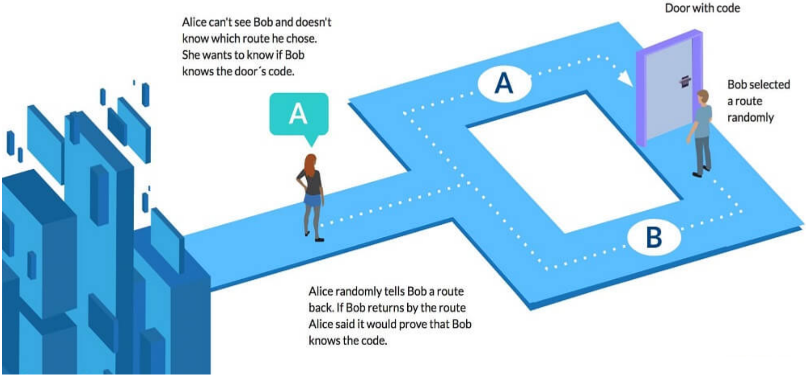
Short Signatures from Regular Syndrome Decoding in the Head

ELIANA CAROZZA, GEOFFROY COUTEAU, ANTOINE JOUX

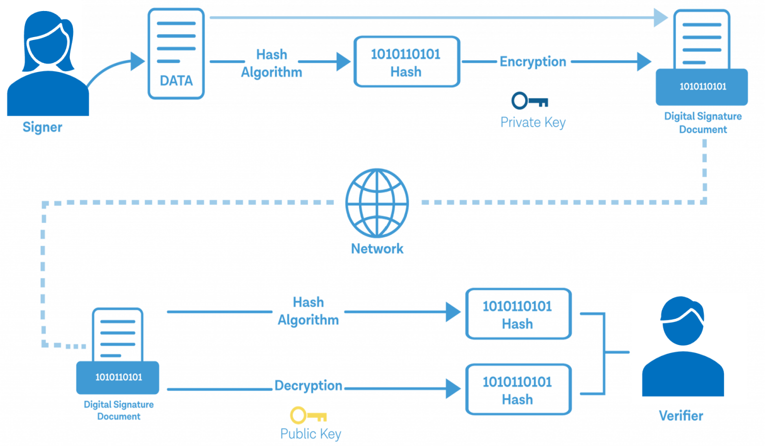
FSMP FONDATION SCIENCES MATHÉMATIQUES DE PARIS
IRIF
Université de Paris
CNRS
CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY
UNIVERSITÄT DES SAARLANDES

Zero Knowledge Proof of Knowledge

Bob wants to prove to Alice that he knows the door code without reveal it to her.



Signature scheme



Syndrome decoding problem

SD definition

The **Syndrome decoding problem** with parameters (K, k, w) is defined as follows:

- (Problem generation)
  Sample $H \leftarrow_r \{0,1\}^{k \times K}$ and $x \leftarrow_r \{x \in \{0,1\}^K : \mathsf{HW}(x) = w\}$.
  Set $y \leftarrow H \cdot x \bmod 2$. Output (H, y).
- (Goal) Given (H, y), find $x \in \{0,1\}^K$ such that
  - $H \cdot x = y \bmod 2$
  - $\mathsf{HW}(x) = w$ over $\mathbb{N}$.

SD DEFINITION

The **Syndrome decoding problem** with parameters (K, k, w) is defined as follows:

- (Problem generation)
  Sample $H \leftarrow_r \{0,1\}^{k \times K}$ and $x \leftarrow_r \{x \in \{0,1\}^K : \mathsf{HW}(x) = w\}$.
  Set $y \leftarrow H \cdot x \bmod 2$. Output (H, y).
- (Goal) Given (H, y), find $x \in \{0,1\}^K$ such that
  - $H \cdot x = y \bmod 2$
  - $\mathsf{HW}(x) = w$ over $\mathbb{N}$.

RSD DEFINITION

The **Regular syndrome decoding problem** is a variant of the SD problem in which the witness is *regular*, i.e. divided into w blocks of size $T = K/w$, each of theme has exactly one non-zero entry.

