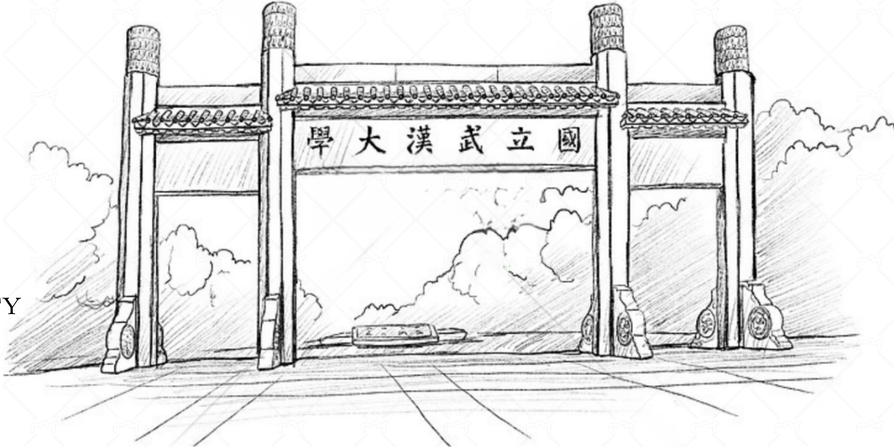
Post-quantum Sigma Protocols and Signatures from Low-Rank Matrix Completions

Jiaming Wen

Houzhen Wang

Huanguo Zhang

Wuhan University



ProvSec 2023

Motivation

Designing Provable and Practical Post-quantum Signature Schemes NOT from Lattices

1. Digital Signature:
   - Message integrity and identity authentication.
   - Quantum Computers $\Rightarrow$ Need to be resistant to classical/quantum adversaries.
   - Schemes standardized by the NIST in July 2022:
     - Dilithium and Falcon: Structural Lattices-based, unknown new attack.
     - SPHINCS+: Hash-based, larger sizes.

Motivation

Designing Provable and Practical Post-quantum Signature Schemes NOT from Lattices

1. Digital Signature:
   - Message integrity and identity authentication.
   - Quantum Computers $\Rightarrow$ Need to be resistant to classical/quantum adversaries.
   - Schemes standardized by the NIST in July 2022:
     - Dilithium and Falcon: Structural Lattices-based, unknown new attack.
     - SPHINCS+: Hash-based, larger sizes.
2. Demand in diversity $\Rightarrow$ NIST's new call for proposals in Sept 2022.

Moody, Dustin (Fed) 7 Sept 2022, 4:16:00 am

to pqc-forum

All, From pqc-forum

NIST is calling for additional digital signature proposals to be considered in the PQC standardization process. NIST is primarily interested in $\boxed{\text{additional general-purpose signature schemes that are not based on structured lattices.}}$ For certain applications, such as certificate transparency, NIST may also be interested in signature schemes that have short signatures and fast verification. NIST is open to receiving additional submissions based on structured lattices, but is intent on diversifying the post-quantum signature standards. As such, any structured lattice-based signature proposal would need to significantly outperform CRYSTALS-Dilithium and FALCON in relevant applications and/or ensure substantial additional security properties to be considered for standardization.

Contribution

Sigma Protocols and Signature Schemes from LRMC

We found there is a hard problem in linear algebra, named

Low-Rank Matrix Completions (LRMC)

could be used in post-quanum crypto designing
- Sigma Protocol
- Signature Scheme – NIST's call

		-1		
			1	
1	1	-1	1	-1
1				-1
		-1		

1	1	-1	1	-1
1	1	-1	1	-1
1	1	-1	1	-1
1	1	-1	1	-1
1	1	-1	1	-1

Goal: Completing the left to the right (low-rank, e.g., $rank = 1$).

