).
20. De Feo, L., Masson, S., Petit, C., Sanso, A.: Verifiable delay functions from supersingular isogenies and pairings. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part I. LNCS, vol. 11921, pp. 248–277. Springer, Heidelberg (Dec 2019).
21. Döttling, N., Garg, S., Malavolta, G., Vasudevan, P.N.: Tight verifiable delay functions. In: Galdi, C., Kolesnikov, V. (eds.) SCN 20. LNCS, vol. 12238, pp. 65–84. Springer, Heidelberg (Sep 2020).
22. Döttling, N., Lai, R.W.F., Malavolta, G.: Incremental proofs of sequential work. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part II. LNCS, vol. 11477, pp. 292–323. Springer, Heidelberg (May 2019).
23. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO'86. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (Aug 1987).
24. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 75–92. Springer, Heidelberg (Aug 2013).
25. Goldreich, O., Goldwasser, S., Halevi, S.: Collision-free hashing from lattice problems. Cryptology ePrint Archive, Report 1996/009 (1996), https://eprint.iacr.org/1996/009
26. Jaques, S., Montgomery, H., Rosie, R., Roy, A.: Time-release cryptography from minimal circuit assumptions. In: Adhikari, A., Küsters, R., Preneel, B. (eds.) INDOCRYPT 2021. vol. 13143, pp. 584–606. Springer (2021)
27. Lai, R.W.F., Malavolta, G., Spooner, N.: Quantum rewinding for many-round protocols. In: Kiltz, E., Vaikuntanathan, V. (eds.) TCC 2022, Part I. LNCS, vol. 13747, pp. 80–109. Springer, Heidelberg (Nov 2022).
28. Lin, H., Pass, R., Soni, P.: Two-round and non-interactive concurrent non-malleable commitments from time-lock puzzles. In: Umans, C. (ed.) 58th FOCS. pp. 576–587. IEEE Computer Society Press (Oct 2017).
29. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 1–23. Springer, Heidelberg (May / Jun 2010).
30. Mahmoody, M., Moran, T., Vadhan, S.P.: Publicly verifiable proofs of sequential work. In: Kleinberg, R.D. (ed.) ITCS 2013. pp. 373–388. ACM (Jan 2013).

31. Malavolta, G., Thyagarajan, S.A.K.: Homomorphic time-lock puzzles and applications. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part I. LNCS, vol. 11692, pp. 620–649. Springer, Heidelberg (Aug 2019).
32. Pietrzak, K.: Simple verifiable delay functions. In: Blum, A. (ed.) ITCS 2019. vol. 124, pp. 60:1–60:15. LIPIcs (Jan 2019).
33. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) 37th ACM STOC. pp. 84–93. ACM Press (May 2005).
34. Rivest, R.L., Shamir, A., Wagner, D.A.: Time-lock puzzles and timed-release crypto. Tech. rep. (1996)
35. Unruh, D.: Computationally binding quantum commitments. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, Part II. LNCS, vol. 9666, pp. 497–527. Springer, Heidelberg (May 2016).
36. Valiant, P.: Incrementally verifiable computation or proofs of knowledge imply time/space efficiency. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 1–18. Springer, Heidelberg (Mar 2008).

## A On Post-Quantum Security of our PoSW

Formally showing that our PoSW is secure against quantum adversaries requires more work than what is presented in Section 4.2. A recent work [27] shows that protocols that satisfy special soundness and a particular notion of binding for the hash function (called *collapsing*) can be proven secure against quantum adversary (when sequentially repeated). Unfortunately, their result is not sufficient for our purposes, since the depth of the extractor scales with the size of the extraction tree, which in particular means that it is polynomial in $T$. Thus, we cannot hope to use this extractor to derive a contradiction against the sequentiality of our function. We leave proving a precise statement in the quantum settings as a fascinating open question. As a first step towards this, in the following we show that the hash function used at each round of our protocol is collapsing. Here we assume familiarity with the basics of quantum information and we refer the reader to [27] for precise definitions of the notions used here. First we recall below the notion of collapsing for hash functions [35].

