Cryptanalysis of Lattice-Based Sequentiality Assumptions and Proofs of Sequential Work

Chris Peikert, *Yi Tang*

Background

Proof of sequential work (PoSW):

- A basic *timed cryptography* primitive [RivestShamirWagner96].
- Prover runs an *inherently sequential* process of depth (parallel time) T.
- Prover convinces a weak verifier with *low running time*, e.g., $O(\log T)$.
- Convincing the verifier should require prover depth $\approx T$.
- Application: energy conservation in blockchains.

Post-quantum PoSW:

- Most prior constructions, from e.g. factoring, are broken by quantum computers.
- Lai and Malavolta (Crypto 2023) give a lattice-based PoSW candidate.

In this work, we break the LM23 assumption, and (almost) the PoSW as well!

Background

Proof of sequential work (PoSW):

- A basic *timed cryptography* primitive [RivestShamirWagner96].
- Prover runs an *inherently sequential* process of depth (parallel time) T.
- Prover convinces a weak verifier with *low running time*, e.g., $O(\log T)$.
- Convincing the verifier should require prover depth $\approx T$.
- Application: energy conservation in blockchains.

Post-quantum PoSW:

- Most prior constructions, from e.g. factoring, are broken by quantum computers.
- Lai and Malavolta (Crypto 2023) give a lattice-based PoSW candidate.

In this work, we break the LM23 assumption, and (almost) the PoSW as well!

Background

Proof of sequential work (PoSW):

- A basic *timed cryptography* primitive [RivestShamirWagner96].
- Prover runs an *inherently sequential* process of depth (parallel time) T.
- Prover convinces a weak verifier with *low running time*, e.g., $O(\log T)$.
- Convincing the verifier should require prover depth $\approx T$.
- Application: energy conservation in blockchains.

Post-quantum PoSW:

- Most prior constructions, from e.g. factoring, are broken by quantum computers.
- Lai and Malavolta (Crypto 2023) give a lattice-based PoSW candidate.

In this work, we break the LM23 assumption, and (almost) the PoSW as well!

Background

Proof of sequential work (PoSW):

- A basic *timed cryptography* primitive [RivestShamirWagner96].
- Prover runs an *inherently sequential* process of depth (parallel time) T.
- Prover convinces a weak verifier with *low running time*, e.g., $O(\log T)$.
- Convincing the verifier should require prover depth $\approx T$.
- Application: energy conservation in blockchains.

Post-quantum PoSW:

- Most prior constructions, from e.g. factoring, are broken by quantum computers.
- Lai and Malavolta (Crypto 2023) give a lattice-based PoSW candidate.

In this work, we break the LM23 assumption, and (almost) the PoSW as well!

Background

Proof of sequential work (PoSW):

- A basic *timed cryptography* primitive [RivestShamirWagner96].
- Prover runs an *inherently sequential* process of depth (parallel time) T.
- Prover convinces a weak verifier with *low running time*, e.g., $O(\log T)$.
- Convincing the verifier should require prover depth $\approx T$.
- Application: energy conservation in blockchains.

Post-quantum PoSW:

- Most prior constructions, from e.g. factoring, are broken by quantum computers.
- Lai and Malavolta (Crypto 2023) give a lattice-based PoSW candidate.

In this work, we break the LM23 assumption, and (almost) the PoSW as well!

Background

Proof of sequential work (PoSW):

- A basic *timed cryptography* primitive [RivestShamirWagner96].
- Prover runs an *inherently sequential* process of depth (parallel time) T.
- Prover convinces a weak verifier with *low running time*, e.g., $O(\log T)$.
- Convincing the verifier should require prover depth $\approx T$.
- Application: energy conservation in blockchains.

Post-quantum PoSW:

- Most prior constructions, from e.g. factoring, are broken by quantum computers.
- Lai and Malavolta (Crypto 2023) give a lattice-based PoSW candidate.

In this work, we break the LM23 assumption, and (almost) the PoSW as well!

Background

Proof of sequential work (PoSW):

- A basic *timed cryptography* primitive [RivestShamirWagner96].
- Prover runs an *inherently sequential* process of depth (parallel time) T.
- Prover convinces a weak verifier with *low running time*, e.g., $O(\log T)$.
- Convincing the verifier should require prover depth $\approx T$.
- Application: energy conservation in blockchains.

Post-quantum PoSW:

- Most prior constructions, from e.g. factoring, are broken by quantum computers.
- Lai and Malavolta (Crypto 2023) give a lattice-based PoSW candidate.

In this work, we break the LM23 assumption, and (almost) the PoSW as well!

Background

Proof of sequential work (PoSW):

- A basic *timed cryptography* primitive [RivestShamirWagner96].
- Prover runs an *inherently sequential* process of depth (parallel time) T.
- Prover convinces a weak verifier with *low running time*, e.g., $O(\log T)$.
- Convincing the verifier should require prover depth $\approx T$.
- Application: energy conservation in blockchains.

Post-quantum PoSW:

- Most prior constructions, from e.g. factoring, are broken by quantum computers.
- Lai and Malavolta (Crypto 2023) give a lattice-based PoSW candidate.

In this work, we break the LM23 assumption, and (almost) the PoSW as well!

Our Results

LM23 PoSW

Assuming *sequential SIS* with norm bound $\approx n^{2 \log T}$ requires depth $\approx T$ to solve, there exists a PoSW that requires prover depth $\approx T$.

Breaking the LM23 sequentiality assumption

Sequential SIS with norm bound $\approx n^{2 \log T}$ can be solved in depth $\tilde{O}_{n,q}(\log T)$.

Moreover, a depth-norm tradeoff breaks a wide range of parameters.

Breaking the LM23 PoSW*

The LM23 PoSW* can be broken in depth $\tilde{O}_{n,q}(\log^2 T)$.

* An essentially identical variant, differing from the original PoSW in only an arbitrary choice that is immaterial to the design and security proof.

Our Results

LM23 PoSW

Assuming *sequential SIS* with norm bound $\approx n^{2 \log T}$ requires depth $\approx T$ to solve, there exists a PoSW that requires prover depth $\approx T$.

Breaking the LM23 sequentiality assumption

Sequential SIS with norm bound $\approx n^{2 \log T}$ can be solved in depth $\tilde{O}_{n,q}(\log T)$.

Moreover, a depth-norm tradeoff breaks a wide range of parameters.

Breaking the LM23 PoSW*

The LM23 PoSW* can be broken in depth $\tilde{O}_{n,q}(\log^2 T)$.

* An essentially identical variant, differing from the original PoSW in only an arbitrary choice that is immaterial to the design and security proof.

Our Results

LM23 PoSW

Assuming *sequential SIS* with norm bound $\approx n^{2\log T}$ requires depth $\approx T$ to solve, there exists a PoSW that requires prover depth $\approx T$.

Breaking the LM23 sequentiality assumption

Sequential SIS with norm bound $\approx n^{2\log T}$ can be solved in depth $\tilde{O}_{n,q}(\log T)$.

Moreover, a depth-norm tradeoff breaks a wide range of parameters.

$\tilde{O}_{n,q}$ hides polylog(n, q).

Breaking the LM23 PoSW*

The LM23 PoSW* can be broken in depth $\tilde{O}_{n,q}(\log^2 T)$.

---

*An essentially identical variant, differing from the original PoSW in only an arbitrary choice that is immaterial to the design and security proof.

Our Results

LM23 PoSW

Assuming *sequential SIS* with norm bound $\approx n^{2\log T}$ requires depth $\approx T$ to solve, there exists a PoSW that requires prover depth $\approx T$.

Breaking the LM23 sequentiality assumption

Sequential SIS with norm bound $\approx n^{2\log T}$ can be solved in depth $\tilde{O}_{n,q}(\log T)$.

Moreover, a depth-norm tradeoff breaks a wide range of parameters.

Breaking the LM23 PoSW*

The LM23 PoSW* can be broken in depth $\tilde{O}_{n,q}(\log^2 T)$.

*An essentially identical variant, differing from the original PoSW in only an arbitrary choice that is immaterial to the design and security proof.

Our Results

LM23 PoSW

Assuming *sequential SIS* with norm bound $\approx n^{2\log T}$ requires depth $\approx T$ to solve, there exists a PoSW that requires prover depth $\approx T$.

Breaking the LM23 sequentiality assumption

Sequential SIS with norm bound $\approx n^{2\log T}$ can be solved in depth $\tilde{O}_{n,q}(\log T)$.

Moreover, a depth-norm tradeoff breaks a wide range of parameters.

Breaking the LM23 PoSW*

The LM23 PoSW* can be broken in depth $\tilde{O}_{n,q}(\log^2 T)$.

*An essentially identical variant, differing from the original PoSW in only an arbitrary choice that is immaterial to the design and security proof.

Sequential Work in LM23

The sequential work: SIS hash $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A} \cdot \mathbf{x}$ iterated T times.

- $f_{\mathbf{A}} : \{0,1\}^m \to \mathbb{Z}_q^n$.
- To iterate, need to map $\mathbb{Z}_q^n \to \{0,1\}^m$.
- Bit expansion $\mathbf{G}_n^{-1}$: replace each $\mathbb{Z}_q$ entry by $\ell := \lceil \log_2 q \rceil$ bits. (So set $m = n \cdot \ell$.)
- "Gadget" vector $\mathbf{g} = (1, 2, \ldots, 2^{\ell-1})$, matrix $\mathbf{G}_n = \mathbf{I}_n \otimes \mathbf{g}$: satisfies $\mathbf{G}_n \cdot \mathbf{G}_n^{-1}(\mathbf{u}) = \mathbf{u}$ for any $\mathbf{u} \in \mathbb{Z}_q^n$.
- Start with given $\mathbf{A}, \mathbf{u}_0$ and output $\mathbf{u}_T$.

Sequential Work in LM23

The sequential work: SIS hash $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A} \cdot \mathbf{x}$ iterated T times.

- $f_{\mathbf{A}} \colon \{0,1\}^m \to \mathbb{Z}_q^n$.
- To iterate, need to map $\mathbb{Z}_q^n \to \{0,1\}^m$.
- Bit expansion $\mathbf{G}_n^{-1}$: replace each $\mathbb{Z}_q$ entry by $\ell := \lceil \log_2 q \rceil$ bits. (So set $m = n \cdot \ell$.)
- "Gadget" vector $\mathbf{g} = (1, 2, \ldots, 2^{\ell-1})$, matrix $\mathbf{G}_n = \mathbf{I}_n \otimes \mathbf{g}$: satisfies $\mathbf{G}_n \cdot \mathbf{G}_n^{-1}(\mathbf{u}) = \mathbf{u}$ for any $\mathbf{u} \in \mathbb{Z}_q^n$.
- Start with given $\mathbf{A}, \mathbf{u}_0$ and output $\mathbf{u}_T$.

Sequential Work in LM23

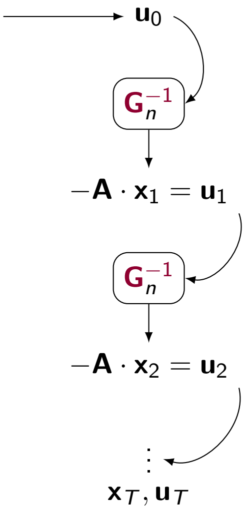
The sequential work: SIS hash $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A} \cdot \mathbf{x}$ iterated T times.

- $f_{\mathbf{A}} \colon \{0,1\}^m \to \mathbb{Z}_q^n$.
- To iterate, need to map $\mathbb{Z}_q^n \to \{0,1\}^m$.
- Bit expansion $\mathbf{G}_n^{-1}$: replace each $\mathbb{Z}_q$ entry by $\ell := \lceil \log_2 q \rceil$ bits. (So set $m = n \cdot \ell$.)
- "Gadget" vector $\mathbf{g} = (1, 2, \ldots, 2^{\ell-1})$, matrix $\mathbf{G}_n = \mathbf{I}_n \otimes \mathbf{g}$: satisfies $\mathbf{G}_n \cdot \mathbf{G}_n^{-1}(\mathbf{u}) = \mathbf{u}$ for any $\mathbf{u} \in \mathbb{Z}_q^n$.
- Start with given $\mathbf{A}, \mathbf{u}_0$ and output $\mathbf{u}_T$.

