Technology Innovation Institute

MinRank-Based Zero-Knowledge proofs and Signatures

Javier Verbel

NIST Crypto Reading Club

August 2023

Cryptography Research Center

Motivation

Motivation

1. **Quantum computers:**
   - Threat for public-key cryptography (e.g. RSA and EC)
   - No **big-enough** computers yet, but
   - Save now decrypt later approach.

2. **New NIST standardization call:**
   - **Type** → post-quantum signatures.
   - **MinRank-based** → MiRitH and MIRA
   - **MinRank attacks** → MEDS, SNOVA, etc.

Outline

1. The MinRank problem
2. Algorithms for MinRank
3. Modern ZK-proofs of MinRank solutions
4. MIRA and MiRitH performance

Technology Innovation Institute

The MinRank problem

The MinRank problem

The MinRank problem

Input: An integer r, and $k+1$ matrices $M_0, M_1, \ldots, M_k \in \mathbb{F}_q^{m \times n}$

Output: $\alpha_1, \ldots, \alpha_k \in \mathbb{F}_q$ such that $rank(M_0 + \sum_{i=1}^{k} \alpha_i M_i) \leq r$

The MinRank problem

Input: An integer r, and $k + 1$ matrices $M_0, M_1, \dots, M_k \in \mathbb{F}_q^{m \times n}$

Output: $\alpha_1, \dots, \alpha_k \in \mathbb{F}_q$ such that $rank(M_0 + \sum_{i=1}^{k} \alpha_i M_i) \leq r$

- MinRank is proven to be NP-complete!

The MinRank problem

Input: An integer r, and $k+1$ matrices $M_0, M_1, \dots, M_k \in \mathbb{F}_q^{m \times n}$

Output: $\alpha_1, \dots, \alpha_k \in \mathbb{F}_q$ such that $rank(M_0 + \sum_{i=1}^{k} \alpha_i M_i) \leq r$

- MinRank is proven to be NP-complete!
- **As decoding problem:**

  Given: - $C = \langle M_1, \dots, M_k \rangle$ a linear code in the rank metric.

  - M_0 (a noisy codeword with error $\leq r$)

  Ask: Decode M_0.

The MinRank problem

Input: An integer r, and $k+1$ matrices $M_0, M_1, \ldots, M_k \in \mathbb{F}_q^{m \times n}$

Output: $\alpha_1, \ldots, \alpha_k \in \mathbb{F}_q$ such that $rank(M_0 + \sum_{i=1}^{k} \alpha_i M_i) \leq r$

- MinRank is proven to be NP-complete!
- **As decoding problem:**
  Given: - $C = \langle M_1, \ldots, M_k \rangle$ a linear code in the rank metric.
  \- M_0 (a noisy codeword with error $\leq r$)

  Ask: Decode M_0.
- Type of instances we used :

The MinRank problem

Input: An integer r, and $k+1$ matrices $M_0, M_1, \dots, M_k \in \mathbb{F}_q^{m \times n}$

Output: $\alpha_1, \dots, \alpha_k \in \mathbb{F}_q$ such that $rank(M_0 + \sum_{i=1}^{k} \alpha_i M_i) \leq r$

- MinRank is proven to be NP-complete!
- **As decoding problem:**

  Given: - $C = \langle M_1, \dots, M_k \rangle$ a linear code in the rank metric.

  \- M_0 (a noisy codeword with error $\leq r$)

  Ask: Decode M_0.
- Type of instances we used :
  - Random matrices
  - Random secret
  - Random E
  - $k = O((n-r)^2)$.

The MinRank in Cryptanalysis

The MinRank in Cryptanalysis

1. (Kipnis-Shamir) 1999: Cryptanalysis of Hidden Field Equations (HFE).

2. NIST first call for post-quantum schemes:

- Cryptanalysis of GeMSS
- “Beaking Rainbow takes a weekend in a Laptop”
- Cryptanalysis of Rollo

3. And many multivariate-
   - 2000 --> TTM
   - 2011 --> HFE, Multi-HFE.
   - 2017 --> ZHFE, HFEV-, HFE-, Rainbow.

The MinRank in Cryptanalysis

1. (Kipnis-Shamir) 1999: Cryptanalysis of Hidden Field Equations (HFE).

2. NIST first call for post-quantum schemes:

Features:

- $M_i \in \mathbb{F}_q^{m \times n}$ and $\alpha_{\boldsymbol{i}} \in \mathbb{F}_{q^{\eta}}$
- M_i are not random.
- Multiple solutions.
- $\#Matrices = k = O(n)$.

3

- 2017 --> ZHFE, HFEV-, HFE-, Rainbow.

