# Lattice-Based Timed Cryptography 

Russell W. F. Lai ${ }^{1}$ and Giulio Malavolta ${ }^{2,3}$<br>${ }^{1}$ Aalto University<br>${ }^{2}$ Bocconi University<br>${ }^{3}$ Max Planck Institute for Security and Privacy

April 7, 2024


#### 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 perform some initial cryptanalysis. We also propose a new template to construct proofs of sequential work, based on lattice techniques.


## 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 [RSW96], timed-commitments [BN00], proofs of sequential work [MMV13], verifiable delay functions [BBBF18], and delay encryption [BD21]. This branch of cryptography has important theoretical implications in the context of non-malleable commitments [LPS17] and in the average-case hardness of the class PPAD $\left[\mathrm{BCH}^{+} 22\right]$, which characterises the complexity of computing a Nash equilibrium. Furthermore, timed cryptography has attracted significant interest in the industry (e.g. [vdf19]), in part due to their large number of practical applications (see [BBBF18, MT19] 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 $N=p q$ 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 3.1.

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 [Ajt96, GGH96], with parameters set in such a way to make the domain and the range of the function coincide. In other words, our base function $f_{\mathbf{A}}: \mathbb{Z}_{q}^{n} \rightarrow \mathbb{Z}_{q}^{n}$ is defined as

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

where $\mathbf{A} \leftarrow \mathbb{Z}_{q}^{n \times m}$, for $m \approx n \log q$, and $\mathbf{G}^{-1}: \mathbb{Z}_{q}^{n} \rightarrow \mathbb{Z}_{2}^{m}$ 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 $(\mathbf{y}, \mathbf{u})$ for $\operatorname{random}(\mathbf{A}, \mathbf{x})$ 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 [CP18].
- 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. As an additional contribution, we construct a simple and efficient proof of sequential work (PoSW), where a prover aims to convince a verifier that it has performed a $T$-steps sequential computation. Crucially, the verifier runtime is only logarithmic in $T$. Our PoSW is based on a variant of the sequential function family introduced above, with an important modification:

- (Relaxed Norm Constraints) Instead of checking that each step of the computation is a binary string, our PoSW will only ensure that is norm is small. This is done for efficiency reasons, since it is much easier to prove the latter property using lattice techniques.

However, subsequent to the publication of this work, attacks against the assumption were discovered, which render the security proof vacuous and even cast doubts on the heuristic security of our PoSW. We discuss in more details the attacks and the current state of affairs of lattice-based sequential functions in Section 2 and Appendix A.

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 NP $\neq P$ but do not imply that NC $\neq P$, 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.

## 2 History of This Work

The idea of using

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

or similar as a candidate sequential function was, to the best of our knowledge, first conceived independently by Bogdanov and Rosen [BR23] but not published. We [LM23] later independently engineered the same idea, made the first attempt of formalising the sequentiality assumption and applying it to build a proof of sequential work (PoSW) protocol. We based the security of the PoSW protocol on a bounded-norm variant of the SIS-sequentiality assumption, which postulates
that no parallel algorithm can find a short solution (as opposed to a binary solution required by the exact variant) to the system of equations induced by the $T$-folded repetition of the function $f_{\mathbf{A}}(\mathbf{x})=-\mathbf{A} \mathbf{G}^{-1}(\mathbf{x}) \bmod q$ in time $o(T) \cdot \operatorname{poly}(\lambda)$.

Subsequently to the publication of this work, Attema and Ducas [AD23] and Peikert and Tang [PT23] independently discovered attacks against the bounded-norm variant (but not the exact variant) of this assumption, as well as the PoSW protocol. Below and in Appendix A, we discuss these attacks in more detail.

The Attema-Ducas Attack. Attema and Ducas [AD23] discovered an attack against the SISsequentiality assumption, which runs in depth $\sqrt{T} \log T \cdot \operatorname{poly}(\lambda)$ and can find solutions of norm at most $m$. Their attack exploits the sparsity of the image vector of the sequential-SIS relation.

The Peikert-Tang Attack. Independently and more generally, Peikert and Tang [PT23] discovered a family of attacks against the SIS-sequentiality assumption, which offers a trade-offs between the parallel time of the solver and the norm of the solution found. Specifically, their attack has depth $k \log _{k} T \cdot \operatorname{poly}(\lambda)$, and can find solutions of norm at most $m^{\left(\log _{k} T\right) / 2}$, for any $1 \leq k \leq T$. For $k=\sqrt{T}$, the Peikert-Tang attack is as performant as that of Attema-Ducas, and has no restriction on the image vector.

Implications on our PoSW Protocol. Underlying our PoSW protocol is a succinct interactive argument system which allows the prover to recursively prove knowledge of a short solution to a sequential-SIS relation. However, for the parameter choices and security proof techniques considered, the norm bound checked by the verifier is around $m^{\log T}$ times larger than the norm of the sequentialSIS solution (known as the "completeness gap"), and the witness extracted by the knowledge extractor in the security proof is yet another $m^{\log T}$ times longer (known as the "soundness gap"). Due to the large soundness gap, the security proof of the PoSW protocol becomes vacuous, because it relies on an instantiation of the SIS-sequentiality assumption which is false (for desirable parameters) due to the above attacks.

In addition to their attack against the SIS-sequentiality assumption, Peikert and Tang [PT23] also discovered attacks against a variant of our PoSW protocol (but not on the protocol presented in this work), with similar norm-time trade-offs as their attacks against the SIS-sequentiality assumption. This casts doubts on the heuristic security of our PoSW protocol, although at present we are not aware of any attack against the protocol.

The (Almost) Exact Sequentiality Assumption. We remark that both attacks discussed above allow to find solutions which are short but non-binary. We are currently unaware of any attack against the exact version of the SIS-sequentiality assumption, where the adversary is required to find a binary (or in general p-ary) solution, or even against the bounded-norm variant with a very strict norm bound of $m^{1 / 2(1-\epsilon)}$ where $\epsilon>0$. This means that the verifiable delay functions proposed in [CLM23] based on the exact version of the assumption is unaffected by the above attacks.

## 3 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 [MMV13, CP18, DLM19] with high concrete efficiency. However, the construction paradigm adopted by these hash-based PoSWs restricts them to only offer a weak sequentiality guarantee: 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, constructions based on unknown order groups (e.g. [Pie19]), which we aim to emulate with lattice-based techniques, seem to resist any adversary running in time at most $(1-\omega(1)) \cdot T$, i.e. no adversary can speed up the computation by any additive factor super-constant in $T$, while still having the verifier run in a fixed polynomial time.

Isogeny-Based Schemes. Recent works have explored constructions of timed cryptography from isogenies over elliptic curves [DMPS19, BD21]. This approach allows one to construct verifiable delay functions (VDF) [BBBF18] and even delay encryption [BD21]. However, such constructions are not post-quantum secure [DMPS19], or they rely on generic composition with succinct non-interactive arguments [CSRT22], 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 [JMRR21]. Incremental verifiable computation [Val08] can be immediately used to construct PoSWs and VDFs given a sequential function [BBBF18, DGMV20], and indistinguishability obfuscation can be used to construct time-lock puzzles [BGJ+16]. While theoretically elegant, such generic constructions use heavy cryptographic machinery and result in schemes that are (concretely) prohibitively inefficient.

### 3.1 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 [Ajt96, GGH96], 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 $f_{\mathbf{A}}: \mathbb{Z}_{q}^{n} \rightarrow \mathbb{Z}_{q}^{n}$ is defined as

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

where $\mathbf{A} \leftarrow \mathbb{Z}_{q}^{n \times m}$, for $m \approx n \log q$, and $\mathbf{G}^{-1}: \mathbb{Z}_{q}^{n} \rightarrow \mathbb{Z}_{2}^{m}$ is the binary decomposition operator. Below, we assume for simplicity that $m=n \log q$.

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 $\mathbf{u} \in \mathbb{Z}_{2}^{m}$ such that

$$
\binom{\mathbf{G}}{\mathbf{A}} \cdot \mathbf{u}=\binom{-\mathbf{x}}{\mathbf{y}} \bmod q
$$

where $\mathbf{G}$ is the binary reconstruction gadget matrix, which in particular is a linear operator.
Generalising, suppose $\mathbf{x}_{T}=f_{\mathbf{A}}^{T}\left(\mathbf{x}_{0}\right)$ is the $T$-fold repeated application of $f_{\mathbf{A}}$ on $\mathbf{x}_{0}$. Writing $\mathbf{x}_{i}=f_{\mathbf{A}}\left(\mathbf{x}_{i-1}\right)$ and $\mathbf{u}_{i}=-\mathbf{G}^{-1}\left(\mathbf{x}_{i}\right)$, we observe the following equivalent relation:

$$
\underbrace{\left(\begin{array}{cccc}
\mathbf{G} & & &  \tag{1}\\
\mathbf{A} & \mathbf{G} & \ddots & \\
& \mathbf{A} & \ddots & \\
& & \ddots & \mathbf{G} \\
& & & \mathbf{A}
\end{array}\right)}_{\mathbf{A}_{T}:=} \cdot \underbrace{\left(\begin{array}{c}
\mathbf{u}_{0} \\
\mathbf{u}_{1} \\
\vdots \\
\vdots \\
\mathbf{u}_{T-1}
\end{array}\right)}_{\mathbf{u}:=}=\left(\begin{array}{c}
-\mathbf{x}_{0} \\
\mathbf{0} \\
\vdots \\
\vdots \\
\mathbf{0} \\
\mathbf{x}_{T}
\end{array}\right) \bmod q \quad \text { and } \quad\left(\begin{array}{c}
\mathbf{u}_{0} \\
\mathbf{u}_{1} \\
\vdots \\
\vdots \\
\mathbf{u}_{T-1}
\end{array}\right) \in \mathbb{Z}_{2}^{m T}
$$

We conjecture and give evidence that, if the short integer solution problem $\mathrm{SIS}_{n, m, q, \mathbb{Z}_{2}}$ 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 $\left(\mathbf{u}_{0}, \ldots, \mathbf{u}_{T-1}, \mathbf{x}_{T}\right)$ satisfying the above relation in parallel time $(1-\omega(1)) \cdot T$.

Sequential Relation and Proof of Sequential Work. In the sequential relation (Eq. (1)) proposed above, enforcing $\mathbf{u} \in \mathbb{Z}_{2}^{m T}$ ensures that for each instance $\mathbf{x}_{0}$ there exists a unique witness $\left(\mathbf{u}_{0}, \ldots, \mathbf{u}_{T-1}, \mathbf{x}_{T}\right)$. 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 model [CMS19], we can obtain a candidate post-quantum VDF. For concrete efficiency, however, it may be beneficial to use a SNARG which natively supports proving lattice relations, e.g. that of [CLM23].

