

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

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

EUFCMA 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

The MinRank 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 $\text{rank}(E) \leq r$

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 $\text{rank}(E) \leq r$

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 $\text{rank}(E) \leq r$

MinRank as decoding problem:

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

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

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 $\text{rank}(E) \leq r$

MinRank as decoding problem:

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

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

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 $\text{rank}(E) \leq r$

MinRank as decoding problem:

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

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

Type of instances we use:

- Random matrices
- Random secret
- Random E

MPC-in-the-Head

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 $x = \sum x_i$

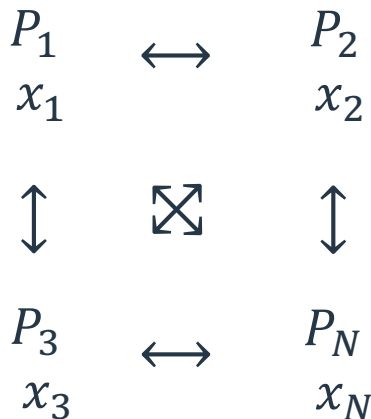
Output:

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



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

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

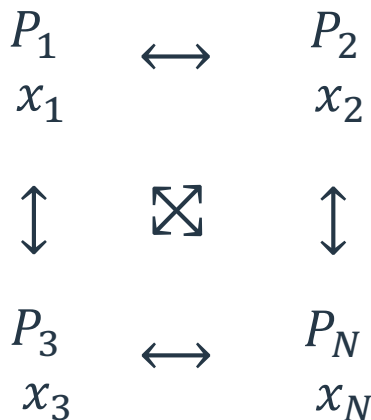
Output:

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



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 = $\Pr[\text{accept} \mid f(x) \neq z]$

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

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

Com_1

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

R

Com_2

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

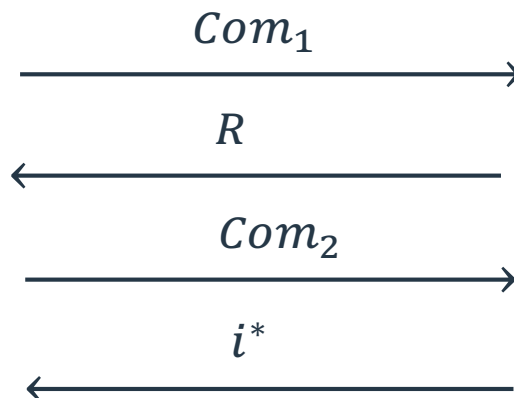
prepare MPC inputs x_i
and commit

Simulate MPC protocol
based on R and commit

Verifier

Sample first challenge R

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

prepare MPC inputs x_i
and commit

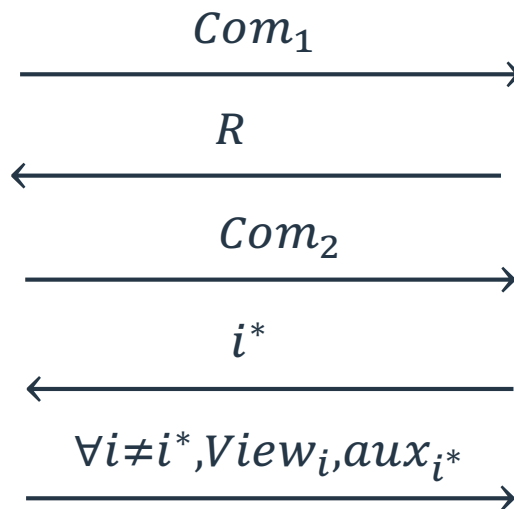
Simulate MPC protocol
based on R and commit

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

Verifier

Sample first challenge R

Sample second challenge i^*



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

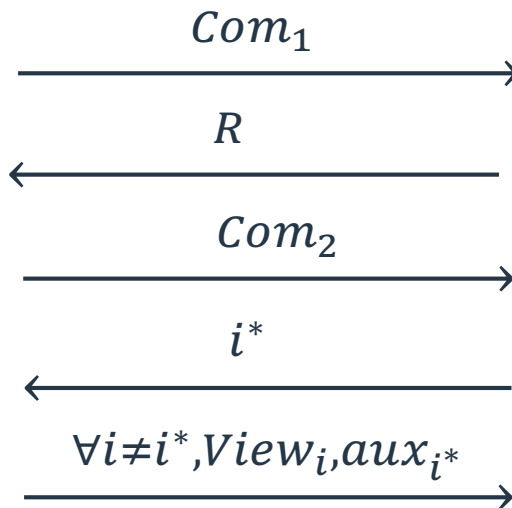
Reveal all views of
Parties $P_i, i \neq i^*$

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 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 $P_i, i \neq i^*$

