MiRitH

(MinRank in the Head)

Gora Adj, Stefano Barbero, Emanuele Bellini,
Andre Esser, Luis Rivera-Zamarripa, Carlo Sanna,
Javier Verbel, Floyd Zweydinger

PQC Workshop Oxford

September 2023

Cryptography Research Center

Overview

Overview

1. Foundation

   EUF-CMA secure in the ROM assuming hardness of MinRank

Overview

1. Foundation

EUF-CMA secure in the ROM assuming hardness of MinRank

2. Approach MPC-in-the-Head:

Overview

1. Foundation

EUF-CMA secure in the ROM assuming hardness of MinRank

2. Approach MPC-in-the-Head:

   1. MPC protocol to verify a shared solution of MinRank
   2. Zero-Knowledge proof of a solution
   3. Signature scheme from Fiat-Shamir transform

Overview

1. Foundation

EUF-CMA secure in the ROM assuming hardness of MinRank

2. Approach MPC-in-the-Head:
    1. MPC protocol to verify a shared solution of MinRank
    2. Zero-Knowledge proof of a solution
    3. Signature scheme from Fiat-Shamir transform

3. Parameters, Security and Performance

Technology Innovation Institute

The MinRank problem

Cryptography
Research Center

tii.ae

MinRank

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

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

MinRank

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

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

MinRank as decoding problem:

MinRank

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

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

MinRank as decoding problem:

Gen. Matrix $G = \begin{pmatrix} Vec(M_1) \\ \vdots \\ Vec(M_k) \end{pmatrix} \in \mathbb{F}_q^{k \times (n \cdot m)}$

$Vec(M_0) = (\alpha_1, \dots, \alpha_k) \cdot G + Vec(E)$, where $rank(E) \leq r$

MinRank

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

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

MinRank as decoding problem:

Gen. Matrix $G = \begin{pmatrix} Vec(M_1) \\ \vdots \\ Vec(M_k) \end{pmatrix} \in \mathbb{F}_q^{k \times (n \cdot m)}$

$Vec(M_0) = (\alpha_1, \dots, \alpha_k) \cdot G + Vec(E)$, where $rank(E) \leq r$

Type of instances we use:

MinRank

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

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

MinRank as decoding problem:

Gen. Matrix $G = \begin{pmatrix} Vec(M_1) \\ \vdots \\ Vec(M_k) \end{pmatrix} \in \mathbb{F}_q^{k \times (n \cdot m)}$

$Vec(M_0) = (\alpha_1, \dots, \alpha_k) \cdot G + Vec(E)$, where $rank(E) \leq r$

Type of instances we use:

- Random matrices
- Random secret
- Random E

MPC-in-the-Head

Cryptography
Research Center

MPC protocol

MPC protocol

Starting point

N-Party MPC protocol

MPC protocol

Starting point

N-Party MPC protocol

Given: function f, value z, share x_i of x

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 **do not** share x

MPC protocol

Starting point

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, value z, share x_i of x

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 **do not** share x

MPC protocol

Starting point

N-Party MPC protocol

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

Given: function f, value z, share x_i of x

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 **do not** 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 of knowledge

(Prover P wants to proof knowledge of x with $f(x) = z$ to V)

MPC-in-the-Head

Given: MPC protocol

Goal: zero-knowledge proof of knowledge

(Prover P wants to proof knowledge of x with $f(x) = z$ to V)

Prover

MPC-in-the-Head

Given: MPC protocol

Goal: zero-knowledge proof of knowledge

(Prover P wants to proof knowledge of x with $f(x) = z$ to V)

Prover

Verifier

MPC-in-the-Head

Given: MPC protocol

Goal: zero-knowledge proof of knowledge

(Prover P wants to proof knowledge of x with $f(x) = z$ to V)

Prover

prepare MPC inputs x_i and commit

$$\xrightarrow{Com_1}$$

Verifier

MPC-in-the-Head

Given: MPC protocol

Goal: zero-knowledge proof of knowledge

(Prover P wants to proof knowledge of x with $f(x) = z$ to V)