Contribution

Sigma Protocols and Signature Schemes from LRMC

In short, we present

1. A Sigma Protocol from LRMC, with soundness error 2/3
2. The first protocol + Sigma Protocol with Helper [Beu20]
   = LRMC-based Sigma Protocol, with soundness error 1/2
3. The second protocol + recent techniques + Fiat-Shamir Transformation [FS86]
   = LRMC-based Signature Scheme, with competitive sizes and simple settings

Outline

- Preliminaries
- LRMC-based Sigma Protocols (with Helper)
- LRMC-based Signature Scheme

Hard Problems

MinRank and 1-MinRank

MinRank Problem

Input: An integer r, and $s+1$ matrices $\mathbf{M}_0; \mathbf{M}_1, \cdots, \mathbf{M}_s \in \mathrm{Mat}_{k,l}(\mathbb{F})$

Output: $\alpha_1, \cdots, \alpha_s \in \mathbb{F}$, such that

$$rank(\mathbf{M}_0 + \sum_{i=1}^{s} \alpha_i \mathbf{M}_i) \leq r$$

Features:

- NP-Complete, and hard for random instances $\Rightarrow$ post-quantum (details later).
- Simple: Based on linear algebra computations, and has high efficiency.
- Extensively studied: Cryptanalysis of Rainbow, GeMSS, HFE/HFEv-, etc.
- 1-MinRank Problem: a special case, requires the rank of $\mathbf{M}_1, \cdots, \mathbf{M}_s$ are 1.

Hard Problems

Low-Rank Matrix Completion (LRMC)

Low-Rank Matrix Completion Problem

Input: An integer r, and a matrix $\mathbf{M} \in \mathrm{Mat}_{k,l}(\mathbb{F})$ with s unfilled entries

Output: $\alpha_1, \cdots, \alpha_s \in \mathbb{F}$, such that completing the remaining to a matrix with rank $\leq r$

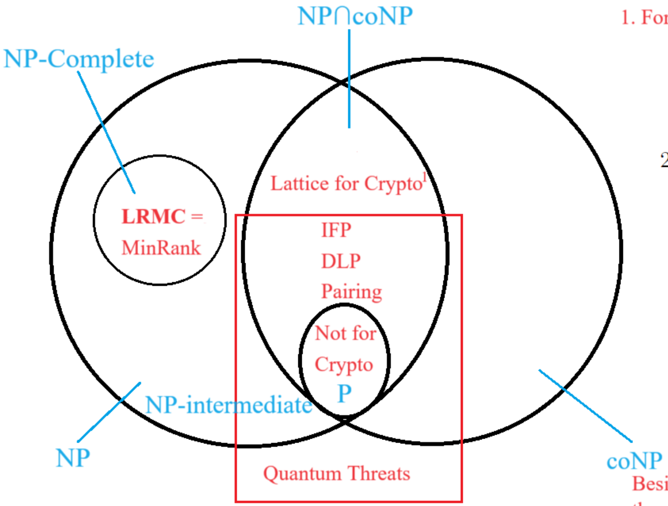
- A toy example, $r = 2, k = 3, l = 4, \mathbb{F} = \mathbb{F}_7$:

$$\mathbf{M} = \begin{bmatrix} 1 & 2 & 4 & 3 \\ 2 & 4 & * & * \\ * & * & 5 & 2 \end{bmatrix}, \quad \mathbf{M}_1 = \begin{bmatrix} 1 & 2 & 4 & 3 \\ 2 & 4 & 4 & 3 \\ 1 & 3 & 5 & 2 \end{bmatrix}, \quad \mathbf{M}_2 = \begin{bmatrix} 1 & 2 & 4 & 3 \\ 2 & 4 & 1 & 6 \\ 3 & 6 & 5 & 2 \end{bmatrix}$$

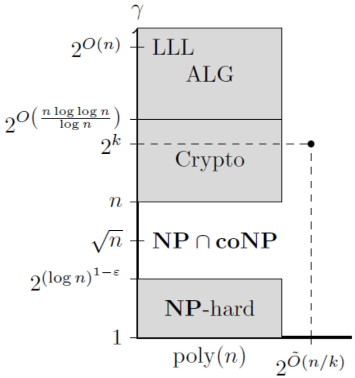
  We have $rank(\mathbf{M}_1) = 3 > 2$, $rank(\mathbf{M}_2) = 1 \leq 2$, and $(1, 6, 3, 6)$ is a solution.
- Equivalence [Der18]: “The instances can be mutual transformed.”