Sequential Work in LM23

The sequential work: SIS hash $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A} \cdot \mathbf{x}$ iterated T times.

- $f_{\mathbf{A}} \colon \{0,1\}^m \to \mathbb{Z}_q^n$.
- To iterate, need to map $\mathbb{Z}_q^n \to \{0,1\}^m$.
- Bit expansion $\mathbf{G}_n^{-1}$: replace each $\mathbb{Z}_q$ entry by $\ell := \lceil \log_2 q \rceil$ bits. (So set $m = n \cdot \ell$.)
- "Gadget" vector $\mathbf{g} = (1, 2, \ldots, 2^{\ell-1})$, matrix $\mathbf{G}_n = \mathbf{I}_n \otimes \mathbf{g}$: satisfies $\mathbf{G}_n \cdot \mathbf{G}_n^{-1}(\mathbf{u}) = \mathbf{u}$ for any $\mathbf{u} \in \mathbb{Z}_q^n$.
- Start with given $\mathbf{A}, \mathbf{u}_0$ and output $\mathbf{u}_T$.



Sequential SIS Problem

$$\mathbf{u}_0 \quad \Rightarrow \cdots \Rightarrow \quad \mathbf{x}_i = \mathbf{G}_n^{-1}(\mathbf{u}_{i-1}) \,,\ \mathbf{u}_i = -\mathbf{A} \cdot \mathbf{x}_i \quad \Rightarrow \cdots \Rightarrow \quad \mathbf{x}_T, \mathbf{u}_T \,.$$

The sequential work can be expressed via a linear system:

$$\underbrace{\begin{pmatrix} \mathbf{G}_n & & & & \\ \mathbf{A} & \mathbf{G}_n & & & \\ & \mathbf{A} & \ddots & & \\ & & \ddots & \mathbf{G}_n & \\ & & & \mathbf{A} & \mathbf{G}_n \\ & & & & \mathbf{A} \end{pmatrix}}_{\mathbf{A}_T \text{ or } \mathbf{A}_T} \cdot \underbrace{\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_T \end{pmatrix}}_{\mathbf{x} \in \mathbb{Z}^{Tm}} = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ -\mathbf{u}_T \end{pmatrix} .$$

Sequential Short Integer Solution (SIS) Problem

Sequential SIS with norm bound B is the (average-case) problem where:

- an instance consists of $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and $\mathbf{u}_0 \leftarrow \mathbb{Z}_q^n$, and
- the goal is to find $\mathbf{x} \in \mathbb{Z}^{Tm}$ with $\|\mathbf{x}\|_\infty \le B$ such that $\mathbf{A}_T \cdot \mathbf{x} = \binom{\mathbf{u}_0}{\mathbf{0}}$.

Sequential SIS Problem

$$\mathbf{u}_0 \quad \Rightarrow \cdots \Rightarrow \quad \mathbf{x}_i = \mathbf{G}_n^{-1}(\mathbf{u}_{i-1})\ ,\ \mathbf{u}_i = -\mathbf{A}\cdot\mathbf{x}_i \quad \Rightarrow \cdots \Rightarrow \quad \mathbf{x}_T, \mathbf{u}_T\ .$$

The sequential work can be expressed via a linear system:

$$\underbrace{\begin{pmatrix} \mathbf{G}_n & & & & \\ \mathbf{A} & \mathbf{G}_n & & & \\ & \mathbf{A} & \ddots & & \\ & & \ddots & \mathbf{G}_n & \\ & & & \mathbf{A} & \mathbf{G}_n \\ & & & & \mathbf{A} \end{pmatrix}}_{\mathbf{A}_T \text{ or } \mathbf{A}_T} \cdot \underbrace{\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_T \end{pmatrix}}_{\mathbf{x}\in\mathbb{Z}^{Tm}} = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ -\mathbf{u}_T \end{pmatrix}.$$

Sequential Short Integer Solution (SIS) Problem

Sequential SIS with norm bound B is the (average-case) problem where:

- an instance consists of $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$ and $\mathbf{u}_0 \leftarrow \mathbb{Z}_q^n$, and
- the goal is to find $\mathbf{x} \in \mathbb{Z}^{Tm}$ with $\|\mathbf{x}\|_\infty \le B$ such that $\mathbf{A}_T \cdot \mathbf{x} = \binom{\mathbf{u}_0}{\mathbf{0}}$.

Sequential SIS Problem

$$\mathbf{u}_0 \quad \Rightarrow \cdots \Rightarrow \quad \mathbf{x}_i = \mathbf{G}_n^{-1}(\mathbf{u}_{i-1}) \;,\; \mathbf{u}_i = -\mathbf{A} \cdot \mathbf{x}_i \quad \Rightarrow \cdots \Rightarrow \quad \mathbf{x}_T, \mathbf{u}_T \;.$$

The sequential work can be expressed via a linear system:

$$\underbrace{\begin{pmatrix} \mathbf{G}_n & & & & \\ \mathbf{A} & \mathbf{G}_n & & & \\ & \mathbf{A} & \ddots & & \\ & & \ddots & \mathbf{G}_n & \\ & & & \mathbf{A} & \mathbf{G}_n \\ & & & & \mathbf{A} \end{pmatrix}}_{\mathbf{A}_T \text{ or } \mathbf{A}_T} \cdot \underbrace{\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_T \end{pmatrix}}_{\mathbf{x} \in \mathbb{Z}^{Tm}} = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ -\mathbf{u}_T \end{pmatrix} .$$

Sequential Short Integer Solution (SIS) Problem

Sequential SIS with norm bound B is the (average-case) problem where:

- an instance consists of $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and $\mathbf{u}_0 \leftarrow \mathbb{Z}_q^n$, and
- the goal is to find $\mathbf{x} \in \mathbb{Z}^{Tm}$ with $\|\mathbf{x}\|_\infty \leq B$ such that $\mathbf{A}_T \cdot \mathbf{x} = \binom{\mathbf{u}_0}{\mathbf{0}}$.

Sequential SIS Problem

$$\mathbf{u}_0 \quad \Rightarrow \cdots \Rightarrow \quad \mathbf{x}_i = \mathbf{G}_n^{-1}(\mathbf{u}_{i-1})\ ,\ \mathbf{u}_i = -\mathbf{A}\cdot\mathbf{x}_i \quad \Rightarrow \cdots \Rightarrow \quad \mathbf{x}_T, \mathbf{u}_T\ .$$

The sequential work can be expressed via a linear system:

$$\underbrace{\begin{pmatrix} \mathbf{G}_n & & & & \\ \mathbf{A} & \mathbf{G}_n & & & \\ & \mathbf{A} & \ddots & & \\ & & \ddots & \mathbf{G}_n & \\ & & & \mathbf{A} & \mathbf{G}_n \\ & & & & \mathbf{A} \end{pmatrix}}_{\mathbf{A}_T \text{ or } \mathbf{A}_T} \cdot \underbrace{\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_T \end{pmatrix}}_{\mathbf{x}\in\mathbb{Z}^{Tm}} = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ -\mathbf{u}_T \end{pmatrix}.$$

Sequential Short Integer Solution (SIS) Problem

Sequential SIS with norm bound B is the (average-case) problem where:

- an instance consists of $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$ and $\mathbf{u}_0 \leftarrow \mathbb{Z}_q^n$, and
- the goal is to find $\mathbf{x} \in \mathbb{Z}^{Tm}$ with $\|\mathbf{x}\|_\infty \leq B$ such that $\mathbf{A}_T \cdot \mathbf{x} = \binom{\mathbf{u}_0}{\mathbf{0}}$.

Sequential SIS Problem

$$\boxed{\mathbf{u}_0 \quad \Rightarrow \cdots \Rightarrow \quad \mathbf{x}_i = \mathbf{G}_n^{-1}(\mathbf{u}_{i-1}) \, , \; \mathbf{u}_i = -\mathbf{A} \cdot \mathbf{x}_i \quad \Rightarrow \cdots \Rightarrow \quad \mathbf{x}_T, \mathbf{u}_T \; .}$$

The sequential work can be expressed via a linear system:

$$\underbrace{\begin{pmatrix} \mathbf{G}_n & & & & \\ \mathbf{A} & \mathbf{G}_n & & & \\ & \mathbf{A} & \ddots & & \\ & & \ddots & \mathbf{G}_n & \\ & & & \mathbf{A} & \mathbf{G}_n \\ & & & & \mathbf{A} \end{pmatrix}}_{\mathbf{A}_T \text{ or } \mathbf{A}_T} \cdot \underbrace{\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_T \end{pmatrix}}_{\mathbf{x} \in \mathbb{Z}^{Tm}} = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ -\mathbf{u}_T \end{pmatrix} .$$

Sequential Short Integer Solution (SIS) Problem

Sequential SIS with norm bound B is the (average-case) problem where:

- an instance consists of $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and $\mathbf{u}_0 \leftarrow \mathbb{Z}_q^n$, and
- the goal is to find $\mathbf{x} \in \mathbb{Z}^{Tm}$ with $\|\mathbf{x}\|_\infty \leq B$ such that $\mathbf{A}_T \cdot \mathbf{x} = \binom{\mathbf{u}_0}{\mathbf{0}}$.

The LM23 PoSW

Goal: prove knowledge of a *short* solution to $\mathbf{A}_T \cdot \mathbf{x} = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ -\mathbf{u}_T \end{pmatrix}$ to a *weak* verifier.

The LM23 PoSW takes a standard "divide and fold" approach.

- Assume for simplicity that $T = 2T' + 1$ is odd.
- $\mathbf{x}$ splits into $\mathbf{x}^t = (\mathbf{x}_1; \ldots; \mathbf{x}_{T'}), \mathbf{x}_{T'+1}, \mathbf{x}^b = (\mathbf{x}_{T'+2}; \ldots; \mathbf{x}_T)$, and correspondingly:

$$\begin{pmatrix} \mathbf{A}_{T'} & & \\ & \begin{matrix} \mathbf{G}_n \\ \mathbf{A} \end{matrix} & \\ & & \mathbf{A}_{T'} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x}^t \\ \mathbf{x}_{T'+1} \\ \mathbf{x}^b \end{pmatrix} = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ -\mathbf{u}_{T'} \\ \\ \\ \end{pmatrix} + \begin{pmatrix} \\ \mathbf{u}_{T'} \\ -\mathbf{u}_{T'+1} \\ \\ \end{pmatrix} + \begin{pmatrix} \\ \\ \mathbf{u}_{T'+1} \\ \mathbf{0} \\ -\mathbf{u}_T \end{pmatrix}.$$

The LM23 PoSW

Goal: prove knowledge of a *short* solution to $\mathbf{A}_T \cdot \mathbf{x} = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ -\mathbf{u}_T \end{pmatrix}$ to a *weak* verifier.

The LM23 PoSW takes a standard "divide and fold" approach.

- Assume for simplicity that $T = 2T' + 1$ is odd.
- $\mathbf{x}$ splits into $\mathbf{x}^t = (\mathbf{x}_1; \ldots; \mathbf{x}_{T'}), \mathbf{x}_{T'+1}, \mathbf{x}^b = (\mathbf{x}_{T'+2}; \ldots; \mathbf{x}_T)$, and correspondingly:

$$\begin{pmatrix} \mathbf{A}_{T'} & & \\ & \begin{matrix} \mathbf{G}_n \\ \mathbf{A} \end{matrix} & \\ & & \mathbf{A}_{T'} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x}^t \\ \mathbf{x}_{T'+1} \\ \mathbf{x}^b \end{pmatrix} = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ -\mathbf{u}_{T'} \\ \\ \\ \end{pmatrix} + \begin{pmatrix} \\ \mathbf{u}_{T'} \\ -\mathbf{u}_{T'+1} \\ \\ \end{pmatrix} + \begin{pmatrix} \\ \\ \mathbf{u}_{T'+1} \\ \mathbf{0} \\ -\mathbf{u}_T \end{pmatrix}.$$

The LM23 PoSW

Goal: prove knowledge of a *short* solution to $\mathbf{A}_T \cdot \mathbf{x} = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ -\mathbf{u}_T \end{pmatrix}$ to a *weak* verifier.

