Syndrome Decoding in the Head: Shorter Signatures from Zero-Knowledge Proofs

Thibauld Feneuil[1,2] Antoine Joux[3] Matthieu Rivain[1]

1. CryptoExperts, Paris, France

2. Sorbonne Université, CNRS, INRIA, Institut de Mathématiques de Jussieu-Paris Rive Gauche, Ouragan, Paris, France

3. CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

Seminar C2. *June 10, 2022.*

Table of Contents

Zero-Knowledge Proofs for Syndrome Decoding

Syndrome Decoding Problem

From (H, y), find $x \in \mathbb{F}^m$ such that

$$y = Hx \quad \text{and} \quad \mathrm{wt}_H(x) \leq w.$$

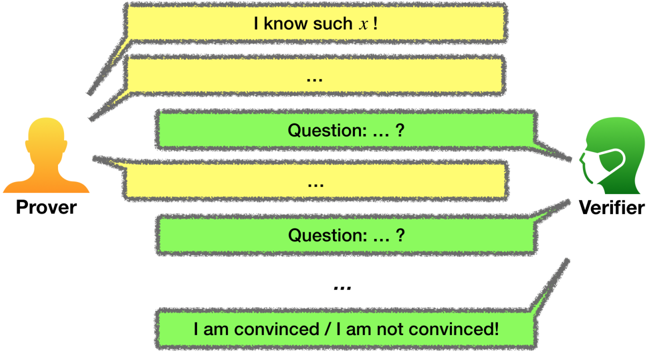
$\mathrm{wt}_H(x) :=$ *nb of non-zero coordinates of* x

Zero-Knowledge Proofs for Syndrome Decoding

Syndrome Decoding Problem

From (H, y), find $x \in \mathbb{F}^m$ such that

$$y = Hx \quad \text{and} \quad \mathrm{wt}_H(x) \leq w.$$

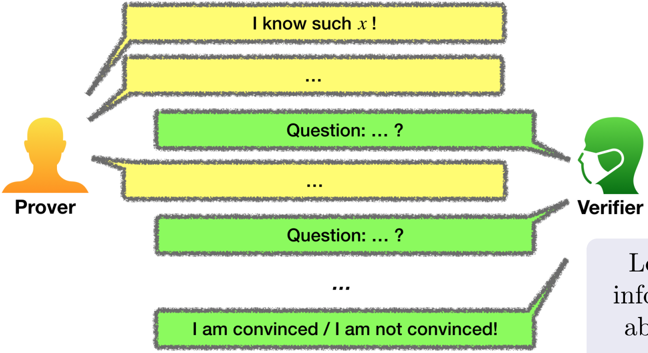


Zero-Knowledge Proofs for Syndrome Decoding

Syndrome Decoding Problem

From (H, y), find $x \in \mathbb{F}^m$ such that

$$y = Hx \quad \text{and} \quad \mathrm{wt}_H(x) \leq w.$$



Learn no information about the secret x.

MPC-in-the-Head Paradigm

MPC-in-the-Head Paradigm

- Generic technique to build *zero-knowledge protocols* using *multi-party computation*.
- Introduced in 2007 by:

  [IKOS07] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. *Zero-knowledge from secure multiparty computation.* STOC 2007.

- Popularized in 2016 by *Picnic*, a candidate of the NIST Post-Quantum Cryptography Standardization.

Sharing of the secret

The secret x satisfies

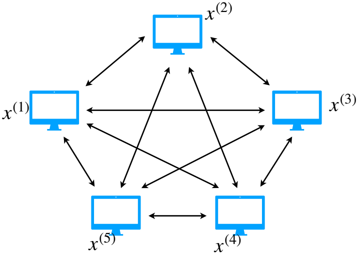
$$y = Hx \qquad \text{and} \qquad \mathrm{wt}_H(x) \leq w.$$

We share it in N parts:

$$x = x^{(1)} + x^{(2)} + \ldots + x^{(N-1)} + x^{(N)}.$$

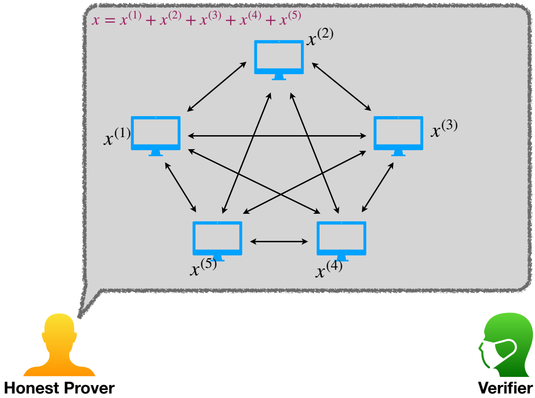
MPC-in-the-Head Paradigm

$x = x^{(1)} + x^{(2)} + x^{(3)} + x^{(4)} + x^{(5)}$

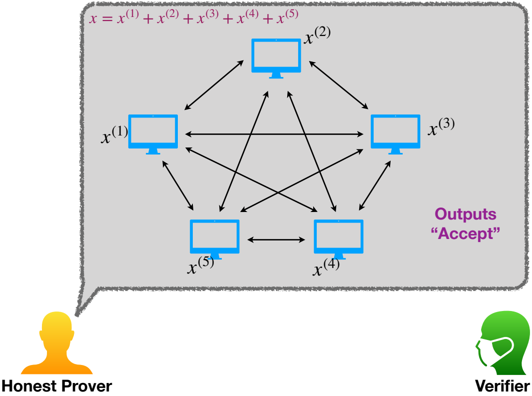


The multi-party computation outputs
- *Accept* if x is a syndrome decoding solution,
- *Reject* otherwise.