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). Towards this, we propose a variant of the sequential relation introduced above with a major change:

- (Bounded-Norm Variant) The binary constraint, i.e. $\mathbf{u} \in \mathbb{Z}_{2}^{m T}$, is relaxed to a bounded-norm constraint, i.e. $\mathbf{u} \in \mathbb{Z}_{\beta}^{m T}$ for some $2 \leq \beta \ll q$.
The relation in Eq. (1) is therefore replaced by

$$
\underbrace{\left(\begin{array}{cccc}
\mathbf{G} & & &  \tag{2}\\
\mathbf{A} & \mathbf{G} & & \\
& & & \\
& \mathbf{A} & \ddots & \\
& & \ddots & \mathbf{G} \\
& & & \mathbf{A}
\end{array}\right)}_{\mathbf{A}_{T}:=} \cdot \underbrace{\left(\begin{array}{c}
\mathbf{u}_{0} \\
\mathbf{u}_{1} \\
\vdots \\
\vdots \\
\mathbf{u}_{T-1}
\end{array}\right)}_{\mathbf{u}:=}=\left(\begin{array}{c}
-\mathbf{x} \\
\mathbf{0} \\
\vdots \\
\vdots \\
\mathbf{0} \\
\mathbf{y}
\end{array}\right) \bmod q \text { and } \quad \mathbf{u} \in \mathbb{Z}_{\beta}^{m T} .
$$

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

$$
\mathbf{A}_{T}=\left(\begin{array}{l|l|l}
\mathbf{A}_{t} & & \\
& \mathbf{G} & \\
& & \mathbf{A} \\
& \mathbf{A}_{t}
\end{array}\right) .
$$

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 [Pie19].

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 [FS87, AFK22] 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 $T=2 t+1$ is odd ${ }^{1}$. The prover sends $\mathbf{u}_{t}$ to the verifier, reducing the linear relation in Eq. (1) to

$$
\mathbf{A}_{t} \cdot\left(\begin{array}{cc}
\mathbf{u}_{0} & \mathbf{u}_{t+1} \\
\vdots & \vdots \\
\mathbf{u}_{t-1} & \mathbf{u}_{T-1}
\end{array}\right)=\left(\begin{array}{cc}
-\mathbf{x} & -\mathbf{A} \cdot \mathbf{u}_{t} \\
\mathbf{0} & \mathbf{0} \\
\vdots & \vdots \\
\vdots & \vdots \\
\mathbf{0} & \mathbf{0} \\
-\mathbf{G} \cdot \mathbf{u}_{t} & \mathbf{y}
\end{array}\right) \bmod q .
$$

The verifier checks that $\mathbf{u}_{t} \in \mathbb{Z}_{\beta}^{m}$. 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

$$
\mathbf{A}_{t}\left(\begin{array}{c}
\mathbf{u}_{0}+\mathbf{u}_{t+1} r \\
\vdots \\
\mathbf{u}_{t-1}+\mathbf{u}_{T-1} r
\end{array}\right)=\left(\begin{array}{c}
-\left(\mathbf{x}+\mathbf{A} \mathbf{u}_{t} r\right) \\
\mathbf{0} \\
\vdots \\
\mathbf{0} \\
\mathbf{y} r-\mathbf{G} \mathbf{u}_{t}
\end{array}\right) \bmod q,\left(\begin{array}{c}
\mathbf{u}_{0}+\mathbf{u}_{t+1} r \\
\vdots \\
\mathbf{u}_{t-1}+\mathbf{u}_{T-1} r
\end{array}\right) \in \mathbb{Z}_{\beta^{\prime}}^{m t}
$$

for an appropriately chosen $\beta^{\prime}>\beta$. 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. [BLNS20, AL21, ACK21]), one could argue that (a parallel repetition [AF22] of) the above protocol allows to convince the verifier that the prover has knowledge of a witness satisfying Eq. (2) with certain norm bound $\beta^{*}>\beta$. However, for typical choices of parameters, such $\beta^{*}$ is some $\lambda^{\omega(1)}$ factor larger than $\beta$ such that the sequentiality assumption becomes vacuous due to the attacks by Attema and

[^0]Ducas [AD23] and by Peikert and Tang [PT23]. We are therefore unable to prove the security of the PoSW protocol from any simpler assumption.

Note that even if we start with $\beta=2$, the above protocol can only convince the verifier about the satisfiability of Eq. (1) with some $\beta^{*}>\beta$, where witnesses are not unique. This is the why our construction of PoSW does not yield a VDF, even though our construction is morally similar to the VDF construction of [Pie19].

## 4 Preliminaries

We denote by $\lambda \in \mathbb{N}$ the security parameter. A function negl(•) 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 $\mathcal{A}=\left\{\mathcal{A}_{\lambda}\right\}_{\lambda \in \mathbb{N}}$, 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 $n \in \mathbb{N}$, denote $[n]:=\{1,2, \ldots, n\}$. For a finite set $S$, we write $U(S)$ for the uniform distribution over $S$. We will use $p, q, s \in \mathbb{N}$ to denote moduli.

We use bold capital letters, e.g. A, to denote matrices, and bold lower-case letters, e.g. $\mathbf{x}$, to denote column vectors. An $m$-dimensional column vector is denoted by $\mathbf{x}=\left(x_{1}, \ldots, x_{m}\right)$.

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 $\mathfrak{f}$-th primitive root of unity for some $\mathfrak{f}=\operatorname{poly}(\lambda)$. The degree of $\mathcal{R}$ is $\varphi=\varphi(\mathfrak{f})$. An element $x \in \mathcal{R}$ is represented by its coefficients encoding $x=\sum_{i=0}^{\varphi-1} x_{i} \zeta^{i}$, and its (infinity) norm is $\|x\|:=\max _{i=0}^{\varphi-1}\left|x_{i}\right|$, where $x_{i} \in \mathbb{Z}$. The norm extends naturally to vectors $\mathbf{u}=\left(u_{1}, \ldots, u_{m}\right) \in \mathcal{R}^{m}$, where $\|\mathbf{u}\|=\max _{i \in[m]}\left\|u_{i}\right\|$. The expansion factor of $\mathcal{R}$ is defined as $\gamma_{\mathcal{R}}:=\max _{a, b \in \mathcal{R}}\|a\| \cdot b\| \| \|$. We will always assume that $\mathfrak{f}$ is a prime-power, and in that case it is known that $\gamma_{\mathcal{R}} \leq 2 \varphi$ [AL21]. For $q \in \mathbb{N}$, define $\mathcal{R}_{q}:=\mathcal{R} / q \mathcal{R}$. By a slight abuse of notation, we identify $\mathcal{R}_{q}$ by

$$
\left\{\sum_{i=0}^{\varphi-1} x_{i} \zeta^{i}: x_{i} \in\{-\lceil q / 2\rceil+1, \ldots,\lfloor q / 2\rfloor\}\right\}
$$

and thus $\|x\| \leq q / 2$ for any $x \in \mathcal{R}_{q}$. The sets of units in $\mathcal{R}$ and $\mathcal{R}_{q}$ are denoted by $\mathcal{R}^{\times}$and $\mathcal{R}_{q}^{\times}$ respectively. 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 4.1 (Adapted from [BJRW20, Lemma 7]). Let $n=\operatorname{poly}(\lambda), p, q \in \mathbb{N}, q$ prime, and $m \geq n \log _{p} q+\omega(\log \lambda)$. The following distributions are statistically close in $\lambda$ :

$$
\left\{\begin{array}{c}
\mathbf{A} \leftarrow \$ \mathcal{R}_{q}^{n \times m} \\
(\mathbf{A}, \mathbf{v}): \mathbf{u} \leftarrow \$ \mathcal{R}_{p}^{m} \\
\mathbf{v}:=\mathbf{A} \cdot \mathbf{u} \bmod q
\end{array}\right\} \quad \text { and } \quad\left\{(\mathbf{A}, \mathbf{v}): \begin{array}{l}
\mathbf{A} \leftarrow \$ \mathcal{R}_{q}^{n \times m} \\
\mathbf{v} \leftarrow \$ \mathcal{R}_{q}^{n}
\end{array}\right\} .
$$

Gadget Matrices. For any $n, p, q \in \mathbb{N}$, let $\ell=\left\lceil\log _{p} q\right\rceil$ and $m=n \ell$. 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

$$
\mathbf{g}_{p, q}^{\mathrm{T}}:=\left(\begin{array}{llll}
1 & p & \ldots & p^{\ell-1}
\end{array}\right), \quad \mathbf{G}_{p, q}:=\mathbf{I}_{n} \otimes \mathbf{g}_{p, q}^{\mathrm{T}}, \quad \mathbf{H}_{p, q}:=\mathbf{I}_{n} \otimes\left(\begin{array}{ccccc}
p & & & & q_{0} \\
-1 & p & & & q_{1} \\
& -1 & \ddots & \ddots & \vdots \\
& & \ddots & p & \vdots \\
& & & -1 & q \ell-1
\end{array}\right)
$$

respectively. Define the operator $\mathbf{G}_{p, q}^{-1}: \mathcal{R}_{q}^{n} \rightarrow \mathcal{R}_{p}^{m}$ which maps $\mathbf{v}=\left(v_{i}\right)_{i=0}^{n-1} \in \mathcal{R}_{q}^{n}$ to the concatenation of its $p$-ary representation $\left(\left(v_{0, j}\right)_{j=0}^{\ell-1}, \ldots,\left(v_{n-1, j}\right)_{j=0}^{\ell-1}\right) \in \mathcal{R}_{p}^{m}$, 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 $\mathbf{G}:=\mathbf{G}_{p, q}$.

Computational Assumptions. In the following we define the ring variant of the well-known short integer solution (SIS) problem [Ajt96].

Assumption 4.2 (Short Integer Solution). Let $\mathcal{R}, n, m, q, s$ be parametrised by $\lambda$. The $\operatorname{SIS}_{\mathcal{R}, n, m, q, s}$ assumption states that for any $\mathbf{v} \in \mathcal{R}_{q}^{n}$ and any PPT adversary $\mathcal{A}$ it holds that

$$
\operatorname{Pr}\left[\begin{array}{l|l}
\mathbf{A} \cdot \mathbf{u}=\mathbf{v} \bmod q \wedge \mathbf{u} \in \mathcal{R}_{\beta}^{m} & \begin{array}{l}
\mathbf{A} \leftarrow \$ \mathcal{R}_{q}^{n \times m} \\
\mathbf{u} \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{v})
\end{array}
\end{array}\right] \leq \operatorname{negl}(\lambda) .
$$

We also recall the learning with errors (LWE) problem [Reg05], and in particular the version over rings [LPR10].