The LM23 PoSW takes a standard "divide and fold" approach.

- Assume for simplicity that $T = 2T' + 1$ is odd.
- $\mathbf{x}$ splits into $\mathbf{x}^t = (\mathbf{x}_1; \ldots; \mathbf{x}_{T'}), \mathbf{x}_{T'+1}, \mathbf{x}^b = (\mathbf{x}_{T'+2}; \ldots; \mathbf{x}_T)$, and correspondingly:

$$\begin{pmatrix} \mathbf{A}_{T'} & & \\ & \begin{matrix} \mathbf{G}_n \\ \mathbf{A} \end{matrix} & \\ & & \mathbf{A}_{T'} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x}^t \\ \mathbf{x}_{T'+1} \\ \mathbf{x}^b \end{pmatrix} = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ -\mathbf{u}_{T'} \\ \\ \\ \end{pmatrix} + \begin{pmatrix} \\ \\ \mathbf{u}_{T'} \\ -\mathbf{u}_{T'+1} \\ \\ \end{pmatrix} + \begin{pmatrix} \\ \\ \\ \mathbf{u}_{T'+1} \\ \mathbf{0} \\ -\mathbf{u}_T \end{pmatrix}.$$

The LM23 PoSW

Goal: prove knowledge of a *short* solution to $\mathbf{A}_T \cdot \mathbf{x} = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ -\mathbf{u}_T \end{pmatrix}$ to a *weak* verifier.

The LM23 PoSW takes a standard "divide and fold" approach.

- Assume for simplicity that $T = 2T' + 1$ is odd.
- $\mathbf{x}$ splits into $\mathbf{x}^t = (\mathbf{x}_1; \ldots; \mathbf{x}_{T'}), \mathbf{x}_{T'+1}, \mathbf{x}^b = (\mathbf{x}_{T'+2}; \ldots; \mathbf{x}_T)$, and correspondingly:

$$\begin{pmatrix} \mathbf{A}_{T'} & & \\ & \mathbf{G}_n & \\ & \mathbf{A} & \\ & & \mathbf{A}_{T'} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x}^t \\ \mathbf{x}_{T'+1} \\ \mathbf{x}^b \end{pmatrix} = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ -\mathbf{u}_{T'} \\ \\ \\ \end{pmatrix} + \begin{pmatrix} \\ \mathbf{u}_{T'} \\ -\mathbf{u}_{T'+1} \\ \\ \end{pmatrix} + \begin{pmatrix} \\ \\ \mathbf{u}_{T'+1} \\ \mathbf{0} \\ -\mathbf{u}_T \end{pmatrix}.$$

The LM23 PoSW

Goal: prove knowledge of a *short* solution to $\mathbf{A}_T \cdot \mathbf{x} = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ -\mathbf{u}_T \end{pmatrix}$ to a *weak* verifier.

The LM23 PoSW takes a standard "divide and fold" approach.

- Assume for simplicity that $T = 2T' + 1$ is odd.
- $\mathbf{x}$ splits into $\mathbf{x}^t = (\mathbf{x}_1; \dots; \mathbf{x}_{T'}), \mathbf{x}_{T'+1}, \mathbf{x}^b = (\mathbf{x}_{T'+2}; \dots; \mathbf{x}_T)$, and correspondingly:

$$\begin{pmatrix} \mathbf{A}_{T'} & & \\ & \mathbf{G}_n & \\ & \mathbf{A} & \\ & & \mathbf{A}_{T'} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x}^t \\ \mathbf{x}_{T'+1} \\ \mathbf{x}^b \end{pmatrix} = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ -\mathbf{u}_{T'} \\ \\ \\ \\ \end{pmatrix} + \begin{pmatrix} \\ \\ \mathbf{u}_{T'} \\ -\mathbf{u}_{T'+1} \\ \\ \\ \end{pmatrix} + \begin{pmatrix} \\ \\ \\ \mathbf{u}_{T'+1} \\ \mathbf{0} \\ -\mathbf{u}_T \end{pmatrix}.$$

The LM23 PoSW

Goal: prove knowledge of a *short* solution to $\mathbf{A}_T \cdot \mathbf{x} = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ -\mathbf{u}_T \end{pmatrix}$ to a *weak* verifier.

The LM23 PoSW takes a standard "divide and fold" approach.

- Assume for simplicity that $T = 2T' + 1$ is odd.
- $\mathbf{x}$ splits into $\mathbf{x}^t = (\mathbf{x}_1; \ldots; \mathbf{x}_{T'}), \mathbf{x}_{T'+1}, \mathbf{x}^b = (\mathbf{x}_{T'+2}; \ldots; \mathbf{x}_T)$, and correspondingly:

$$\begin{pmatrix} \mathbf{A}_{T'} & & \\ & \begin{matrix} \mathbf{G}_n \\ \mathbf{A} \end{matrix} & \\ & & \mathbf{A}_{T'} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x}^t \\ \mathbf{x}_{T'+1} \\ \mathbf{x}^b \end{pmatrix} = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ -\mathbf{u}_{T'} \\ \\ \\ \end{pmatrix} + \begin{pmatrix} \\ \\ \mathbf{u}_{T'} \\ -\mathbf{u}_{T'+1} \\ \\ \end{pmatrix} + \begin{pmatrix} \\ \\ \\ \mathbf{u}_{T'+1} \\ \mathbf{0} \\ -\mathbf{u}_T \end{pmatrix}.$$

The LM23 PoSW*, Folding and Norm Bounds

$$\mathbf{A}_{T'} \cdot \mathbf{x}^t = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ -\mathbf{u}_{T'} \end{pmatrix}, \quad \begin{pmatrix} \mathbf{G}_n \\ \mathbf{A} \end{pmatrix} \cdot \mathbf{x}_{T'+1} = \begin{pmatrix} \mathbf{u}_{T'} \\ -\mathbf{u}_{T'+1} \end{pmatrix}, \quad \mathbf{A}_{T'} \cdot \mathbf{x}^b = \begin{pmatrix} \mathbf{u}_{T'+1} \\ \mathbf{0} \\ -\mathbf{u}_T \end{pmatrix}.$$

- Prover reveals $\mathbf{x}_{T'+1}$, and verifier checks that it is short.
- Verifier sends a random challenge c with $|c| \leq \gamma = \Omega(n)$.
- Prover and verifier fold by c as follows, and recurse to prove:

$$\mathbf{A}_{T'} \cdot \underbrace{(c \cdot \mathbf{x}^t + \mathbf{x}^b)}_{\mathbf{x}'} = \begin{pmatrix} \mathbf{u}'_0 \\ \mathbf{0} \\ -\mathbf{u}'_{T'} \end{pmatrix} = \begin{pmatrix} c \cdot \mathbf{u}_0 + \mathbf{u}_{T'+1} \\ \mathbf{0} \\ -(c \cdot \mathbf{u}_{T'} + \mathbf{u}_T) \end{pmatrix}.$$

Norm bounds:

- In each round, $\|\mathbf{x}\|$ grows by $\leq 2|c| \leq 2\gamma$, so the final norm bound is $(2\gamma)^{\log T}$.
- Reduction loses a similar factor, so is from sequential SIS with norm bound $(2\gamma)^{2 \log T}$.
- Our attacks crucially exploit the gap between these bounds and honest $\|\mathbf{x}\| = 1$.

The LM23 PoSW*, Folding and Norm Bounds

$$\mathbf{A}_{T'} \cdot \mathbf{x}^t = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ -\mathbf{u}_{T'} \end{pmatrix}, \quad \begin{pmatrix} \mathbf{G}_n \\ \mathbf{A} \end{pmatrix} \cdot \mathbf{x}_{T'+1} = \begin{pmatrix} \mathbf{u}_{T'} \\ -\mathbf{u}_{T'+1} \end{pmatrix}, \quad \mathbf{A}_{T'} \cdot \mathbf{x}^b = \begin{pmatrix} \mathbf{u}_{T'+1} \\ \mathbf{0} \\ -\mathbf{u}_T \end{pmatrix}.$$

- Prover reveals $\mathbf{x}_{T'+1}$, and verifier checks that it is short.
- Verifier sends a random challenge c with $|c| \leq \gamma = \Omega(n)$.
- Prover and verifier fold by c as follows, and recurse to prove:

$$\mathbf{A}_{T'} \cdot \underbrace{(c \cdot \mathbf{x}^t + \mathbf{x}^b)}_{\mathbf{x}'} = \begin{pmatrix} \mathbf{u}'_0 \\ \mathbf{0} \\ -\mathbf{u}'_{T'} \end{pmatrix} = \begin{pmatrix} c \cdot \mathbf{u}_0 + \mathbf{u}_{T'+1} \\ \mathbf{0} \\ -(c \cdot \mathbf{u}_{T'} + \mathbf{u}_T) \end{pmatrix}.$$

Norm bounds:

- In each round, $\|\mathbf{x}\|$ grows by $\leq 2|c| \leq 2\gamma$, so the final norm bound is $(2\gamma)^{\log T}$.
- Reduction loses a similar factor, so is from sequential SIS with norm bound $(2\gamma)^{2 \log T}$.
- Our attacks crucially exploit the gap between these bounds and honest $\|\mathbf{x}\| = 1$.

The LM23 PoSW*, Folding and Norm Bounds

$$\mathbf{A}_{T'} \cdot \mathbf{x}^t = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ -\mathbf{u}_{T'} \end{pmatrix}, \quad \begin{pmatrix} \mathbf{G}_n \\ \mathbf{A} \end{pmatrix} \cdot \mathbf{x}_{T'+1} = \begin{pmatrix} \mathbf{u}_{T'} \\ -\mathbf{u}_{T'+1} \end{pmatrix}, \quad \mathbf{A}_{T'} \cdot \mathbf{x}^b = \begin{pmatrix} \mathbf{u}_{T'+1} \\ \mathbf{0} \\ -\mathbf{u}_T \end{pmatrix}.$$

- Prover reveals $\mathbf{x}_{T'+1}$, and verifier checks that it is short.
- Verifier sends a random challenge c with $|c| \leq \gamma = \Omega(n)$.
- Prover and verifier fold by c as follows, and recurse to prove:

$$\mathbf{A}_{T'} \cdot \underbrace{(c \cdot \mathbf{x}^t + \mathbf{x}^b)}_{\mathbf{x}'} = \begin{pmatrix} \mathbf{u}'_0 \\ \mathbf{0} \\ -\mathbf{u}'_{T'} \end{pmatrix} = \begin{pmatrix} c \cdot \mathbf{u}_0 + \mathbf{u}_{T'+1} \\ \mathbf{0} \\ -(c \cdot \mathbf{u}_{T'} + \mathbf{u}_T) \end{pmatrix}.$$

Norm bounds:

- In each round, $\|\mathbf{x}\|$ grows by $\leq 2|c| \leq 2\gamma$, so the final norm bound is $(2\gamma)^{\log T}$.
- Reduction loses a similar factor, so is from sequential SIS with norm bound $(2\gamma)^{2 \log T}$.
- Our attacks crucially exploit the gap between these bounds and honest $\|\mathbf{x}\| = 1$.

The LM23 PoSW*, Folding and Norm Bounds

$$\mathbf{A}_{T'} \cdot \mathbf{x}^t = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ -\mathbf{u}_{T'} \end{pmatrix}, \quad \begin{pmatrix} \mathbf{G}_n \\ \mathbf{A} \end{pmatrix} \cdot \mathbf{x}_{T'+1} = \begin{pmatrix} \mathbf{u}_{T'} \\ -\mathbf{u}_{T'+1} \end{pmatrix}, \quad \mathbf{A}_{T'} \cdot \mathbf{x}^b = \begin{pmatrix} \mathbf{u}_{T'+1} \\ \mathbf{0} \\ -\mathbf{u}_T \end{pmatrix}.$$

- Prover reveals $\mathbf{x}_{T'+1}$, and verifier checks that it is short.
- Verifier sends a random challenge c with $|c| \leq \gamma = \Omega(n)$.
- Prover and verifier fold by c as follows, and recurse to prove:

$$\mathbf{A}_{T'} \cdot \underbrace{(c \cdot \mathbf{x}^t + \mathbf{x}^b)}_{\mathbf{x}'} = \begin{pmatrix} \mathbf{u}'_0 \\ \mathbf{0} \\ -\mathbf{u}'_{T'} \end{pmatrix} = \begin{pmatrix} c \cdot \mathbf{u}_0 + \mathbf{u}_{T'+1} \\ \mathbf{0} \\ -(c \cdot \mathbf{u}_{T'} + \mathbf{u}_T) \end{pmatrix}.$$

Norm bounds:

- In each round, $\|\mathbf{x}\|$ grows by $\leq 2|c| \leq 2\gamma$, so the final norm bound is $(2\gamma)^{\log T}$.
- Reduction loses a similar factor, so is from sequential SIS with norm bound $(2\gamma)^{2 \log T}$.
- Our attacks crucially exploit the gap between these bounds and honest $\|\mathbf{x}\| = 1$.

