Building MPCitH-based Signatures with Some Classical Hardness Assumptions

Thibauld Feneuil

CryptoExperts, Paris, France

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

NIST. *February 7, 2023.*

Table of Contents

Methodology

Hard Problem

↓ ***MPC-in-the-Head***

Zero Knowledge Proof of Knowledge

↓ ***Fiat-Shamir Transform***

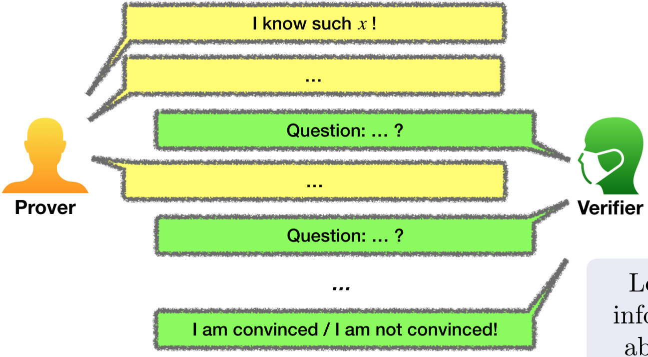
Signature Scheme

Zero-Knowledge Proofs of Knowledge

Let have a circuit C and an output y.
Problem: find x such that $C(x) = y$.

Zero-Knowledge Proofs of Knowledge

Let have a circuit C and an output y.
Problem: find x such that $C(x) = y$.



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 former candidate of the NIST Post-Quantum Cryptography Standardization.

Sharing of the secret

The secret x satisfies

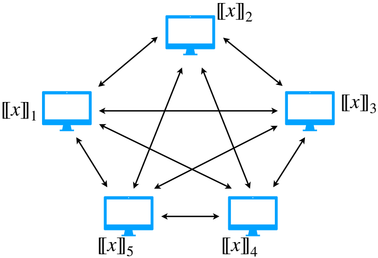
$$y = C(x).$$

We share it in N parts:

$$x = [\![x]\!]_1 + [\![x]\!]_2 + \ldots + [\![x]\!]_{N-1} + [\![x]\!]_N.$$

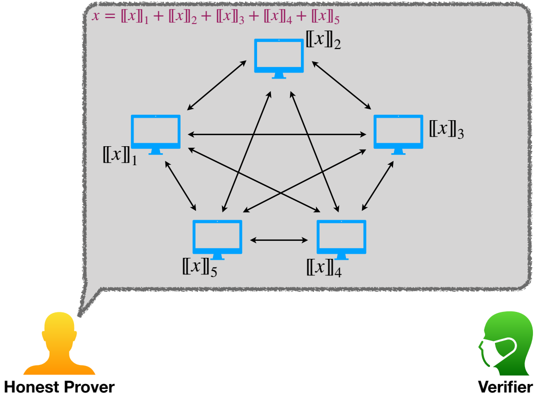
MPC-in-the-Head Paradigm

$x = [\![x]\!]_1 + [\![x]\!]_2 + [\![x]\!]_3 + [\![x]\!]_4 + [\![x]\!]_5$

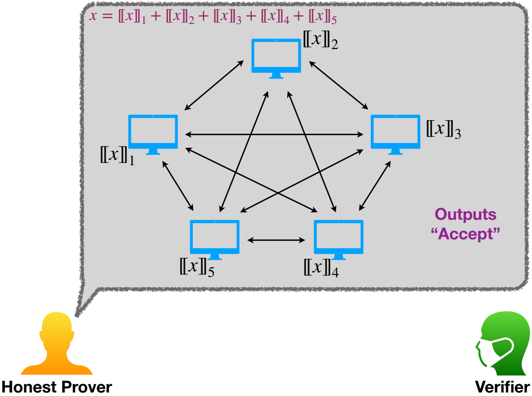


The multi-party computation outputs
- *Accept* if x satisfies $y = C(x)$,
- *Reject* otherwise.