$$\text{MinRank} \Leftrightarrow \text{1-MinRank} \Leftrightarrow \text{LRMC}$$

Hard Problems
The Complexity Comparisons



1. For standard lattices. we have SVPγ



From Vinod Vaikuntanathan's slides in CS294, UCBerkeley

Besides, standard lattices are more reliable than structural lattices (Dilithium and Falcon)

Building Blocks

3-move Sigma Protocol [CD95]

Ronald Cramer

Ivan Damgård

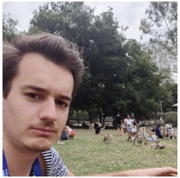
Prover(x, w)		**Verifier**(x)
$(\mathsf{com}, \mathsf{P}_{\mathsf{state}}) \leftarrow \mathcal{P}_1(x, w)$	$\xrightarrow{\mathsf{com}}$	
	$\xleftarrow{\mathsf{ch}}$	$\mathsf{ch} \leftarrow\!\!\$\ \mathcal{C}$
$\mathsf{rsp} \leftarrow \mathcal{P}_2(\mathsf{P}_{\mathsf{state}}, \mathsf{ch})$	$\xrightarrow{\mathsf{rsp}}$	
		$\mathcal{V}(x, \mathsf{com}, \mathsf{ch}, \mathsf{rsp}) = \mathrm{Accept}/\mathrm{Reject}$

Goal: Prove the knowledge w such that $(x, w) \in \mathcal{R}$

- Completeness
- Soundness
- Special Honest-Verifier Zero-Knowledge (SHVZK)

Building Blocks

From 3-move Sigma Protocol to Sigma Protocol with Helper [Beu20]



Ward Beullens

Helper(x)

$\mathsf{seed} \leftarrow\!\!\$\ \{0,1\}^{\lambda}$

$\mathsf{aux} \leftarrow \mathsf{Setup}(\mathsf{seed})$

Sends seed to **Prover** and aux to **Verifier**

Prover(x, w, seed)		**Verifier**(x, aux)
$(\mathsf{com}, \mathsf{P_{state}}) \leftarrow \mathcal{P}_1(x, w, \mathsf{seed})$	$\xrightarrow{\mathsf{com}}$	
	$\xleftarrow{\mathsf{ch}}$	$\mathsf{ch} \leftarrow\!\!\$\ \mathcal{C}$
$\mathsf{rsp} \leftarrow \mathcal{P}_2(\mathsf{P_{state}}, \mathsf{ch})$	$\xrightarrow{\mathsf{rsp}}$	
		$\mathcal{V}(x, \mathsf{aux}, \mathsf{com}, \mathsf{ch}, \mathsf{rsp}) = \text{Accept}/\text{Reject}$

Helper: Trusted by the **Prover** and the **Verifier**.

Goal: Prove the knowledge w such that $(x, w) \in \mathcal{R}$.

Building Blocks

From 3-move Sigma Protocol to Sigma Protocol with Helper [Beu20]

Goal: Prove the knowledge w such that $(x, w) \in \mathcal{R}$.

Security properties of Sigma Protocol with Helper:

- **Completeness. Prover** holds w is always Accepted.
- **2-Special Soundness.** The witness w can be efficiently extracted, i.e., there exists a polynomial-time knowledge extractor $\mathcal{E}$ to use two valid transcripts

$$(x, \mathsf{aux}, \mathsf{com}, \mathsf{ch}, \mathsf{rsp}) \text{ and } (x, \mathsf{aux}, \mathsf{com}, \mathsf{ch}', \mathsf{rsp}').$$

  — Soundness error p: Any efficient adversary $\mathcal{A}(1^\lambda, x)$ passes the protocol with prob. $\leq p + negl(\lambda)$.

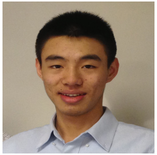
- **HV Zero-Knowledge.** There exists a polynomial-time simulator $\mathcal{S}(x)$ that produces transcripts indistinguishable from ones by **Prover**(x, w).