Syndrome decoding problem

SD definition

The **Syndrome decoding problem** with parameters (K, k, w) is defined as follows:

- (Problem generation)
  Sample $H \leftarrow_r \{0,1\}^{k \times K}$ and $x \leftarrow_r \{x \in \{0,1\}^K : \mathrm{HW}(x) = w\}$.
  Set $y \leftarrow H \cdot x \bmod 2$. Output (H, y).
- (Goal) Given (H, y), find $x \in \{0,1\}^K$ such that
  - $H \cdot x = y \bmod 2$
  - $\mathrm{HW}(x) = w$ over $\mathbb{N}$.

RSD definition

The **Regular syndrome decoding problem** is a variant of the SD problem in which the witness is *regular*, i.e. divided into w blocks of size $T = K/w$, each of theme has exactly one non-zero entry.

Our point of view

Let K, k, w be three integers, with $K > k > w$. Given $H \in \{0,1\}^{k \times K}$ and $y \in \{0,1\}^k$, find regular $x \in \{0,1\}^K$ s.t.:

- $H \cdot x = y$ over $\mathbb{F}_2$
- $\langle 1, x \rangle = w$ over $\mathbb{Z}_T$.

What is MPC?

Let $P_1, \cdots, P_n$ be n parties, each one with a private information $p_1, \cdots, p_n$. For a public function g, an n-party protocol allows them to compute

$$g(p_1, \cdots, p_n)$$

without revealing their own secret inputs.

WHAT IS MPC?

Let $P_1, \cdots, P_n$ be n parties, each one with a private information $p_1, \cdots, p_n$. For a public function g, an n-party protocol allows them to compute

$$g(p_1, \cdots, p_n)$$

without revealing their own secret inputs.

MPC IN-THE-HEAD

For a public value y, it is possible to produce an honest-verifier zero-knowledge argument of knowledge of a witness x s.t. $f(x) = y$ using an n-party protocol for a function g related to f.

What is MPC?

Let $P_1,\cdots,P_n$ be n parties, each one with a private information $p_1,\cdots,p_n$. For a public function g, an n-party protocol allows them to compute

$$g(p_1,\cdots,p_n)$$

without revealing their own secret inputs.

MPC in-the-Head

For a public value y, it is possible to produce an honest-verifier zero-knowledge argument of knowledge of a witness x s.t. $f(x)=y$ using an n-party protocol for a function g related to f.

In our context $f(x)=(H\cdot x \bmod 2, \langle 1,x\rangle \bmod T)=(y,w)$.

WHAT IS MPC?

Let $P_1, \cdots, P_n$ be n parties, each one with a private information $p_1, \cdots, p_n$. For a public function g, an n-party protocol allows them to compute

$$g(p_1, \cdots, p_n)$$

without revealing their own secret inputs.

MPC IN-THE-HEAD

For a public value y, it is possible to produce an honest-verifier zero-knowledge argument of knowledge of a witness x s.t. $f(x) = y$ using an n-party protocol for a function g related to f.

In our context $f(x) = (H \cdot x \bmod 2, \langle 1, x \rangle \bmod T) = (y, w)$.

The prover:

- Shares $[\![x]\!]_2 = (x_1, \cdots, x_n)$ s.t. $\sum_{i=1}^{n} x_i = x$ among n virtual parties,
- Computes $g(x_1, \cdots, x_n) = f\left(\sum_i x_i\right) = \left(\sum_i H \cdot x_i, \sum_i \langle 1, x_i \rangle\right)$ over appropriate ring.

ZKPoK from MPC

P

V

x_1
$y_1 = Hx_1 \mod 2$
$w_1 = \langle 1, x_1 \rangle \mod T$

P_1

x_n
$y_n = Hx_n \mod 2$
$w_n = \langle 1, x_n \rangle \mod T$

P_n

P_2

x_2
$y_2 = Hx_2 \mod 2$
$w_2 = \langle 1, x_2 \rangle \mod T$

P_i

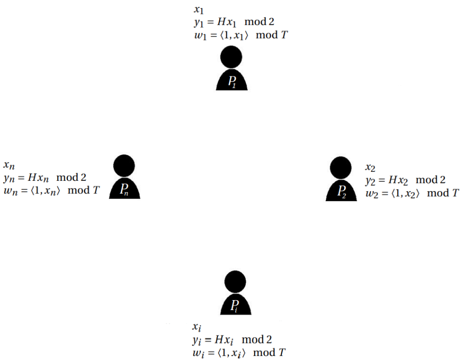
x_i
$y_i = Hx_i \mod 2$
$w_i = \langle 1, x_i \rangle \mod T$

ZKPoK from MPC

P

V

Asks to see all views but one.