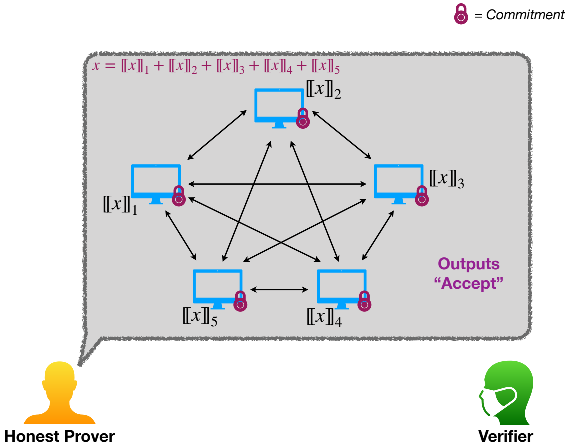
MPC-in-the-Head Paradigm



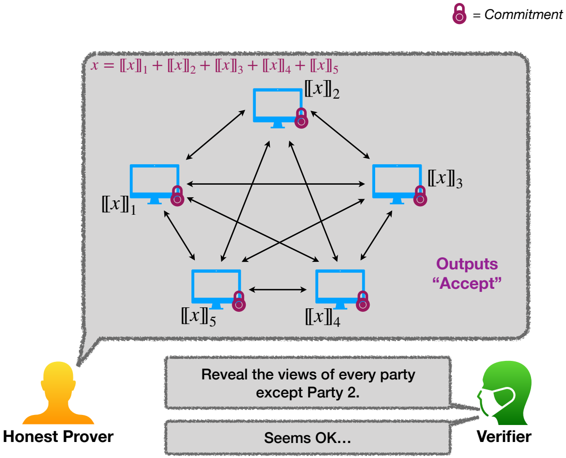
MPC-in-the-Head Paradigm



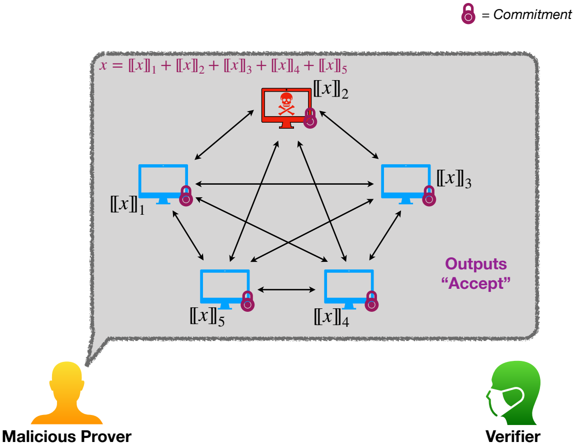
MPC-in-the-Head Paradigm



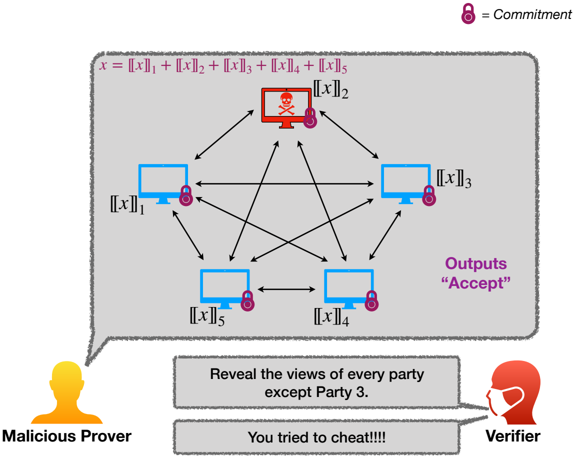
MPC-in-the-Head Paradigm



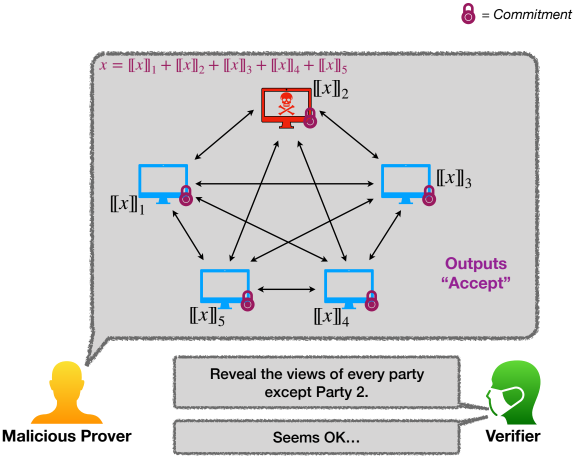
MPC-in-the-Head Paradigm



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

From ID scheme to signature scheme

To get a signature scheme, we use

☞ the Fiat-Shamir Transformation.

The First MPCitH-based Signatures

Scheme Name	Year	$\lvert\mathsf{sgn}\rvert$	Assumption
Picnic1 [CDG+17]	2016	32.1 KB	LowMC (partial)
Picnic2 [KKW18]	2018	12.1 KB	
Picnic3 [KZ20b]	2019	12.3 KB	LowMC (full)
Helium+LowMC [KZ22]	2022	6.4 - 9.2 KB*	
BBQ [dDOS19]	2019	30.9 KB	AES
Banquet [BdK+21]	2021	13.0 - 17.1 KB*	
Limbo-Sign [dOT21]	2021	14.2 - 17.9 KB*	
Helium+AES [KZ22]	2022	9.7 - 14.4 KB*	
Rainier [DKR+21]	2021	5.9 - 8.1 KB*	Rain
BN++Rain [KZ22]	2022	4.9 - 6.4 KB*	

*sizes given for a range of 32-256 parties.

Table of Contents

Signature with Syndrome Decoding Problem

Idea:

Instead of relying on AES or on MPC-friendly primitives, we can rely on hard problems from asymmetric crypto.

The case of the Syndrome Decoding in Hamming metric:

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

Rephrase the constraint

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

The multi-party computation must check that the vector x satisfies

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

Rephrase the constraint

The multi-party computation must check that the vector x satisfies

$$y = Hx$$

and

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

where

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

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

Rephrase the constraint

Let us assume that there exists $Q, P \in \mathbb{F}_{\text{poly}}[X]$ s.t.

$$S \cdot Q = P \cdot F \qquad \text{and} \qquad \deg Q = w$$

where

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

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

Rephrase the constraint

Let us assume that there exists $Q, P \in \mathbb{F}_{\text{poly}}[X]$ s.t.

$$S \cdot Q = P \cdot F \qquad \text{and} \qquad \deg Q = w$$

where

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.

$$\text{wt}_H(x) := \#\{i : x_i \neq 0\} \leq w$$

Rephrase the constraint

Such polynomial Q can be built as

$$Q := Q' \cdot \underbrace{\prod_{i : x_i \neq 0} (X - \gamma_i)}_{\text{The non-zero positions of } x \text{ are encoding as roots.}}$$

And $P := \frac{S \cdot Q}{F}$ since F divides $S \cdot Q$.