Assumption 4.3 (Learning with Errors). Let $\mathcal{R}, n, m, q, \chi$ be parametrised by $\lambda$. The (normal form of the) $\mathrm{LWE}_{\mathcal{R}, n, m, q, \chi}$ assumption states that for any PPT adversary $\mathcal{A}$ it holds that

$$
\left.\operatorname{Pr}\left[\mathcal{A}(\mathbf{A}, \mathbf{b})=1 \left\lvert\, \begin{array}{l}
\mathbf{A} \leftarrow \$ \mathcal{R}_{q}^{n \times m} \\
\mathbf{s} \leftarrow \$ \chi^{n} \\
\mathbf{e} \leftarrow \$ \chi^{m} \\
\mathbf{b}^{\mathrm{T}}:=\mathbf{s}^{\mathrm{T}} \cdot \mathbf{A}+\mathbf{e}^{\mathrm{T}} \bmod q
\end{array}\right.\right]-\operatorname{Pr}\left[\mathcal{A}(\mathbf{A}, \mathbf{b})=1 \left\lvert\, \begin{array}{l}
\mathbf{A} \leftarrow \$ \mathcal{R}_{q}^{n \times m} \\
\mathbf{b} \leftarrow \$ \mathcal{R}_{q}^{m}
\end{array}\right.\right] \right\rvert\, \leq \operatorname{negl}(\lambda) .
$$

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 4.4 (Decisional Short Integer Solution). Let $\mathcal{R}, n, m, q, \chi$ be parametrised by $\lambda$. The $\mathrm{dSIS}_{\mathcal{R}, n, m, q, \chi}$ assumption states that for any PPT adversary $\mathcal{A}$ it holds that

$$
\left.\operatorname{Pr}\left[\begin{array}{l|l}
\mathcal{A}(\mathbf{A}, \mathbf{v})=1 & \begin{array}{l}
\mathbf{A} \leftarrow \$ \mathcal{R}_{q}^{n \times m} \\
\mathbf{u} \leftarrow \$ \chi^{m} \\
\mathbf{v}:=\mathbf{A} \cdot \mathbf{u} \bmod q
\end{array}
\end{array}\right]-\operatorname{Pr}\left[\mathcal{A}(\mathbf{A}, \mathbf{v})=1 \left\lvert\, \begin{array}{l}
\mathbf{A} \leftarrow \$ \mathcal{R}_{q}^{n \times m} \\
\mathbf{v} \leftarrow \$ \mathcal{R}_{q}^{n}
\end{array}\right.\right] \right\rvert\, \leq \operatorname{negl}(\lambda)
$$

Lemma 4.5. If $m=n+\Omega(\lambda)$ and the $\operatorname{LWE}_{\mathcal{R}, n, m, q, \chi}$ assumption holds, then the $\operatorname{dSIS}_{\mathcal{R}, n, m, q, \chi}$ assumption holds.

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

We next analyse the distribution of $(\mathbf{A}, \mathbf{v})$. First, since $\overline{\mathbf{A}}$ 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 $(\overline{\mathbf{A}}, \overline{\mathbf{b}})$ is an LWE sample, then $\overline{\mathbf{b}}$ is of the form $\overline{\mathbf{b}}^{\mathrm{T}}=\mathbf{s}^{\mathrm{T}} \cdot \overline{\mathbf{A}}+\mathbf{e}^{\mathrm{T}}$ 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 $(\overline{\mathbf{A}}, \overline{\mathbf{b}})$ is a random sample, then $\mathbf{v}=\mathbf{A} \cdot \overline{\mathbf{b}} \bmod q$ is uniformly random.

## 5 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. We consider two variants of the assumptions, that we describe below.

Our Exact Sequential Function. For any $\mathbf{A} \in \mathcal{R}_{q}^{n \times m}$, define the function $f_{\mathbf{A}}: \mathcal{R}_{q}^{m} \rightarrow \mathcal{R}_{q}^{m}$ as

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

For $T \in \mathbb{N}$, the $T$-fold recursive evaluation of $f_{\mathbf{A}}$ is defined as:

$$
f_{\mathbf{A}, \mathbf{z}}^{T}(\mathbf{x}):=\underbrace{f_{\mathbf{A}}\left(\ldots f _ { \mathbf { A } } \left(f_{\mathbf{A}}\right.\right.}_{T \text { times }}(\mathbf{x})) \ldots)
$$

We conjecture that for uniformly distributed $(\mathbf{A}, \mathbf{x})$, the evaluation of $f_{\mathbf{A}}^{T}(\mathbf{x})$ take sequential time at least $\Omega(T) .{ }^{2}$

Our Sequential Relation. We also consider a relaxed variant of the assumption as stated above, that will be useful in the context of our PoSW. Compared to the above function, we introduce one modification: We only require that the transcript is of small norm. To formally state our conjecture, it is convenient to define

$$
\mathbf{A}_{T}:=\underbrace{\left(\begin{array}{ccccc}
\mathbf{G} & & & & \\
\mathbf{A} & \mathbf{G} & & & \\
& \mathbf{A} & \ddots & & \\
& & \ddots & \ddots & \mathbf{G} \\
& & & & \mathbf{A}
\end{array}\right)}_{T \text { columns }}
$$

[^1]Let $\mathbf{x}_{0}:=\mathbf{x}$. Observe that if the evaluation of each step of $f_{\mathbf{A}, \mathbf{z}}^{T}(\mathbf{x})$ is split into two sub-steps as $\mathbf{u}_{i-1}=-\mathbf{G}^{-1}\left(\mathbf{x}_{i-1}\right) \in \mathcal{R}_{p}^{m}$ and $\mathbf{x}_{i}=\mathbf{A} \mathbf{u}_{i-1} \bmod q$ for all $i \in[T]$, then

$$
\mathbf{A}_{T}\left(\begin{array}{c}
\mathbf{u}_{0} \\
\mathbf{u}_{1} \\
\vdots \\
\vdots \\
\mathbf{u}_{T-1}
\end{array}\right)=\left(\begin{array}{cccc}
\mathbf{G} & & & \\
\mathbf{A} & \mathbf{G} & & \\
& \mathbf{A} & \ddots & \\
& & \ddots & \mathbf{G} \\
& & & \mathbf{A}
\end{array}\right)\left(\begin{array}{c}
\mathbf{u}_{0} \\
\mathbf{u}_{1} \\
\vdots \\
\mathbf{u}_{T-1}
\end{array}\right)=\left(\begin{array}{c}
-\mathbf{x}_{0} \\
\mathbf{0} \\
\vdots \\
\vdots \\
\mathbf{0} \\
\mathbf{x}_{T}
\end{array}\right) \bmod q \quad \text { and } \quad\left(\begin{array}{c}
\mathbf{u}_{0} \\
\mathbf{u}_{1} \\
\vdots \\
\vdots \\
\mathbf{u}_{T-1}
\end{array}\right) \in \mathcal{R}_{p}^{m T}
$$

Furthermore, if $q=p^{\ell}$, we observe that the preimage $\mathbf{u}:=\left(\begin{array}{lll}\mathbf{u}_{0}^{\mathrm{T}} & \mathbf{u}_{1}^{\mathrm{T}} \quad \ldots & \mathbf{u}_{T-1}^{\mathrm{T}}\end{array}\right)^{\mathrm{T}}$, and hence the evaluation result $\mathbf{x}_{T}$, are unique. Moreover, for any $\beta \geq p$, we see that $\mathbf{u} \in \mathcal{R}_{p}^{m T}$ implies the relaxed condition $\mathbf{u} \in \mathcal{R}_{\beta}^{m T}$.

Formally, we state a family of conjectures parametrised by $\mathcal{R}, n, m, p, q, \beta, \sigma$ as follows.
Assumption 5.1 (SIS-Sequentiality). Let $\mathcal{R}, n, p, q, \beta, \sigma$ be parametrised by $\lambda$, where $\beta: \mathbb{N} \rightarrow \mathbb{R}^{+}$ and $\sigma: \mathbb{N} \rightarrow \mathbb{N}$ are functions, and let $m=n\left\lceil\log _{p} q\right\rceil$. For any $T \in \mathbb{N}$, if the $\operatorname{SIS}_{\mathcal{R}, n, m, q, \beta, \sigma}$ assumption holds for $\beta=\beta(T)$, then for all polynomial-size adversary $\mathcal{A}$ it holds that

$$
\operatorname{Pr}\left[\begin{array}{lllll}
\quad \mathbf{A}_{T} \cdot \mathbf{u}=\left(\begin{array}{llll}
-\mathbf{x}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} & \ldots & \mathbf{0}^{\mathrm{T}} \\
\mathbf{y}^{\mathrm{T}}
\end{array}\right)^{\mathrm{T}} \bmod q & \mathbf{A} \leftarrow \$ \mathcal{R}_{q}^{n \times m} \\
\wedge \mathbf{u} \in \mathcal{R}_{\beta(T)}^{m T} & \mathbf{x} \leftarrow \$ \mathcal{R}_{q}^{n} \\
\wedge \operatorname{Depth}(\mathcal{A})<\sigma(T) & & (\mathbf{y}, \mathbf{u})=\mathcal{A}(\mathbf{A}, \mathbf{x})
\end{array}\right] \leq \operatorname{neg}(\lambda) .
$$

The security proof of the PoSW of the published version of this work [LM23] relies on the assumption for $\beta=\left(2 \gamma_{\mathcal{R}}\right)^{2 \log T} \cdot p$. However, the attack by Attema and Ducas [AD23] shows that the assumption is false for $\beta=\gamma_{\mathcal{R}} m p / 2$ and $\sigma=O(\sqrt{T} \log T) \cdot \operatorname{poly}(\lambda)$. More generally, the attack by Peikert and Tang [PT23] falsifies the assumption in the case where $\beta>\left(\gamma_{\mathcal{R}} m\right)^{\left(\log _{k} T\right) / 2}$ and $\sigma(T)>\left(k \log _{k} T\right) \cdot \operatorname{poly}(\lambda)$ for some arity parameter $1 \leq k \leq T$. In particular, the Peikert-Tang attack has the same performance as that of Attema-Ducas when $k=\sqrt{T}$ and $p=2$.