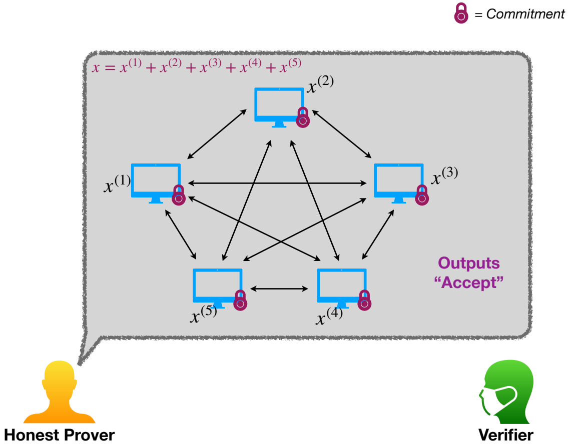
MPC-in-the-Head Paradigm



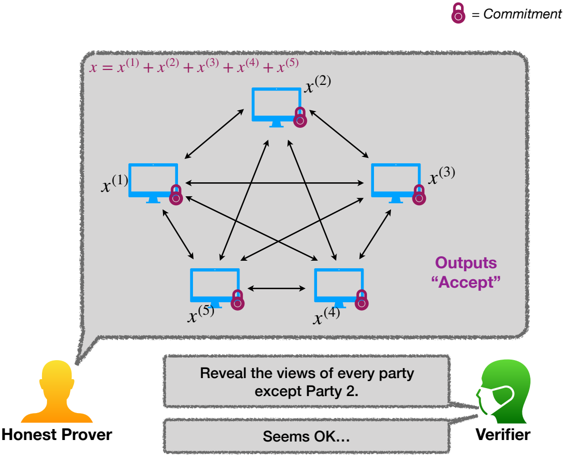
MPC-in-the-Head Paradigm



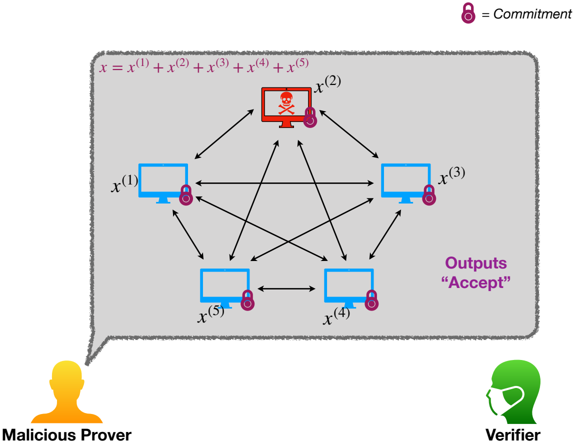
MPC-in-the-Head Paradigm



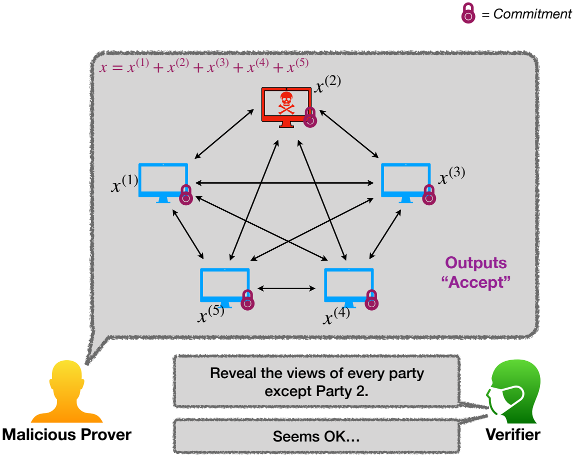
MPC-in-the-Head Paradigm



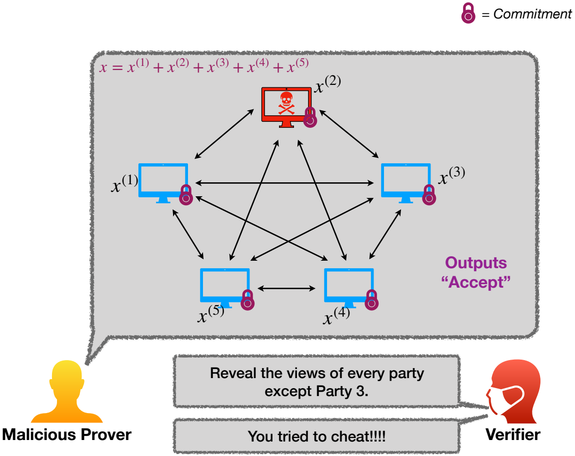
MPC-in-the-Head Paradigm



MPC-in-the-Head Paradigm



MPC-in-the-Head Paradigm



MPC-in-the-Head Paradigm

Soundness error:

$$\frac{1}{N}$$

Proof size: depends on the multi-party computation protocol

Two possible trade-offs:

- Repeat the protocol many times:

  fast proofs, but large proofs

- Take a larger N:

  short proofs, but slow proofs

Table of Contents

Rephrase the constraint

The secret x satisfies

$$\underbrace{y = Hx}_{\text{linear, easy to prove}} \qquad \text{and} \qquad \underbrace{\mathrm{wt}_H(x) \leq w}_{\text{non-linear, hard to prove}}$$

Rephrase the constraint

Let $x \in \mathbb{F}_{\text{SD}}^m$.
To show that $\text{wt}_H(x) \leq w$, we prove there exists $Q \in \mathbb{F}_{\text{poly}}[X]$ s.t.

$$\begin{cases} x_1 \cdot Q(\gamma_1) = 0 \\ x_2 \cdot Q(\gamma_2) = 0 \\ \vdots \\ x_m \cdot Q(\gamma_m) = 0 \end{cases}$$

where

- the degree of Q is **exactly** w,
- $\mathbb{F}_{\text{poly}}$ is a field extension of $\mathbb{F}_{\text{SD}}$,
- $\gamma_1, \ldots, \gamma_m$ are distinct elements of $\mathbb{F}_{\text{poly}}$.

Rephrase the constraint

Let $x \in \mathbb{F}_{\mathrm{SD}}^m$.
To prove that $\mathrm{wt}_H(x) \leq w$, we prove there exists $Q \in \mathbb{F}_{\mathrm{poly}}[X]$ s.t.

$$\begin{cases} S(\gamma_1) \cdot Q(\gamma_1) = 0 \\ S(\gamma_2) \cdot Q(\gamma_2) = 0 \\ \vdots \\ S(\gamma_m) \cdot Q(\gamma_m) = 0 \end{cases}$$

where

the degree of Q is exactly w,

$\mathbb{F}_{\mathrm{poly}}$ is a field extension of $\mathbb{F}_{\mathrm{SD}}$,

$\gamma_1, \ldots, \gamma_m$ are distinct elements of $\mathbb{F}_{\mathrm{poly}}$,

S is built by interpolation such that

$$\forall i, \ S(\gamma_i) = x_i.$$

Rephrase the constraint

Let $x \in \mathbb{F}_{\text{SD}}^m$.
To prove that $\text{wt}_H(x) \leq w$, we prove there exists $Q \in \mathbb{F}_{\text{poly}}[X]$ s.t.

$$S \cdot Q \text{ is equal to zero on } \{\gamma_1, \ldots, \gamma_m\}.$$

where

- the degree of Q is exactly w,
- $\mathbb{F}_{\text{poly}}$ is a field extension of $\mathbb{F}_{\text{SD}}$,
- $\gamma_1, \ldots, \gamma_m$ are distinct elements of $\mathbb{F}_{\text{poly}}$,
- S is built by interpolation such that

$$\forall i, \; S(\gamma_i) = x_i.$$

Rephrase the constraint

If the prover convinces the verifier that there exists $Q, P \in \mathbb{F}_{\text{poly}}[X]$ s.t.

$$S \cdot Q = P \cdot F$$

where

the degree of Q is exactly w,

S is built by interpolation such that $\forall i,\ S(\gamma_i) = x_i$,

$F := \prod_{i=1}^{m}(X - \gamma_i)$,

then, the verifier deduces that

$$\forall i \leq m,\ (Q \cdot S)(\gamma_i) = P(\gamma_i) \cdot F(\gamma_i) = 0$$
$$\Rightarrow\ \forall i \leq m,\ Q(\gamma_i) = 0 \ \text{ or } \ S(\gamma_i) = x_i = 0$$

Rephrase the constraint

If the prover convinces the verifier that there exists $Q, P \in \mathbb{F}_{\text{poly}}[X]$ s.t.

$$S \cdot Q = P \cdot F$$

where

- the degree of Q is exactly w,
- S is built by interpolation such that $\forall i,\ S(\gamma_i) = x_i$,
- $F := \prod_{i=1}^{m}(X - \gamma_i)$,

then, the verifier deduces that

$$\forall i \leq m,\ (Q \cdot S)(\gamma_i) = P(\gamma_i) \cdot F(\gamma_i) = 0$$
$$\Rightarrow\ \forall i \leq m,\ Q(\gamma_i) = 0 \ \text{ or } \ S(\gamma_i) = x_i = 0$$

i.e.

$$\mathrm{wt}_H(x) \leq w$$

Rephrase the constraint

The solution x of the syndrome decoding problem must satisfy

$$\mathbf{y} = \mathbf{Hx}$$

and

$$\exists\ \mathbf{Q}, \mathbf{P} \text{ two polynomials} : \mathbf{SQ} = \mathbf{PF} \text{ and } \deg \mathbf{Q} = \mathbf{w}$$

where

S is defined by interpolation such that $\forall i,\ S(\gamma_i) = x_i$,
and $F := \prod_{i=1}^{m}(X - \gamma_i)$.