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

Guidelines for the MPC Protocol

We want to build a MPC protocol which checks 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

We want to build a MPC protocol which checks 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

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

Inputs of the MPC protocol: x_A, Q, P.
Aim of the MPC protocol:

Check that x_A corresponds to a syndrome decoding solution.

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

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

Interpolation Formula:

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

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. Get a random point r from $\mathbb{F}_{\text{points}}$ (field extension of $\mathbb{F}_{\text{poly}}$).
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.

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 deduce $[\![x]\!] := \begin{pmatrix} [\![x_A]\!] \\ [\![x_B]\!] \end{pmatrix}$.
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 from $\mathbb{F}_{\text{points}}$ (field extension of $\mathbb{F}_{\text{poly}}$).
4. Compute

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

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

Analysis

Even if x_A does not describe a SD solution (implying that $S \cdot Q \neq P \cdot F$), the MPC protocol can output ACCEPT if

Case 1 :

$$S(r) \cdot Q(r) = P(r) \cdot F(r)$$

which occurs with probability (Schwartz-Zippel Lemma)

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

Analysis

Even if x_A does not describe a SD solution (implying that $S \cdot Q \neq P \cdot F$), the MPC protocol can output Accept if

Case 1 :

$$S(r) \cdot Q(r) = P(r) \cdot F(r)$$

which occurs with probability (Schwartz-Zippel Lemma)

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

Case 2 : the [BN20] protocol fails, which occurs with probability

$$\frac{1}{|\mathbb{F}_{\text{points}}|}.$$

Summary

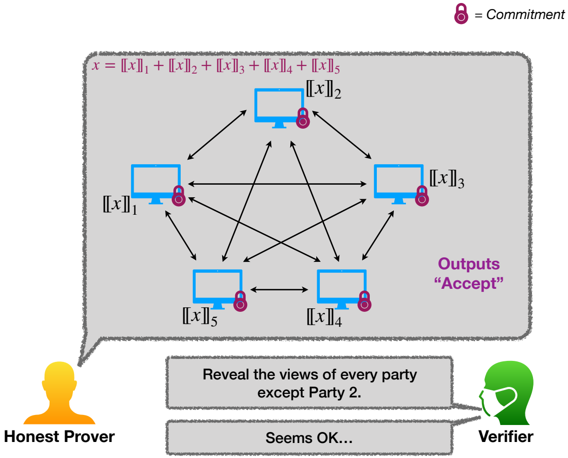
The MPC protocol π checks that (x_A, Q, P) describes a solution of the SD instance (H, y).

	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 $\mathrm{wt}_H(x) \le w$		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.	$\xleftarrow{r}$	
	$\xrightarrow{\text{broadcast messages}}$	$i^* \overset{\$}{\leftarrow} \{1, \ldots, N\}$
	$\xleftarrow{i^*}$	
	$\xrightarrow{\text{all } V_i \text{ for } i \neq i^*}$	
		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:

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

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

Security of the signature

Fiat-Shamir Transform:

5-round Identification Scheme ⇒ Signature

Attack of [KZ20]:

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

[KZ20a] 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 $\mathbb{F}_{poly} = \mathbb{F}_{2^8}$.

Obtained Performances

Scheme Name	$\lvert \mathsf{sgn} \rvert$	$\lvert \mathsf{pk} \rvert$	t_{sgn}	t_{verif}
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

Obtained Performances

Scheme Name	$\lvert\mathsf{sgn}\rvert$	$\lvert\mathsf{pk}\rvert$	t_{sgn}	t_{verif}
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

Number of parties: $N = 256$
Number of repetitions: $\tau = 17$

Obtained Performances

Scheme Name	|sgn|	|pk|	t_{sgn}	t_{verif}
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

Number of parties: $N = 32$
Number of repetitions: $\tau = 27$

Comparison Code-based Signatures (1/2)

Scheme Name	$\|\text{sgn}\|$	$\|\text{pk}\|$	t_{sgn}	t_{verif}
BGKS21	24.1 KB	0.1 KB	-	-
BGKS21	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	$\|\text{sgn}\|$	$\|\text{pk}\|$	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

Signature Security

- Keys = Generic Instances of the considered problem (no structure).
- Forgery in the *Random Oracle Model*:

$$\mathrm{Adv}^{\text{EUF-KO}} \leq \varepsilon_{\text{OWF}} + \frac{(\tau \cdot N + 1)Q^2}{2^{2\lambda}} + \underbrace{\mathrm{Prob}[X + Y = \tau]}_{\text{[KZ20a]'s attack}}$$

$$\mathrm{Adv}^{\text{EUF-CMA}} \leq \mathrm{Adv}^{\text{EUF-KO}} + Q_s \cdot \left(\tau \cdot \varepsilon_{\text{PRG}} + \varepsilon_{\text{Tree}} + \frac{Q}{2^{\kappa}}\right)$$

[BdK+21] Carsten Baum, Cyprien Delpech de Saint Guilhem, Daniel Kales, Emmanuela Orsini, Peter Scholl, and Greg Zaverucha. *Banquet: Short and Fast Signatures from AES*. PKC 2021.

[KZ22] Daniel Kales, and Greg Zaverucha. *Efficient Lifting for Shorter Zero-Knowledge Proofs and Post-Quantum Signatures*. Eprint 2022/282.

Signature Security

☞ Forgery in the ***Quantum*** *Random Oracle Model*:

[DFM20] Jelle Don, Serge Fehr, and Christian Majenz. *The measure-and-reprogram technique 2.0: Multi-round fiat-shamir and more.* Crypto 2020.