To avoid both attacks, one should set $\beta$ and $\sigma$ to fall outside the regime of

$$
\beta>\left(\gamma_{\mathcal{R}} m\right)^{\left(\log _{k} T\right) / 2} \quad \text { and } \quad \sigma(T)>\left(k \log _{k} T\right) \cdot \operatorname{poly}(\lambda)
$$

For example, setting $\beta \leq\left(\gamma_{\mathcal{R}} m\right)^{\frac{1}{2(1-\epsilon)}}$ for some $0 \leq \epsilon=\epsilon(\lambda, T)<1$ forces the Peikert-Tang attack to run in depth at least $\frac{T^{1-\epsilon}}{1-\epsilon} \cdot \operatorname{poly}(\lambda)$. We refer to Appendix A for more details on the attacks.

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

### 5.1 Evidence of Sequentiality

To substantiate the plausibility of our exact 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 5.2 (Collision Resistance). If the $\operatorname{SIS}_{\mathcal{R}, n, m, q, p}$ problem is hard for $m=n \cdot\left\lceil\log _{p} q\right\rceil$, then $f_{\mathbf{A}}$ is collision resistant.

Proof. The proof is a trivial reduction from the $\mathrm{SI}_{\mathcal{R}, n, m, q, 2 p-1}$ problem. Let $\mathbf{A}$ be an instance of $\operatorname{SIS}_{\mathcal{R}, n, m, q, 2 p-1}$. If $\mathbf{x}, \mathbf{x}^{\prime} \in \mathcal{R}_{q}^{n}$ are distinct vectors such that $f_{\mathbf{A}}(\mathbf{x})=f_{\mathbf{A}}\left(\mathbf{x}^{\prime}\right)$, write $\mathbf{u}=-\mathbf{G}^{-1}(\mathbf{x})$ and $\mathbf{u}^{\prime}=-\mathbf{G}^{-1}\left(\mathbf{x}^{\prime}\right)$, we have $\mathbf{A} \cdot \mathbf{u}=\mathbf{A} \cdot \mathbf{u}^{\prime} \bmod q$. In other words, we have $\mathbf{A} \cdot\left(\mathbf{u}-\mathbf{u}^{\prime}\right)=\mathbf{0} \bmod q$ and $\mathbf{u}-\mathbf{u}^{\prime} \in \mathcal{R}_{2 p-1}^{m}$.

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.

Theorem 5.3 (Uniformity Preserving for Large $q \lesssim p^{k}$ ). Let $q$ be a prime of the form $q=p^{k}-r$ where $k>\log _{p} q+2 \lambda / n, r=\operatorname{poly}(\lambda), 0<r<p^{k}-p^{k-1}$, and $1 / q=\operatorname{negl}(\lambda)$. 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 $d=\operatorname{poly}(\lambda)$ be the degree of the ring $\mathcal{R}$. The statistical distance of the two distributions is given by

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

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

$$
\begin{aligned}
\left(\frac{q^{n}}{p^{m}}\right)^{d}=\left(\frac{q^{n}}{q^{n k}}\right)^{d}=\left(\frac{q}{p^{k}}\right)^{d n}=\left(\frac{p^{k}-r}{p^{k}}\right)^{d n} & =\left(1-\frac{r}{p^{k}}\right)^{d n}>\left(1-\frac{r}{q}\right)^{d n} \\
& \geq 1-\frac{r d n}{q} \geq 1-\operatorname{neg}(\lambda)
\end{aligned}
$$

In other words, we have $\Delta \leq \operatorname{negl}(\lambda)$. Since $k>\log _{p} q+2 \lambda / n$, we have $m=n k>n \log _{p} q+2 \lambda$. The result then follows from the leftover hash lemma (Lemma 4.1).

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 5.4 (Uniformity Preserving for $q=p^{k}$ ). Let $q=p^{k}$ for some $k \in \mathbb{N}$. If the $\operatorname{LWE}_{\mathcal{R}, n, m, q, U\left(\mathcal{R}_{p}\right)}$ assumption holds for $m=n k$, 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 $\operatorname{LWE}_{\mathcal{R}, n, m, q, U\left(\mathcal{R}_{p}\right)}$ 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. $U\left(\left(\mathbf{g}^{\mathbf{T}}\right)^{-1}\left(\mathcal{R}_{q}\right)\right)$.

More Heuristic Evidence. We offer more heuristic evidence that the function is indeed sequential. First, it was shown in [JMRR21] 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 [GSW13], 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. [CLMQ21] 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. ${ }^{3}$ 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.

### 5.2 Cryptanalysis

We discuss some attack strategies and why they do not apply to our candidate sequential function. In Appendix A, we summarise attacks discovered by other authors.

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

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

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.

[^2]Gluing Parallel Threads. Another (related) attack strategy is to glue together two parallel computation transcripts. For example, suppose $T=2 t+1$, 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 $\mathbf{x}_{t} \leftarrow f_{\mathbf{A}}^{t}(\mathbf{x})$ and $\mathbf{y} \leftarrow f_{\mathbf{A}}^{t}\left(\mathbf{x}^{*}\right)$. To obtain a consistent transcript, the adversary must now find a vector $\mathbf{u}_{t} \in \mathcal{R}^{m}$ such that:

$$
\binom{\mathbf{G}}{\mathbf{A}} \cdot \mathbf{u}_{t}=\binom{-\mathbf{x}_{t}}{\mathbf{x}^{*}} \bmod q \quad \text { and } \quad \mathbf{u}_{t} \in \mathcal{R}_{\beta}^{m}
$$

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

### 5.3 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 [DGMV20], 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:

$$
\left(\begin{array}{ccccc}
\mathbf{G} & & & & \\
\mathbf{A} & \mathbf{G} & & & \\
& \mathbf{A} & \ddots & \\
& & \ddots & \mathbf{G} \\
& & & \mathbf{A}
\end{array}\right) \cdot \mathbf{u}=\left(\begin{array}{c}
-\mathbf{x} \\
\mathbf{0} \\
\vdots \\
\vdots \\
\mathbf{0} \\
\mathbf{y}
\end{array}\right) \bmod q \quad \text { and } \quad \mathbf{u} \in \mathcal{R}_{p}^{m T}
$$

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 $\left(\prod_{a \in \mathcal{R}_{p}}\left(u_{i}-a\right)=0\right)_{i=1}^{m T}$ which reduces to $\left(u_{i} \cdot\left(u_{i}-1\right)=0\right)_{i=1}^{m T}$ for $(p, \mathcal{R})=(2, \mathbb{Z})$. We expect that recent constructions of efficient succinct arguments for structured relations, e.g. [BSCG ${ }^{+} 19$, CLM23], 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.

## 6 Proof of Sequential Work

We recall the definition of proofs of sequential work (PoSW) and present a construction based on the new lattice-based sequential relation. Due to the attacks by Attema and Ducas [AD23] and by Peikert and Tang [PT23] against the SIS-sequentiality assumption, the construction is not known to be provably secure against any simpler assumption. Moreover, we warn that due to another attack of Peikert and Tang [PT23] against a close variant of the PoSW, whatever security the construction below has might be fragile.

### 6.1 Definitions

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

| Gen(pp) | $\left\langle\operatorname{Eval}\left(\mathrm{x}, 1^{T}\right), \operatorname{Verify}(\mathrm{x}, T)\right\rangle$ |
| :---: | :---: |
| $\begin{aligned} & \mathbf{A} \leftarrow \$ \mathcal{R}_{q}^{n \times m} \\ & \mathbf{x} \leftarrow \$ \mathcal{R}_{q}^{n} \\ & \mathrm{x}:=(\mathbf{A}, \mathbf{x}) \\ & \text { return } \times \end{aligned}$ | $\begin{aligned} & \text { Eval }: \\ & \begin{aligned} & \mathbf{x}_{0}:=\mathbf{x} \\ & \quad \text { for } i \in\{0, \ldots, T-1\} \text { do } \\ & \quad \mathbf{u}_{i}:=-\mathbf{G}^{-1}\left(\mathbf{x}_{i}\right) \\ & \mathbf{x}_{i+1}:=\mathbf{A} \cdot \mathbf{u}_{i} \bmod q \\ & \text { wit }:=\left(\mathbf{u}_{i}\right)_{i=0}^{T-1} \quad \text { send } \mathbf{y}:=\mathbf{x}_{T} \\ &\langle\text { Eval, Verify }\rangle \\ & \quad \text { params }:=\mathbf{A} \\ & \text { stmt }:=(\mathbf{x}, \mathbf{y}, p, T) \\ & \text { for } j \in[\lambda / \log \lambda] \text { do } \\ & b_{j} \leftarrow\langle\mathcal{P}(\text { params, stmt, wit }), \mathcal{V}(\text { params, stmt })\rangle \end{aligned} \end{aligned}$ $\text { Verify : return }\left(b_{1} \wedge \ldots \wedge b_{\lambda / \log \lambda}\right)$ |

Figure 1: Construction of proof of sequential work.

Definition 6.1 ((Interactive) Proof of Sequential Work (PoSW)). An (interactive) proof of sequential work (PoSW) is a tuple of PPT algorithms/protocols (Gen, 〈Eval, Verify〉) with the following syntax:

- $x \leftarrow \operatorname{Gen}\left(1^{\lambda}\right)$ : The instance generation algorithm inputs the security parameter $\lambda \in \mathbb{N}$ and generates a problem instance $x$.
- $b \leftarrow\left\langle\operatorname{Eval}\left(\mathrm{x}, 1^{T}\right)\right.$, $\left.\operatorname{Verify}(\mathrm{x}, T)\right\rangle$ : The evaluation-verification protocol is run between the the interactive evaluation and verification algorithms. Both algorithms input an instance 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 $x \in \operatorname{Gen}\left(1^{\lambda}\right)$, the circuit-depth of Eval (as a function of $(\lambda, T)$ ) satisfies

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

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

$$
\left\langle\operatorname{Eval}\left(\mathrm{x}, 1^{T}\right), \operatorname{Verify}(\mathrm{x}, T)\right\rangle=1
$$

Sequential Soundness. Let $\sigma: \mathbb{N} \rightarrow \mathbb{N}$ be parametrised by $\lambda$. The PoSW is said to be sequentialsound with sequentiality $\sigma$ if, for any pair of PPT adversaries $\left(\mathcal{A}_{0}, \mathcal{A}_{1}\right)$, it holds that

$$
\operatorname{Pr}\left[\left.\begin{array}{c|l}
\left\langle\mathcal{A}_{1}(\mathrm{x}), \operatorname{Verify}(\mathrm{x}, T)\right\rangle=1 & \mathrm{x} \leftarrow \operatorname{Gen}\left(1^{\lambda}\right) \\
\left.\wedge \operatorname{Depth}\left(\mathcal{A}_{1}\right)<\sigma(T)\right)
\end{array} \right\rvert\, \begin{array}{l}
\text { st }, T) \leftarrow \mathcal{A}_{0}\left(1^{\lambda}\right)
\end{array}\right] \leq \operatorname{negl}(\lambda) .
$$