Building Blocks

Removing the Helper – Using Cut-and-choose technique [KKW18] to simulate the Helper

Jonathan Katz

Vladimir Kolesnikov



Xiao Wang

Prover(x, w)		**Verifier**(x)
For $i \in \{1, \cdots, s\}$:		
$\mathsf{seed}_i \leftarrow_\$ \{0,1\}^\lambda$	$\xrightarrow{\mathsf{com}_i, \mathsf{aux}_i, \forall i}$	Samples $I \subset \{1, \cdots, s\}, \lvert I \rvert = \tau$
$\mathsf{aux} \leftarrow \mathbf{Setup}(\mathsf{seed}_i)$	$\xleftarrow{I, \{\mathsf{ch}_i\}_{i \in I}}$	Checks $\mathsf{aux}_i = \mathbf{Setup}(\mathsf{seed}_i), \forall i \notin I$
com_i and rsp_i as before	$\xrightarrow{\{\mathsf{seed}_i\}_{i \notin I}, \{\mathsf{rsp}_i\}_{i \in I}}$	Validates $(x, \mathsf{aux}_i, \mathsf{com}_i, \mathsf{ch}_i, \mathsf{rsp}_i)_{i \in I}$

Building Blocks

From Sigma Protocol to Signature Scheme – Using Fiat-Shamir Transformation [FS86]

Amos Fiat

Adi Shamir

$$\textbf{Prover}(x,\ w,\ \mathsf{msg}) \quad \begin{array}{c} \xrightarrow{\mathsf{com}} \\ \xleftarrow{\mathsf{ch} = \mathsf{H}(\mathsf{com}, \mathsf{msg})} \\ \xrightarrow{\mathsf{rsp}} \end{array} \quad \Longrightarrow (\mathsf{com}, \mathsf{rsp}) \text{ is a signature for } \mathsf{msg}$$

The transcripts of the simulated protocol

Outline

2 LRMC-based Sigma Protocols (with Helper)

- Preliminaries
- LRMC-based Sigma Protocols (with Helper)
- LRMC-based Signature Scheme

Our Sigma Protocol

How to obtain a LRMC instance

Prover obtains a LRMC instance for crypto construction, as follows:

1. **Prover** chooses a random matrix $\mathbf{A} = (a_{i,j}) \leftarrow \mathrm{Mat}_{k,l}(\mathbb{F})$, s.t. $rank(\mathbf{A}) = r$;
2. **Prover** removes s entries $(a_{i_1,j_1}, a_{i_2,j_2}, \cdots, a_{i_s,j_s})$ to obtain a partially filled matrix $\mathbf{A}^-$.

Let the public key is $\mathbf{A}^-$, and the witness is $(a_{i_1,j_1}, a_{i_2,j_2}, \cdots, a_{i_s,j_s}) = (a_{i_t,j_t})$ for $1 \leq t \leq s$. Then, the completed matrix $\mathbf{A}$ is a solution for the LRMC Problem.

Recall: Low-Rank Matrix Completion Problem

Input: An integer r, and a matrix $\mathbf{M} \in \mathrm{Mat}_{k,l}(\mathbb{F})$ with s unfilled entries

Output: $\alpha_1, \cdots, \alpha_s \in \mathbb{F}$, such that completing the remaining to a matrix with rank $\leq r$

Our Sigma Protocol

How to design a Zero-Knowledge Protocol to prove the Relation – Hiding the witness

Prover proves a relation $(x, w) = (\mathbf{A}^-, a_{i_1,j_1}, a_{i_2,j_2}, \cdots, a_{i_s,j_s})$ in zero-knowledge, as follows:

1. **Prover** breaks it into $\mathbf{A}^- = \mathbf{A}_1^- + \mathbf{A}_2^-$, where $\mathbf{A}_1^-, \mathbf{A}_2^-$ are also partially filled, s.t. $\mathbf{A}_1^-$ is 1 at its filled entries, and divides $a_{i_t,j_t} = \alpha_{i_t,j_t} + \beta_{i_t,j_t}$ for $1 \leq t \leq s$;

$$\underbrace{\begin{bmatrix} 1 & 2 & 4 & 3 \\ 2 & 4 & * & * \\ * & * & 5 & 2 \end{bmatrix}}_{\mathbf{A}^-} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & * & * \\ * & * & 1 & 1 \end{bmatrix}}_{\mathbf{A}_1^-} + \underbrace{\begin{bmatrix} 0 & 1 & 3 & 2 \\ 1 & 3 & * & * \\ * & * & 4 & 1 \end{bmatrix}}_{\mathbf{A}_2^-}$$