Algorithms for MinRank

Cryptography
Research Center

tii.ae

Combinatorial

Combinatorial

$$\boxed{E} = \boxed{M_0} + \sum_{i=1}^{k} \alpha_i \boxed{M_i}$$

Combinatorial

$$\boxed{E} = \boxed{M_0} + \sum_{i=1}^{k} \alpha_i \boxed{M_i}$$

- $\forall i, j$:
  $e_{i,j} = Linear_{i,j}\,(\alpha_1, \dots, \alpha_k)$

Combinatorial

$$\left(\boxed{E} = \boxed{M_0} + \sum_{i=1}^{k} \alpha_i \boxed{M_i} \right) \underbrace{\boldsymbol{v}}_{\in \text{kernel}(E)}$$

- $\forall i, j$:
  $e_{i,j} = Linear_{i,j}(\alpha_1, \dots, \alpha_k)$

Combinatorial

$$\underbrace{\left(\boxed{E} = \boxed{M_0} + \sum_{i=1}^{k} \alpha_i \boxed{M_i} \right) \mathbf{v}}_{\mathbf{v} \in \text{kernel}(E)}$$

$$\Downarrow$$

$$\mathbf{0} = M_0 \cdot \mathbf{v} + \sum_{i=1}^{k} \alpha_i \left(M_i \cdot \mathbf{v}\right)$$

- $\forall i, j$:
  $e_{i,j} = Linear_{i,j}\,(\alpha_1, \ldots, \alpha_k)$

Combinatorial

$$\underbrace{\left(\boxed{E} = \boxed{M_0} + \sum_{i=1}^{k} \alpha_i \boxed{M_i} \right) \mathbf{v}}_{\in \mathrm{kernel}(E)}$$

- $\forall i, j$:
  $e_{i,j} = Linear_{i,j}(\alpha_1, \dots, \alpha_k)$

$$\Downarrow$$

$$\mathbf{0} = M_0 \cdot \mathbf{v} + \sum_{i=1}^{k} \alpha_i (M_i \cdot \mathbf{v}) \quad \rightarrow \quad m \text{ linear eqs in the } \alpha_i$$

Combinatorial

$$\underbrace{\left(\boxed{E} = \boxed{M_0} + \sum_{i=1}^{k} \alpha_i \boxed{M_i}\right)\mathbf{v}}_{\mathbf{v} \in \mathrm{kernel}(E)}$$

- $\forall i, j$:
  $e_{i,j} = Linear_{i,j}(\alpha_1, \dots, \alpha_k)$

$$\Downarrow$$

$$\mathbf{0} = M_0 \cdot \mathbf{v} + \textstyle\sum_{i=1}^{k} \alpha_i \, (M_i \cdot \mathbf{v}) \quad \rightarrow \quad \textit{m linear eqs in the } \alpha_i$$

1. Naive approaches → Guess the α_i or the $e_{i,j}$
2. Kernel Search → Guess **enough** L.I vectors in $\mathrm{kernel}(E)$.

Combinatorial

$$\left(\boxed{E} = \boxed{M_0} + \sum_{i=1}^{k} \alpha_i \boxed{M_i} \right) \underbrace{\boldsymbol{v}}_{\in \mathrm{kernel}(E)}$$

- $\forall i, j$:
  $e_{i,j} = Linear_{i,j}\,(\alpha_1, \dots, \alpha_k)$

$$\Downarrow$$

$$\mathbf{0} = M_0 \cdot \boldsymbol{v} + \textstyle\sum_{i=1}^{k} \alpha_i \,(M_i \cdot \boldsymbol{v}\,)$$ $\rightarrow$ *m linear eqs in the* α_i

1. Naive approaches → Guess the α_i or the $e_{i,j}$
2. Kernel Search → Guess **enough** L.I vectors in $\mathrm{kernel}(E)$.

3. Hybrid approach → ($basic\ case$)

$$\left(\boxed{E} = \boxed{M_0} + \sum_{i=1}^{k} \alpha_i \boxed{M_i} \right)$$

Combinatorial

$$\left(\boxed{E} = \boxed{M_0} + \sum_{i=1}^{k} \alpha_i \boxed{M_i} \right) \underbrace{\boldsymbol{v}}_{\in \operatorname{kernel}(E)}$$

- $\forall i, j$:
  $e_{i,j} = Linear_{i,j}(\alpha_1, \dots, \alpha_k)$

$$\Downarrow$$

$$\mathbf{0} = M_0 \cdot \boldsymbol{v} + \sum_{i=1}^{k} \alpha_i \, (M_i \cdot \boldsymbol{v}) \quad \rightarrow \quad \textit{m linear eqs in the } \alpha_i$$

1. Naive approaches → Guess the α_i or the $e_{i,j}$
2. Kernel Search → Guess **enough** L.I vectors in $\operatorname{kernel}(E)$.

3. Hybrid approach → $(\textit{basic case})$

$$\left(\boxed{E} = \boxed{M_0} + \sum_{i=1}^{k} \alpha_i \boxed{M_i} \right) \underbrace{\boxed{\begin{matrix} *** \\ *** \end{matrix}}}_{\textbf{Non-singular}}$$

Combinatorial

$$\Bigg(\boxed{E} = \boxed{M_0} + \sum_{i=1}^{k} \alpha_i \boxed{M_i}\Bigg) \underbrace{\boldsymbol{v}}_{\in \, \mathrm{kernel}(E)}$$

- $\forall i, j$:
  $e_{i,j} = Linear_{i,j}\,(\alpha_1, \dots, \alpha_k)$

$$\Downarrow$$

$$\mathbf{0} = M_0 \cdot \boldsymbol{v} + \sum_{i=1}^{k} \alpha_i \,(M_i \cdot \boldsymbol{v}) \quad \rightarrow \quad \textit{m linear eqs in the } \alpha_i$$

1. Naive approaches → Guess the α_i or the $e_{i,j}$
2. Kernel Search → Guess **enough** L.I vectors in $\mathrm{kernel}(E)$.