Guidelines for the MPC Protocol

We want to build a MPC protocol which check if some vector is a syndrome decoding solution.

Let us assume $H = (H'|I)$. We split x as $\begin{pmatrix} x_A \\ x_B \end{pmatrix}$.
We have $y = Hx$, so

$$x_B = y - H'x_A.$$

Guidelines for the MPC Protocol

Inputs: x_A, Q, P.

1. Build $x_B := y - H' x_A$ and deduce $x := \begin{pmatrix} x_A \\ x_B \end{pmatrix}$.

   We have
   $$y = Hx.$$

Guidelines for the MPC Protocol

Inputs: x_A, Q, P.

1. Build $x_B := y - H'x_A$ and deduce $x := \begin{pmatrix} x_A \\ x_B \end{pmatrix}$.
2. Build the polynomial S by interpolation such that

$$\forall i \in \{1, \dots, m\}, S(\gamma_i) = x_i.$$

Guidelines for the MPC Protocol

Inputs: x_A, Q, P.

1. Build $x_B := y - H'x_A$ and deduce $x := \begin{pmatrix} x_A \\ x_B \end{pmatrix}$.
2. Build the polynomial S by interpolation such that

$$\forall i \in \{1, \ldots, m\}, S(\gamma_i) = x_i.$$

3. Check that $S \cdot Q = P \cdot F$.

Guidelines for the MPC Protocol

Inputs: x_A, Q, P.

1. Build $x_B := y - H'x_A$ and deduce $x := \begin{pmatrix} x_A \\ x_B \end{pmatrix}$.
2. Build the polynomial S by interpolation such that

$$\forall i \in \{1, \ldots, m\}, S(\gamma_i) = x_i.$$

3. Check that $S \cdot Q = P \cdot F$.

	Output of π	
	ACCEPT	REJECT
A good witness	1	0
Not a good witness	0	1

Guidelines for the MPC Protocol

Inputs: x_A, Q, P.

1. Build $x_B := y - H' x_A$ and deduce $x := \begin{pmatrix} x_A \\ x_B \end{pmatrix}$.
2. Build the polynomial S by interpolation such that
$$\forall i \in \{1, \dots, m\}, S(\gamma_i) = x_i.$$
3. Get a random point $r \in \mathbb{F}_{\text{points}}$ and check that $S(r) \cdot Q(r) = P(r) \cdot F(r)$.

$\mathbb{F}_{points}$ is a field extension of $\mathbb{F}_{poly}$.

Guidelines for the MPC Protocol

Inputs: x_A, Q, P.

1. Build $x_B := y - H' x_A$ and deduce $x := \begin{pmatrix} x_A \\ x_B \end{pmatrix}$.
2. Build the polynomial S by interpolation such that

$$\forall i \in \{1, \dots, m\}, S(\gamma_i) = x_i.$$

3. Get a random point $r \in \mathbb{F}_{\text{points}}$ and check that $S(r) \cdot Q(r) = P(r) \cdot F(r)$.

Schwartz-Zippel Lemma: If $S \cdot Q \neq P \cdot F$, then

$$\Pr_{r \xleftarrow{\$} \mathbb{F}_{\text{points}}} [S(r) \cdot Q(r) = P(r) \cdot F(r)] \leq \frac{m + w - 1}{|\mathbb{F}_{\text{points}}|}$$

Guidelines for the MPC Protocol

Inputs: x_A, Q, P.

1. Build $x_B := y - H'x_A$ and deduce $x := \begin{pmatrix} x_A \\ x_B \end{pmatrix}$.
2. Build the polynomial S by interpolation such that

$$\forall i \in \{1, \dots, m\}, S(\gamma_i) = x_i.$$

3. Get a random point $r \in \mathbb{F}_{\text{points}}$ and check that $S(r) \cdot Q(r) = P(r) \cdot F(r)$.

	Output of π	
	ACCEPT	REJECT
A good witness	1	0
Not a good witness	p	$1 - p$

with $p \leq \frac{m+w-1}{|\mathbb{F}_{\text{points}}|}$ by the **Schwartz-Zippel Lemma**.

Guidelines for the MPC Protocol

Inputs: x_A, Q, P.

1. Build $x_B := y - H' x_A$ and deduce $x := \begin{pmatrix} x_A \\ x_B \end{pmatrix}$.
2. Build the polynomial S by interpolation such that

$$\forall i \in \{1, \dots, m\}, S(\gamma_i) = x_i.$$

3. Get a random point $r \in \mathbb{F}_{\text{points}}$.
4. Compute $S(r)$, $Q(r)$ and $P(r)$.
5. Using [BN20], check that $S(r) \cdot Q(r) = P(r) \cdot F(r)$.

[BN20] Carsten Baum and Ariel Nof. *Concretely-efficient zero-knowledge arguments for arithmetic circuits and their application to lattice-based cryptography.* PKC 2020.