[DFMS21] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. *Online-extractability in the quantum random-oracle model.* Eprint 2021/280.

Table of Contents

Recent Optimizations

☞ Usage of additive sharings with a hypercube approach

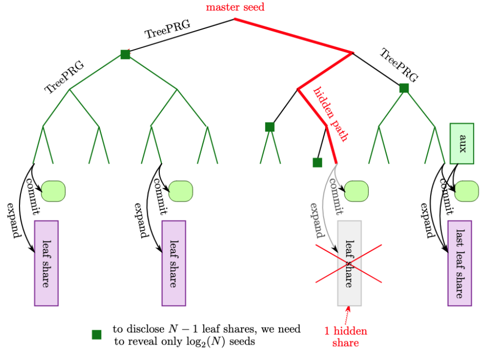
[AGH+22] Carlos Aguilar-Melchor, Nicolas Gama, James Howe, Andreas Hülsing, David Joseph, Dongze Yue. *The Return of the SDitH.* Eprint 2022/1645.

☞ Usage of low-threshold Shamir's secret sharings

[FR22] Thibauld Feneuil, Matthieu Rivain. *Threshold Linear Secret Sharing to the Rescue of MPC-in-the-Head.* Eprint 2022/1407.

Using additive sharings in a hypercube approach

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



(Eprint 2022/1645)

Using additive sharings in a hypercube approach

How to generate two N-sharings of a given value?

- ☞ Option 1: With two seed trees of N seeds.

  $$\textsc{Cost} = 2\log_2 N \text{ seeds} + 2 \text{ auxiliary states.}$$

Using additive sharings in a hypercube approach

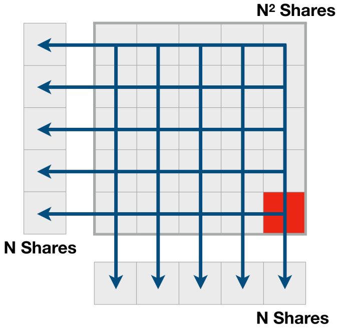
How to generate two N-sharings of a given value?

- ☞ Option 1: With two seed trees of N seeds.

  $$\textsc{Cost} = 2\log_2 N \text{ seeds} + 2 \text{ auxiliary states.}$$

- ☞ Option 2: With a large seed tree of N^2 seeds [AGH+22].

  $$\textsc{Cost} = \log_2(N^2) \text{ seeds} + 1 \text{ auxiliary state.}$$

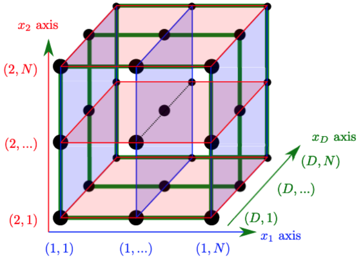


Using additive sharings in a hypercube approach

If we want to have a protocol with a soundness error of $\frac{1}{N}$, we can emulate the MPC protocol $D := \log_2(N)$ times on 2-sharings with the same auxiliary state:

$$\textsc{Soundness Error} := \left(\frac{1}{2}\right)^{\log_2 N} = \frac{1}{N}.$$

Thus, instead of emulating N parties to achieve a soundness error of $1/N$, we run only $2\log_2 N$ parties.



The $D \times N$ main party slices

Comparison over SDitH

Comparison over SDitH – variant $\mathbb{F}_{256}$:

Variant	$\lvert\mathsf{sgn}\rvert$	t_{sgn}	t_{verif}
Standard - Fast ($N = 32$)	11.5 KB	$\approx$ 6 ms	$\approx$ 6 ms
Standard - Short ($N = 256$)	8.26 KB	$\approx$ 25 ms	$\approx$ 25 ms
Hypercube - Fast ($N = 32$)	11.5 KB	$\approx$ 4 ms	$\approx$ 4 ms
Hypercube - Short ($N = 256$)	8.26 KB	$\approx$ 7 ms	$\approx$ 7 ms

Using Shamir's secret sharings

Idea: use a Shamir's (ℓ, N)-secret sharing and reveal only ℓ shares to the verifier (instead of $N-1$) [FR22].

To share $s \in \mathbb{F}$,

- sample $r_1, r_2, \ldots, r_\ell$ uniformly from $\mathbb{F}$,
- build the polynomial $P(X) = s + \sum_{k=1}^{\ell} r_k X^k$,
- set the share $[\![s]\!]_i$ as $P(e_i)$, where e_i is publicly known.

Resulting proof of knowledge:

☞ Correctness: ok.

Using Shamir's secret sharings

Idea: use a Shamir's (ℓ, N)-secret sharing and reveal only ℓ shares to the verifier (instead of $N-1$) [FR22].

To share $s \in \mathbb{F}$,

- sample $r_1, r_2, \ldots, r_\ell$ uniformly from $\mathbb{F}$,
- build the polynomial $P(X) = s + \sum_{k=1}^{\ell} r_k X^k$,
- set the share $[\![s]\!]_i$ as $P(e_i)$, where e_i is publicly known.

Resulting proof of knowledge:

☞ Correctness: ok.

☞ Zero-knowledge: ok, since we reveal only ℓ parties.

Using Shamir's secret sharings

Idea: use a Shamir's (ℓ, N)-secret sharing and reveal only ℓ shares to the verifier (instead of $N-1$) [FR22].