### 6.2 Construction

Let $S \subseteq \mathcal{R}$ be a subtractive set where $\|r\|=1$ for all $r \in S$. We first construct a core protocol $\langle\mathcal{P}$ (params, stmt, wit), $\mathcal{V}$ (params, stmt) $\rangle$, where params, stmt, wit are respectively parameters, statements, and witnesses of the form

$$
\text { params }=\mathbf{A}, \quad \text { stmt }=(\mathbf{x}, \mathbf{y}, \beta, T), \quad \text { wit }=\left(\mathbf{u}_{i}\right)_{i=0}^{T-1}
$$

for proving the relation

$$
\mathbf{A}_{T} \mathbf{u}=(-\mathbf{x}, \mathbf{0}, \ldots, \mathbf{0}, \mathbf{y}) \bmod q \quad \text { and } \quad \mathbf{u} \in \mathcal{R}_{\beta}^{m T}
$$

recursively as follows:

- If $T=1, \mathcal{P}$ sends $\mathbf{u}_{0}$ and $\mathcal{V}$ returns 1 if

$$
\binom{\mathbf{G}}{\mathbf{A}} \cdot \mathbf{u}_{0}=\binom{-\mathbf{x}}{\mathbf{y}} \bmod q \quad \text { and } \quad \mathbf{u}_{0} \in \mathcal{R}_{\beta}^{m}
$$

- If $T=2 t>1$ :
$-\mathcal{P}$ sends $\mathbf{u}_{T-1}$.
- $\mathcal{V}$ checks that
$\mathbf{A} \cdot \mathbf{u}_{T-1}=\mathbf{y} \bmod q$
and
$\mathbf{u}_{T-1} \in \mathcal{R}_{\beta}^{m}$
and returns 0 if not.
$-\mathcal{P}$ and $\mathcal{V}$ compute $\mathbf{y}^{\prime}=-\mathbf{G} \cdot \mathbf{u}_{T-1}$ and set stmt $:=\left(\mathbf{x}, \mathbf{y}^{\prime}, \beta, T-1\right)$.
$-\mathcal{P}$ sets wit $:=\left(\mathbf{u}_{i}\right)_{i=0}^{T-2}$.
$-\operatorname{Run}\left\langle\mathcal{P}\left(\right.\right.$ params, stmt $^{\prime}$, wit $\left.^{\prime}\right), \mathcal{V}\left(\right.$ params, stmt $\left.\left.{ }^{\prime}\right)\right\rangle$.
- If $T=2 t+1>1$ :
$-\mathcal{P}$ sends $\mathbf{u}_{t}$.
$-\mathcal{V}$ checks that $\mathbf{u}_{t} \in \mathcal{R}_{\beta}^{m}$, returns 0 if not.
- $\mathcal{V}$ samples $r \leftarrow \Phi S$ and sends $r$ to $\mathcal{P}$.
$-\mathcal{P}$ and $\mathcal{V}$ compute the following:
* $\mathbf{x}^{\prime}:=-\mathbf{x}_{0}-\mathbf{A} \cdot \mathbf{u}_{t} \cdot r \bmod q$.
* $\mathbf{y}^{\prime}:=\mathbf{x}_{T} \cdot r-\mathbf{G} \cdot \mathbf{u}_{t} \bmod q$.
* $\beta^{\prime}:=2 \gamma_{\mathcal{R}} \beta$.
* stmt ${ }^{\prime}:=\left(\mathbf{x}^{\prime}, \mathbf{y}^{\prime}, \beta^{\prime}, t\right)$.
- $\mathcal{P}$ computes
* $\mathbf{u}_{i}^{\prime}:=\mathbf{u}_{i}+\mathbf{u}_{t+i+1} \cdot r$ for all $i \in\{0, \ldots, t-1\}$.
* wit $^{\prime}:=\left(\mathbf{u}_{i}^{\prime}\right)_{i=0}^{t-1}$.
$-\operatorname{Run}\left\langle\mathcal{P}\left(\right.\right.$ params, stmt $^{\prime}$, wit $\left.^{\prime}\right), \mathcal{V}\left(\right.$ params, stmt $\left.\left.^{\prime}\right)\right\rangle$.
The PoSW protocol is then specified in Fig. 1.

Completeness. We show that the above core protocol and therefore the PoSW are complete.
It is easy to check that, given an instance ( $\mathbf{A}, \mathbf{x}$ ), the evaluator produces an output $\mathbf{y}$ and a witness $\mathbf{u}=\left(\mathbf{u}_{i}\right)_{i=0}^{T-1}$ satisfying

$$
\mathbf{A}_{T} \mathbf{u}=(-\mathbf{x}, \mathbf{0}, \ldots, \mathbf{0}, \mathbf{y}) \bmod q \quad \text { and } \quad \mathbf{u} \in \mathcal{R}_{p}^{m T}
$$

Indeed, by construction, we have for $i \in\{0, \ldots, T-1\}$,

$$
\begin{aligned}
\mathbf{u}_{i} \in-\mathbf{G}^{-1}\left(\mathbf{x}_{i}\right) & \in \mathcal{R}_{2}^{m} \\
\mathbf{G u} & =-\mathbf{x}_{i} \bmod q, \\
\mathbf{A} \mathbf{u}_{i} & =\mathbf{x}_{i+1} \bmod q
\end{aligned}
$$

Therefore, for $i \in\{0, \ldots, T-2\}$,

$$
\mathbf{A} \mathbf{u}_{i}+\mathbf{G} \mathbf{u}_{i+1}=\mathbf{0} \bmod q .
$$

Finally, by construction, we have

$$
\mathbf{y}=\mathbf{x}_{T}=\mathbf{A} \mathbf{u}_{T-1} \bmod q .
$$

It remains to check that the core protocol is complete for the relation

$$
\mathbf{A}_{T} \mathbf{u}=(-\mathbf{x}, \mathbf{0}, \ldots, \mathbf{0}, \mathbf{y}) \bmod q \quad \text { and } \quad \mathbf{u} \in \mathcal{R}_{\beta}^{m T}
$$

for any $\beta$.
The cases $T=1$ and $T=2 t>1$ are obvious. For the case $T=2 t+1>1$, observe the relations

$$
\begin{aligned}
\mathbf{A}_{t}\left(\begin{array}{cc}
\mathbf{u}_{0} & \mathbf{u}_{t-1} \\
\vdots & \vdots \\
\mathbf{u}_{t+1} & \mathbf{u}_{2 t}
\end{array}\right) & =\left(\begin{array}{cc}
-\mathbf{x}_{0} & -\mathbf{A} \cdot \mathbf{u}_{t} \\
\vdots & \vdots \\
-\mathbf{G} \mathbf{u}_{t} & \mathbf{x}_{T}
\end{array}\right) \bmod q, \\
\mathbf{A}_{t}\left(\begin{array}{c}
\mathbf{u}_{0}+\mathbf{u}_{t-1} \cdot r \\
\vdots \\
\mathbf{u}_{t+1}+\mathbf{u}_{2 t} \cdot r
\end{array}\right) & =\left(\begin{array}{c}
-\mathbf{x}_{0}-\mathbf{A} \cdot \mathbf{u}_{t} \cdot r \\
\vdots \\
-\mathbf{G} \mathbf{u}_{t}+\mathbf{x}_{T} \cdot r
\end{array}\right) \bmod q
\end{aligned}
$$

where the folded witness $\left(\mathbf{u}_{i}+\mathbf{u}_{t+i+1} \cdot r\right)$ is of norm at most $2 \gamma_{\mathcal{R}} \beta$. The claim thus follows by induction.

Soundness. As mentioned in the beginning of Section 6, our security proof of the PoSW is vacuous since it relies on an instantiation of the SIS-sequentiality assumption with parameters vulnerable to known attacks. We keep the original soundness analysis below for completeness.

In the following we show that the core protocol is $(2,2, \ldots, 2)$-special sound. First, we recall the definition of special soundness and a useful fact from [AL21].

Definition $6.2\left(\left(k_{1}, \ldots, k_{\mu}\right)\right.$-out-of- $\left(N_{1}, \ldots, N_{\mu}\right)$-Special Soundness [AF22]). 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 $\left(k_{1}, \ldots, k_{\mu}\right)$-out-of- $\left(N_{1}, \ldots, N_{\mu}\right)$-special-sound if there exists a polynomial-time algorithm that, on input a statement stmt and a ( $k_{1}, \ldots, k_{\mu}$ )-tree of accepting transcripts, outputs a witness wit such that (stmt; wit) $\in \Phi$. We also say $(\mathcal{P}, \mathcal{V})$ is $\left(k_{1}, \ldots, k_{\mu}\right)$-special-sound.

For the (standard) definitions of public-coin protocol and trees of accepting transcripts we refer to [AF22].

Lemma 6.3 ([AL21]). Let $\mathcal{R}=\mathbb{Z}\left[\zeta_{d+1}\right]$ be the $(d+1)$-th cyclotomic ring where $(d+1)$ is prime. The set $S=\left\{\mu_{1}, \ldots, \mu_{d}\right\} \subseteq \mathcal{R}^{\times}$where $\mu_{i}=\left(\zeta^{i}-1\right) /(\zeta-1)$ is subtractive, i.e. for any $r_{0}, r_{1} \in S$ it holds that $\left(r_{1}-r_{0}\right)^{-1} \in \mathcal{R}$. Furthermore, for any $r_{0}, r_{1} \in S$, we have $\frac{r_{0}}{r_{1}-r_{0}}, \frac{r_{1}}{r_{1}-r_{0}}, \frac{1}{r_{1}-r_{0}} \in \mathcal{R}_{2}$.
Lemma 6.4 (Special Soundness). The above folding argument is $(2,2, \ldots, 2)$-special sound for the relation