2. **Prover** completes $\mathbf{A}_1^-$ ($\mathbf{A}_2^-$) with α_{i_1,j_1} (β_{i_1,j_1}) to obtain $\mathbf{A}_1$ ($\mathbf{A}_2$), i.e.,

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & \alpha_{i_1,j_1} & \alpha_{i_2,j_2} \\ \alpha_{i_3,j_3} & \alpha_{i_4,j_4} & 1 & 1 \end{bmatrix}, \mathbf{A}_2 = \begin{bmatrix} 0 & 1 & 3 & 2 \\ 1 & 3 & \beta_{i_1,j_1} & \beta_{i_2,j_2} \\ \beta_{i_3,j_3} & \beta_{i_4,j_4} & 4 & 1 \end{bmatrix}$$

We have $\mathbf{A} = \mathbf{A}_1 + \mathbf{A}_2$.

Our Sigma Protocol

How to design a Zero-Knowledge Protocol to prove the Relation – Stern-like framework [Ste93]

Based on: $\mathbf{A} = \mathbf{A}_1 + \mathbf{A}_2 \Rightarrow \mathbf{PAQ} = (\mathbf{P}\mathbf{A}_1\mathbf{Q} + \mathbf{Y}) + (\mathbf{P}\mathbf{A}_2\mathbf{Q} - \mathbf{Y}), \forall\, \mathbf{P}, \mathbf{Q}, \mathbf{Y}$

Prover$(x = \mathbf{A}^-, \; w = (a_{i_t,j_t}))$ | **Verifier**$(x = \mathbf{A}^-)$

$\mathbf{c}_0 := \mathsf{Com}(r_0, \mathbf{P}, \mathbf{Q}, \mathbf{Y})$

$\mathbf{c}_1 := \mathsf{Com}(r_1, \mathbf{P}\mathbf{A}_1\mathbf{Q} + \mathbf{Y})$ $\quad \xrightarrow{\mathsf{com} := (\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2)}$

$\mathbf{c}_2 := \mathsf{Com}(r_2, \mathbf{P}\mathbf{A}_2\mathbf{Q} - \mathbf{Y})$ $\quad \xleftarrow{\mathsf{ch}}$ $\quad \mathsf{ch} \xleftarrow{\$} \{0, 1, 2\}$

If $\mathsf{ch} = 0$, then reveals $\mathbf{c}_1, \mathbf{c}_2$

$\mathsf{rsp} := (r_1, r_2, \mathbf{P}\mathbf{A}_1\mathbf{Q} + \mathbf{Y}, \mathbf{P}\mathbf{A}_2\mathbf{Q} - \mathbf{Y})$

If $\mathsf{ch} = 1$, then reveals $\mathbf{c}_0, \mathbf{c}_2$

$\mathsf{rsp} := (r_0, r_2, \mathbf{P}, \mathbf{Q}, \mathbf{Y}, \beta = (\beta_{i_t,j_t}))$ $\quad \xrightarrow{\mathsf{rsp}}$ $\quad$ *Accpet/Reject?*

If $\mathsf{ch} = 2$, then reveals $\mathbf{c}_0, \mathbf{c}_1$

$\mathsf{rsp} := (r_0, r_1, \mathbf{P}, \mathbf{Q}, \mathbf{Y}, \alpha = (\alpha_{i_t,j_t}))$

Our Sigma Protocol with Helper

How to decrease the soundness error from 2/3 to 1/2 [Beu20]

Based on: $\mathbf{A} = \mathbf{A}_1 + \mathbf{A}_2 \Rightarrow \mathbf{PAQ} = (\mathbf{P}\mathbf{A}_1\mathbf{Q} + \mathbf{Y}) + (\mathbf{P}\mathbf{A}_2\mathbf{Q} - \mathbf{Y}), \forall\, \mathbf{P}, \mathbf{Q}, \mathbf{Y}$