The LM23 PoSW*, Folding and Norm Bounds

$$\mathbf{A}_{T'} \cdot \mathbf{x}^t = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ -\mathbf{u}_{T'} \end{pmatrix}, \quad \begin{pmatrix} \mathbf{G}_n \\ \mathbf{A} \end{pmatrix} \cdot \mathbf{x}_{T'+1} = \begin{pmatrix} \mathbf{u}_{T'} \\ -\mathbf{u}_{T'+1} \end{pmatrix}, \quad \mathbf{A}_{T'} \cdot \mathbf{x}^b = \begin{pmatrix} \mathbf{u}_{T'+1} \\ \mathbf{0} \\ -\mathbf{u}_T \end{pmatrix}.$$

- Prover reveals $\mathbf{x}_{T'+1}$, and verifier checks that it is short.
- Verifier sends a random challenge c with $|c| \leq \gamma = \Omega(n)$.
- Prover and verifier fold by c as follows, and recurse to prove:

$$\mathbf{A}_{T'} \cdot \underbrace{(c \cdot \mathbf{x}^t + \mathbf{x}^b)}_{\mathbf{x}'} = \begin{pmatrix} \mathbf{u}'_0 \\ \mathbf{0} \\ -\mathbf{u}'_{T'} \end{pmatrix} = \begin{pmatrix} c \cdot \mathbf{u}_0 + \mathbf{u}_{T'+1} \\ \mathbf{0} \\ -(c \cdot \mathbf{u}_{T'} + \mathbf{u}_T) \end{pmatrix}.$$

* The original LM23 PoSW differs *only* by multiplying c to the second/bottom half.

Norm bounds:

- In each round, $\|\mathbf{x}\|$ grows by $\leq 2|c| \leq 2\gamma$, so the final norm bound is $(2\gamma)^{\log T}$.
- Reduction loses a similar factor, so is from sequential SIS with norm bound $(2\gamma)^{2\log T}$.
- Our attacks crucially exploit the gap between these bounds and honest $\|\mathbf{x}\| = 1$.

The LM23 PoSW*, Folding and Norm Bounds

$$\mathbf{A}_{T'} \cdot \mathbf{x}^t = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ -\mathbf{u}_{T'} \end{pmatrix}, \quad \begin{pmatrix} \mathbf{G}_n \\ \mathbf{A} \end{pmatrix} \cdot \mathbf{x}_{T'+1} = \begin{pmatrix} \mathbf{u}_{T'} \\ -\mathbf{u}_{T'+1} \end{pmatrix}, \quad \mathbf{A}_{T'} \cdot \mathbf{x}^b = \begin{pmatrix} \mathbf{u}_{T'+1} \\ \mathbf{0} \\ -\mathbf{u}_T \end{pmatrix}.$$

- Prover reveals $\mathbf{x}_{T'+1}$, and verifier checks that it is short.
- Verifier sends a random challenge c with $|c| \leq \gamma = \Omega(n)$.
- Prover and verifier fold by c as follows, and recurse to prove:

$$\mathbf{A}_{T'} \cdot \underbrace{(c \cdot \mathbf{x}^t + \mathbf{x}^b)}_{\mathbf{x}'} = \begin{pmatrix} \mathbf{u}'_0 \\ \mathbf{0} \\ -\mathbf{u}'_{T'} \end{pmatrix} = \begin{pmatrix} c \cdot \mathbf{u}_0 + \mathbf{u}_{T'+1} \\ \mathbf{0} \\ -(c \cdot \mathbf{u}_{T'} + \mathbf{u}_T) \end{pmatrix}.$$

Norm bounds:

- In each round, $\|\mathbf{x}\|$ grows by $\leq 2|c| \leq 2\gamma$, so the final norm bound is $(2\gamma)^{\log T}$.
- Reduction loses a similar factor, so is from sequential SIS with norm bound $(2\gamma)^{2 \log T}$.
- Our attacks crucially exploit the gap between these bounds and honest $\|\mathbf{x}\| = 1$.

The LM23 PoSW*, Folding and Norm Bounds

$$\mathbf{A}_{T'} \cdot \mathbf{x}^t = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ -\mathbf{u}_{T'} \end{pmatrix}, \quad \begin{pmatrix} \mathbf{G}_n \\ \mathbf{A} \end{pmatrix} \cdot \mathbf{x}_{T'+1} = \begin{pmatrix} \mathbf{u}_{T'} \\ -\mathbf{u}_{T'+1} \end{pmatrix}, \quad \mathbf{A}_{T'} \cdot \mathbf{x}^b = \begin{pmatrix} \mathbf{u}_{T'+1} \\ \mathbf{0} \\ -\mathbf{u}_T \end{pmatrix}.$$

- Prover reveals $\mathbf{x}_{T'+1}$, and verifier checks that it is short.
- Verifier sends a random challenge c with $|c| \leq \gamma = \Omega(n)$.
- Prover and verifier fold by c as follows, and recurse to prove:

$$\mathbf{A}_{T'} \cdot \underbrace{(c \cdot \mathbf{x}^t + \mathbf{x}^b)}_{\mathbf{x}'} = \begin{pmatrix} \mathbf{u}'_0 \\ \mathbf{0} \\ -\mathbf{u}'_{T'} \end{pmatrix} = \begin{pmatrix} c \cdot \mathbf{u}_0 + \mathbf{u}_{T'+1} \\ \mathbf{0} \\ -(c \cdot \mathbf{u}_{T'} + \mathbf{u}_T) \end{pmatrix}.$$

Norm bounds:

- In each round, $\|\mathbf{x}\|$ grows by $\leq 2|c| \leq 2\gamma$, so the final norm bound is $(2\gamma)^{\log T}$.
- Reduction loses a similar factor, so is from sequential SIS with norm bound $(2\gamma)^{2 \log T}$.
- Our attacks crucially exploit the gap between these bounds and honest $\|\mathbf{x}\| = 1$.

The LM23 PoSW*, Folding and Norm Bounds

$$\mathbf{A}_{T'} \cdot \mathbf{x}^t = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ -\mathbf{u}_{T'} \end{pmatrix}, \quad \begin{pmatrix} \mathbf{G}_n \\ \mathbf{A} \end{pmatrix} \cdot \mathbf{x}_{T'+1} = \begin{pmatrix} \mathbf{u}_{T'} \\ -\mathbf{u}_{T'+1} \end{pmatrix}, \quad \mathbf{A}_{T'} \cdot \mathbf{x}^b = \begin{pmatrix} \mathbf{u}_{T'+1} \\ \mathbf{0} \\ -\mathbf{u}_T \end{pmatrix}.$$

- Prover reveals $\mathbf{x}_{T'+1}$, and verifier checks that it is short.
- Verifier sends a random challenge c with $|c| \leq \gamma = \Omega(n)$.
- Prover and verifier fold by c as follows, and recurse to prove:

$$\mathbf{A}_{T'} \cdot \underbrace{(c \cdot \mathbf{x}^t + \mathbf{x}^b)}_{\mathbf{x}'} = \begin{pmatrix} \mathbf{u}'_0 \\ \mathbf{0} \\ -\mathbf{u}'_{T'} \end{pmatrix} = \begin{pmatrix} c \cdot \mathbf{u}_0 + \mathbf{u}_{T'+1} \\ \mathbf{0} \\ -(c \cdot \mathbf{u}_{T'} + \mathbf{u}_T) \end{pmatrix}.$$

Norm bounds:

- In each round, $\|\mathbf{x}\|$ grows by $\leq 2|c| \leq 2\gamma$, so the final norm bound is $(2\gamma)^{\log T}$.
- Reduction loses a similar factor, so is from sequential SIS with norm bound $(2\gamma)^{2 \log T}$.
- Our attacks crucially exploit the gap between these bounds and honest $\|\mathbf{x}\| = 1$.

Our Attacks, High-level Idea

We construct a "somewhat short" [MP12]-style *trapdoor* $\mathbf{R}$ for $\mathbf{A}_T$ such that

$$\mathbf{A}_T \cdot \mathbf{R} = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \end{pmatrix} .$$

We construct $\mathbf{R}$ in a recursive "divide and conquer" manner so that it takes low depth!

With such $\mathbf{R}$, we then compute a similarly short $\mathbf{x} = \mathbf{R} \cdot \mathbf{G}_n^{-1}(\mathbf{u}_0)$, which satisfies

$$\mathbf{A}_T \cdot \mathbf{x} = \mathbf{A}_T \cdot \mathbf{R} \cdot \mathbf{G}_n^{-1}(\mathbf{u}_0) = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \end{pmatrix} \cdot \mathbf{G}_n^{-1}(\mathbf{u}_0) = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \end{pmatrix} .$$

This directly solves sequential SIS for a wide range of parameters, including LM23.

To break the LM23 PoSW*, we similarly compute a solution $\mathbf{x}$ that interacts well with the folding, and simply run the honest prover with it.

Our Attacks, High-level Idea

We construct a "somewhat short" [MP12]-style *trapdoor* $\mathbf{R}$ for $\mathbf{A}_T$ such that

$$\mathbf{A}_T \cdot \mathbf{R} = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \end{pmatrix} .$$

We construct $\mathbf{R}$ in a recursive "divide and conquer" manner so that it takes low depth!

With such $\mathbf{R}$, we then compute a similarly short $\mathbf{x} = \mathbf{R} \cdot \mathbf{G}_n^{-1}(\mathbf{u}_0)$, which satisfies

$$\mathbf{A}_T \cdot \mathbf{x} = \mathbf{A}_T \cdot \mathbf{R} \cdot \mathbf{G}_n^{-1}(\mathbf{u}_0) = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \end{pmatrix} \cdot \mathbf{G}_n^{-1}(\mathbf{u}_0) = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \end{pmatrix} .$$

This directly solves sequential SIS for a wide range of parameters, including LM23.

To break the LM23 PoSW*, we similarly compute a solution $\mathbf{x}$ that interacts well with the folding, and simply run the honest prover with it.

Our Attacks, High-level Idea

We construct a "somewhat short" [MP12]-style *trapdoor* $\mathbf{R}$ for $\mathbf{A}_T$ such that

$$\mathbf{A}_T \cdot \mathbf{R} = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \end{pmatrix} .$$

We construct $\mathbf{R}$ in a recursive "divide and conquer" manner so that it takes low depth!

With such $\mathbf{R}$, we then compute a similarly short $\mathbf{x} = \mathbf{R} \cdot \mathbf{G}_n^{-1}(\mathbf{u}_0)$, which satisfies

$$\mathbf{A}_T \cdot \mathbf{x} = \mathbf{A}_T \cdot \mathbf{R} \cdot \mathbf{G}_n^{-1}(\mathbf{u}_0) = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \end{pmatrix} \cdot \mathbf{G}_n^{-1}(\mathbf{u}_0) = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \end{pmatrix} .$$

This directly solves sequential SIS for a wide range of parameters, including LM23.

To break the LM23 PoSW*, we similarly compute a solution $\mathbf{x}$ that interacts well with the folding, and simply run the honest prover with it.

Our Attacks, High-level Idea

We construct a "somewhat short" [MP12]-style *trapdoor* $\mathbf{R}$ for $\mathbf{A}_T$ such that

$$\mathbf{A}_T \cdot \mathbf{R} = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \end{pmatrix} .$$

We construct $\mathbf{R}$ in a recursive "divide and conquer" manner so that it takes low depth!