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

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 $T_{i}:=2^{\mu-i+1}-1$ 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 node ${ }_{0}$ and node $_{1}$ be siblings at depth- $i$ associated with the challenges $r_{0}$ and $r_{1}$ respectively, and let node $\epsilon_{\epsilon}$ be the parent node of node ${ }_{0}$ and node $_{1}$. From the tree of accepting transcripts, $\mathcal{E}$ fetches the vectors $\mathbf{x}_{0}^{(\text {node })}, \mathbf{x}_{T_{\text {depth (node })}^{(\text {note }}}^{\text {( }}, \mathbf{u}_{t}^{(\text {node })}$ associated to each node node $\in\left\{\right.$ node $_{\epsilon}$, node $_{0}$, node $\left.{ }_{1}\right\}$ recursively defined such that $\mathbf{x}_{0}^{\text {root }}=\mathbf{x}_{0}, \mathbf{x}_{T_{0}}^{\text {root }}=\mathbf{x}_{T}$, and

$$
\binom{\mathbf{x}_{0}^{\left(\text {node }_{b}\right)}}{\mathbf{x}_{T_{i}}^{\left(\text {node }_{b}\right)}}=\binom{\mathbf{x}_{0}^{\left(\text {node }_{\epsilon}\right)}}{-\mathbf{G} \cdot \mathbf{u}_{T_{i}}^{\left(\text {node }_{\epsilon}\right)}}+\binom{\mathbf{A} \cdot \mathbf{u}_{T_{i}}^{\left(\text {node }_{\epsilon}\right)}}{\mathbf{x}_{T_{i-1}}^{\text {node } \left._{\epsilon}\right)}} \cdot r_{b} \bmod q .
$$

Suppose the vector $\left(\mathbf{u}_{0}^{\text {node }_{b}}, \ldots, \mathbf{u}_{T_{i}-1}^{\text {node }_{b}}\right.$ ) extracted at node ${ }_{b}$ for $b \in\{0,1\}$ satisfies

$$
\mathbf{A}_{T_{i}} \cdot\left(\begin{array}{c}
\mathbf{u}_{0}^{\text {node }_{b}} \\
\vdots \\
\mathbf{u}_{T_{i}-1}^{\text {node }_{b}}
\end{array}\right)=\left(\begin{array}{c}
-\mathbf{x}_{0}^{\left(\text {node }_{b}\right)} \\
\mathbf{0} \\
\vdots \\
\mathbf{0} \\
\mathbf{x}_{T_{i}}^{\text {node } \left._{b}\right)}
\end{array}\right) \bmod q \quad \text { and } \quad\left(\begin{array}{c}
\mathbf{u}_{0}^{\text {node }_{b}} \\
\vdots \\
\mathbf{u}_{T_{i}-1}^{\text {node }_{b}}
\end{array}\right) \in \mathcal{R}_{\left(2 \gamma_{\mathcal{R}}\right)^{2 \mu-i . \beta}}^{m T_{i}} .
$$

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

$$
\mathbf{A}_{T_{i}} \cdot\left(\begin{array}{c}
\mathbf{u}_{0}^{\text {node }_{b}} \\
\vdots \\
\mathbf{u}_{T_{i}-1}^{\text {node }_{b}}
\end{array}\right)=\left(\begin{array}{c}
-\mathbf{x}_{0}^{\left(\text {node }_{\epsilon}\right)} \\
\mathbf{0} \\
\vdots \\
\mathbf{0} \\
-\mathbf{G} \cdot \mathbf{u}_{T_{i}}^{\left(\text {node }_{\epsilon}\right)}
\end{array}\right)+\left(\begin{array}{c}
-\mathbf{A} \cdot \mathbf{u}_{T_{i}}^{\left(\text {node }_{\epsilon}\right)} \\
\mathbf{0} \\
\vdots \\
\mathbf{0} \\
\mathbf{x}_{T}^{\left(\text {node }_{\epsilon}\right)}
\end{array}\right) \cdot r_{b} \bmod q .
$$

Let

$$
\left(\begin{array}{cc}
\mathbf{u}_{0} & \mathbf{u}_{t+1} \\
\vdots & \vdots \\
\vdots & \vdots \\
\mathbf{u}_{T_{i}-1} & \mathbf{u}_{T_{i-1}-1}
\end{array}\right)=\left(\begin{array}{cc}
\mathbf{u}_{0}^{\text {node }_{0}} & \mathbf{u}_{0}^{\text {node }_{1}} \\
\vdots & \vdots \\
\vdots & \vdots \\
\mathbf{u}_{T_{i}-1}^{\text {node }} & \mathbf{u}_{T_{i}-1}^{\text {node }}
\end{array}\right) \cdot\left(\begin{array}{cc}
\frac{r_{1}}{r_{1}-r_{0}} & \frac{-1}{r_{1}-r_{0}} \\
\frac{r_{1}-r_{0}}{} & \frac{1}{r_{1}-r_{0}}
\end{array}\right) .
$$

It follows that

$$
\mathbf{A}_{T_{i}} \cdot\left(\begin{array}{cc}
\mathbf{u}_{0} & \mathbf{u}_{t+1} \\
\vdots & \vdots \\
\mathbf{u}_{T_{i}-1} & \mathbf{u}_{T_{i-1}-1}
\end{array}\right)=\left(\begin{array}{cc}
-\mathbf{x}_{0}^{\left(\text {node }_{\epsilon}\right)} & -\mathbf{A} \cdot \mathbf{u}_{t}^{\left(\text {node }_{\epsilon}\right)} \\
\mathbf{0} & \mathbf{0} \\
\vdots & \vdots \\
\vdots & \mathbf{0} \\
-\mathbf{G} \cdot \mathbf{u}_{T_{i}}^{\left(\text {node }_{\epsilon}\right)} & \mathbf{x}_{T_{i-1}}^{\left(\text {node }_{\epsilon}\right)}
\end{array}\right) \bmod q,
$$

or equivalently

$$
\mathbf{A}_{T_{i-1}} \cdot\left(\begin{array}{c}
\mathbf{u}_{0} \\
\vdots \\
\mathbf{u}_{T_{i-1}-1}
\end{array}\right)=\left(\begin{array}{c}
-\mathbf{x}_{0}^{\left(\text {node }_{\epsilon}\right)} \\
\mathbf{0} \\
\vdots \\
\mathbf{0} \\
\mathbf{x}_{T_{i-1}}^{\left(\text {node }_{\epsilon}\right)}
\end{array}\right) \bmod q,
$$

and $\left(\mathbf{u}_{0}, \ldots, \mathbf{u}_{T-1}\right) \in \mathcal{R}_{\left(2 \gamma_{\mathcal{R}}\right)^{2 \mu-i+1 . \beta}}$, where the inclusion is due to Lemma 6.3.
By recursion, $\mathcal{E}$ extracts at the root node a vector $\left(\mathbf{u}_{0}, \ldots, \mathbf{u}_{T-1}\right)$ satisfying

$$
\mathbf{A}_{T}\left(\begin{array}{c}
\mathbf{u}_{0} \\
\vdots \\
\mathbf{u}_{T-1}
\end{array}\right)=\left(\begin{array}{c}
-\mathbf{x}_{0} \\
\mathbf{0} \\
\vdots \\
\vdots \\
\mathbf{0} \\
\mathbf{x}_{T}
\end{array}\right) \bmod q,\left(\begin{array}{c}
\mathbf{u}_{0} \\
\vdots \\
\mathbf{u}_{T-1}
\end{array}\right) \in \mathcal{R}_{\left(2 \gamma_{\mathcal{R}}\right)^{2 \mu \beta}}^{m T} \subset \mathcal{R}_{\left(2 \gamma_{\mathcal{R}}\right)^{2 \log T_{\beta}}}
$$

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 6.5 (Soundness (Vacuous due to [PT23])). There exists $p(\lambda) \in \operatorname{poly}(\lambda)$ and $\sigma^{\prime}(T)=$ $\sigma(T) \cdot p(\lambda)$ such that if $\operatorname{SIS}_{\mathcal{R}, n, m, q, \beta}$ assumption and the SIS-sequentiality assumption with sequentiality $\sigma^{\prime}$ hold, then the PoSW constructed in Fig. 1 is sequentially sound with sequentiality $\sigma$.

Sketch. The theorem follows from Lemma 6.4 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 [AF22] 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 [AF22] is depth-preserving, i.e. there exists a polynomial $p(\lambda) \in \operatorname{poly}(\lambda)$ such that the knowledge extractor $\mathcal{E}_{\mathcal{A}}$ has depth $\operatorname{Depth}\left(\mathcal{E}_{\mathcal{A}}\right) \leq p(\lambda) \cdot \operatorname{Depth}(\mathcal{A})$. Suppose there exists a polynomial-size adversary $\mathcal{A}$ which breaks the sequential-soundness of the PoSW with sequentiality $\sigma$, then the above shows that $\mathcal{E}_{\mathcal{A}}$ is a polynomial-size adversary which breaks the SIS-sequentiality assumption with sequentiality $\sigma^{\prime}$.

In Appendix B, 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 5.2. G.M. wishes to thank Chis Peikert for discussions on the sequentiality assumption. R.L. wishes to thank Thomas Attema for sharing an earlier draft of his attack. 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, by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA - 390781972, and by the European Research Council through an ERC Starting Grant (Grant agreement No. 101077455, ObfusQation).