**Definition 3 (Collapsing).** *Let* $\mathsf{H}$ *be a (keyed) hash function. We say that* $\mathsf{H}$ *is collapsing if for any efficient (quantum) adversary* $\mathcal{A}$

$$\left|\Pr\left[\mathsf{Collapsing}^0_{\mathcal{A}}(1^\lambda)=1\right]-\Pr\left[\mathsf{Collapsing}^1_{\mathcal{A}}(1^\lambda)=1\right]\right|\leq \mathsf{negl}(\lambda),$$

*where the experiment* $\mathsf{Collapsing}^b_{\mathcal{A}}$ *is defined as follows:*

$\underline{\mathsf{Collapsing}^b_{\mathcal{A}}(1^\lambda)}$*:*

- *Sample a key* $k$ *and send it over to* $\mathcal{A}$*.*
- $\mathcal{A}$ *replies with a classical bitstring* $y$ *and a quantum state on a register* $\mathcal{X}$*.*
- *Let* $U_{k,y}$ *be the unitary that acts on* $\mathcal{X}$ *and a fresh ancilla, and CNOTs into the fresh ancilla the bit that determines whether the output of* $\mathsf{H}_k(\cdot)$ *equals* $y$ *and the input belongs to the appropriate domain. Apply* $U_{k,y}$*, measure the ancilla, and apply* $U^\dagger_{k,y}$*.*
- *If the output of the measurement is* $0$*, then abort the experiment. Else proceed.*
- *If* $b=0$ *do nothing.*
- *If* $b=1$ *measure the register* $\mathcal{X}$ *in the computational basis, discard the result.*
- *Return to* $\mathcal{A}$ *all registers and output whichever bit* $\mathcal{A}$ *outputs.*

In [27] it is shown that the SIS-based hash function is collapsing, assuming the hardness of the LWE problem.

**Lemma 5 ([27]).** *If the LWE problem is hard, then the function* $\mathsf{H}_{\mathbf{A}}$ *defined as*

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

*where* $\mathbf{A}\leftarrow\!\!\$\,\mathcal{R}_q^{n\times m}$*, is collapsing.*

We are now ready to prove that the hash function used in our folding argument is collapsing.

**Lemma 6 (Collapsing).** *Let $t$ be a polynomial in the security parameter. If the LWE problem is hard, then the function $\mathsf{H}_{\mathbf{A}}$ defined as*

$$\mathsf{H}_{\mathbf{A}}(\mathbf{u}) = \underbrace{\begin{pmatrix} \mathbf{G} & & & \\ \mathbf{A} & \mathbf{G} & & \\ & \mathbf{A} & \ddots & \\ & & \ddots & \mathbf{G} \\ & & & \mathbf{A} \end{pmatrix}}_{t \text{ columns}} \cdot \mathbf{u} \bmod q$$

*where $\mathbf{A} \leftarrow\!\!\$\, \mathcal{R}_q^{n \times m}$, is collapsing.*

*Proof.* The proof follows by a standard hybrid argument. Let us split the input $\mathbf{u} \in \mathcal{R}_p^{tm}$ in $t$ blocks $(\mathbf{u}_1, \ldots, \mathbf{u}_t)$ where $\mathbf{u}_i \in \mathcal{R}_p^m$ and let $\mathcal{X}_1 \otimes \cdots \otimes \mathcal{X}_t$ be the corresponding registers. In the last hybrid, the challenger does not measure any of the registers (this corresponds to $\mathsf{Collapsing}^1_{\mathcal{A}}$). The $i$-th hybrid is defined as the previous one, except that the challenger only measures registers $\mathcal{X}_1 \otimes \cdots \otimes \mathcal{X}_i$ in the computational basis. Note that the 0-th hybrid corresponds to the experiment $\mathsf{Collapsing}^0_{\mathcal{A}}$. What is left to be shown is that consequent hybrids are computationally indistinguishable.

For hybrids $t$ and $t-1$, indistinguishability follows directly from the collapsing property of $\mathbf{A}$ (Lemma 5). For other hybrids, it suffices to observe that the $i$-th block of the output $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_t)$ is computed as:

$$\begin{aligned} \mathbf{y}_i &= \mathbf{A} \cdot \mathbf{u}_i + \mathbf{G} \cdot \mathbf{u}_{i+1} \bmod q \\ \mathbf{y}_i - \mathbf{G} \cdot \mathbf{u}_{i+1} &= \mathbf{A} \cdot \mathbf{u}_i \bmod q \end{aligned}$$

and therefore indistinguishability follows one again by Lemma 5, since $\mathbf{u}_{i+1}$ can be computed as the result of the measurement on register $\mathcal{X}_{i+1}$ (which we assume it is measured). □