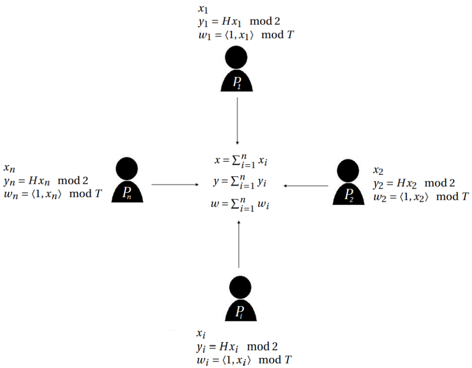


ZKPoK from MPC

P

V

Asks to see all views but one.



How to convert $[\![x]\!]_2$ into $[\![x]\!]_T$

$$[\![x]\!]_T = z \cdot [\![1-r]\!]_T + (1-z) \cdot [\![r]\!]_T$$

where r is random and $[\![z]\!]_2 = [\![r]\!]_2 + [\![x]\!]_2$.

How to convert $[\![x]\!]_2$ into $[\![x]\!]_T$

$$[\![x]\!]_T = z \cdot [\![1-r]\!]_T + (1-z) \cdot [\![r]\!]_T$$

where r is random and $[\![z]\!]_2 = [\![r]\!]_2 + [\![x]\!]_2$.

Prepocessing material: $\mathbf{s} = [\![r]\!]_2, \mathbf{t} = [\![r]\!]_T$.

Round 1 (P)

- $([\![x]\!]_2, [\![\mathbf{r}]\!]_2, [\![\mathbf{r}]\!]_T) = ((\mathbf{x}_1, \cdots, \mathbf{x}_n), (\mathbf{s}_1, \cdots, \mathbf{s}_n), (\mathbf{t}_1, \cdots, \mathbf{t}_n))$.

Idea of the 5 round protocol

Round 1 (P)

- $([\![x]\!]_2, [\![\mathbf{r}]\!]_2, [\![\mathbf{r}]\!]_T) = ((\mathbf{x}_1, \cdots, \mathbf{x}_n), (\mathbf{s}_1, \cdots, \mathbf{s}_n), (\mathbf{t}_1, \cdots, \mathbf{t}_n))$.
- $c_i \leftarrow \mathsf{Commit}(\mathbf{x}_i, \mathbf{s}_i, \mathbf{t}_i)$ for $i = 1$ to n.

Round 1 (P)

- $([\![x]\!]_2, [\![\mathbf{r}]\!]_2, [\![\mathbf{r}]\!]_T) = ((\mathbf{x}_1, \cdots, \mathbf{x}_n), (\mathbf{s}_1, \cdots, \mathbf{s}_n), (\mathbf{t}_1, \cdots, \mathbf{t}_n))$.
- $c_i \leftarrow \text{Commit}(\mathbf{x}_i, \mathbf{s}_i, \mathbf{t}_i)$ for $i = 1$ to n.

Round 2 (V) Does something in order to verify the preprocessing phase.

Round 1 (P)

- $([\![x]\!]_2, [\![\mathbf{r}]\!]_2, [\![\mathbf{r}]\!]_T) = ((\mathbf{x}_1, \cdots, \mathbf{x}_n), (\mathbf{s}_1, \cdots, \mathbf{s}_n), (\mathbf{t}_1, \cdots, \mathbf{t}_n))$.
- $c_i \leftarrow \mathsf{Commit}(\mathbf{x}_i, \mathbf{s}_i, \mathbf{t}_i)$ for $i = 1$ to n.

Round 2 (V) Does something in order to verify the preprocessing phase.
Round 3 (P) Runs the online phase of the MPC in the head.