Helper: $\mathbf{P}, \mathbf{Q}, \mathbf{Y} \leftarrow \mathsf{PRG}(\mathsf{seed})$, $\mathbf{c}_0 := \mathsf{Com}(r_0, \mathbf{P}, \mathbf{Q}, \mathbf{Y}), \mathbf{c}_1 := \mathsf{Com}(r_1, \mathbf{P}\mathbf{A}_1\mathbf{Q} + \mathbf{Y})$

Prover$(x = \mathbf{A}^-, w = (a_{i_t,j_t}), \mathsf{seed})$		**Verifier**$(x = \mathbf{A}^-, (\mathbf{c}_0, \mathbf{c}_1))$
$\mathbf{c}_2 := \mathsf{Com}(r_1, \mathbf{P}\mathbf{A}_1\mathbf{Q} + \mathbf{Y})$		
	$\xrightarrow{\mathsf{com} := \mathbf{c}_2}$	
	$\xleftarrow{\mathsf{ch}}$	$\mathsf{ch} \leftarrow\!\!\$\ \{0, 1\}$
If $\mathsf{ch} = 0$, then reveals $\mathbf{c}_1, \mathbf{c}_2$ $\mathsf{rsp} := (r_1, r_2, \mathbf{P}\mathbf{A}_1\mathbf{Q} + \mathbf{Y}, \mathbf{P}\mathbf{A}_2\mathbf{Q} - \mathbf{Y})$ If $\mathsf{ch} = 1$, then reveals $\mathbf{c}_0, \mathbf{c}_2$ $\mathsf{rsp} := (r_0, r_2, \mathbf{P}, \mathbf{Q}, \mathbf{Y}, \beta = (\beta_{i_t,j_t}))$		
	$\xrightarrow{\mathsf{rsp}}$	*Accpet/Reject?*

Our Sigma Protocol with Helper

Further Optimizations [Beu20, BESV22]

We take similar tricks to optimize the sizes [Beu20, BESV22], includes:

- Using **cut-and-choose technique** [KKW18] to drop the pre-processing, and remove the helper.
- Using **Merkle Tree** to compress and recompute the commitments.
- Using **Binary Tree** to optimize the transmission of seeds.
- Using several **MPC tricks** to improve parallel repetitions.

Our Sigma Protocol with Helper

Further Optimizations [Beu20, BESV22]

We take similar tricks to optimize the sizes [Beu20, BESV22], includes:

- Using **cut-and-choose technique** [KKW18] to drop the pre-processing, and remove the helper.
- Using **Merkle Tree** to compress and recompute the commitments.
- Using **Binary Tree** to optimize the transmission of seeds.
- Using several **MPC tricks** to improve parallel repetitions.

Pierre de Fermat

"I have a proof of the theorem, but there is not enough space in this margin."– Pierre de Fermat

Please check our paper for more details :)

Outline

- Preliminaries
- LRMC-based Sigma Protocols (with Helper)
- LRMC-based Signature Scheme

Our Signature Scheme

From Sigma Protocol to Signature Scheme – Using Fiat-Shamir Transformation [FS86]

$$\textbf{Prover}(x, w, \mathsf{msg}) \quad \begin{array}{c} \xrightarrow{\mathsf{com}} \\ \xleftarrow{\mathsf{ch} = \mathsf{H}(\mathsf{com}, \mathsf{msg})} \\ \xrightarrow{\mathsf{rsp}} \end{array}$$

$\Longrightarrow$ $(\mathsf{com}, \mathsf{rsp})$ is a signature for msg

The transcripts of the simulated protocol

To sign on the message μ, **Signer** executes the following steps:

1. Runs **the first move** of the Sigma Protocol to generate com ;
2. Computes $\mathsf{ch} = \mathsf{H}(\mathsf{com}, \mu)$;
3. Runs **the third move** of the Sigma Protocol to obtain rsp, and the signature on μ is $(\mathsf{com}, \mathsf{rsp})$.

Our Signature Scheme

Parameters: [BESV22] (PQCrypto'22) is the state-of-the-art, [Cou01] (AC'01) is the pioneering.