Com_1

Fiat-Shamir: Signature
Scheme

i^*

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

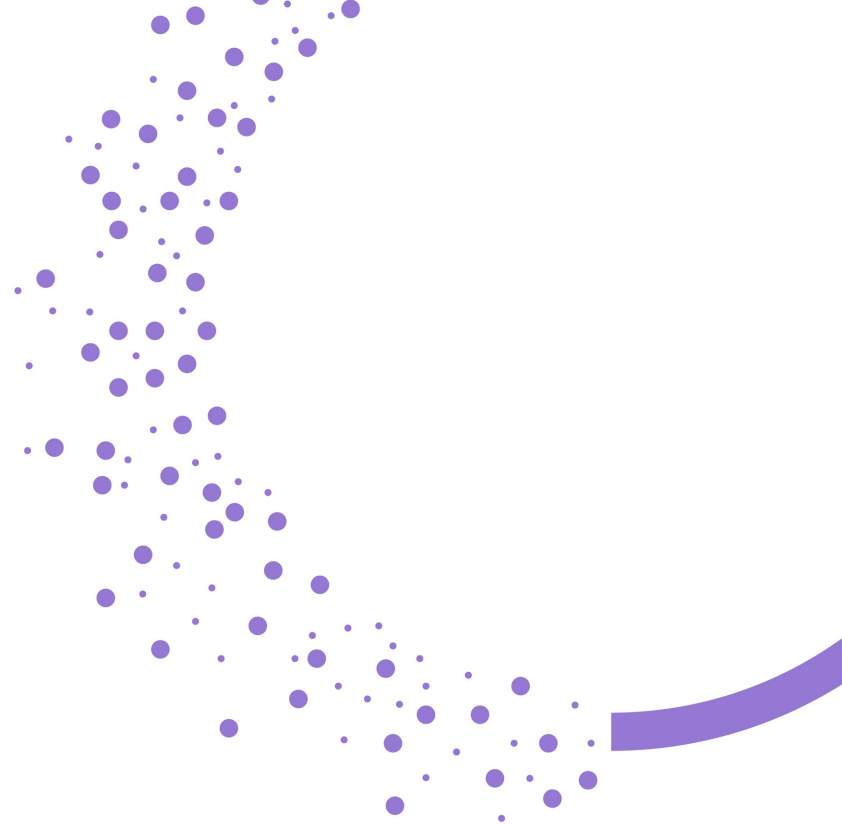
Verifier

Sample first challenge R

Sample second challenge i^*

Check validity of Com_j

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!

$$\beta_i = \alpha_i$$

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

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

Solving system \Rightarrow Solving MinRank!

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

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!

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

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

\Leftrightarrow

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

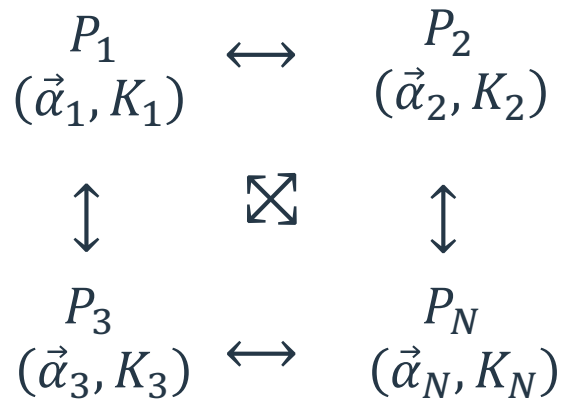
$\vec{\alpha}$ solution of MinRank problem M_0, M_1, \dots, M_k

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

MPC Protocol to verify MinRank Solution

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

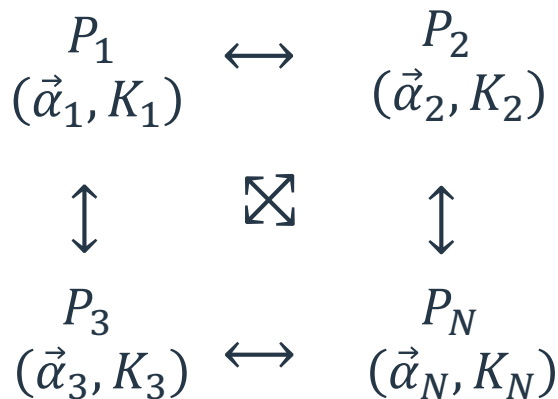
$$\vec{\alpha} = \sum_{i=1}^N \vec{\alpha}_i \quad \text{and} \quad K = \sum_{i=1}^N K_i$$



MPC Protocol to verify MinRank Solution

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

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



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

$$M_{\vec{\alpha}}^L = -M_{\vec{\alpha}}^R \cdot K$$

Output:

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

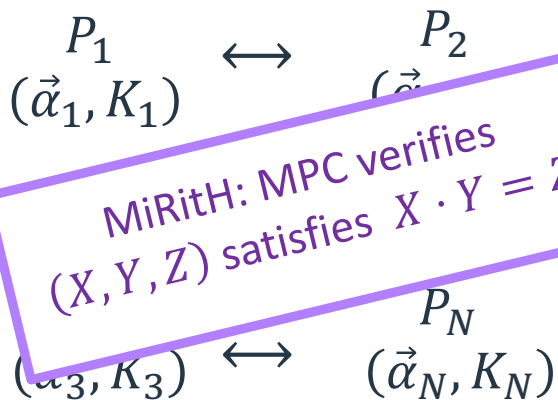
reject : P_i 's think they **don't** share $(\vec{\alpha}, K)$

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

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



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

$$M_{\vec{\alpha}}^L = -M_{\vec{\alpha}}^R \cdot K$$

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

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

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

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

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**

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}

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

Security, Parameters and Performance

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 $\text{rank}(E) \leq r$

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 $\text{rank}(E) \leq r$

1. Kernel Search (combinatorial) : Guess vectors in $\text{kernel}(E)$

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 $\text{rank}(E) \leq r$

1. Kernel Search (combinatorial) : Guess vectors in $\text{kernel}(E)$
2. Support-Minors (algebraic) : Model as bilinear system of equations

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 $\text{rank}(E) \leq r$

1. Kernel Search (combinatorial) : Guess vectors in $\text{kernel}(E)$
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: $E := M_0 + \sum_{i=1}^k \alpha_i M_i \in \mathbb{F}_q^{m \times n}$, and $\text{rank}(E) \leq r$

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

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

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 $\text{rank}(E) \leq r$

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

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