3. Hybrid approach → $(basic\ case)$

$$\Bigg(\boxed{E} = \boxed{M_0} + \sum_{i=1}^{k} \alpha_i \boxed{M_i}\Bigg) \underbrace{\left[\begin{array}{c|c} \boldsymbol{v} & \begin{matrix} *** \\ *** \end{matrix} \end{array}\right]}_{\text{Non-singular}}$$

Combinatorial

$$\left(\boxed{E} = \boxed{M_0} + \sum_{i=1}^{k} \alpha_i \boxed{M_i} \right) \underbrace{\mathbf{v}}_{\in \, \mathrm{kernel}(E)}$$

- $\forall i,j$:
  $e_{i,j} = Linear_{i,j}(\alpha_1, \dots, \alpha_k)$

$$\Downarrow$$

$$\mathbf{0} = M_0 \cdot \boldsymbol{v} + \textstyle\sum_{i=1}^{k} \alpha_i \, (M_i \cdot \boldsymbol{v}) \quad \rightarrow \quad \textit{m linear eqs in the } \alpha_i$$

1. Naive approaches → Guess the α_i or the $e_{i,j}$
2. Kernel Search → Guess **enough** L.I vectors in $\mathrm{kernel}(E)$.

3. Hybrid approach → ($\textit{basic case}$)

$$\left(\boxed{E} = \boxed{M_0} + \sum_{i=1}^{k} \alpha_i \boxed{M_i} \right) \underbrace{\left(\mathbf{v} \;\middle|\; \begin{matrix} *** \\ *** \end{matrix} \right)}_{\text{Non-singular}} \implies$$

MR problem with one **known** column of E'

Hybrid approach and Algebraic

Hybrid approach and Algebraic

- Hybrid → Guess $\boldsymbol{l_v}$ of the $\alpha_i{}'s$, and $\boldsymbol{a}$ vectors in $kernel(E)$.

$$MinRank\ (m\times n, k, r) = q^{l_v + ar}\ MinRank(m\times(n - a), k - am - l_v, r).$$

- **Algebraic:**

Hybrid approach and Algebraic

- Hybrid → Guess $\boldsymbol{l_v}$ of the $\alpha_i{}'s$, and $\boldsymbol{a}$ vectors in $kernel(E)$.

$$MinRank\ (m\times n, k, r) = q^{l_v + ar}\ MinRank(m\times(n-a), k - am - l_v, r).$$

- **Algebraic:**

Hybrid approach and Algebraic

- Hybrid → Guess $\boldsymbol{l_v}$ of the $\alpha_i's$, and $\boldsymbol{a}$ vectors in $kernel(E)$.

$$MinRank\ (m\times n, k, r) = \ q^{l_v+ar}\ MinRank(m\times(n\ -a\ ), k-am\ -l_v\ , r).$$

- **Algebraic:**

1. Minors → Minor_of_size_(r+1) $\left(M_0 + \sum_{\ell=1}^{k} \alpha_\ell M_\ell\right) = 0$

Hybrid approach and Algebraic

- Hybrid → Guess $\boldsymbol{l_v}$ of the $\alpha_i's$, and $\boldsymbol{a}$ vectors in $kernel(E)$.

$$MinRank\ (m\times n, k, r) = q^{l_v+ar}\ MinRank(m\times(n-a), k-am-l_v, r).$$

- **Algebraic:**

1. Minors → Minor_of_size_(r+1) $\left(M_0 + \sum_{\ell=1}^{k} \alpha_\ell M_\ell\right) = 0 \implies$ degree-(r+1) polys in α_i

Hybrid approach and Algebraic

- Hybrid → Guess $\boldsymbol{l_v}$ of the $\alpha_i's$, and $\boldsymbol{a}$ vectors in $kernel(E)$.

$$MinRank\ (m\times n, k, r) = q^{l_v+ar}\ MinRank(m\times(n-a), k-am-l_v, r).$$

- **Algebraic:**

1. Minors → Minor_of_size_(r+1) $\left| M_0 + \sum_{\ell=1}^{k} \alpha_\ell M_\ell \right| = 0 \implies$ degree-(r+1) polys in α_i
2. Kipnis-Shamir → $\left(M_0 + \sum_{\ell=1}^{k} \alpha_\ell M_\ell\right) \cdot \begin{pmatrix} I_{n-r} \\ K \end{pmatrix} = 0$

Hybrid approach and Algebraic

- Hybrid → Guess $\boldsymbol{l_v}$ of the $\alpha_i{}'s$, and $\boldsymbol{a}$ vectors in $kernel(E)$.

$$MinRank\ (m \times n, k, r) = q^{l_v + ar}\ MinRank(m \times (n - a), k - am - l_v, r).$$

- **Algebraic:**

1. Minors → Minor_of_size_(r+1) $\left(M_0 + \sum_{\ell=1}^{k} \alpha_\ell M_\ell \right) = 0 \implies$ degree-(r+1) polys in α_i
2. Kipnis-Shamir → $\left(M_0 + \sum_{\ell=1}^{k} \alpha_\ell M_\ell \right) \cdot \begin{pmatrix} I_{n-r} \\ K \end{pmatrix} = 0 \implies$ bilinear in α_i and entries of K.