Prover

prepare MPC inputs x_i and commit

$$\xrightarrow{Com_1}$$

$$\xleftarrow{R}$$

Verifier

Sample first challenge R

MPC-in-the-Head

Given: MPC protocol

Goal: zero-knowledge proof of knowledge

(Prover P wants to proof knowledge of x with $f(x) = z$ to V)

Prover

prepare MPC inputs x_i and commit

Simulate MPC protocol based on R and commit

$$Com_1 \longrightarrow$$

$$R \longleftarrow$$

$$Com_2 \longrightarrow$$

Verifier

Sample first challenge R

MPC-in-the-Head

Given: MPC protocol

Goal: zero-knowledge proof of knowledge

(Prover P wants to proof knowledge of x with $f(x) = z$ to 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^*

MPC-in-the-Head

Given: MPC protocol

Goal: zero-knowledge proof of knowledge

(Prover P wants to proof knowledge of x with $f(x) = z$ to 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 $P_i, i \neq i^*$	$\xrightarrow{\forall i \neq i^*, View_i, aux_{i^*}}$	

MPC-in-the-Head

Given: MPC protocol

Goal: zero-knowledge proof of knowledge

(Prover P wants to proof knowledge of x with $f(x) = z$ to 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 of knowledge

(Prover P wants to proof knowledge of x with $f(x) = z$ to V)

Prover

prepare MPC inputs x_i and commit

Simulate MPC protocol based on R and commit

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

$\xrightarrow{Com_1}$

Fiat-Shamir: Signature Scheme

$\xleftarrow{i^*}$

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

Verifier

Sample first challenge R

Sample second challenge i^*

Check validity of Com_j

Design rationale

Kipnis-Shamir modelling

Kipnis-Shamir modelling

Models MinRank as a bilinear system

Kipnis-Shamir modelling

Models MinRank as a bilinear system

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

Kipnis-Shamir modelling

Models MinRank as a bilinear system

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

Solving system $\Rightarrow$ Solving MinRank!

Kipnis-Shamir modelling

Models MinRank as a bilinear system

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

Solving system $\Rightarrow$ Solving MinRank!

$$\boxed{\beta_i = \alpha_i}$$

Kipnis-Shamir modelling

Models MinRank as a bilinear system

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

Solving system $\Rightarrow$ Solving MinRank!

Kipnis-Shamir modelling

Models MinRank as a bilinear system

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

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

Models MinRank as a bilinear system

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

Solving system $\Rightarrow$ Solving MinRank!

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

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

$\iff$

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

MPC Protocol to verify MinRank Solution

MPC Protocol to verify MinRank Solution

$\vec{\alpha}$ 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 to verify MinRank Solution

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

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

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

MPC Protocol to verify MinRank Solution

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

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

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

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 to verify MinRank Solution

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

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

P_1 $(\vec{\alpha}_1, K_1)$ $\longleftrightarrow$ P_2 $(\vec{\alpha}$

MiRitH: MPC verifies (X, Y, Z) satisfies $X \cdot Y = Z$

$(\vec{\alpha}_3, K_3)$ $\longleftrightarrow$ 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 $(\vec{\alpha}, K)$

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

Verifying Matrix Multiplication Triplets

Verifying Matrix Multiplication Triplets

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 Multiplication Triplets

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 Multiplication Triplets

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

Verifying Matrix Multiplication Triplets

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

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 Multiplication Triplets

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

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 Multiplication Triplets

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

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**

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}

MiRitH Summary

MiRitH Summary

1. Kinpis-Shamir modelling

MiRitH Summary

1. Kinpis-Shamir modelling

2. MPC for matrix-triplet verification

MiRitH Summary

1. Kinpis-Shamir modelling
2. MPC for matrix-triplet verification
3. MPC-in-the-Head (incl. hypercube, seedtrees, etc.)

MiRitH Summary

1. Kinpis-Shamir modelling
2. MPC for matrix-triplet verification
3. MPC-in-the-Head (incl. hypercube, seedtrees, etc.)
4. Fiat-Shamir transform

Technology Innovation Institute

Security, Parameters and Performance

Cryptography
Research Center

Attacks

Attacks

Find $\alpha \in \mathbb{F}_q^k$ such that: $E := M_0 + \sum_{i=1}^{k} \alpha_i M_i \in \mathbb{F}_q^{m \times n}$, and $rank(E) \leq r$