Sharing of the MPC input

$$\begin{array}{ccccccccccc} & & \mathcal{P}_1 & & \mathcal{P}_2 & & \dots & & \mathcal{P}_N & \\ x_A & = & [\![x_A]\!]_1 & + & [\![x_A]\!]_2 & + & \dots & + & [\![x_A]\!]_N & \in \mathbb{F}_{\mathrm{SD}}^k \\ Q & = & [\![Q]\!]_1 & + & [\![Q]\!]_2 & + & \dots & + & [\![Q]\!]_N & \in \mathbb{F}_{\mathrm{poly}}[X] \\ P & = & [\![P]\!]_1 & + & [\![P]\!]_2 & + & \dots & + & [\![P]\!]_N & \in \mathbb{F}_{\mathrm{poly}}[X] \end{array}$$

Operations on sharings

Addition: $[\![v_1 + v_2]\!] = [\![v_1]\!] + [\![v_2]\!]$

$$\forall i,\ [\![v_1 + v_2]\!]_i := [\![v_1]\!]_i + [\![v_2]\!]_i$$

Operations on sharings

Addition: $[\![v_1 + v_2]\!] = [\![v_1]\!] + [\![v_2]\!]$

$$\forall i,\ [\![v_1 + v_2]\!]_i := [\![v_1]\!]_i + [\![v_2]\!]_i$$

Addition with a constant: $[\![v + \alpha]\!] = [\![v]\!] + \alpha$

$$\begin{cases} [\![v + \alpha]\!]_1 := [\![v]\!]_1 + \alpha \\ [\![v + \alpha]\!]_i := [\![v]\!]_i \quad \text{for } i \neq 1 \end{cases}$$

Operations on sharings

Addition: $[\![v_1 + v_2]\!] = [\![v_1]\!] + [\![v_2]\!]$

$$\forall i,\ [\![v_1 + v_2]\!]_i := [\![v_1]\!]_i + [\![v_2]\!]_i$$

Addition with a constant: $[\![v + \alpha]\!] = [\![v]\!] + \alpha$

$$\begin{cases} [\![v + \alpha]\!]_1 := [\![v]\!]_1 + \alpha \\ [\![v + \alpha]\!]_i := [\![v]\!]_i \quad \text{for } i \neq 1 \end{cases}$$

Multiplication by a constant: $[\![\alpha \cdot v]\!] = \alpha \cdot [\![v]\!]$

$$\forall i,\ [\![\alpha \cdot v]\!]_i := \alpha \cdot [\![v]\!]_i$$

The MPC Protocol

Inputs of the party $\mathcal{P}_i$: $[\![x_A]\!]_i$, $[\![Q]\!]_i$ and $[\![P]\!]_i$.

1. Compute $[\![x_B]\!] = y - H'[\![x_A]\!]$, and then deduce $[\![x]\!]$.

The MPC Protocol

Inputs of the party $\mathcal{P}_i$: $[\![x_A]\!]_i$, $[\![Q]\!]_i$ and $[\![P]\!]_i$.

1. Compute $[\![x_B]\!] = y - H'[\![x_A]\!]$, and then deduce $[\![x]\!]$.
2. Compute $[\![S]\!]$ from $[\![x]\!]$ by interpolation such that

$$\forall i \in \{1, \dots, m\}, S(\gamma_i) = x_i.$$

The MPC Protocol

Inputs of the party $\mathcal{P}_i$: $[\![x_A]\!]_i$, $[\![Q]\!]_i$ and $[\![P]\!]_i$.

1. Compute $[\![x_B]\!] = y - H'[\![x_A]\!]$, and then deduce $[\![x]\!]$.
2. Compute $[\![S]\!]$ from $[\![x]\!]$ thanks to

$$[\![S(X)]\!] = \sum_i [\![x_i]\!] \cdot \prod_{\ell \neq i} \frac{X - \gamma_\ell}{\gamma_i - \gamma_\ell} .$$

The MPC Protocol

Inputs of the party $\mathcal{P}_i$: $[\![x_A]\!]_i$, $[\![Q]\!]_i$ and $[\![P]\!]_i$.

1. Compute $[\![x_B]\!] = y - H'[\![x_A]\!]$, and then deduce $[\![x]\!]$.
2. Compute $[\![S]\!]$ from $[\![x]\!]$ thanks to

$$[\![S(X)]\!] = \sum_i [\![x_i]\!] \cdot \prod_{\ell \neq i} \frac{X - \gamma_\ell}{\gamma_i - \gamma_\ell} .$$

3. Get a random point $r \in \mathbb{F}_{\text{points}}$ from a trusted source.
4. Compute

$$\begin{cases} [\![S(r)]\!] = [\![S]\!](r) \\ [\![Q(r)]\!] = [\![Q]\!](r) \\ [\![P(r)]\!] = [\![P]\!](r) \end{cases}$$

5. Using [BN20], check that $S(r) \cdot Q(r) = P(r) \cdot F(r)$.

Summary

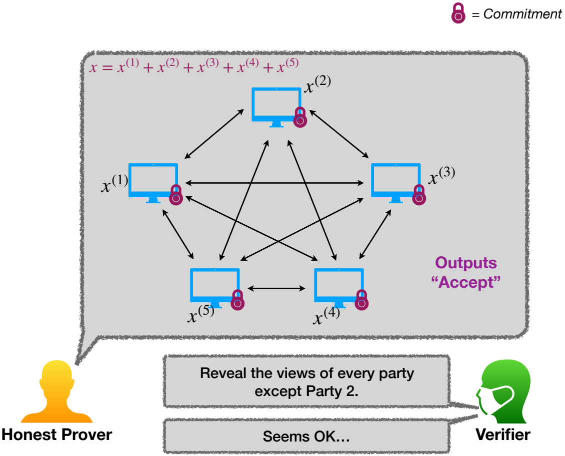
The MPC protocol π checks that $([\![x_A]\!], [\![Q]\!], [\![P]\!])$ describes a solution of the SD instance (H, y).