Hybrid approach and Algebraic

- Hybrid → Guess $\boldsymbol{l_v}$ of the $\alpha_i{}'s$, and $\boldsymbol{a}$ vectors in $kernel(E)$.

$$MinRank\ (m{\times}n, k, r) = q^{l_v+ar}\ MinRank(m{\times}(n-a), k-am-l_v, r).$$

- **Algebraic:**

1. Minors → Minor_of_size_(r+1) $\left(M_0 + \sum_{\ell=1}^{k} \alpha_\ell M_\ell\right) = 0 \implies$ degree-(r+1) polys in α_i

2. Kipnis-Shamir → $\left(M_0 + \sum_{\ell=1}^{k} \alpha_\ell M_\ell\right) \cdot \begin{pmatrix} I_{n-r} \\ K \end{pmatrix} = 0 \implies$ bilinear in α_i and entries of K.

3. Support-Minors →

Hybrid approach and Algebraic

- Hybrid → Guess $\boldsymbol{l_v}$ of the $\alpha_i's$, and $\boldsymbol{a}$ vectors in $kernel(E)$.

$$MinRank\ (m \times n, k, r) = q^{l_v + ar}\ MinRank(m \times (n - a), k - am - l_v, r).$$

- **Algebraic:**

1. Minors → Minor_of_size_(r+1) $\left(M_0 + \sum_{\ell=1}^{k} \alpha_\ell M_\ell\right) = 0 \implies$ degree-(r+1) polys in α_i

2. Kipnis-Shamir → $\underbrace{\left(M_0 + \sum_{\ell=1}^{k} \alpha_\ell M_\ell\right)}_{M} \cdot \begin{pmatrix} I_{n-r} \\ K \end{pmatrix} = 0 \implies$ bilinear in α_i and entries of K.

3. Support-Minors →

$$\forall i \quad \text{Minors} \begin{pmatrix} \boxed{i\text{-th row of } M} \\ \boxed{C = \text{Gen of row space } E} \end{pmatrix}$$

Hybrid approach and Algebraic

- Hybrid → Guess $\boldsymbol{l_v}$ of the $\alpha_i's$, and $\boldsymbol{a}$ vectors in $kernel(E)$.

$$MinRank\ (m\times n, k, r) = q^{l_v+ar}\ MinRank(m\times(n-a), k-am-l_v, r).$$

- **Algebraic:**

1. Minors → Minor_of_size_(r+1) $\left(M_0 + \sum_{\ell=1}^{k} \alpha_\ell M_\ell\right) = 0 \implies$ degree-(r+1) polys in α_i
2. Kipnis-Shamir → $\underbrace{\left(M_0 + \sum_{\ell=1}^{k} \alpha_\ell M_\ell\right)}_{M} \cdot \begin{pmatrix} I_{n-r} \\ K \end{pmatrix} = 0 \implies$ bilinear in α_i and entries of K.
3. Support-Minors →

$$\forall i \quad \text{Minors} \begin{pmatrix} \boxed{i\text{-th row of } M} \\ \boxed{C = \text{Gen of row space } E} \end{pmatrix} = 0$$

Hybrid approach and Algebraic

- Hybrid → Guess $\boldsymbol{l_v}$ of the $\alpha_i's$, and $\boldsymbol{a}$ vectors in $kernel(E)$.

$$MinRank\ (m\times n, k, r) = q^{l_v + ar}\ MinRank(m\times(n-a), k-am-l_v, r).$$

- **Algebraic:**

1. Minors → Minor_of_size_(r+1) $\left(M_0 + \sum_{\ell=1}^{k} \alpha_\ell M_\ell\right) = 0 \implies$ degree-(r+1) polys in α_i
2. Kipnis-Shamir → $\underbrace{\left(M_0 + \sum_{\ell=1}^{k} \alpha_\ell M_\ell\right)}_{M} \cdot \begin{pmatrix} I_{n-r} \\ K \end{pmatrix} = 0 \implies$ bilinear in α_i and entries of K.
3. Support-Minors →

$\forall i$ Minors $\begin{pmatrix} \boxed{i\text{-th row of } M} \\ \boxed{C = \text{Gen of row space } E} \end{pmatrix} = 0 \implies$ bilinear in α_i and minors of C.

Zero-Knowledge proofs

Cryptography
Research Center

tii.ae

ZK proofs

ZK proofs

- Executed by **Prover**(s, w) and **Verifier**(s)
- **Goal**: Proof (s, w) satisfy a relation R.
- **Requires:**
  - **ZK**: No info of w is leaked to **Verifier.**
  - **Soundness**:

    Someone without w cheats with

    $$\text{Prob} \leq \text{Soundness Error}$$

ZK proofs

- Executed by **Prover**(s, w) and **Verifier**(s)
- **Goal**: Proof (s, w) satisfy a relation R.
- **Requires:**
  - **ZK**: No info of w is leaked to **Verifier.**
  - **Soundness**:

    Someone without w cheats with

    $$\text{Prob} \leq \text{Soundness Error}$$

ZK proofs for MinRank

$$s = (M_0, \ldots, M_k), \quad w = (\alpha_1, \ldots, \alpha_k)$$

$$R \rightarrow \ Rank\left(M_0 + \Sigma_{i=1}^{k} \alpha_i M_i\right) \leq r$$

ZK proofs

- Executed by **Prover**(s, w) and **Verifier**(s)
- **Goal**: Proof (s, w) satisfy a relation R.
- **Requires:**
  - **ZK**: No info of w is leaked to **Verifier.**
  - **Soundness**:

    Someone without w cheats with

    $\text{Prob} \leq \text{Soundness Error}$

ZK proofs for MinRank

$$s = (M_0, \ldots, M_k),\ \ w = (\alpha_1, \ldots, \alpha_k)$$

$$R \rightarrow\ Rank\left(M_0 + \Sigma_{i=1}^{k} \alpha_i M_i\right) \leq r$$

Previous MinRank ZK proofs

Authors	Type	Year	S. error
Courtois	3-pass	2001	2/3
Bellini-Esser-Sanna-Verbel	3-pass + Helper	2022	1/2
Adj-Rivera-Verbel	MPCitH	2022	O(1/N)
Feneuil	MPCitH	2022	O(1/N)

MPC protocol

MPC protocol

N- Party MPC protocol

MPC protocol

N- Party MPC protocol

Given: function f and value z

Goal: Verify if $f(x) = z$, with $x = \sum x_i$

Output:

accept : P_i's think they **do** share x.

reject : P_i's think they **don't** share x

MPC protocol

N- Party MPC protocol

$$
\begin{array}{ccc}
P_1 & \longleftrightarrow & P_2 \\
x_1 & & x_2 \\
\updownarrow & \boxtimes & \updownarrow \\
P_3 & \longleftrightarrow & P_N \\
x_3 & & x_N
\end{array}
$$

Given: function f and value z

Goal: Verify if $f(x) = z$, with $x = \sum x_i$

Output:

- accept : P_i's think they **do** share x.
- reject : P_i's think they **don't** share x

MPC protocol

N- Party MPC protocol

$$\begin{array}{ccc} P_1 & & P_2 \\ x_1 & \longleftrightarrow & x_2 \\ \updownarrow & ⤧ & \updownarrow \\ P_3 & & P_N \\ x_3 & \longleftrightarrow & x_N \end{array}$$

Given: function f and value z

Goal: Verify if $f(x) = z$, with $x = \sum x_i$

Output:

- accept : P_i's think they **do** share x.
- reject : P_i's think they **don't** share x

False-Positive-Rate = Pr[**accept** | $f(x) \neq z$]

No information on x_i leaked to P_j for $j \neq i$

MPC-in-the-Head

MPC-in-the-Head

Given: MPC protocol

Goal: zero-knowledge proof solution

(Prover P wants to identify to verifier V)

MPC-in-the-Head

Given: MPC protocol

Goal: zero-knowledge proof solution

(Prover P wants to identify to verifier V)

Prover

prepare MPC inputs x_i and commit

Simulate MPC protocol based on R and commit

Reveal all views of Parties $P_i, i \neq i^*$

MPC-in-the-Head

Given: MPC protocol

Goal: zero-knowledge proof solution

(Prover P wants to identify to verifier V)

Prover

prepare MPC inputs x_i and commit

Simulate MPC protocol based on R and commit

Reveal all views of Parties $P_i, i \neq i^*$

Verifier

Sample first challenge R

Sample second challenge i^*

Check validity of Com_j

MPC-in-the-Head

Given: MPC protocol

Goal: zero-knowledge proof solution

(Prover P wants to identify to verifier V)

Prover		**Verifier**
prepare MPC inputs x_i and commit	$\xrightarrow{Com_1}$	
	$\xleftarrow{R}$	Sample first challenge R
Simulate MPC protocol based on R and commit	$\xrightarrow{Com_2}$	
	$\xleftarrow{i^*}$	Sample second challenge i^*
Reveal all views of Parties $P_i, i \neq i^*$	$\xrightarrow{\forall i \neq i^*, View_i, aux_{i^*}}$	Check validity of Com_j

MPC-in-the-Head

Given: MPC protocol

Goal: zero-knowledge proof solution

(Prover P wants to identify to verifier V)

Prover

prepare MPC inputs x_i and commit

Simulate MPC protocol based on R and commit

Reveal all views of Parties $P_i, i \neq i^*$

Com_1

Fiat-Shamir: Signature Scheme

i^*

$\forall i \neq i^*, View_i, aux_{i^*}$

Verifier

Sample first challenge R

Sample second challenge i^*

Check validity of Com_j

Kipnis-Shamir modelling (1999)

Kipnis-Shamir modelling (1999)

Models MinRank as a bilinear system

Kipnis-Shamir modelling (1999)

Models MinRank as a bilinear system

$$\left(M_0 + \sum_{\ell=1}^{k} \alpha_\ell M_\ell\right) \cdot \begin{pmatrix} I_{n-r} \\ K \end{pmatrix} = 0$$

Kipnis-Shamir modelling (1999)

Models MinRank as a bilinear system

$$\left(M_0 + \sum_{\ell=1}^{k} \alpha_\ell M_\ell\right) \cdot \begin{pmatrix} I_{n-r} \\ K \end{pmatrix} = 0$$

Solving system $\Rightarrow$ Solving MinRank!