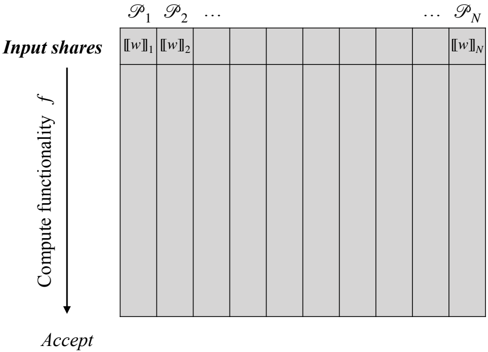
To share $s \in \mathbb{F}$,

- sample $r_1, r_2, \ldots, r_\ell$ uniformly from $\mathbb{F}$,
- build the polynomial $P(X) = s + \sum_{k=1}^{\ell} r_k X^k$,
- set the share $[\![s]\!]_i$ as $P(e_i)$, where e_i is publicly known.

Resulting proof of knowledge:

- ☞ Correctness: ok.
- ☞ Zero-knowledge: ok, since we reveal only ℓ parties.
- ☞ Soundness: ?

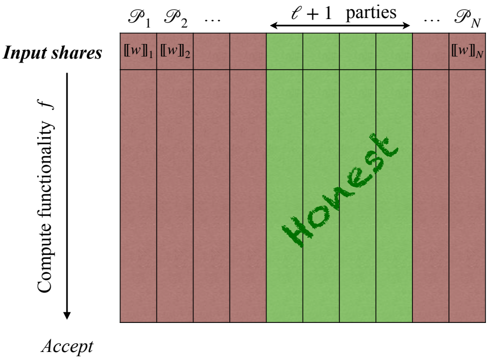
Using Shamir's secret sharings



Assumptions:
- Only broadcast
- Only linear operations on shares

Cheat on less than $N-\ell$ parties	?
Cheat on more than $N-\ell$ parties	?
Cheat on exactly $N-\ell$ parties	?

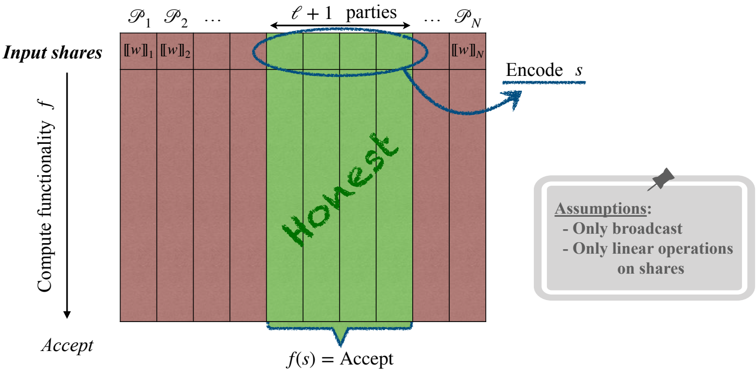
Using Shamir's secret sharings



Assumptions:
- Only broadcast
- Only linear operations on shares

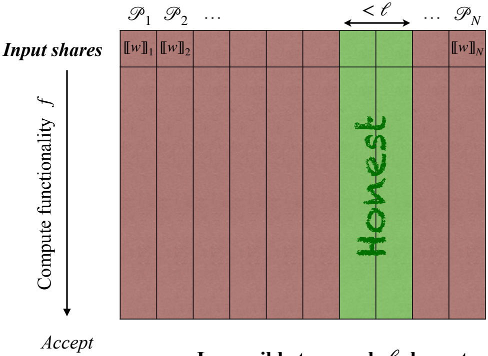
Cheat on less than $N-\ell$ parties	**?**
Cheat on more than $N-\ell$ parties	?
Cheat on exactly $N-\ell$ parties	?

Using Shamir's secret sharings



Cheat on less than $N - \ell$ parties	***Impossible***
Cheat on more than $N - \ell$ parties	**?**
Cheat on exactly $N - \ell$ parties	**?**

Using Shamir's secret sharings

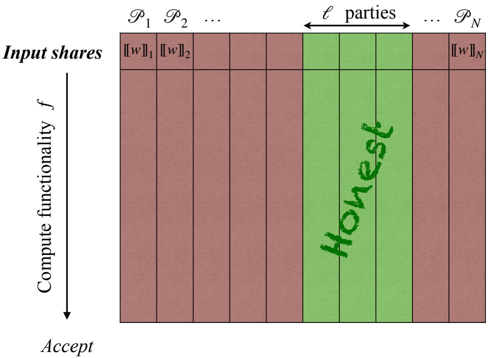


Assumptions:
- **Only broadcast**
- **Only linear operations on shares**

Impossible to reveal ℓ honest parties!

Cheat on less than $N - \ell$ parties	***Impossible***
Cheat on more than $N - \ell$ parties	***Useless***
Cheat on exactly $N - \ell$ parties	**?**

Using Shamir's secret sharings



Assumptions:
- Only broadcast
- Only linear operations on shares

Cheat on less than $N-\ell$ parties	***Impossible***
Cheat on more than $N-\ell$ parties	***Useless***
Cheat on exactly $N-\ell$ parties	***OK***

Using Shamir's secret sharings

Soundness error:

$$\frac{1}{\binom{N}{N-\ell}} = \frac{1}{\binom{N}{\ell}}$$

- ☞ No seed tree to generate the input shares
- ☞ A Merkle tree to commit the N input shares (by repetition)
- ☞ **A verifier re-emulates only ℓ parties by repetition (instead of $N-1$)**
- ☞ A prover needs to emulate only $\ell+1$ parties by repetition (instead of N)

Restriction: $N \leq |\mathbb{F}|$.