With such $\mathbf{R}$, we then compute a similarly short $\mathbf{x} = \mathbf{R} \cdot \mathbf{G}_n^{-1}(\mathbf{u}_0)$, which satisfies

$$\mathbf{A}_T \cdot \mathbf{x} = \mathbf{A}_T \cdot \mathbf{R} \cdot \mathbf{G}_n^{-1}(\mathbf{u}_0) = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \end{pmatrix} \cdot \mathbf{G}_n^{-1}(\mathbf{u}_0) = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \end{pmatrix} .$$

This directly solves sequential SIS for a wide range of parameters, including LM23.

To break the LM23 PoSW*, we similarly compute a solution $\mathbf{x}$ that interacts well with the folding, and simply run the honest prover with it.

Our Attacks, High-level Idea

We construct a "somewhat short" [MP12]-style *trapdoor* $\mathbf{R}$ for $\mathbf{A}_T$ such that

$$\mathbf{A}_T \cdot \mathbf{R} = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \end{pmatrix} .$$

We construct $\mathbf{R}$ in a recursive "divide and conquer" manner so that it takes low depth!

With such $\mathbf{R}$, we then compute a similarly short $\mathbf{x} = \mathbf{R} \cdot \mathbf{G}_n^{-1}(\mathbf{u}_0)$, which satisfies

$$\mathbf{A}_T \cdot \mathbf{x} = \mathbf{A}_T \cdot \mathbf{R} \cdot \mathbf{G}_n^{-1}(\mathbf{u}_0) = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \end{pmatrix} \cdot \mathbf{G}_n^{-1}(\mathbf{u}_0) = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \end{pmatrix} .$$

This directly solves sequential SIS for a wide range of parameters, including LM23.

To break the LM23 PoSW*, we similarly compute a solution $\mathbf{x}$ that interacts well with the folding, and simply run the honest prover with it.

Our Attacks, High-level Idea

We construct a "somewhat short" [MP12]-style *trapdoor* $\mathbf{R}$ for $\mathbf{A}_T$ such that

$$\mathbf{A}_T \cdot \mathbf{R} = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \end{pmatrix} .$$

We construct $\mathbf{R}$ in a recursive "divide and conquer" manner so that it takes low depth!

With such $\mathbf{R}$, we then compute a similarly short $\mathbf{x} = \mathbf{R} \cdot \mathbf{G}_n^{-1}(\mathbf{u}_0)$, which satisfies

$$\mathbf{A}_T \cdot \mathbf{x} = \mathbf{A}_T \cdot \mathbf{R} \cdot \mathbf{G}_n^{-1}(\mathbf{u}_0) = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \end{pmatrix} \cdot \mathbf{G}_n^{-1}(\mathbf{u}_0) = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \end{pmatrix} .$$

This directly solves sequential SIS for a wide range of parameters, including LM23.

To break the LM23 PoSW*, we similarly compute a solution $\mathbf{x}$ that interacts well with the folding, and simply run the honest prover with it.

Low-depth Recursive Construction of Trapdoors

Suppose we have a block lower-triangular matrix $\mathbf{L}$ (e.g., $\mathbf{L} = \mathbf{A}_T$),
and by recursion *in parallel* have sub-trapdoors $\mathbf{R}_0, \mathbf{R}_1$, as follows:

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}_0 & \\ \boxed{\begin{matrix}\mathbf{W}\\ \mathbf{0}\end{matrix}} & \mathbf{L}_1 \end{pmatrix} \; ; \quad \mathbf{L}_0\mathbf{R}_0 = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \end{pmatrix}, \quad \mathbf{L}_1\mathbf{R}_1 = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \end{pmatrix}.$$

Then we construct trapdoor $\mathbf{R}$ for $\mathbf{L}$ as:

$$\begin{pmatrix} \mathbf{L}_0 & \\ \boxed{\begin{matrix}\mathbf{W}\\ \mathbf{0}\end{matrix}} & \mathbf{L}_1 \end{pmatrix} \overbrace{\begin{pmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \cdot \mathbf{G}_n^{-1}(-\mathbf{W}\mathbf{R}_0) \end{pmatrix}}^{\mathbf{R}, \text{ in depth } \tilde{O}_{n,q}(1)} = \begin{pmatrix} & \boxed{\begin{matrix}\mathbf{G}_n\\ \mathbf{0}\end{matrix}} \\ \boxed{\begin{matrix}\mathbf{W}\\ \mathbf{0}\end{matrix}} \cdot \mathbf{R}_0 + & \boxed{\begin{matrix}\mathbf{G}_n\\ \mathbf{0}\end{matrix}} \cdot \mathbf{G}_n^{-1}(-\mathbf{W}\mathbf{R}_0) \end{pmatrix} = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}.$$

(The base case is $\mathbf{L} = \mathbf{G}_n = \mathbf{A}_1$, which has trivial trapdoor $\mathbf{R} = \mathbf{I}$.)

Low-depth Recursive Construction of Trapdoors

Suppose we have a block lower-triangular matrix $\mathbf{L}$ (e.g., $\mathbf{L} = \mathbf{A}_T$), and by recursion *in parallel* have sub-trapdoors $\mathbf{R}_0, \mathbf{R}_1$, as follows:

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}_0 & \\ \boxed{\begin{matrix}\mathbf{W}\\ \mathbf{0}\end{matrix}} & \mathbf{L}_1 \end{pmatrix} \; ; \quad \mathbf{L}_0 \mathbf{R}_0 = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \end{pmatrix} \, , \quad \mathbf{L}_1 \mathbf{R}_1 = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \end{pmatrix} \, .$$

Then we construct trapdoor $\mathbf{R}$ for $\mathbf{L}$ as:

$$\begin{pmatrix} \mathbf{L}_0 & \\ \boxed{\begin{matrix}\mathbf{W}\\ \mathbf{0}\end{matrix}} & \mathbf{L}_1 \end{pmatrix} \overbrace{\begin{pmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \cdot \mathbf{G}_n^{-1}(-\mathbf{W}\mathbf{R}_0) \end{pmatrix}}^{\mathbf{R}, \text{ in depth } \tilde{O}_{n,q}(1)} = \begin{pmatrix} \boxed{\begin{matrix}\mathbf{G}_n\\ \mathbf{0}\end{matrix}} \\ \boxed{\begin{matrix}\mathbf{W}\\ \mathbf{0}\end{matrix}} \cdot \mathbf{R}_0 + \boxed{\begin{matrix}\mathbf{G}_n\\ \mathbf{0}\end{matrix}} \cdot \mathbf{G}_n^{-1}(-\mathbf{W}\mathbf{R}_0) \end{pmatrix} = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} .$$

(The base case is $\mathbf{L} = \mathbf{G}_n = \mathbf{A}_1$, which has trivial trapdoor $\mathbf{R} = \mathbf{I}$.)

Low-depth Recursive Construction of Trapdoors

Suppose we have a block lower-triangular matrix $\mathbf{L}$ (e.g., $\mathbf{L} = \mathbf{A}_T$), and by recursion *in parallel* have sub-trapdoors $\mathbf{R}_0, \mathbf{R}_1$, as follows:

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}_0 & \\ \boxed{\begin{matrix}\mathbf{W} \\ \mathbf{0}\end{matrix}} & \mathbf{L}_1 \end{pmatrix} \; ; \quad \mathbf{L}_0\mathbf{R}_0 = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \end{pmatrix} , \quad \mathbf{L}_1\mathbf{R}_1 = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \end{pmatrix} .$$

Then we construct trapdoor $\mathbf{R}$ for $\mathbf{L}$ as:

$$\begin{pmatrix} \mathbf{L}_0 & \\ \boxed{\begin{matrix}\mathbf{W} \\ \mathbf{0}\end{matrix}} & \mathbf{L}_1 \end{pmatrix} \overbrace{\begin{pmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \cdot \mathbf{G}_n^{-1}(-\mathbf{W}\mathbf{R}_0) \end{pmatrix}}^{\mathbf{R}\text{, in depth } \tilde{O}_{n,q}(1)} = \begin{pmatrix} & \boxed{\begin{matrix}\mathbf{G}_n \\ \mathbf{0}\end{matrix}} \\ \boxed{\begin{matrix}\mathbf{W} \\ \mathbf{0}\end{matrix}} \cdot \mathbf{R}_0 + & \boxed{\begin{matrix}\mathbf{G}_n \\ \mathbf{0}\end{matrix}} \cdot \mathbf{G}_n^{-1}(-\mathbf{W}\mathbf{R}_0) \end{pmatrix} = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} .$$

(The base case is $\mathbf{L} = \mathbf{G}_n = \mathbf{A}_1$, which has trivial trapdoor $\mathbf{R} = \mathbf{I}$.)

Low-depth Recursive Construction of Trapdoors

Suppose we have a block lower-triangular matrix $\mathbf{L}$ (e.g., $\mathbf{L} = \mathbf{A}_T$), and by recursion *in parallel* have sub-trapdoors $\mathbf{R}_0, \mathbf{R}_1$, as follows:

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}_0 & \\ \boxed{\begin{matrix}\mathbf{W} \\ \mathbf{0}\end{matrix}} & \mathbf{L}_1 \end{pmatrix} \; ; \quad \mathbf{L}_0\mathbf{R}_0 = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \end{pmatrix} , \quad \mathbf{L}_1\mathbf{R}_1 = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \end{pmatrix} .$$

Then we construct trapdoor $\mathbf{R}$ for $\mathbf{L}$ as:

$$\begin{pmatrix} \mathbf{L}_0 & \\ \boxed{\begin{matrix}\mathbf{W} \\ \mathbf{0}\end{matrix}} & \mathbf{L}_1 \end{pmatrix} \overbrace{\begin{pmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \cdot \mathbf{G}_n^{-1}(-\mathbf{W}\mathbf{R}_0) \end{pmatrix}}^{\mathbf{R}\text{, in depth } \tilde{O}_{n,q}(1)} = \begin{pmatrix} & \boxed{\begin{matrix}\mathbf{G}_n \\ \mathbf{0}\end{matrix}} \\ \boxed{\begin{matrix}\mathbf{W} \\ \mathbf{0}\end{matrix}} \cdot \mathbf{R}_0 + & \boxed{\begin{matrix}\mathbf{G}_n \\ \mathbf{0}\end{matrix}} \cdot \mathbf{G}_n^{-1}(-\mathbf{W}\mathbf{R}_0) \end{pmatrix} = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} .$$

(The base case is $\mathbf{L} = \mathbf{G}_n = \mathbf{A}_1$, which has trivial trapdoor $\mathbf{R} = \mathbf{I}$.)

Low-depth Recursive Construction of Trapdoors

Suppose we have a block lower-triangular matrix $\mathbf{L}$ (e.g., $\mathbf{L} = \mathbf{A}_T$), and by recursion *in parallel* have sub-trapdoors $\mathbf{R}_0, \mathbf{R}_1$, as follows:

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}_0 & \\ \boxed{\begin{matrix}\mathbf{W} \\ \mathbf{0}\end{matrix}} & \mathbf{L}_1 \end{pmatrix} \; ; \quad \mathbf{L}_0\mathbf{R}_0 = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \end{pmatrix} , \quad \mathbf{L}_1\mathbf{R}_1 = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \end{pmatrix} .$$

Then we construct trapdoor $\mathbf{R}$ for $\mathbf{L}$ as:

$$\begin{pmatrix} \mathbf{L}_0 & \\ \boxed{\begin{matrix}\mathbf{W} \\ \mathbf{0}\end{matrix}} & \mathbf{L}_1 \end{pmatrix} \overbrace{\begin{pmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \cdot \mathbf{G}_n^{-1}(-\mathbf{W}\mathbf{R}_0) \end{pmatrix}}^{\mathbf{R}\text{, in depth } \tilde{O}_{n,q}(1)} = \begin{pmatrix} & \boxed{\begin{matrix}\mathbf{G}_n \\ \mathbf{0}\end{matrix}} \\ \boxed{\begin{matrix}\mathbf{W} \\ \mathbf{0}\end{matrix}} \cdot \mathbf{R}_0 + \boxed{\begin{matrix}\mathbf{G}_n \\ \mathbf{0}\end{matrix}} \cdot \mathbf{G}_n^{-1}(-\mathbf{W}\mathbf{R}_0) \end{pmatrix} = \begin{pmatrix} \mathbf{G}_n \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} .$$

(The base case is $\mathbf{L} = \mathbf{G}_n = \mathbf{A}_1$, which has trivial trapdoor $\mathbf{R} = \mathbf{I}$.)