	Output of π	
	Accept	Reject
A good witness	1	0
Not a good witness	p	$1-p$

where

$$p = \underbrace{\frac{m+w-1}{|\mathbb{F}_{\text{points}}|}}_{\substack{\text{false positive} \\ \text{from Schwartz-Zippel}}} + \left(1 - \frac{m+w-1}{|\mathbb{F}_{\text{points}}|}\right) \cdot \underbrace{\frac{1}{|\mathbb{F}_{\text{points}}|}}_{\substack{\text{false positive} \\ \text{from [BN20]}}}$$

MPC-in-the-Head paradigm



MPC-in-the-Head paradigm

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
H, y, x such that $y = Hx$ and $\mathrm{wt}_H(x) \leq w$		H, y
Prepare Q, P. $\textsc{Com}_i \leftarrow \mathrm{Com}(\text{inputs of } \mathcal{P}_i)$	$\xrightarrow{\textsc{Com}_1,\ldots,\textsc{Com}_N}$	$r \in \mathbb{F}_{\text{points}}$
Run the MPC protocol π for each party.	$\xleftarrow{r}$	
	$\xrightarrow{\text{broadcast messages}}$	$i^* \xleftarrow{\$} \{1, \ldots, N\}$
	$\xleftarrow{i^*}$	
	$\xrightarrow{\text{all } V_i \text{ for } i \neq i^*}$	
		Check that the views are consistent. Check that the MPC output is $\textsc{Accept}$

Zero-Knowledge Protocol

Soundness error:

$$p + (1 - p) \cdot \frac{1}{N}$$

Zero-Knowledge Protocol

Soundness error:

$$p + (1-p)\cdot\frac{1}{N}$$

Proof size:

- Inputs of $N-1$ parties:

$$\begin{array}{ccccccccccc} & & \mathcal{P}_1 & & \mathcal{P}_2 & & \dots & & \mathcal{P}_{N-1} & & \mathcal{P}_N \\ x_A & = & [\![x_A]\!]_1 & + & [\![x_A]\!]_2 & + & \dots & + & [\![x_A]\!]_{N-1} & + & [\![x_A]\!]_N \\ Q & = & [\![Q]\!]_1 & + & [\![Q]\!]_2 & + & \dots & + & [\![Q]\!]_{N-1} & + & [\![Q]\!]_N \\ P & = & [\![P]\!]_1 & + & [\![P]\!]_2 & + & \dots & + & [\![P]\!]_{N-1} & + & [\![P]\!]_N \\ & & \uparrow & & \uparrow & & & & \uparrow & & \\ & & \mathsf{seed}_1 & & \mathsf{seed}_2 & & & & \mathsf{seed}_{N-1} & & \end{array}$$

Zero-Knowledge Protocol

Soundness error:

$$p + (1-p) \cdot \frac{1}{N}$$

Proof size:

- Inputs of $N-1$ parties:
  - Party $i < N$: a seed of λ bits
  - Last party:

$$\underbrace{k \cdot \log_2 |\mathbb{F}_{\text{SD}}|}_{[\![x_A]\!]_N} + \underbrace{2w \cdot \log_2 |\mathbb{F}_{\text{poly}}|}_{[\![Q]\!]_N, [\![P]\!]_N} + \underbrace{\lambda}_{[\![a]\!]_N, [\![b]\!]_N} + \underbrace{\log_2 |\mathbb{F}_{\text{points}}|}_{[\![c]\!]_N}$$

Zero-Knowledge Protocol

Soundness error:

$$p + (1-p) \cdot \frac{1}{N}$$

Proof size:

- Inputs of $N-1$ parties:
  - Party $i < N$: a seed of λ bits
  - Last party:

$$\underbrace{k \cdot \log_2 |\mathbb{F}_{\text{SD}}|}_{[\![x_A]\!]_N} + \underbrace{2w \cdot \log_2 |\mathbb{F}_{\text{poly}}|}_{[\![Q]\!]_N, [\![P]\!]_N} + \underbrace{\lambda}_{[\![a]\!]_N, [\![b]\!]_N} + \underbrace{\log_2 |\mathbb{F}_{\text{points}}|}_{[\![c]\!]_N}$$

- Communication between parties: 2 elements of $\mathbb{F}_{\text{points}}$.
- 2 hash digests ($2 \times 2\lambda$ bits),
- Some commitment randomness + COM_{i^*}

State of the art about ZK PoK for SD

Only for unstructured syndrom decoding problems.