Comparison over SDitH

Comparison over SDitH – variant $\mathbb{F}_{256}$:

Variant	$\lvert \textsf{sgn} \rvert$	$t_{\textsf{sgn}}$	$t_{\textsf{verif}}$
Standard - Fast ($N = 32$)	11.5 KB	≈ 6 ms	≈ 6 ms
Standard - Short ($N = 256$)	8.26 KB	≈ 25 ms	≈ 25 ms
Hypercube - Fast ($N = 32$)	11.5 KB	≈ 4 ms	≈ 4 ms
Hypercube - Short ($N = 256$)	8.26 KB	≈ 7 ms	≈ 7 ms
Shamir's Secret Sharing ($N = 256$)	9.97 KB	≈ 3 ms	≈ 0.4 ms

Remark: **non-isochronous implementation**. Ongoing efforts are currently done to propose isochronous and optimized implementations of SDitH.

Remark: the two optimizations do not seem to be compatible with each other.

Table of Contents

Exploring other problems

- ☞ [Fen22] Thibauld Feneuil. *Building MPCitH-based Signatures from MQ, MinRank, Rank SD and PKP.* Eprint 2022/1512.

- ☞ [FMRV22] Thibauld Feneuil, Jules Maire, Matthieu Rivain and Damien Vergnaud. *Zero-Knowledge Protocols for the Subset Sum Problem from MPC-in-the-Head with Rejection.* Asiacrypt 2022.

Multivariate Quadratic Problem

Multivariate Quadratic Problem

From $(A_1, \ldots, A_m, b_1, \ldots b_m, y_1, \ldots, y_m)$, find $x \in \mathbb{F}_q^n$ such that

$$\forall i \leq m, \; y_i = x^T A_i x + b_i^T x.$$

The multi-party computation must check that the vector x satisfies

$$\begin{aligned} y_1 &= x^T A_1 x + b_1^T x \\ y_2 &= x^T A_2 x + b_2^T x \\ &\vdots \\ y_m &= x^T A_m x + b_m^T x \end{aligned}$$

Multivariate Quadratic Problem - Signature schemes

Instance	Protocol Name	Variant	Parameters			Sig. Size
			N	M	τ	
$q=4$, $m=88$, $n=88$	MudFish	-	4	191	68	14 640 B
	Mesquite	Fast	8	187	49	9 578 B
		Short	32	389	28	**8 609 B**
	Fen22	Fast	32	-	40	10 764 B
		Short	256	-	25	9 064 B
$q=256$, $m=40$, $n=40$	MudFish	Fast	8	176	51	15 958 B
		Short	16	250	36	13 910 B
	Mesquite	Fast	8	187	49	11 339 B
		Short	32	389	28	9 615 B
	Fen22	Fast	32	-	36	8 488 B
		Short	256	-	25	**7 114 B**

MinRank Problem

MinRank Problem

From $(M_0, M_1, \ldots, M_k)$, find $\alpha \in \mathbb{F}_q^k$ such that

$$\mathrm{rank}(M_0 + \sum_{i=1}^{k} \alpha_i M_i) \leq r.$$

MPC protocols

The multi-party computation must check that a matrix $M \in \mathbb{F}_q^{m \times n}$ has a rank of at most r.

MPC protocols

The multi-party computation must check that a matrix $M \in \mathbb{F}_q^{m\times n}$ has a rank of at most r.

Rank Decomposition:

A matrix $M \in \mathbb{F}_q^{n\times m}$ has a rank of at most r
iff there exists $T \in \mathbb{F}_q^{n\times r}$ and $R \in \mathbb{F}_q^{r\times m}$ such that $M = TR$.

MPC protocols

The multi-party computation must check that a matrix $M \in \mathbb{F}_q^{m \times n}$ has a rank of at most r. Rewrite M as $(x_1, \ldots, x_n) \in \mathbb{F}_{q^m}^n$.

Rank Decomposition:

$$\text{A matrix } M \in \mathbb{F}_q^{n \times m} \text{ has a rank of at most } r$$
$$\textit{iff} \text{ there exists } T \in \mathbb{F}_q^{n \times r} \text{ and } R \in \mathbb{F}_q^{r \times m} \text{ such that } M = TR.$$

Linearized Polynomials:

$$\text{A matrix } M \in \mathbb{F}_q^{n \times m} \text{ has a rank of at most } r$$
$$\Leftrightarrow \text{ there exists a linear subspace } U \text{ of } \mathbb{F}_{q^m} \text{ of dimension } r$$
$$\text{such that } \{x_1, \ldots, x_n\} \subset U.$$
$$\Leftrightarrow \text{ there exists a monic } q\text{-polynomial } L_U \text{ of degree } q^r$$
$$\text{such that } x_1, \ldots, x_n \text{ are roots of } L_U.$$

Remark: Computing $[\![v^q]\!]$ from $[\![v]\!]$ is free.