Solving Sequential SIS in Low Depth

Recall: Breaking the LM23 Sequentiality Assumption

Sequential SIS with norm bound $(2\gamma)^{2 \log T}$ can be solved in depth $\tilde{O}_{n,q}(\log T)$.

By our recursive construction $\mathbf{R} = \begin{pmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \cdot \mathbf{G}_n^{-1}(\star) \end{pmatrix}$, at each level of the recursion, $\|\mathbf{R}\|$ grows by a factor of $\|\mathbf{G}_n^{-1}(\star)\| \leq O(m)$, and the depth is $\tilde{O}_{n,q}(1)$.

So our attack finds a solution:

- with norm $O(m)^{\log T} \leq (2\gamma)^{2 \log T}$ (for $m = o(n^2) = o(\gamma^2)$, a common setting),
- in depth $\tilde{O}_{n,q}(1) \cdot \log T = \tilde{O}_{n,q}(\log T)$.

More generally, norm $O(m)^{\log_k T}$ in depth $\tilde{O}_{n,q}(k \log_k T)$ for any $2 \leq k \leq T$.

- With $k = T^{\varepsilon}$, *polynomial* norm $O(m)^{1/\varepsilon}$ in *small polynomial* depth $\tilde{O}_{n,q}(T^{\varepsilon})$.

Solving Sequential SIS in Low Depth

Recall: Breaking the LM23 Sequentiality Assumption

Sequential SIS with norm bound $(2\gamma)^{2\log T}$ can be solved in depth $\tilde{O}_{n,q}(\log T)$.

By our recursive construction $\mathbf{R} = \begin{pmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \cdot \mathbf{G}_n^{-1}(\star) \end{pmatrix}$, at each level of the recursion, $\|\mathbf{R}\|$ grows by a factor of $\|\mathbf{G}_n^{-1}(\star)\| \leq O(m)$, and the depth is $\tilde{O}_{n,q}(1)$.

So our attack finds a solution:

- with norm $O(m)^{\log T} \leq (2\gamma)^{2\log T}$ (for $m = o(n^2) = o(\gamma^2)$, a common setting),
- in depth $\tilde{O}_{n,q}(1) \cdot \log T = \tilde{O}_{n,q}(\log T)$.

More generally, norm $O(m)^{\log_k T}$ in depth $\tilde{O}_{n,q}(k \log_k T)$ for any $2 \leq k \leq T$.

- With $k = T^{\varepsilon}$, *polynomial* norm $O(m)^{1/\varepsilon}$ in *small polynomial* depth $\tilde{O}_{n,q}(T^{\varepsilon})$.

Solving Sequential SIS in Low Depth

Recall: Breaking the LM23 Sequentiality Assumption

Sequential SIS with norm bound $(2\gamma)^{2\log T}$ can be solved in depth $\tilde{O}_{n,q}(\log T)$.

By our recursive construction $\mathbf{R} = \begin{pmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \cdot \mathbf{G}_n^{-1}(\star) \end{pmatrix}$, at each level of the recursion, $\|\mathbf{R}\|$ grows by a factor of $\|\mathbf{G}_n^{-1}(\star)\| \leq O(m)$, and the depth is $\tilde{O}_{n,q}(1)$.

So our attack finds a solution:

- with norm $O(m)^{\log T} \leq (2\gamma)^{2\log T}$ (for $m = o(n^2) = o(\gamma^2)$, a common setting),
- in depth $\tilde{O}_{n,q}(1) \cdot \log T = \tilde{O}_{n,q}(\log T)$.

More generally, norm $O(m)^{\log_k T}$ in depth $\tilde{O}_{n,q}(k \log_k T)$ for any $2 \leq k \leq T$.

- With $k = T^{\varepsilon}$, *polynomial* norm $O(m)^{1/\varepsilon}$ in *small polynomial* depth $\tilde{O}_{n,q}(T^{\varepsilon})$.

Solving Sequential SIS in Low Depth

Recall: Breaking the LM23 Sequentiality Assumption

Sequential SIS with norm bound $(2\gamma)^{2\log T}$ can be solved in depth $\tilde{O}_{n,q}(\log T)$.

By our recursive construction $\mathbf{R} = \begin{pmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \cdot \mathbf{G}_n^{-1}(\star) \end{pmatrix}$, at each level of the recursion, $\|\mathbf{R}\|$ grows by a factor of $\|\mathbf{G}_n^{-1}(\star)\| \leq O(m)$, and the depth is $\tilde{O}_{n,q}(1)$.

So our attack finds a solution:

- with norm $O(m)^{\log T} \leq (2\gamma)^{2\log T}$ (for $m = o(n^2) = o(\gamma^2)$, a common setting),
- in depth $\tilde{O}_{n,q}(1) \cdot \log T = \tilde{O}_{n,q}(\log T)$.

More generally, norm $O(m)^{\log_k T}$ in depth $\tilde{O}_{n,q}(k \log_k T)$ for any $2 \leq k \leq T$.

- With $k = T^{\varepsilon}$, *polynomial* norm $O(m)^{1/\varepsilon}$ in *small polynomial* depth $\tilde{O}_{n,q}(T^{\varepsilon})$.

Breaking the LM23 PoSW*

Recall: in the LM23 PoSW, the first check is $\|\mathbf{x}_{T/2}\| \leq 1$, for the middle point;
the second check is $\|c \cdot \mathbf{x}_{T/4} + \mathbf{x}_{3T/4}\| \leq 2\gamma$, for the folding of the quarter points; etc.

Issue: our recursive construction $\mathbf{R} = \begin{pmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \cdot \mathbf{G}_n^{-1}(\star) \end{pmatrix}$ does not have a norm "profile" that works for the folding.

Breaking the LM23 PoSW*

Recall: in the LM23 PoSW, the first check is $\|\mathbf{x}_{T/2}\| \leq 1$, for the middle point; the second check is $\|c \cdot \mathbf{x}_{T/4} + \mathbf{x}_{3T/4}\| \leq 2\gamma$, for the folding of the quarter points; etc.

Issue: our recursive construction $\mathbf{R} = \begin{pmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \cdot \mathbf{G}_n^{-1}(\star) \end{pmatrix}$ does not have a norm "profile" that works for the folding.

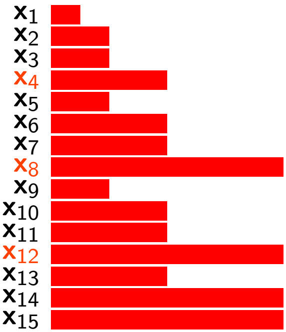
Breaking the LM23 PoSW*

Recall: in the LM23 PoSW, the first check is $\|\mathbf{x}_{T/2}\| \leq 1$, for the middle point; the second check is $\|c \cdot \mathbf{x}_{T/4} + \mathbf{x}_{3T/4}\| \leq 2\gamma$, for the folding of the quarter points; etc.

Issue: our recursive construction $\mathbf{R} = \begin{pmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \cdot \mathbf{G}_n^{-1}(\star) \end{pmatrix}$ does not have a norm "profile" that works for the folding.

Profile needed in folding:

Profile from our recursion:

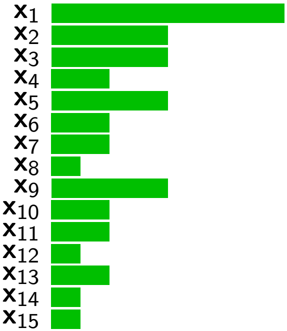


Breaking the LM23 PoSW*

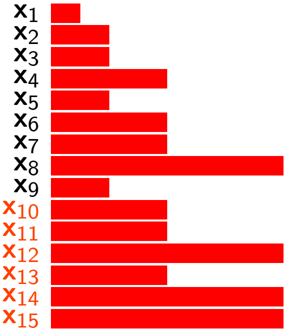
Recall: in the LM23 PoSW, the first check is $\|\mathbf{x}_{T/2}\| \leq 1$, for the middle point; the second check is $\|c \cdot \mathbf{x}_{T/4} + \mathbf{x}_{3T/4}\| \leq 2\gamma$, for the folding of the quarter points; etc.

Issue: our recursive construction $\mathbf{R} = \begin{pmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \cdot \mathbf{G}_n^{-1}(\star) \end{pmatrix}$ does not have a norm “profile” that works for the folding.

Profile needed in original folding:



Profile from our recursion:



First of All, A More Accurate Picture

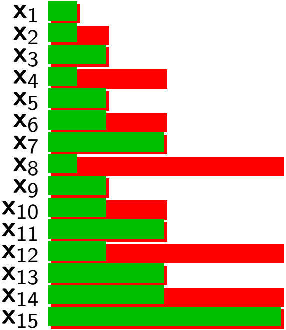
Note the different scales: base γ for folding and base $m \leq \gamma^2$ for our recursion.

Profiles needed in folding / from our recursion, calibrated:

First of All, A More Accurate Picture

Note the different scales: base γ for folding and base $m \leq \gamma^2$ for our recursion.

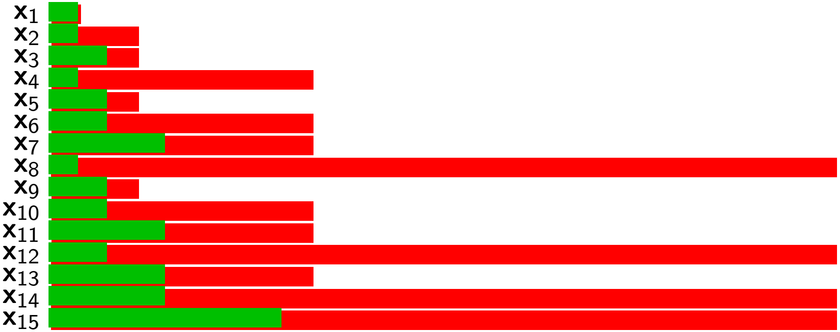
Profiles needed in folding / from our recursion, calibrated:



First of All, A More Accurate Picture

Note the different scales: base γ for folding and base $m \leq \gamma^2$ for our recursion.

Profiles needed in folding / from our recursion, calibrated:



Matching the Profiles

Approach: carefully divide $\mathbf{L}$ *unevenly* into $\mathbf{L}_0, \mathbf{L}_1, \ldots, \mathbf{L}_{k-1}$.

Attempt 1: $k = 3$, divide by $T = T' + 1 + T'$.

Profiles needed in folding / from attempt 1:

Matching the Profiles

Approach: carefully divide $\mathbf{L}$ *unevenly* into $\mathbf{L}_0, \mathbf{L}_1, \ldots, \mathbf{L}_{k-1}$.

Attempt 1: $k = 3$, divide by $T = T' + 1 + T'$.

Profiles needed in folding / from attempt 1:

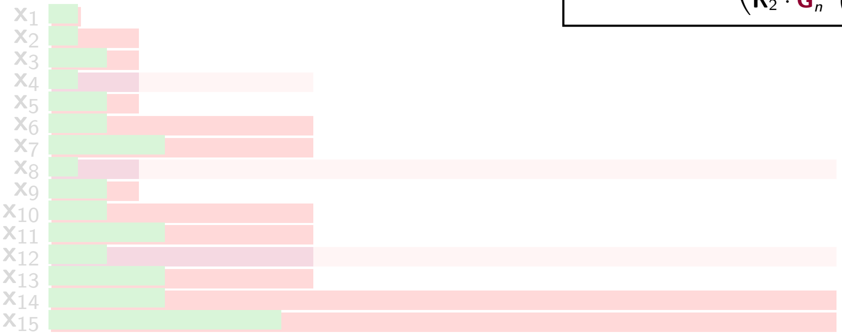
Matching the Profiles

Approach: carefully divide $\mathbf{L}$ *unevenly* into $\mathbf{L}_0, \mathbf{L}_1, \ldots, \mathbf{L}_{k-1}$.