## References

[ACK21] Thomas Attema, Ronald Cramer, and Lisa Kohl. A compressed $\Sigma$-protocol theory for lattices. In Tal Malkin and Chris Peikert, editors, CRYPTO 2021, Part II, volume 12826 of LNCS, pages 549-579, Virtual Event, August 2021. Springer, Heidelberg.
[AD23] Thomas Attema and Léo Ducas. Personal communication, 2023.
[AF22] Thomas Attema and Serge Fehr. Parallel repetition of $\left(k_{1}, \ldots, k_{\mu}\right)$-special-sound multi-round interactive proofs. In Yevgeniy Dodis and Thomas Shrimpton, editors, CRYPTO 2022, Part I, volume 13507 of LNCS, pages 415-443. Springer, Heidelberg, August 2022.
[AFK22] Thomas Attema, Serge Fehr, and Michael Klooß. Fiat-shamir transformation of multiround interactive proofs. In Eike Kiltz and Vinod Vaikuntanathan, editors, TCC 2022, Part I, volume 13747 of LNCS, pages 113-142. Springer, Heidelberg, November 2022.
[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In 28th ACM STOC, pages 99-108. ACM Press, May 1996.
[AL21] Martin R. Albrecht and Russell W. F. Lai. Subtractive sets over cyclotomic rings limits of Schnorr-like arguments over lattices. In Tal Malkin and Chris Peikert, editors, CRYPTO 2021, Part II, volume 12826 of LNCS, pages 519-548, Virtual Event, August 2021. Springer, Heidelberg.
[BBBF18] Dan Boneh, Joseph Bonneau, Benedikt Bünz, and Ben Fisch. Verifiable delay functions. In Hovav Shacham and Alexandra Boldyreva, editors, CRYPTO 2018, Part I, volume 10991 of LNCS, pages 757-788. Springer, Heidelberg, August 2018.
$\left[\mathrm{BCH}^{+} 22\right]$ Nir Bitansky, Arka Rai Choudhuri, Justin Holmgren, Chethan Kamath, Alex Lombardi, Omer Paneth, and Ron D. Rothblum. PPAD is as hard as LWE and iterated squaring. In Eike Kiltz and Vinod Vaikuntanathan, editors, TCC 2022, Part II, volume 13748 of LNCS, pages 593-622. Springer, Heidelberg, November 2022.
[BD21] Jeffrey Burdges and Luca De Feo. Delay encryption. In Anne Canteaut and FrançoisXavier Standaert, editors, EUROCRYPT 2021, Part I, volume 12696 of LNCS, pages 302-326. Springer, Heidelberg, October 2021.
[BGJ $\left.{ }^{+} 16\right]$ Nir Bitansky, Shafi Goldwasser, Abhishek Jain, Omer Paneth, Vinod Vaikuntanathan, and Brent Waters. Time-lock puzzles from randomized encodings. In Madhu Sudan, editor, ITCS 2016, pages 345-356. ACM, January 2016.
[BJRW20] Katharina Boudgoust, Corentin Jeudy, Adeline Roux-Langlois, and Weiqiang Wen. Towards classical hardness of module-LWE: The linear rank case. In Shiho Moriai and Huaxiong Wang, editors, ASIACRYPT 2020, Part II, volume 12492 of LNCS, pages 289-317. Springer, Heidelberg, December 2020.
[BLNS20] Jonathan Bootle, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. A non-PCP approach to succinct quantum-safe zero-knowledge. In Daniele Micciancio and Thomas Ristenpart, editors, CRYPTO 2020, Part II, volume 12171 of LNCS, pages 441-469. Springer, Heidelberg, August 2020.
[BN00] Dan Boneh and Moni Naor. Timed commitments. In Mihir Bellare, editor, CRYPTO 2000, volume 1880 of $L N C S$, pages 236-254. Springer, Heidelberg, August 2000.
[BR23] Andrej Bogdanov and Alon Rosen. Personal communication, 2023.
[ $\left.\mathrm{BSCG}^{+} 19\right]$ Eli Ben-Sasson, Alessandro Chiesa, Lior Goldberg, Tom Gur, Michael Riabzev, and Nicholas Spooner. Linear-size constant-query iops for delegating computation. In Dennis Hofheinz and Alon Rosen, editors, Theory of Cryptography (TCC), 2019.
[CLM23] Valerio Cini, Russell W. F. Lai, and Giulio Malavolta. Lattice-based succinct arguments from vanishing polynomials - (extended abstract). In Helena Handschuh and Anna Lysyanskaya, editors, CRYPTO 2023, Part II, volume 14082 of LNCS, pages 72-105. Springer, Heidelberg, August 2023.
[CLMQ21] Yilei Chen, Alex Lombardi, Fermi Ma, and Willy Quach. Does fiat-shamir require a cryptographic hash function? In Tal Malkin and Chris Peikert, editors, CRYPTO 2021, Part IV, volume 12828 of LNCS, pages 334-363, Virtual Event, August 2021. Springer, Heidelberg.
[CMS19] Alessandro Chiesa, Peter Manohar, and Nicholas Spooner. Succinct arguments in the quantum random oracle model. In Dennis Hofheinz and Alon Rosen, editors, TCC 2019, Part II, volume 11892 of $L N C S$, pages 1-29. Springer, Heidelberg, December 2019.
[CP18] Bram Cohen and Krzysztof Pietrzak. Simple proofs of sequential work. In Jesper Buus Nielsen and Vincent Rijmen, editors, EUROCRYPT 2018, Part II, volume 10821 of LNCS, pages 451-467. Springer, Heidelberg, April / May 2018.
[CSRT22] Jorge Chávez-Saab, Francisco Rodríguez-Henríquez, and Mehdi Tibouchi. Verifiable isogeny walks: Towards an isogeny-based postquantum VDF. In Riham AlTawy and Andreas Hülsing, editors, SAC 2021, volume 13203 of LNCS, pages 441-460. Springer, Heidelberg, September / October 2022.
[DGMV20] Nico Döttling, Sanjam Garg, Giulio Malavolta, and Prashant Nalini Vasudevan. Tight verifiable delay functions. In Clemente Galdi and Vladimir Kolesnikov, editors, SCN 20, volume 12238 of $L N C S$, pages 65-84. Springer, Heidelberg, September 2020.
[DLM19] Nico Döttling, Russell W. F. Lai, and Giulio Malavolta. Incremental proofs of sequential work. In Yuval Ishai and Vincent Rijmen, editors, EUROCRYPT 2019, Part II, volume 11477 of LNCS, pages 292-323. Springer, Heidelberg, May 2019.
[DMPS19] Luca De Feo, Simon Masson, Christophe Petit, and Antonio Sanso. Verifiable delay functions from supersingular isogenies and pairings. In Steven D. Galbraith and Shiho Moriai, editors, ASIACRYPT 2019, Part I, volume 11921 of LNCS, pages 248-277. Springer, Heidelberg, December 2019.
[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, CRYPTO'86, volume 263 of LNCS, pages 186-194. Springer, Heidelberg, August 1987.
[GGH96] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Collision-free hashing from lattice problems. Cryptology ePrint Archive, Report 1996/009, 1996. https://eprint.iacr. org/1996/009.
[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, CRYPTO 2013, Part I, volume 8042 of LNCS, pages 75-92. Springer, Heidelberg, August 2013.
[JMRR21] Samuel Jaques, Hart Montgomery, Razvan Rosie, and Arnab Roy. Time-release cryptography from minimal circuit assumptions. In Avishek Adhikari, Ralf Küsters, and Bart Preneel, editors, INDOCRYPT 2021, volume 13143, pages 584-606. Springer, 2021.
[LM23] Russell W. F. Lai and Giulio Malavolta. Lattice-based timed cryptography. In Helena Handschuh and Anna Lysyanskaya, editors, CRYPTO 2023, Part V, volume 14085 of LNCS, pages 782-804. Springer, Heidelberg, August 2023.
[LMS22] Russell W. F. Lai, Giulio Malavolta, and Nicholas Spooner. Quantum rewinding for many-round protocols. In Eike Kiltz and Vinod Vaikuntanathan, editors, TCC 2022, Part I, volume 13747 of LNCS, pages 80-109. Springer, Heidelberg, November 2022.
[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, EUROCRYPT 2010, volume 6110 of LNCS, pages 1-23. Springer, Heidelberg, May / June 2010.
[LPS17] Huijia Lin, Rafael Pass, and Pratik Soni. Two-round and non-interactive concurrent non-malleable commitments from time-lock puzzles. In Chris Umans, editor, 58th FOCS, pages 576-587. IEEE Computer Society Press, October 2017.
[MMV13] Mohammad Mahmoody, Tal Moran, and Salil P. Vadhan. Publicly verifiable proofs of sequential work. In Robert D. Kleinberg, editor, ITCS 2013, pages 373-388. ACM, January 2013.
[MT19] Giulio Malavolta and Sri Aravinda Krishnan Thyagarajan. Homomorphic time-lock puzzles and applications. In Alexandra Boldyreva and Daniele Micciancio, editors, CRYPTO 2019, Part I, volume 11692 of LNCS, pages 620-649. Springer, Heidelberg, August 2019.
[Pie19] Krzysztof Pietrzak. Simple verifiable delay functions. In Avrim Blum, editor, ITCS 2019, volume 124, pages 60:1-60:15. LIPIcs, January 2019.
[PT23] Chris Peikert and Yi Tang. Cryptanalysis of lattice-based sequentiality assumptions and proofs of sequential work. IACR Cryptol. ePrint Arch., page 1880, 2023.
[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, 37th ACM STOC, pages 84-93. ACM Press, May 2005.
[RSW96] Ronald L. Rivest, Adi Shamir, and David A. Wagner. Time-lock puzzles and timedrelease crypto. Technical report, 1996.
[Unr16] Dominique Unruh. Computationally binding quantum commitments. In Marc Fischlin and Jean-Sébastien Coron, editors, EUROCRYPT 2016, Part II, volume 9666 of LNCS, pages 497-527. Springer, Heidelberg, May 2016.
[Val08] Paul Valiant. Incrementally verifiable computation or proofs of knowledge imply time/space efficiency. In Ran Canetti, editor, TCC 2008, volume 4948 of LNCS, pages 1-18. Springer, Heidelberg, March 2008.
[vdf19] Vdf alliance, 2019. Accessed in June 2023.

## A Attacks against SIS-Sequentiality

In [LM23], it was implicitly assumed that the SIS-sequentiality assumption (Assumption 5.1) holds for any choice of $\beta$ and $\sigma$ with $\sigma(T)=(1-\omega(1)) \cdot T \cdot \operatorname{poly}(\lambda)$. In particular, the sequential-soundness of the PoSW was proven in [LM23] based on the assumption with $\beta=\left(2 \gamma_{\mathcal{R}}\right)^{2 \log T} \cdot p$. Below, we discuss attacks against the SIS-sequentiality assumption and its "offline" variant.

Offline Preprocessing Attack. This attack is suggested to us by anonymous reviewers. Let us start with the simplified settings where we fix the matrix $\mathbf{A}$, instead of sampling it as part of the instance $(\mathbf{A}, \mathbf{x})$. The idea of the attack is to precompute a witness for $\left(\mathbf{A}, p^{i} \cdot \mathbf{e}_{j}\right)$, for $i=0, \ldots,\left\lceil\log _{p} q\right\rceil-1$ 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 $\left(\mathbf{u}_{0}^{(i, j)}, \ldots, \mathbf{u}_{T-1}^{(i, j)}, \mathbf{y}^{(i, j)}\right)$. Note that $\mathbf{u}_{k}^{(i, j)} \in \mathcal{R}_{p}^{m}$. Upon receiving $\mathbf{x} \in \mathcal{R}_{q}^{n}$, decompose $\mathbf{x}$ into

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

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

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

Note that $\mathbf{u}_{k} \in \mathcal{R}_{\gamma_{\mathcal{R}} m p}^{m}$, and furthermore the parallel runtime of the attack (after the preprocessing phase) is $\log T \cdot \operatorname{poly}(\lambda)$. This would break a hypothetical version of our sequentiality assumption, where we fix $\mathbf{A}$ for all instances, for $\beta=\gamma_{\mathcal{R}} m p$ and $\sigma=\log T \cdot \operatorname{poly}(\lambda)$.

Online Preprocessing Attack. Next, we describe an online attack by Attema and Ducas [AD23] which, given an instance $(\mathbf{A}, \mathbf{x})$, finds a solution $(\mathbf{y}, \mathbf{u})$ with $\mathbf{u} \in \mathcal{R}_{\gamma_{\mathcal{R}} m p}^{m T}$ in $\operatorname{depth} \sqrt{T} \log T \cdot \operatorname{poly}(\lambda)$.