Protocol	**Year**	**Assumption**	**Soundness err.**
Stern's	1993	SD	$2/3$
Véron's	1997	SD	$2/3$
CVE's	2010	SD on $\mathbb{F}_q$	$\approx 1/2$

State of the art about ZK PoK for SD

Only for unstructured syndrom decoding problems.

Protocol	**Year**	**Assumption**	**Soundness err.**
Stern's	1993	SD	$2/3$
Véron's	1997	SD	$2/3$
CVE's	2010	SD on $\mathbb{F}_q$	$\approx 1/2$
GPS's	2021	SD on $\mathbb{F}_q$	$\approx 1/N$

[GPS21] Shay Gueron, Edoardo Persichetti, and Paolo Santini. *Designing a Practical Code-based Signature Scheme from Zero-Knowledge Proofs with Trusted Setup*. Cryptography 2022.

State of the art about ZK PoK for SD

Only for unstructured syndrom decoding problems.

Protocol	Year	Assumption	Soundness err.
Stern's	1993	SD	$2/3$
Véron's	1997	SD	$2/3$
CVE's	2010	SD on $\mathbb{F}_q$	$\approx 1/2$
GPS's	2021	SD on $\mathbb{F}_q$	$\approx 1/N$
FJR21's	2021	SD	$\approx 1/N$

$$\sigma = \sigma_N \circ \sigma_{N-1} \circ \ldots \circ \sigma_3 \circ \sigma_2 \circ \sigma_1$$

[FJR21] Thibauld Feneuil, Antoine Joux, and Matthieu Rivain. *Shared Permutation for Syndrome Decoding: New Zero-Knowledge Protocol and Code-Based Signature.* Eprint 2021/1576.

State of the art about ZK PoK for SD

Only for unstructured syndrom decoding problems.

Protocol	Year	Assumption	Soundness err.
Stern's	1993	SD	$2/3$
Véron's	1997	SD	$2/3$
CVE's	2010	SD on $\mathbb{F}_q$	$\approx 1/2$
GPS's	2021	SD on $\mathbb{F}_q$	$\approx 1/N$
FJR21's	2021	SD	$\approx 1/N$
BGKM's	2022	SD	$\approx 1/N$

[BGKM22] Loïc Bidoux, Philippe Gaborit, Mukul Kulkarni, Victor Mateu. *Code-based Signatures from New Proofs of Knowledge for the Syndrome Decoding Problem.* arXiv 2110.05005.

State of the art about ZK PoK for SD

Only for unstructured syndrom decoding problems.

Protocol	Year	Assumption	Soundness err.
Stern's	1993	SD	$2/3$
Véron's	1997	SD	$2/3$
CVE's	2010	SD on $\mathbb{F}_q$	$\approx 1/2$
GPS's	2021	SD on $\mathbb{F}_q$	$\approx 1/N$
FJR21's	2021	SD	$\approx 1/N$
BGKM's	2022	SD	$\approx 1/N$
FJR22's	2022	SD	$\approx 1/N$

Prove $\mathrm{wt}_H(x) \leq w$, not $\mathrm{wt}_H(x) = w$.

$$Q(X) = \prod_{i: x_i \neq 0} (X - \gamma_i), \quad \deg Q = w$$

[FJR22] Thibauld Feneuil, Antoine Joux, Matthieu Rivain. *Syndrome Decoding in the Head: Shorter Signatures from Zero-Knowledge Proofs.* Crypto 2022.

Comparison Zero-Knowledge Protocol for SD

Name Protocol	Year	Instance 1	Instance 2
Stern	1993	37.4 KB	46.1 KB
Véron	1997	31.7 KB	38.7 KB
CVE10	2010	-	37.4 KB
GPS21 (short)	2021	-	**15.2 KB**
GPS21 (fast)	2021	-	19.9 KB
FJR21 (short)	2021	**13.6 KB**	16.4 KB
FJR21 (fast)	2021	20.7 KB	25.6 KB
FJR22 (short)	2022	**9.7 KB**	**6.9 KB**
FJR22 (fast)	2022	14.4 KB	9.7 KB
Field size q		2	256
Code length m		1280	208
Code dimension k		$m/2$	$m/2$
Hamming weight w		132	78
Security level λ		128	128

Prove only an inequality

Table of Contents

Fiat-Shamir Transform

Signature algorithm:

Inputs:

- x such that $y = Hx$ and $\mathrm{wt}_H(x) \leq w$
- the message mess to sign

1. Prepare the witness, *i.e.* the polynomials P and Q.
2. Commit to party's inputs in distinct commitments $\mathrm{COM}_1, \ldots, \mathrm{COM}_N$.
3. $r = \mathrm{Hash}(\mathsf{mess}, \mathsf{salt}, \mathrm{COM}_1, \ldots, \mathrm{COM}_N)$.
4. Run the MPC protocol π for each party.
5. $i^* = \mathrm{Hash}(\mathsf{mess}, \mathsf{salt}, r, \text{broadcast messages})$.
6. Build the signature with the views of all the parties except the party i^*.