MinRank Problem

Instance	Protocol Name	Variant	Parameters			Sig. Size
			N	M	τ	
$q = 16$, $m = 16$, $n = 16$, $k = 142$, $r = 4$	Cou01	-	-	-	219	52 430 B
		Optimized	-	-	219	28 575 B
	SINY22	-	-	-	128	50 640 B
		Optimized	-	-	128	28 128 B
	BESV22	-	-	256	128	26 405 B
	BG22	Fast	8	187	49	13 644 B
		Short	32	389	28	10 937 B
	ARZV22	Fast	32	-	28	10 116 B
		Short	256	-	18	7 422 B
	Fen22 (RD)	Fast	32	-	33	9 288 B
		Short	256	-	19	7 122 B
	Fen22 (LP)	Fast	32	-	28	7 204 B
		Short	256	-	18	**5 518 B**

Rank Syndrome Decoding Problem

Rank Syndrome Decoding Problem

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

$$y = Hx \quad \text{and} \quad \text{rank}(x) \leq r.$$

- ☞ Using the rank decomposition
- ☞ Using q-polynomials

Rank Syndrome Decoding Problem

Instance	Protocol Name	Variant	Parameters N	M	τ	Sig. Size
$q = 2$, $m = 31$, $n = 30$, $k = 15$, $r = 9$	Stern	-	-	-	219	31 358 B
	Véron	-	-	-	219	27 115 B
	FJR21	Fast	8	187	49	19 328 B
		Short	32	389	28	14 181 B
	BG22	Fast	8	187	49	15 982 B
		Short	32	389	28	12 274 B
	Fen22 (RD)	Fast	32	-	33	11 000 B
		Short	256	-	21	8 543 B
	Fen22 (LP)	Fast	32	-	30	7 376 B
		Short	256	-	20	**5 899 B**
Ideal RSL	BG22	Fast	32	-	27	9 392 B
		Short	256	-	17	6 754 B

Subset Sum Problem

Subset Sum Problem

From (w, t), find a vector x such that

$$\langle w, x \rangle = t \mod q \quad \text{and} \quad x \in \{0,1\}^n.$$

The multi-party computation must check that the vector x satisfies

$$\langle w, x \rangle = t \mod q \quad \text{and} \quad x \in \{0,1\}^n.$$

Problem: q is very large ($q \approx 2^{256}$).

Subset Sum Problem

Subset Sum Problem

From (w, t), find a vector x such that

$$\langle w, x \rangle = t \mod q \quad \text{and} \quad x \in \{0,1\}^n.$$

The multi-party computation must check that the vector x satisfies

$$\langle w, x \rangle = t \mod q \quad \text{and} \quad x \in \{0,1\}^n.$$

Problem: q is very large ($q \approx 2^{256}$).
Solution: Use an additive sharing over integers **with rejection**.
[FMRV22] Thibauld Feneuil, Jules Maire, Matthieu Rivain and Damien Vergnaud. *Zero-Knowledge Protocols for the Subset Sum Problem from MPC-in-the-Head with Rejection.* Asiacrypt 2022.

Subset Sum Problem

Instance	Protocol Name	Variant	Parameters			Sig. Size
			N	M	τ	
$q = 2^{256}$ $n = 256$	Sha86	-	-	-	219	$\approx$ 1.2 MB
	LNSW13	-	-	-	219	$\approx$ 2.3 MB
	Beu20	-	1024	4040	14	$\approx$ 120 KB
	FMRV22	C&C	64	514	28	$\approx$ 21 KB*
		Short	256	-	29	$\approx$ 28 KB*
	FMRV22 + Optim	Fast	32	-	28	$\approx$ 29 KB*
		Short	256	-	19	$\approx$ 18 KB*

*sizes given for a rejection rate which is less than 2%.

Conclusion

Security Assumption	Scheme	Achieved sizes (in KB)
Subset Sum	[FMRV22]	18 – 29
Legendre PRF	[Bd20]	12.2 – 14.8
AES	[KZ22]	9.7 – 14.4
Permuted Kernel	[BG22]	8.6 – 9.7
Syndrome Decoding *(Hamm.)*	[FJR22]	8.3 – 11.5
LowMC	[KZ22]	6.4 – 9.2
Multivariate Quadratic	[Fen22]	6.9 – 8.3
Higher-Power Residue Characters	[Bd20]	6.3 – 7.8
Syndrome Decoding *(Rank)*	[Fen22]	5.8 – 7.2
Min Rank	[Fen22]	5.4 – 7.0
[BHH01] PRF	[FMRV22]	4.8 – 6.5
Rain [DKR+21]	[KZ22]	4.9 – 6.4

Sizes given for a range of 32-256 parties.

Estimation of the running time:

for 256 parties, 2-10 ms for signing (with [AGH+22]).

Conclusion

Security Assumption	Scheme	Achieved sizes (in KB)
Subset Sum	[FMRV22]	18 – 29
Legendre PRF	[Bd20]	12.2 – 14.8
AES	[KZ22]	9.7 – 14.4
Permuted Kernel	[BG22]	8.6 – 9.7
Syndrome Decoding *(Hamm.)*	[FJR22]	8.3 – 11.5
LowMC	[KZ22]	6.4 – 9.2
Multivariate Quadratic	[Fen22]	6.9 – 8.3
Higher-Power Residue Characters	[Bd20]	6.3 – 7.8
Syndrome Decoding *(Rank)*	[Fen22]	5.8 – 7.2
Min Rank	[Fen22]	5.4 – 7.0
[BHH01] PRF	[FMRV22]	4.8 – 6.5
Rain [DKR+21]	[KZ22]	4.9 – 6.4

Sizes given for a range of 32-256 parties.

Estimation of the running time:

for 256 parties, 2-10 ms for signing (with [AGH+22]).

Thank you for your attention!

References

[ABG+19] Nicolas Aragon, Olivier Blazy, Philippe Gaborit, Adrien Hauteville, and Gilles Zémor. *Durandal: A Rank Metric Based Signature Scheme.* Eurocrypt 2019.