The attack consists of a preprocessing phase and an iterative phase. Upon receiving ( $\mathbf{A}, \mathbf{x}$ ) and denoting $\mathbf{x}_{0}:=\mathbf{x}$, perform the above preprocessing on $\mathbf{A}$ for $\sqrt{T}$ steps to compute witnesses $\left(\mathbf{u}_{0}^{(i, j)}, \ldots, \mathbf{u}_{\sqrt{T}-1}^{(i, j)}, \mathbf{y}^{(i, j)}\right)$ for $i=0, \ldots,\left\lceil\log _{p} q\right\rceil-1$ and $j=1, \ldots, n$. Then, in the iterative phase, for $\ell=1, \ldots, \sqrt{T}$, decompose

$$
\mathbf{x}_{(\ell-1) \sqrt{T}}=\sum_{i=0}^{\left\lceil\log _{p} q\right\rceil-1} \sum_{j=1}^{n} x_{(\ell-1) \sqrt{T}}^{(i, j)} \cdot p^{i} \cdot \mathbf{e}_{j}
$$

and compute

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

Finally, output $\left(\mathbf{x}_{T}, \mathbf{u}\right)$, where $\mathbf{u}^{\mathrm{T}}:=\left(\mathbf{u}_{0}^{\mathrm{T}}, \ldots, \mathbf{u}_{T-1}^{\mathrm{T}}\right)$.
Clearly, for each $k=1, \ldots, T$, we have $\mathbf{u}_{k} \in \mathcal{R}_{\gamma \mathcal{R} m p}^{m}$ as in the preprocessing attack. For the runtime, note that the preprocessing of $\mathbf{A}$ for $\sqrt{T}$ steps takes takes depth $\sqrt{T} \cdot \operatorname{poly}(\lambda)$. Then, each of
the $\sqrt{T}$ iterative steps takes $\log T \cdot \operatorname{poly}(\lambda)$ depth. The entire attack therefore takes $\sqrt{T} \log T \cdot \operatorname{poly}(\lambda)$ depth.

We remark that the above attack seems to rely on the sparsity of the image

$$
\left(\begin{array}{lllll}
-\mathbf{x}_{0}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} & \ldots & \mathbf{0}^{\mathrm{T}} & \mathbf{x}_{T}^{\mathrm{T}}
\end{array}\right)^{\mathrm{T}}
$$

and in particular that it can be decomposed into

$$
\sum_{\ell=1}^{\sqrt{T}}\left(\begin{array}{lllllllllll}
\mathbf{0}^{\mathrm{T}} & \ldots & \mathbf{0}^{\mathrm{T}} & -\mathbf{x}_{(\ell-1) \sqrt{T}}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} & \ldots & \mathbf{0}^{\mathrm{T}} & \mathbf{x}_{\ell \sqrt{T}}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} & \ldots & \mathbf{0}^{\mathrm{T}}
\end{array}\right)^{\mathrm{T}}
$$

Preimage Sampling Attack. We summarise a family of preimage sampling attacks by Peikert and Tang [PT23] which, given an instance (A, x), finds a solution $(\mathbf{y}, \mathbf{u})$ with $\mathbf{u} \in \mathcal{R}_{\left(\gamma_{\mathcal{R}} m\right)^{m T}{ }^{\left(\log _{k} T\right) / 2} \text { in }}$ depth $k \log _{k} T \cdot \operatorname{poly}(\lambda)$, for any $1 \leq k \leq T$. Note that the family of attacks is parametrised by a parameter $k$ which describes a tradeoff between the parallel runtime and the norm of the solution. In particular, when $k=\sqrt{T}$, this attack has the same performance as that of Attema and Ducas.

In essence, the attack works by first computing a trapdoor of the matrix

$$
\mathbf{B}_{T}:=\underbrace{\left(\begin{array}{ccccc}
\mathbf{G} & & & & \\
\mathbf{A} & \mathbf{G} & & & \\
& \ddots & \ddots & & \\
& & \ddots & \ddots & \\
& & & \mathbf{A} & \mathbf{G}
\end{array}\right)}_{T \text { columns }}
$$

and then using the trapdoor to sample a short preimage $\mathbf{u}^{\mathrm{T}}=\left(\mathbf{u}_{0}^{\mathrm{T}}, \ldots, \mathbf{u}_{T-1}^{\mathrm{T}}\right)$ such that

$$
\mathbf{B}_{T} \mathbf{u}=\left(-\mathbf{x}_{0}^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}, \ldots, \boldsymbol{0}^{\mathrm{T}}\right)^{\mathrm{T}} \bmod q
$$

The output is then $(\mathbf{y}, \mathbf{u})$ where $\mathbf{y}:=\mathbf{A} \mathbf{u}_{T-1} \bmod q$.
The computation of a trapdoor of $\mathbf{B}_{T}$ is based on the following observation. For $i \in[k]$, let $\mathbf{C}_{i} \in \mathcal{R}_{q}^{n t_{i} \times m t_{i}}$ be matrices with trapdoors $\mathbf{R}_{i} \in \mathcal{R}^{m t_{i} \times m t_{i}}$ so that $\mathbf{C}_{i} \mathbf{R}_{i}=\mathbf{I}_{t_{i}} \otimes \mathbf{G} \bmod q$. For $i, j \in[k]$ with $i>j$, let $\mathbf{W}_{i, j} \in \mathcal{R}_{q}^{n t_{i} \times m t_{j}}$ be arbitrary matrices. Define

$$
\mathbf{C}:=\left(\begin{array}{ccccc}
\mathbf{C}_{1} & & & \\
\mathbf{W}_{2,1} & \mathbf{C}_{2} & & \\
\vdots & \vdots & \ddots & \\
& \ddots & \\
\mathbf{W}_{k, 1} & \mathbf{W}_{k, 2} & \cdots \cdots & \mathbf{C}_{k}
\end{array}\right)
$$

Observe that the matrix

$$
\mathbf{R}:=\left(\begin{array}{ccccc}
\mathbf{R}_{1} & & & & \\
& \mathbf{R}_{2} & & & \\
& & \ddots & \\
& & & \ddots & \mathbf{R}_{k}
\end{array}\right)\left(\begin{array}{cccccc}
\mathbf{I}_{m t_{1}} & & & \\
\mathbf{S}_{2,1} & \mathbf{I}_{m t_{2}} & & \\
\vdots & \vdots & \ddots & \\
& & & & \ddots & \\
\mathbf{S}_{k, 1} & \mathbf{S}_{k, 2} & \ldots & \mathbf{I}_{m t_{k}}
\end{array}\right)=\left(\begin{array}{cccc}
\mathbf{R}_{1} & & & \\
\mathbf{R}_{2} \mathbf{S}_{2,1} & \mathbf{R}_{2} & & \\
\vdots & \vdots & \ddots & \\
\mathbf{R}_{k} \mathbf{S}_{k, 1} & \mathbf{R}_{k} \mathbf{S}_{k, 2} & \ldots & \ddots
\end{array}\right)
$$

where

$$
\mathbf{S}_{i, j}:=-\left(\mathbf{I}_{t_{i}} \otimes \mathbf{G}\right)^{-1}\left(\mathbf{W}_{i, j} \mathbf{R}_{j}+\sum_{j<j^{\prime}<i} \mathbf{W}_{i, j^{\prime}} \mathbf{R}_{j^{\prime}} \mathbf{S}_{j^{\prime}, j}\right)
$$

is a trapdoor of $\mathbf{C}$, i.e. $\mathbf{C R}=\mathbf{I}_{\sum_{i=1}^{k} t_{i}} \otimes \mathbf{G} \bmod q$.
Given the above observation, the idea is to recursively decompose $\mathbf{B}_{T}$ into $k$-by- $k$ block matrices until arriving at the block matrix

$$
\mathbf{B}_{k}=\underbrace{\left(\begin{array}{ccccc}
\mathbf{G} & & & & \\
\mathbf{A} & \mathbf{G} & & & \\
& \ddots & \ddots & & \\
& & \ddots & \ddots & \\
& & & \mathbf{A} & \mathbf{G}
\end{array}\right)}_{k \text { columns }}
$$

whose diagonal blocks admit trivial trapdoors, i.e. $\mathbf{G I}_{m}=\mathbf{G}$.
We remark that the above attack, as pointed out by Peikert and Tang [PT23], is oblivious (up to minor optimisation) to the form of the image

$$
\left(\begin{array}{llll}
-\mathrm{x}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} & \ldots & \mathbf{0}^{\mathrm{T}}
\end{array}\right)^{\mathrm{T}} .
$$

## B 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 6.2. A recent work [LMS22] 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 [LMS22] for precise definitions of the notions used here. First we recall below the notion of collapsing for hash functions [Unr16].

Definition B. 1 (Collapsing). Let H be a (keyed) hash function. We say that H is collapsing if for any efficient (quantum) adversary $\mathcal{A}$

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

where the experiment Collapsing ${ }_{\mathcal{A}}^{b}$ is defined as follows:
Collapsing ${ }_{\mathcal{A}}^{b}\left(1^{\lambda}\right)$ :

- 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 $\mathrm{H}_{k}(\cdot)$ equals $y$ and the input belongs to the appropriate domain. Apply $U_{k, y}$, measure the ancilla, and apply $U_{k, y}^{\dagger}$.
- 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 [LMS22] it is shown that the SIS-based hash function is collapsing, assuming the hardness of the LWE problem.

Lemma B. 2 ([LMS22]). If the LWE problem is hard, then the function $\mathrm{H}_{\mathbf{A}}$ defined as

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

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 B. 3 (Collapsing). Let $t$ be a polynomial in the security parameter. If the LWE problem is hard, then the function $H_{A}$ defined as

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

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 $\mathbf{u} \in \mathcal{R}_{p}^{t m}$ in $t$ blocks $\left(\mathbf{u}_{1}, \ldots, \mathbf{u}_{t}\right)$ 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 Collapsing ${ }_{\mathcal{A}}^{1}$ ). 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 Collapsing $\mathcal{A}_{\mathcal{A}}$. 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 B.2). For other hybrids, it suffices to observe that the $i$-th block of the output $\mathbf{y}=\left(\mathbf{y}_{1}, \ldots, \mathbf{y}_{t}\right)$ 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 B.2, 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).


[^0]:    ${ }^{1}$ 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.

[^1]:    ${ }^{2}$ Note that this exact version of the conjecture is unaffected by known attacks [AD23, PT23].

[^2]:    ${ }^{3}$ Although the signature scheme presented in [CLMQ21] 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.