Kipnis-Shamir modelling (1999)

Models MinRank as a bilinear system

$$\underbrace{\left(M_0 + \sum_{\ell=1}^{k} \alpha_\ell M_\ell\right)}_{M_{\vec{\alpha}}} \cdot \begin{pmatrix} I_{n-r} \\ K \end{pmatrix} = 0$$

Solving system $\Rightarrow$ Solving MinRank!

$$M_{\vec{\alpha}} \cdot \begin{pmatrix} I_{n-r} \\ K \end{pmatrix} = 0 \iff M_{\vec{\alpha}}^{L} = -M_{\vec{\alpha}}^{R} \cdot K$$

Kipnis-Shamir modelling (1999)

Models MinRank as a bilinear system

$$\underbrace{\left(M_0 + \sum_{\ell=1}^{k} \alpha_\ell M_\ell\right)}_{M_{\vec{\alpha}}} \cdot \begin{pmatrix} I_{n-r} \\ K \end{pmatrix} = 0$$

Solving system $\Rightarrow$ Solving MinRank!

$$M_{\vec{\alpha}} \cdot \binom{I_{n-r}}{K} = 0 \iff M_{\vec{\alpha}}^{L} = -M_{\vec{\alpha}}^{R} \cdot K$$

Knowledge of MinRank solution $\vec{\alpha}$

$\iff$

Knowledge of K such that $M_{\vec{\alpha}}^{L} = -M_{\vec{\alpha}}^{R} \cdot K$

MPC Protocol for MiRitH

MPC Protocol for MiRitH

$\vec{\alpha}$ solution of MinRank problem $M_0, M_1, \dots, M_k$

$\vec{\alpha} = \sum_{i=1}^{N} \vec{\alpha}_i$ and $K = \sum_{i=1}^{N} K_i$

MPC Protocol for MiRitH

$\vec{\alpha}$ solution of MinRank problem $M_0, M_1, \ldots, M_k$

$\vec{\alpha} = \sum_{i=1}^{N} \vec{\alpha}_i$ and $K = \sum_{i=1}^{N} K_i$

$$
\begin{array}{ccc}
\begin{matrix} P_1 \\ (\vec{\alpha}_1, K_1) \end{matrix} & \longleftrightarrow & \begin{matrix} P_2 \\ (\vec{\alpha}_2, K_2) \end{matrix} \\
\updownarrow & \boxtimes & \updownarrow \\
\begin{matrix} P_3 \\ (\vec{\alpha}_3, K_3) \end{matrix} & \longleftrightarrow & \begin{matrix} P_N \\ (\vec{\alpha}_N, K_N) \end{matrix}
\end{array}
$$

MPC Protocol for MiRitH

$\vec{\alpha}$ solution of MinRank problem $M_0, M_1, \ldots, M_k$

$\vec{\alpha} = \sum_{i=1}^{N} \vec{\alpha}_i$ and $K = \sum_{i=1}^{N} K_i$

P_1 $(\vec{\alpha}_1, K_1)$	$\longleftrightarrow$	P_2 $(\vec{\alpha}_2, K_2)$
$\updownarrow$		$\updownarrow$
P_3 $(\vec{\alpha}_3, K_3)$	$\longleftrightarrow$	P_N $(\vec{\alpha}_N, K_N)$

Goal: Verify parties share $(\vec{\alpha}, K)$ s.t.

$$M^L_{\vec{\alpha}} = -M^R_{\vec{\alpha}} \cdot K$$

Output:

- accept : P_i's think they **do** share $(\vec{\alpha}, K)$
- reject : P_i's think they **don't** share $(\vec{\alpha}, K)$

MPC Protocol for MiRitH

$\vec{\alpha}$ solution of MinRank problem $M_0, M_1, \ldots, M_k$

$\vec{\alpha} = \sum_{i=1}^{N} \vec{\alpha}_i$ and $K = \sum_{i=1}^{N} K_i$

$$\begin{array}{ccc} P_1 & & P_2 \\ (\vec{\alpha}_1, K_1) & \longleftrightarrow & (\vec{\alpha}_2, K_2) \\ \updownarrow & \boxtimes & \updownarrow \\ P_3 & & P_N \\ (\vec{\alpha}_3, K_3) & \longleftrightarrow & (\vec{\alpha}_N, K_N) \end{array}$$

Goal: Verify parties share $(\vec{\alpha}, K)$ s.t.

$$M^L_{\vec{\alpha}} = -M^R_{\vec{\alpha}} \cdot K$$

Output:

accept : P_i's think they **do** share $(\vec{\alpha}, K)$

reject : P_i's think they **don't** share $(\vec{\alpha}, K)$

No information on $(\vec{\alpha}_i, K_i)$ leaked

MPC Protocol for MiRitH

Matrix-Product MPC verifies (X, Y, Z) satisfies $X \cdot Y = Z$

$\vec{\alpha}$ solution of MinRank problem $M_0, M_1, \ldots, M_k$

$\vec{\alpha} = \sum_{i=1}^{N} \vec{\alpha}_i$ and $K = \sum_{i=1}^{N} K_i$

$$\begin{array}{ccc} P_1 & & P_2 \\ (\vec{\alpha}_1, K_1) & \longleftrightarrow & (\vec{\alpha}_2, K_2) \\ \updownarrow & \boxtimes & \updownarrow \\ P_3 & & P_N \\ (\vec{\alpha}_3, K_3) & \longleftrightarrow & (\vec{\alpha}_N, K_N) \end{array}$$

Goal: Verify parties share $(\vec{\alpha}, K)$ s.t.

$$M^L_{\vec{\alpha}} = -M^R_{\vec{\alpha}} \cdot K$$

Output:

accept : P_i's think they **do** share $(\vec{\alpha}, K)$

reject : P_i's think they **don't** share $(\vec{\alpha}, K)$

No information on $(\vec{\alpha}_i, K_i)$ leaked

Verifying Matrix-Product

Verifying Matrix-Product

Given : Party i holds matrices Z_i, X_i, Y_i, C_i and A_i

Goal : Verify that $Z = X \cdot Y$

Verifying Matrix-Product

Given : Party i holds matrices Z_i, X_i, Y_i, C_i and A_i

Goal : Verify that $Z = X \cdot Y$

C, A auxiliary matrices s.t. $C = A \cdot Y$

Verifying Matrix-Product

C, A auxiliary matrices s.t. $C = A \cdot Y$

Given : Party i holds matrices Z_i, X_i, Y_i, C_i and A_i

Goal : Verify that $Z = X \cdot Y$

MPC-Protocol (Similar to [KZ22] over $\mathbb{F}_q$)