Attempt 1: $k = 3$, divide by $T = T' + 1 + T'$.

Recursion $\mathbf{R} = \begin{pmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \cdot \mathbf{G}_n^{-1}(\star) \\ \mathbf{R}_2 \cdot \mathbf{G}_n^{-1}(\dagger) \end{pmatrix}$ for $k = 3$.

Profiles needed in folding / from attempt 1:

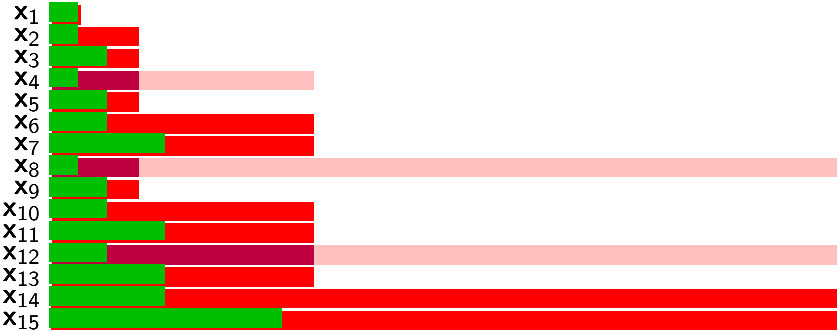


Matching the Profiles

Approach: carefully divide $\mathbf{L}$ *unevenly* into $\mathbf{L}_0, \mathbf{L}_1, \ldots, \mathbf{L}_{k-1}$.

Attempt 1: $k = 3$, divide by $T = T' + 1 + T'$.

Profiles needed in folding / from attempt 1:



"Booting" the Attack: A Direct Solution Technique

Issue: The first check is $\|\mathbf{x}_{T/2}\| \leq 1$. We take $\mathbf{x} = \mathbf{R} \cdot \mathbf{G}_n^{-1}(\mathbf{u}_0)$ so $\|\mathbf{x}_{T/2}\| \leq \|\mathbf{R}_{T/2}\|$.
But we cannot get $\|\mathbf{R}_{T/2}\| \leq 1$ even with "honest" middle point (divide by "$+1+$").

Our solution: to be "fully honest"—apply *direct solution*: $\mathbf{x} = \begin{pmatrix} \mathbf{x}^t \\ \mathbf{R}_1 \cdot \mathbf{G}_n^{-1}(-\mathbf{W}\mathbf{x}^t) \end{pmatrix}$.

- Recursively solve $\mathbf{x}^t$, with base case $\mathbf{x}_0 = \mathbf{G}_n^{-1}(\mathbf{u}_0)$.
- For computing $\mathbf{R}_1$ (*in parallel*), use the same trapdoor recursion as before.
- Roughly the same depth as trapdoor recursion.
- Similar generalization to larger k as trapdoor recursion.

  E.g., recursion $\mathbf{x} = \begin{pmatrix} \mathbf{x}^t \\ \mathbf{R}_1 \cdot \mathbf{G}_n^{-1}(\star) \\ \mathbf{R}_2 \cdot \mathbf{G}_n^{-1}(\dagger) \end{pmatrix}$ for $k = 3$.

"Booting" the Attack: A Direct Solution Technique

Issue: The first check is $\|\mathbf{x}_{T/2}\| \leq 1$. We take $\mathbf{x} = \mathbf{R} \cdot \mathbf{G}_n^{-1}(\mathbf{u}_0)$ so $\|\mathbf{x}_{T/2}\| \leq \|\mathbf{R}_{T/2}\|$. But we cannot get $\|\mathbf{R}_{T/2}\| \leq 1$ even with "honest" middle point (divide by "$+1+$").

Our solution: to be "fully honest"—apply *direct solution*: $\mathbf{x} = \begin{pmatrix} \mathbf{x}^t \\ \mathbf{R}_1 \cdot \mathbf{G}_n^{-1}(-\mathbf{W}\mathbf{x}^t) \end{pmatrix}$.

- Recursively solve $\mathbf{x}^t$, with base case $\mathbf{x}_0 = \mathbf{G}_n^{-1}(\mathbf{u}_0)$.
- For computing $\mathbf{R}_1$ (*in parallel*), use the same trapdoor recursion as before.
- Roughly the same depth as trapdoor recursion.
- Similar generalization to larger k as trapdoor recursion.

  E.g., recursion $\mathbf{x} = \begin{pmatrix} \mathbf{x}^t \\ \mathbf{R}_1 \cdot \mathbf{G}_n^{-1}(\star) \\ \mathbf{R}_2 \cdot \mathbf{G}_n^{-1}(\dagger) \end{pmatrix}$ for $k = 3$.

"Booting" the Attack: A Direct Solution Technique

Issue: The first check is $\|\mathbf{x}_{T/2}\| \leq 1$. We take $\mathbf{x} = \mathbf{R} \cdot \mathbf{G}_n^{-1}(\mathbf{u}_0)$ so $\|\mathbf{x}_{T/2}\| \leq \|\mathbf{R}_{T/2}\|$. But we cannot get $\|\mathbf{R}_{T/2}\| \leq 1$ even with "honest" middle point (divide by "$+1+$").

Our solution: to be "fully honest"—apply *direct solution*: $\mathbf{x} = \begin{pmatrix} \mathbf{x}^t \\ \mathbf{R}_1 \cdot \mathbf{G}_n^{-1}(-\mathbf{W}\mathbf{x}^t) \end{pmatrix}$.

- Recursively solve $\mathbf{x}^t$, with base case $\mathbf{x}_0 = \mathbf{G}_n^{-1}(\mathbf{u}_0)$.
- For computing $\mathbf{R}_1$ (*in parallel*), use the same trapdoor recursion as before.
- Roughly the same depth as trapdoor recursion.
- Similar generalization to larger k as trapdoor recursion.

  E.g., recursion $\mathbf{x} = \begin{pmatrix} \mathbf{x}^t \\ \mathbf{R}_1 \cdot \mathbf{G}_n^{-1}(\star) \\ \mathbf{R}_2 \cdot \mathbf{G}_n^{-1}(\dagger) \end{pmatrix}$ for $k = 3$.

"Booting" the Attack: A Direct Solution Technique

Issue: The first check is $\|\mathbf{x}_{T/2}\| \le 1$. We take $\mathbf{x} = \mathbf{R} \cdot \mathbf{G}_n^{-1}(\mathbf{u}_0)$ so $\|\mathbf{x}_{T/2}\| \le \|\mathbf{R}_{T/2}\|$. But we cannot get $\|\mathbf{R}_{T/2}\| \le 1$ even with "honest" middle point (divide by "$+1+$").

Our solution: to be "fully honest"—apply *direct solution*: $\mathbf{x} = \begin{pmatrix} \mathbf{x}^t \\ \mathbf{R}_1 \cdot \mathbf{G}_n^{-1}(-\mathbf{W}\mathbf{x}^t) \end{pmatrix}$.

- Recursively solve $\mathbf{x}^t$, with base case $\mathbf{x}_0 = \mathbf{G}_n^{-1}(\mathbf{u}_0)$.
- For computing $\mathbf{R}_1$ (*in parallel*), use the same trapdoor recursion as before.
- Roughly the same depth as trapdoor recursion.
- Similar generalization to larger k as trapdoor recursion.

  E.g., recursion $\mathbf{x} = \begin{pmatrix} \mathbf{x}^t \\ \mathbf{R}_1 \cdot \mathbf{G}_n^{-1}(\star) \\ \mathbf{R}_2 \cdot \mathbf{G}_n^{-1}(\dagger) \end{pmatrix}$ for $k = 3$.

“Booting” the Attack: A Direct Solution Technique

Issue: The first check is $\|\mathbf{x}_{T/2}\| \leq 1$. We take $\mathbf{x} = \mathbf{R} \cdot \mathbf{G}_n^{-1}(\mathbf{u}_0)$ so $\|\mathbf{x}_{T/2}\| \leq \|\mathbf{R}_{T/2}\|$. But we cannot get $\|\mathbf{R}_{T/2}\| \leq 1$ even with “honest” middle point (divide by “$+1+$”).

Our solution: to be “fully honest”—apply *direct solution*: $\mathbf{x} = \begin{pmatrix} \mathbf{x}^t \\ \mathbf{R}_1 \cdot \mathbf{G}_n^{-1}(-\mathbf{W}\mathbf{x}^t) \end{pmatrix}$.

- Recursively solve $\mathbf{x}^t$, with base case $\mathbf{x}_0 = \mathbf{G}_n^{-1}(\mathbf{u}_0)$.
- For computing $\mathbf{R}_1$ (*in parallel*), use the same trapdoor recursion as before.
- Roughly the same depth as trapdoor recursion.
- Similar generalization to larger k as trapdoor recursion.

  E.g., recursion $\mathbf{x} = \begin{pmatrix} \mathbf{x}^t \\ \mathbf{R}_1 \cdot \mathbf{G}_n^{-1}(\star) \\ \mathbf{R}_2 \cdot \mathbf{G}_n^{-1}(\dagger) \end{pmatrix}$ for $k = 3$.

"Booting" the Attack: A Direct Solution Technique

Issue: The first check is $\|\mathbf{x}_{T/2}\| \leq 1$. We take $\mathbf{x} = \mathbf{R} \cdot \mathbf{G}_n^{-1}(\mathbf{u}_0)$ so $\|\mathbf{x}_{T/2}\| \leq \|\mathbf{R}_{T/2}\|$. But we cannot get $\|\mathbf{R}_{T/2}\| \leq 1$ even with "honest" middle point (divide by "$+1+$").

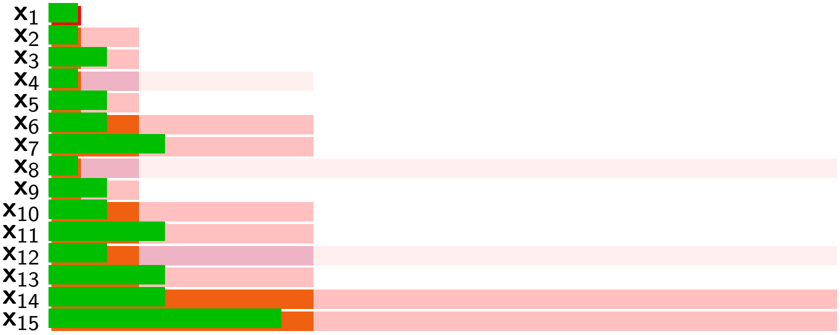
Our solution: to be "fully honest"—apply *direct solution*: $\mathbf{x} = \begin{pmatrix} \mathbf{x}^t \\ \mathbf{R}_1 \cdot \mathbf{G}_n^{-1}(-\mathbf{W}\mathbf{x}^t) \end{pmatrix}$.

- Recursively solve $\mathbf{x}^t$, with base case $\mathbf{x}_0 = \mathbf{G}_n^{-1}(\mathbf{u}_0)$.
- For computing $\mathbf{R}_1$ (*in parallel*), use the same trapdoor recursion as before.
- Roughly the same depth as trapdoor recursion.
- Similar generalization to larger k as trapdoor recursion.

  E.g., recursion $\mathbf{x} = \begin{pmatrix} \mathbf{x}^t \\ \mathbf{R}_1 \cdot \mathbf{G}_n^{-1}(\star) \\ \mathbf{R}_2 \cdot \mathbf{G}_n^{-1}(\dagger) \end{pmatrix}$ for $k = 3$.

Matching the Profiles, Attempt 1 + Direct Solution

Profiles needed in folding / from attempt 1 with direct solution:



Matching the Profiles, Attempt 2

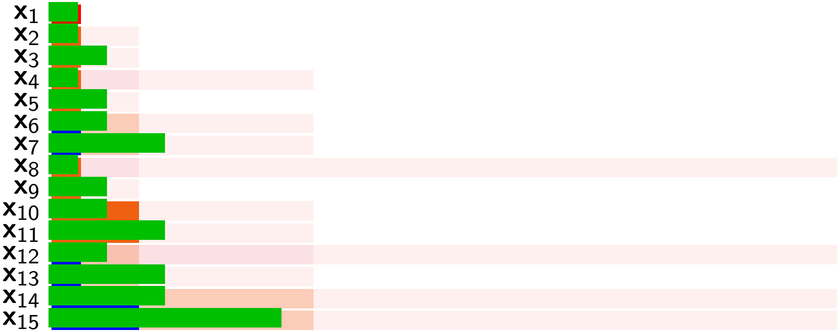
Attempt 2: further divide at $3T/4$, so divide by $T = T' + 1 + T'' + 1 + T''$.