Parameter Set		I	II	III
λ [Security parameter]		128	192	256
q [Order of finite field $\mathbb{F} = \mathbb{F}_q$]		16	16	16
(k, l) [Dimensions of matrix $\mathbf{A}$]		(14, 14)	(17, 17)	(20, 20)
r [Rank of matrix $\mathbf{A}$]		4	6	6
s [Unfilled number of matrix $\mathbf{A}$]		108	130	208
Public Key Size (B)	This work	44	80	96
	[BESV22]	60 (by seeds)	104 (by seeds)	128 (by seeds)
	[Cou01]	114 (by seeds)	169 (by seeds)	232 (by seeds)
Signature Size (KB)	This work	24	54	97
	[BESV22]	24	54	97
	[Cou01]	55	118	221
PK + Sig Storage (KB)	This work	24	54	97
	[BESV22]	34	72	137
	[Cou01]	65	136	261

Our Signature Scheme

Comparsions

- **Comparing with MinRank-based schemes [BESV22, Cou01].**
  - **Storage-Lower**: LRMC-based avoid seeds for the PK generation, leading a significant reduction in total storage costs of the public key and signature when actual signing, e.g., more than 30% for the Parameter Set I.
  - **Time-Shorter**: LRMC-based avoid linear combinations and matrix-vector multiplications between hundreds of matrices $\in \mathrm{Mat}_{k,l}(\mathbb{F}_q)$ to recover the PK, saving considerable time.
  - **Conceptually-Simpler**: LRMC-based are more intuitive and succinct, only one (partially completed) matrix $\mathbf{A} \in \mathrm{Mat}_{k,l}(\mathbb{F}_q)$ in the system parameters, instead of $s+1$ matrices in the same dimensions.
- **Comparing with NIST Standards.**
  - SPHINCS+: Sizes are in the same magnitude, e.g., $\lambda = 128$
    - 44B vs 32B in Public Key Size
    - 24KB vs 17KB in Signature Size
  - Dilithium and Falcon: The underlying hard problem LRMC is NP-Complete, providing stronger security guarantee than problems over Structural Lattices.

Retrospect and Prospect

Following the Research Philosophy of Modern Cryptography [GSC+23]

Conclusions: In this work, we present

1. A new NP-Complete problem – LRMC, for crypto designing.
2. A 3-move ZK proof for the solution of LRMC.
3. Decreasing the soundness error from 2/3 to 1/2.
4. A signature scheme from LRMC.

Future work:

Self-Cultivation

- New Construction:
  - ZK protocol for LRMC with smaller error.
  - Trapdoor LRMC-based signatures.
- New Foundation: New hard problems with better sizes/efficiencies.
- New Definition: Formalizing new primitives, and instantiating them from assumptions (LRMC, Lattice, Pairing).

Q&A

Thank you for listening!

Jiaming Wen
Website: https://jiamiwen.github.io
E-mail: wenjm@whu.edu.cn

Emanuele Bellini, Andre Esser, Carlo Sanna, and Javier A. Verbel.
MR-DSS - smaller minrank-based (ring-)signatures.
PQCrypto, 2022.

Ward Beullens.
Sigma protocols for MQ, PKP and SIS, and fishy signature schemes.
EUROCRYPT, 2020.

Ronald Cramer and Ivan Damgård.
Secure signature schemes based on interactive protocols.
CRYPTO, 1995.

Nicolas T. Courtois.
Efficient zero-knowledge authentication based on a linear algebra problem minrank.
ASIACRYPT, 2001.

Harm Derksen.
On the equivalence between low-rank matrix completion and tensor rank.
Linear and Multilinear Algebra, 2018.

Amos Fiat and Adi Shamir.
How to prove yourself: Practical solutions to identification and signature problems.
CRYPTO, 1986.

Fuchun Guo, Willy Susilo, Xiaofeng Chen, Peng Jiang, Jianchang Lai, and Zhen Zhao.
Research philosophy of modern cryptography.
IACR Cryptol. ePrint Arch., 2023.

Jonathan Katz, Vladimir Kolesnikov, and Xiao Wang.
Improved non-interactive zero knowledge with applications to post-quantum signatures.
ACM CCS, 2018.

Jacques Stern.
A new identification scheme based on syndrome decoding.
CRYPTO, 1993.