[AGH+22] Carlos Aguilar-Melchor, Nicolas Gama, James Howe, Andreas Hülsing, David Joseph, Dongze Yue. *The Return of the SDitH.* Eprint 2022/1645.

[ARZV22] Gora Adj, Luis Rivera-Zamarripa, and Javier Verbel. *Minrank in the head: Short signatures from zero-knowledge proofs.* Eprint 2022/1501.

[Cou01] Nicolas Courtois. *Efficient zero-knowledge authentication based on a linear algebra problem MinRank.* Asiacrypt 2001.

[Beu20] Ward Beullens. *LESS-FM: Sigma protocols for MQ, PKP and SIS, and Fishy signature schemes.* Eurocrypt 2020.

References

[Bd20] Ward Beullens and Cyprien de Saint Guilhem. *LegRoast: Efficient post-quantum signatures from the Legendre PRF.* PQC 2020.

[BBPS21] Alessandro Barenghi, Jean-François Biasse, Edoardo Persichetti, and Paolo Santini. *LESS-FM: Fine-Tuning Signatures from the Code Equivalence Problem.* PQC 2021.

[BdK+21] Carsten Baum, Cyprien Delpech de Saint Guilhem, Daniel Kales, Emmanuela Orsini, Peter Scholl, and Greg Zaverucha. *Banquet: Short and Fast Signatures from AES.* PKC 2021.

[BDNS21] Gustavo Banegas, Thomas Debris-Alazard, Milena Nedeljković, and Benjamin Smith. *Wavelet: Code-based postquantum signatures with fast verification on microcontrollers.* Eprint 2021/1432.

References

[BG22] Loïc Bidoux and Philippe Gaborit. *Compact post-quantum signatures from proofs of knowledge leveraging structure for the pkp, sd and rsd problems.* arXiv 2204.02915.

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

[BGKS21] Loïc Bidoux, Philippe Gaborit, Mukul Kulkarni, and Nicolas Sendrier. *Quasi-Cyclic Stern Proof of Knowledge.* 2022 IEEE ISIT.

[BHH01] D. Boneh, S. Halevi, and N. Howgrave-Graham. *The modular inversion hidden number problem.* Asiacrypt 2001.

References

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

[CDG+17] Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. *Post-quantum zero-knowledge and signatures from symmetric-key primitives.* CCS 2017.

[dDOS19] Cyprien de Saint Guilhem, Lauren De Meyer, Emmanuela Orsini, and Nigel P. Smart. *BBQ: Using AES in picnic signatures.* SAC 2019.

[DFM20] Jelle Don, Serge Fehr, and Christian Majenz. *The measure-and-reprogram technique 2.0: Multi-round fiat-shamir and more.* Crypto 2020.

References

[DFMS21] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. *Online-extractability in the quantum random-oracle model.* Eprint 2021/280.

[DKR+21] C. Dobraunig, D. Kales, C. Rechberger, M. Schofnegger, and G. Zaverucha. *Shorter signatures based on tailor-made minimalist symmetric-key crypto.* CCS 2022.

[dOT21] Cyprien de Saint Guilhem, Emmanuela Orsini, and Titouan Tanguy. *Limbo: Efficient zero-knowledge MPCitH-based arguments.* CCS 2021.

[DST19] Thomas Debris, Nicolas Sendrier, and Jean-Pierre Tillich. *Wave: A New Family of Trapdoor One-Way Preimage Sampleable Functions Based on Codes.* Asiacrypt 2019.

[Fen22] Thibauld Feneuil. *Building MPCitH-based Signatures from MQ, MinRank, Rank SD and PKP.* Eprint 2022/1512.

References

[FJR21] Thibauld Feneuil, Antoine Joux, and Matthieu Rivain. *Shared Permutation for Syndrome Decoding: New Zero-Knowledge Protocol and Code-Based Signature.* Designs, Codes and Cryptography.

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

[FMRV22] Thibauld Feneuil, Jules Maire, Matthieu Rivain and Damien Vergnaud. *Zero-Knowledge Protocols for the Subset Sum Problem from MPC-in-the-Head with Rejection.* Asiacrypt 2022.

[FR22] Thibauld Feneuil, Matthieu Rivain. *Threshold Linear Secret Sharing to the Rescue of MPC-in-the-Head.* Eprint 2022/1407.

References

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

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

[KKW18] J. Katz, V. Kolesnikov, and X. Wang. *Improved non-interactive zero knowledge with applications to post-quantum signatures.* CCS 2018.

[KZ20a] Daniel Kales and Greg Zaverucha. *An attack on some signature schemes constructed from five-pass identification schemes.* CANS 2020.

[KZ20b] Daniel Kales and Greg Zaverucha. *Improving the performance of the Picnic signature scheme.* TCHES 2020.

References

[KZ22] Daniel Kales, and Greg Zaverucha. *Efficient Lifting for Shorter Zero-Knowledge Proofs and Post-Quantum Signatures.* Eprint 2022/282.

[LNSW13] S. Ling, K. Nguyen, D. Stehlé, and H. Wang. *Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications.* PKC 2013.

[Sha86] A. Shamir. *A zero-knowledge proof for knapsacks.* 1986.

[SINY22] Bagus Santoso, Yasuhiko Ikematsu, Shuhei Nakamura, and Takanori Yasuda. *Three-Pass Identification Scheme Based on MinRank Problem with Half Cheating Probability.* arXiv 2205.03255.

[Wan22] William Wang. *Shorter Signatures from MQ.* Eprint 2022/344.