Verifying Matrix-Product

C, A auxiliary matrices s.t. $C = A \cdot Y$

Given : Party i holds matrices Z_i, X_i, Y_i, C_i and A_i

Goal : Verify that $Z = X \cdot Y$

MPC-Protocol (Similar to [KZ22] over $\mathbb{F}_q$)

1. Select a random $R \in \mathbb{F}_q^{t \times n}$
2. $S_i = R \cdot X_i + A_i$
3. Broadcast S_i to obtain S
4. $V_i = S \cdot Y_i - R \cdot Z_i - C_i$
5. Broadcast V_i to obtain V
5. **accept** if $V = 0$, otherwise, **reject**

Verifying Matrix-Product

C, A auxiliary matrices s.t. $C = A \cdot Y$

Given : Party i holds matrices Z_i, X_i, Y_i, C_i and A_i

Goal : Verify that $Z = X \cdot Y$

MPC-Protocol (Similar to [KZ22] over $\mathbb{F}_q$)

R is verifier's first challenge

1. Select a random $R \in \mathbb{F}_q^{t \times n}$
2. $S_i = R \cdot X_i + A_i$
3. Broadcast S_i to obtain S
4. $V_i = S \cdot Y_i - R \cdot Z_i - C_i$
5. Broadcast V_i to obtain V
5. **accept** if $V = 0$, otherwise, **reject**

Verifying Matrix-Product

Given : Party i holds matrices Z_i, X_i, Y_i, C_i and A_i

Goal : Verify that $Z = X \cdot Y$

C, A auxiliary matrices s.t. $C = A \cdot Y$

MPC-Protocol (Similar to [KZ22] over $\mathbb{F}_q$)

1. Select a random $R \in \mathbb{F}_q^{t \times n}$
2. $S_i = R \cdot X_i + A_i$
3. Broadcast S_i to obtain S
4. $V_i = S \cdot Y_i - R \cdot Z_i - C_i$
5. Broadcast V_i to obtain V
5. **accept** if $V = 0$, otherwise, **reject**

R is verifier's first challenge

Correctness : If $Z = X \cdot Y$ and $C = A \cdot Y$, then parties **accept**

False-Positive rate: If not, the Parties **accept** with prob. q^{-t}

Linearized Polynomials

Linearized Polynomials

- **Definition** $L: \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}$ of the form $L(X) = X^{q^r} + \beta_1 X^{q^{r-1}} + \ldots + \beta_r X$

- **Fact 1:** Roots of L form a r-dimensional $\mathbb{F}_q$ - subspace of $\mathbb{F}_{q^m}$.
  - 0 is a solution
  - If x , y $\in \mathbb{F}_{q^m}$ are solutions then ax + by solution with a, b $\in \mathbb{F}_q$

- **Fact 2:** $\mathbb{F}_q^m \cong \mathbb{F}_{q^m}$

- **Fact 3:**

Linearized Polynomials

- **Definition** $L: \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}$ of the form $L(X) = X^{q^r} + \beta_1 X^{q^{r-1}} + \ldots + \beta_r X$

- **Fact 1:** Roots of L form a r-dimensional $\mathbb{F}_q$ - subspace of $\mathbb{F}_{q^m}$.
  - 0 is a solution
  - If x , y $\in \mathbb{F}_{q^m}$ are solutions then ax + by solution with a, b $\in \mathbb{F}_q$

- **Fact 2:** $\mathbb{F}_q^m \cong \mathbb{F}_{q^m}$

- **Fact 3:** $E = M_0 + \sum_{\ell=1}^{k} \alpha_\ell M_\ell$ and represent E as $(e_1, \ldots, e_n) \in \mathbb{F}_{q^m}^n$

$$Rank(E) \leq r \Leftrightarrow L(e_1) = \cdots = L(e_n) = 0$$

MPC Protocol for MIRA

MPC Protocol for MIRA

$\vec{\alpha} = \sum_{i=1}^{N} \vec{\alpha}_i$

$L = \sum_{i=1}^{N} L_i$, where L_i linearized poly.

MPC Protocol for MIRA

$\vec{\alpha} = \sum_{i=1}^{N} \vec{\alpha}_i$

$L = \sum_{i=1}^{N} L_i$, where L_i linearized poly.

$$\begin{array}{ccc}
P_1 & & P_2 \\
(\vec{\alpha}_1, L_1) & \longleftrightarrow & (\vec{\alpha}_2, L_2) \\
\updownarrow & ⤧ & \updownarrow \\
P_3 & & P_N \\
(\vec{\alpha}_3, L_3) & \longleftrightarrow & (\vec{\alpha}_N, L_N)
\end{array}$$

MPC Protocol for MIRA

$\vec{\alpha} = \sum_{i=1}^{N} \vec{\alpha}_i$

$L = \sum_{i=1}^{N} L_i$, where L_i linearized poly.