Security of the signature

5-round Identification Scheme ⇒ Signature

Attack of [KZ20]:

$$\mathrm{cost}_{\mathrm{forge}} := \min_{\tau_1,\tau_2:\tau_1+\tau_2=\tau} \left\{ \frac{1}{\sum_{i=\tau_1}^{\tau} \binom{\tau}{i} p^i (1-p)^{\tau-i}} + N^{\tau_2} \right\}$$

[KZ20] Daniel Kales and Greg Zaverucha. *An attack on some signature schemes constructed from five-pass identification schemes.* CANS 2020.

Parameters selected

Variant 1: SD over $\mathbb{F}_2$,

$$(m, k, w) = (1280, 640, 132)$$

We have $\mathbb{F}_{poly} = \mathbb{F}_{2^{11}}$.

Parameters selected

Variant 1: SD over $\mathbb{F}_2$,

$$(m, k, w) = (1280, 640, 132)$$

We have $\mathbb{F}_{poly} = \mathbb{F}_{2^{11}}$.

Variant 2: SD over $\mathbb{F}_2$,

$$(m, k, w) = (1536, 888, 120)$$

but we split $x := (x_1 \mid \ldots \mid x_6)$ into 6 chunks and we prove that $\mathrm{wt}_H(x_i) \leq \frac{w}{6}$ for all i.

We have $\mathbb{F}_{poly} = \mathbb{F}_{2^8}$.

Parameters selected

Variant 3: SD over $\mathbb{F}_{2^8}$,

$$(m, k, w) = (256, 128, 80)$$

$$We\ have\ \mathbb{F}_{poly} = \mathbb{F}_{2^8}.$$

Performances

	Security Assumption	Computation Field
Variant 1	Over $\mathbb{F}_2$	$\mathbb{F}_{2048}$
Variant 2	Over $\mathbb{F}_2$	$\mathbb{F}_{256}$
Variant 3	Over $\mathbb{F}_{256}$	$\mathbb{F}_{256}$

Two trade-offs:

Fast: $N = 32$, $\tau = 27$

Short: $N = 256$, $\tau = 17$

Comparison Code-based Signatures (1/2)

Scheme Name	$\lvert \mathsf{sgn} \rvert$	$\lvert \mathsf{pk} \rvert$	t_{sgn}	t_{verif}
BGS21	24.1 KB	0.1 KB	-	-
BGS21	22.5 KB	1.7 KB	-	-
GPS21 - 256	22.2 KB	0.11 KB	-	-
GPS21 - 1024	19.5 KB	0.12 KB	-	-
FJR21 (fast)	22.6 KB	0.09 KB	13 ms	12 ms
FJR21 (short)	16.0 KB	0.09 KB	62 ms	57 ms
BGKM22 - Sig1	23.7 KB	0.1 KB	-	-
BGKM22 - Sig2	20.6 KB	0.2 KB	-	-
FJR22 - $\mathbb{F}_2$ (fast)	15.6 KB	0.09 KB	-	-
FJR22 - $\mathbb{F}_2$ (short)	10.9 KB	0.09 KB	-	-
FJR22 - $\mathbb{F}_2$ (fast)	17.0 KB	0.09 KB	13 ms	13 ms
FJR22 - $\mathbb{F}_2$ (short)	11.8 KB	0.09 KB	64 ms	61 ms
FJR22 - $\mathbb{F}_{256}$ (fast)	11.5 KB	0.14 KB	6 ms	6 ms
FJR22 - $\mathbb{F}_{256}$ (short)	**8.26 KB**	0.14 KB	30 ms	27 ms

Comparison Code-based Signatures (2/2)

Scheme Name	$\|\mathsf{sgn}\|$	$\|\mathsf{pk}\|$	t_{sgn}	t_{verif}
Durandal - I	3.97 KB	14.9 KB	4 ms	5 ms
Durandal - II	4.90 KB	18.2 KB	5 ms	6 ms
LESS-FM - I	15.2 KB	9.78 KB	-	-
LESS-FM - II	5.25 KB	205 KB	-	-
LESS-FM - III	10.39 KB	11.57 KB	-	-
Wave	**2.07 KB**	3.1 MB	$\geq$ 300 ms	2 ms
Wavelet	**0.91 KB**	3.1 MB	$\geq$ 300 ms	$\leq$ 1 ms
FJR22 - $\mathbb{F}_2$ (fast)	15.6 KB	0.09 KB	-	-
FJR22 - $\mathbb{F}_2$ (short)	10.9 KB	0.09 KB	-	-
FJR22 - $\mathbb{F}_2$ (fast)	17.0 KB	0.09 KB	13 ms	13 ms
FJR22 - $\mathbb{F}_2$ (short)	11.8 KB	0.09 KB	64 ms	61 ms
FJR22 - $\mathbb{F}_{256}$ (fast)	11.5 KB	0.14 KB	6 ms	6 ms
FJR22 - $\mathbb{F}_{256}$ (short)	**8.26 KB**	**0.14 KB**	30 ms	27 ms

Conclusion

Summary

- ☞ New signature scheme with Syndrome Decoding
- ☞ Conservative scheme (SD on random linear codes)
- ☞ Small "signature size + public key size"

Future Work

- ☞ Optimize the signature implementation.
- ☞ Search (aggressive) parameter sets which provide better performances.

More details in `https://eprint.iacr.org/2022/188`.

Contact: `thibauld.feneuil@cryptoexperts.com`