Profiles needed in folding / from attempt 2 (with direct solution):

Matching the Profiles, Attempt 2

Attempt 2: further divide at $3T/4$, so divide by $T = T' + 1 + T'' + 1 + T''$.

Profiles needed in folding / from attempt 2 (with direct solution):



Matching the Profiles, Attempt 3

Attempt 3: further divide at $5T/8$, so divide by $T = T' + 1 + T''' + 1 + T''' + 1 + T''$.

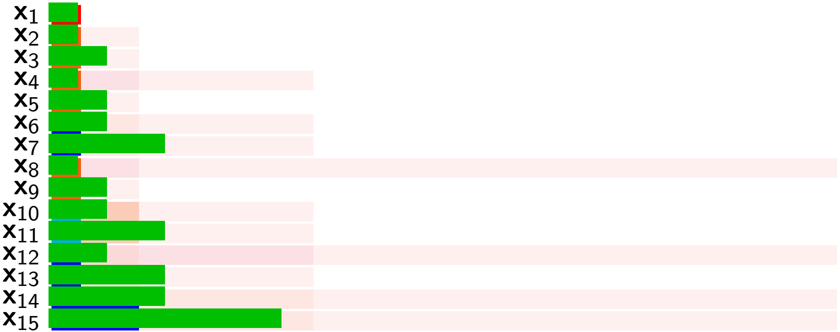
Profiles needed in folding / from attempt 3 (with direct solution):

Done!?

Matching the Profiles, Attempt 3

Attempt 3: further divide at $5T/8$, so divide by $T = T' + 1 + T''' + 1 + T''' + 1 + T''$.

Profiles needed in folding / from attempt 3 (with direct solution):

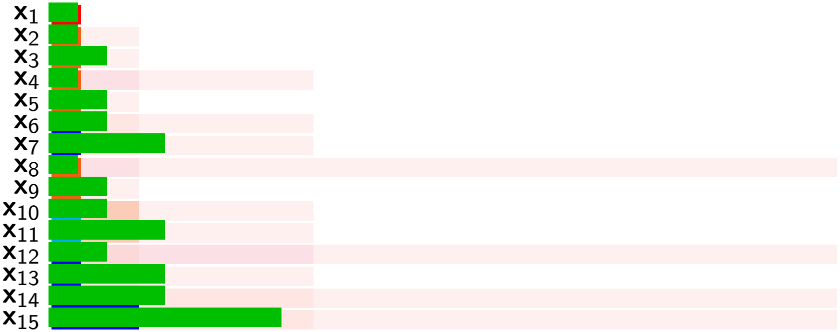


Done!?

Matching the Profiles, Attempt 3

Attempt 3: further divide at $5T/8$, so divide by $T = T' + 1 + T''' + 1 + T''' + 1 + T''$.

Profiles needed in folding / from attempt 3 (with direct solution):



Done!?

Matching the Profiles, Final Attempt

For larger T, we need to continue and further divide at all "$(2^i + 1)/2^{i+1}$-points".

(We have seen the $3/4$- and $5/8$-points.)

We finally take "attempt $\log T$":

- Uses $k \leq 2\log T + 1 = O(\log T)$ at each level of the recursion.
- (Still) has $O(\log T)$ levels.
- Breaks the LM23 PoSW* in depth $\tilde{O}_{n,q}(\log^2 T)$.

Matching the Profiles, Final Attempt

For larger T, we need to continue and further divide at all "$(2^i + 1)/2^{i+1}$-points".

(We have seen the $3/4$- and $5/8$-points.)

We finally take "attempt $\log T$":

- Uses $k \leq 2\log T + 1 = O(\log T)$ at each level of the recursion.
- (Still) has $O(\log T)$ levels.
- Breaks the LM23 PoSW* in depth $\tilde{O}_{n,q}(\log^2 T)$.

Matching the Profiles, Final Attempt

For larger T, we need to continue and further divide at all "$(2^i + 1)/2^{i+1}$-points".

(We have seen the $3/4$- and $5/8$-points.)

We finally take "attempt $\log T$":

- Uses $k \leq 2\log T + 1 = O(\log T)$ at each level of the recursion.
- (Still) has $O(\log T)$ levels.
- Breaks the LM23 PoSW* in depth $\tilde{O}_{n,q}(\log^2 T)$.

Attacking Larger Parameter Space

We assumed $m \leq \gamma^2$ for both attacks, and this can be relaxed.
(This was by $m = o(n^2)$ and $\gamma = \Omega(n)$, a common setting.)

Recall: for our SIS attack, we achieve norm $O(m)^{\log_k T}$ in depth $\tilde{O}_{n,q}(k \log_k T)$.

- With large enough constant k, we achieve depth $\tilde{O}_{n,q}(\log T)$ for any $m = \mathrm{poly}(\gamma)$.
- Also recall: norm $O(m)^{1/\varepsilon}$ in depth $\tilde{O}_{n,q}(T^\varepsilon)$, extending to $m \leq (2\gamma)^{2\varepsilon \log T}$.

For our PoSW* attack:

- We achieve polylogarithmic depth $\tilde{O}_{n,q}(\log^{2+a} T)$ for $m \leq \gamma^{2+a}$, for any $a \in \mathbb{N}$.
- By "further dividing" at more points, still the same profile-matching approach.

Attacking Larger Parameter Space

We assumed $m \leq \gamma^2$ for both attacks, and this can be relaxed.
(This was by $m = o(n^2)$ and $\gamma = \Omega(n)$, a common setting.)

Recall: for our SIS attack, we achieve norm $O(m)^{\log_k T}$ in depth $\tilde{O}_{n,q}(k \log_k T)$.

- With large enough constant k, we achieve depth $\tilde{O}_{n,q}(\log T)$ for any $m = \text{poly}(\gamma)$.
- Also recall: norm $O(m)^{1/\varepsilon}$ in depth $\tilde{O}_{n,q}(T^\varepsilon)$, extending to $m \leq (2\gamma)^{2\varepsilon \log T}$.

For our PoSW* attack:

- We achieve polylogarithmic depth $\tilde{O}_{n,q}(\log^{2+a} T)$ for $m \leq \gamma^{2+a}$, for any $a \in \mathbb{N}$.
- By "further dividing" at more points, still the same profile-matching approach.

Attacking Larger Parameter Space

We assumed $m \leq \gamma^2$ for both attacks, and this can be relaxed.
(This was by $m = o(n^2)$ and $\gamma = \Omega(n)$, a common setting.)

Recall: for our SIS attack, we achieve norm $O(m)^{\log_k T}$ in depth $\tilde{O}_{n,q}(k \log_k T)$.

- With large enough constant k, we achieve depth $\tilde{O}_{n,q}(\log T)$ for any $m = \text{poly}(\gamma)$.
- Also recall: norm $O(m)^{1/\varepsilon}$ in depth $\tilde{O}_{n,q}(T^\varepsilon)$, extending to $m \leq (2\gamma)^{2\varepsilon \log T}$.

For our PoSW* attack:

- We achieve polylogarithmic depth $\tilde{O}_{n,q}(\log^{2+a} T)$ for $m \leq \gamma^{2+a}$, for any $a \in \mathbb{N}$.
- By "further dividing" at more points, still the same profile-matching approach.

Attacking Larger Parameter Space

We assumed $m \leq \gamma^2$ for both attacks, and this can be relaxed.
(This was by $m = o(n^2)$ and $\gamma = \Omega(n)$, a common setting.)

Recall: for our SIS attack, we achieve norm $O(m)^{\log_k T}$ in depth $\tilde{O}_{n,q}(k \log_k T)$.

- With large enough constant k, we achieve depth $\tilde{O}_{n,q}(\log T)$ for any $m = \mathrm{poly}(\gamma)$.
- Also recall: norm $O(m)^{1/\varepsilon}$ in depth $\tilde{O}_{n,q}(T^\varepsilon)$, extending to $m \leq (2\gamma)^{2\varepsilon \log T}$.

For our PoSW* attack:

- We achieve polylogarithmic depth $\tilde{O}_{n,q}(\log^{2+a} T)$ for $m \leq \gamma^{2+a}$, for any $a \in \mathbb{N}$.
- By "further dividing" at more points, still the same profile-matching approach.

Attacking Larger Parameter Space

We assumed $m \leq \gamma^2$ for both attacks, and this can be relaxed.
(This was by $m = o(n^2)$ and $\gamma = \Omega(n)$, a common setting.)

Recall: for our SIS attack, we achieve norm $O(m)^{\log_k T}$ in depth $\tilde{O}_{n,q}(k \log_k T)$.

- With large enough constant k, we achieve depth $\tilde{O}_{n,q}(\log T)$ for any $m = \mathrm{poly}(\gamma)$.
- Also recall: norm $O(m)^{1/\varepsilon}$ in depth $\tilde{O}_{n,q}(T^\varepsilon)$, extending to $m \leq (2\gamma)^{2\varepsilon \log T}$.

For our PoSW* attack:

- We achieve polylogarithmic depth $\tilde{O}_{n,q}(\log^{2+a} T)$ for $m \leq \gamma^{2+a}$, for any $a \in \mathbb{N}$.
- By "further dividing" at more points, still the same profile-matching approach.

Attacking Larger Parameter Space

We assumed $m \leq \gamma^2$ for both attacks, and this can be relaxed.
(This was by $m = o(n^2)$ and $\gamma = \Omega(n)$, a common setting.)

Recall: for our SIS attack, we achieve norm $O(m)^{\log_k T}$ in depth $\tilde{O}_{n,q}(k \log_k T)$.

- With large enough constant k, we achieve depth $\tilde{O}_{n,q}(\log T)$ for any $m = \mathrm{poly}(\gamma)$.
- Also recall: norm $O(m)^{1/\varepsilon}$ in depth $\tilde{O}_{n,q}(T^\varepsilon)$, extending to $m \leq (2\gamma)^{2\varepsilon \log T}$.

For our PoSW* attack:

- We achieve polylogarithmic depth $\tilde{O}_{n,q}(\log^{2+a} T)$ for $m \leq \gamma^{2+a}$, for any $a \in \mathbb{N}$.
- By "further dividing" at more points, still the same profile-matching approach.

Open Questions

Is there attack against the original LM23 PoSW?
(I.e., challenge c on second half.)

Or can we prove its soundness from other plausible (lattice) assumptions?
(A proof would need to rely on the position of c.)

Can we construct lattice-based timed cryptography differently?
(Recently, TLP by Agrawalr, Malavolta, and Zhang (Crypto 2024).)

Open Questions

Is there attack against the original LM23 PoSW?
(I.e., challenge c on second half.)

Or can we prove its soundness from other plausible (lattice) assumptions?
(A proof would need to rely on the position of c.)

Can we construct lattice-based timed cryptography differently?
(Recently, TLP by Agrawalr, Malavolta, and Zhang (Crypto 2024).)

Open Questions

Is there attack against the original LM23 PoSW?
(I.e., challenge c on second half.)

Or can we prove its soundness from other plausible (lattice) assumptions?
(A proof would need to rely on the position of c.)

Can we construct lattice-based timed cryptography differently?
(Recently, TLP by Agrawalr, Malavolta, and Zhang (Crypto 2024).)

References

S. Agrawalr, G. Malavolta, and T. Zhang.
Time-lock puzzles from lattices.
In *CRYPTO*, pages 425–456. 2024.

R. W. F. Lai and G. Malavolta.
Lattice-based timed cryptography.
In *CRYPTO*, pages 782–804. 2023.

D. Micciancio and C. Peikert.
Trapdoors for lattices: Simpler, tighter, faster, smaller.
In *EUROCRYPT*, pages 700–718. 2012.

R. L. Rivest, A. Shamir, and D. A. Wagner.
Time-lock puzzles and timed-release crypto.
Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA, 1996.