Attacks

Find $\alpha \in \mathbb{F}_q^k$ such that: $E \coloneqq M_0 + \Sigma_{i=1}^k \alpha_i M_i \in \mathbb{F}_q^{m \times n}$, and $rank(E) \leq r$

1. Kernel Search (combinatorial) : Guess vectors in $kernel(E)$

Attacks

Find $\alpha \in \mathbb{F}_q^k$ such that: $E \coloneqq M_0 + \Sigma_{i=1}^k \alpha_i M_i \in \mathbb{F}_q^{m \times n}$, and $rank(E) \leq r$

1. Kernel Search (combinatorial) : Guess vectors in $kernel(E)$

2. Support-Minors (algebraic) : Model as bilinear system of equations

Attacks

Find $\alpha \in \mathbb{F}_q^k$ such that: $E \coloneqq M_0 + \Sigma_{i=1}^k \alpha_i M_i \in \mathbb{F}_q^{m \times n}$, and $rank(E) \leq r$

1. Kernel Search (combinatorial) : Guess vectors in $kernel(E)$
2. Support-Minors (algebraic) : Model as bilinear system of equations
3. Big-k (combinatorial) : Guess entries of E

Attacks

Find $\alpha \in \mathbb{F}_q^k$ such that: $E \coloneqq M_0 + \Sigma_{i=1}^{k} \alpha_i \, M_i \in \mathbb{F}_q^{m \times n}$, and $rank(E) \leq r$

1. Kernel Search (combinatorial) : Guess vectors in $kernel(E)$

2. Support-Minors (algebraic) : Model as bilinear system of equations

3. Big-k (combinatorial) : Guess entries of E

Hybrid approach: Guess some of the $\alpha_i{}'s$, and some vectors in $kernel(E)$

Attacks

Find $\alpha \in \mathbb{F}_q^k$ such that: $E := M_0 + \Sigma_{i=1}^k \alpha_i M_i \in \mathbb{F}_q^{m \times n}$, and $rank(E) \leq r$

1. Kernel Search (combinatorial) : Guess vectors in $kernel(E)$

2. Support-Minors (algebraic) : Model as bilinear system of equations

3. Big-k (combinatorial) : Guess entries of E

Hybrid approach: Guess some of the $\alpha_i{}'s$, and some vectors in $kernel(E)$

→ MinRank instance of smaller dimension

Category I MinRank Parameters

Category I MinRank Parameters

Category	set	q	$m = n$	k	r
I	a	16	15	78	6
I	b	16	16	142	4

Parameters of the underlying MinRank instance

Category I MinRank Parameters

Category	set	q	$m = n$	k	r
I	a	16	15	78	6
I	b	16	16	142	4

Parameters of the underlying MinRank instance

Category	set	Kernel-Search	Support Minors	Big-k
I	a	151	**144**	154
I	b	**159**	165	226

Complexity estimates for proposed parameters, with linear algebra constant equal to 3 in KS and Big-k, equal 2.81 for Strassen in SM.

Performance

Performance

Category I a	Sig. Size	Pk size	Key Gen.	Sign	Verify
Short	5.7 kB	129 Bytes	~53.000	~23 MCycles	
Fast	7.7 kB			~ 3 MCycles	

Performance

Category I a	Sig. Size	Pk size	Key Gen.	Sign	Verify
Short	5.7 kB	129 Bytes	~53.000	~23 MCycles	
Fast	7.7 kB			~ 3 MCycles	

Category I b	Sig. Size	Pk size	Key Gen.	Sign	Verify
Short	6.3 kB	129 Bytes	~53.000	~24 MCycles	
Fast	8.8 kB			~ 4 MCycles	

Performance

Category I a	Sig. Size	Pk size	Key Gen.	Sign	Verify
Short	5.7 kB	129 Bytes	~53.000	~23 MCycles	
Fast	7.7 kB			~ 3 MCycles	

Category I b	Sig. Size	Pk size	Key Gen.	Sign	Verify
Short	6.3 kB	129 Bytes	~53.000	~24 MCycles	
Fast	8.8 kB			~ 4 MCycles	

Thank You!

Questions?