Goal: Verify parties share $(\vec{\alpha}, L)$ s.t.

$$L(e_i) = 0 \ \forall i,$$

$e_1, \dots, e_n \in \mathbb{F}_{q^m}$ representing the cols of $E = M_0 + \sum_{\ell=1}^{k} \alpha_\ell M_\ell$.

$$\begin{tikzcd}
\begin{matrix} P_1 \\ (\vec{\alpha}_1, L_1) \end{matrix} \arrow[r, leftrightarrow] \arrow[d, leftrightarrow] \arrow[dr, leftrightarrow] & \begin{matrix} P_2 \\ (\vec{\alpha}_2, L_2) \end{matrix} \arrow[d, leftrightarrow] \arrow[dl, leftrightarrow] \\
\begin{matrix} P_3 \\ (\vec{\alpha}_3, L_3) \end{matrix} \arrow[r, leftrightarrow] & \begin{matrix} P_N \\ (\vec{\alpha}_N, L_N) \end{matrix}
\end{tikzcd}$$

MPC Protocol for MIRA

$\vec{\alpha} = \sum_{i=1}^{N} \vec{\alpha}_i$

$L = \sum_{i=1}^{N} L_i$, where L_i linearized poly.

Goal: Verify parties share $(\vec{\alpha}, L)$ s.t.

$$L(e_i) = 0 \ \forall i,$$

$e_1, \dots, e_n \in \mathbb{F}_{q^m}$ representing the cols of $E = M_0 + \sum_{\ell=1}^{k} \alpha_\ell M_\ell$.

$$\begin{array}{ccc} P_1 & \longleftrightarrow & P_2 \\ (\vec{\alpha}_1, L_1) & & (\vec{\alpha}_2, L_2) \\ \updownarrow & ⤡⤢ & \updownarrow \\ P_3 & & P_N \\ (\vec{\alpha}_3, L_3) & \longleftrightarrow & (\vec{\alpha}_N, L_N) \end{array}$$

1. Select a random $\gamma_1, \dots, \gamma_n \in \mathbb{F}_{q^m}$.
2. Run an MPC to verify that $\sum_{j=1}^{n} \gamma_j L(e_j) = 0$
   - Set $\vec{\beta}_i \in \mathbb{F}_{q^m}^r$ as the coef. L_i
   - Parties locally use $\vec{\alpha}_i$ to compute

     $$z_i \in \mathbb{F}_{q^m} \text{ and } \vec{w}_i \in \mathbb{F}_{q^m}^r$$

   - Run Matrix-Product MPC to check

     $$z = \vec{\beta} \cdot \vec{w}$$

Technology Innovation Institute

Performance

Category I: SPHINS+ vs MinRank-based

Category I: SPHINS+ vs MinRank-based

Scheme	Variant	$\|sig\|$ (bytes)	$\|pk\|$ (bytes)	Signing (million-cycles)	Verify (million-cycles)
SPHINCS+	fast	17.1 K	0.03 K	33.6	2.1
	short	7.9 K		644.7	0.8
MIRA	fast	**7.4 K**	**0.08 K**	43.7	43.1
	short	**5.6 K**	**0.08 K**	51.8	49.4
MiRitH	fast	7.9 K	0.13 K	**5.2**	**4.7**
	short	5.7 K	0.13 K	**31.9**	**31.5**

Comparitsion: Some MPCitH/ZK candidates (cat I)

Comparitsion: Some MPCitH/ZK candidates (cat I)

Scheme	Variant	$\lvert sig \rvert$ (bytes)	$\lvert pk \rvert$ (bytes)	Sign time (million-cycles)	Verify time (million-cycles)	Security Assumption
MQOM	short	6.3 K	0.05K	32.8	29.5	MQ
RYDE	short	6.0 K	0.1 K	23.4	20.1	Rank-SD
LESS	large-pk	5.4 K	95.9 K	206	213	Linear Code Equiv.
CROSS	short	7.6 K	0.04 K	11.0	7.8	Restricted-SD
SDitH	short	8.2 K	0.12 K	13.4	12.5	d-split-SD
FEAST	short	**4.5** K	0.03 K	53 **ms**	53 **ms**	AES
PERK	short	6.1 K	0.24 K	36.0	25.0	Permuted Kernel
MiRitH	short	5.7 K	0.13 K	31.9	31.5	MinRank
MIRA	short	5.6 K	0.08 K	51.8	49.4	MinRank

Comparitsion: Some MPCitH/ZK candidates (cat I)

Scheme	Variant	$\|sig\|$ (bytes)	$\|pk\|$ (bytes)	Sign time (million-cycles)	Verify time (million-cycles)	Security Assumption
MQOM	short	6.3 K	0.05K	32.8	29.5	MQ
RYDE	short	6.0 K	0.1 K	23.4	20.1	Rank-SD
LESS	large-pk	5.4 K	95.9 K	206	213	Linear Code Equiv.
CROSS	short	7.6 K	0.04 K	11.0	7.8	Restricted-SD
SDitH	short	8.2 K	0.12 K	13.4	12.5	d-split-SD
FEAST	short	**4.5** K	0.03 K	53 **ms**	53 **ms**	AES
PERK	short	6.1 K	0.24 K	36.0	25.0	Permuted Kernel
MiRitH	short	5.7 K	0.13 K	31.9	31.5	MinRank
MIRA	short	5.6 K	0.08 K	51.8	49.4	MinRank

Thank You!

Questions?