Round 1 (P)

- $([\![x]\!]_2, [\![\mathbf{r}]\!]_2, [\![\mathbf{r}]\!]_T) = ((\mathbf{x}_1, \cdots, \mathbf{x}_n), (\mathbf{s}_1, \cdots, \mathbf{s}_n), (\mathbf{t}_1, \cdots, \mathbf{t}_n))$.
- $c_i \leftarrow \mathsf{Commit}(\mathbf{x}_i, \mathbf{s}_i, \mathbf{t}_i)$ for $i = 1$ to n.

Round 2 (V) Does something in order to verify the preprocessing phase.
Round 3 (P) Runs the online phase of the MPC in the head.
Round 4 (V) Chooses $d \in [n]$.

Idea of the 5 round protocol

Round 1 (P)

- $([\![x]\!]_2, [\![\mathbf{r}]\!]_2, [\![\mathbf{r}]\!]_T) = ((\mathbf{x}_1, \cdots, \mathbf{x}_n), (\mathbf{s}_1, \cdots, \mathbf{s}_n), (\mathbf{t}_1, \cdots, \mathbf{t}_n))$.
- $c_i \leftarrow \text{Commit}(\mathbf{x}_i, \mathbf{s}_i, \mathbf{t}_i)$ for $i = 1$ to n.

Round 2 (V) Does something in order to verify the preprocessing phase.
Round 3 (P) Runs the online phase of the MPC in the head.
Round 4 (V) Chooses $d \in [n]$.
Round 5 (P) Opens c_j for $j \neq d$.

Idea of the 5 round protocol

Round 1 (P)

- $([\![x]\!]_2, [\![\mathbf{r}]\!]_2, [\![\mathbf{r}]\!]_T) = ((\mathbf{x}_1, \cdots, \mathbf{x}_n), (\mathbf{s}_1, \cdots, \mathbf{s}_n), (\mathbf{t}_1, \cdots, \mathbf{t}_n))$.
- $c_i \leftarrow \text{Commit}(\mathbf{x}_i, \mathbf{s}_i, \mathbf{t}_i)$ for $i = 1$ to n.

Round 2 (V) Does something in order to verify the preprocessing phase.
Round 3 (P) Runs the online phase of the MPC in the head.
Round 4 (V) Chooses $d \in [n]$.
Round 5 (P) Opens c_j for $j \neq d$.

Each $\mathbf{t}_i$ is a $K \log T$ term.

Idea of the 5 round protocol

Round 1 (P)

- $([\![x]\!]_2, [\![\mathbf{r}]\!]_2, [\![\mathbf{r}]\!]_T) = ((\mathbf{x}_1, \cdots, \mathbf{x}_n), (\mathbf{s}_1, \cdots, \mathbf{s}_n), (\mathbf{t}_1, \cdots, \mathbf{t}_n))$.
- $c_i \leftarrow \mathsf{Commit}(\mathbf{x}_i, \mathbf{s}_i, \mathbf{t}_i)$ for $i = 1$ to n.

Round 2 (V) Does something in order to verify the preprocessing phase.
Round 3 (P) Runs the online phase of the MPC in the head.
Round 4 (V) Chooses $d \in [n]$.
Round 5 (P) Opens c_j for $j \neq d$.

Each $\mathbf{t}_i$ is a $K \log T$ term.

SD

- $T = K$;
- Sharing $\mathbf{t}_i$ requires $K \log K$ bits.

Round 1 (P)

- $([\![x]\!]_2, [\![\mathbf{r}]\!]_2, [\![\mathbf{r}]\!]_T) = ((\mathbf{x}_1, \cdots, \mathbf{x}_n), (\mathbf{s}_1, \cdots, \mathbf{s}_n), (\mathbf{t}_1, \cdots, \mathbf{t}_n))$.
- $c_i \leftarrow \text{Commit}(\mathbf{x}_i, \mathbf{s}_i, \mathbf{t}_i)$ for $i = 1$ to n.

Round 2 (V) Does something in order to verify the preprocessing phase.
Round 3 (P) Runs the online phase of the MPC in the head.
Round 4 (V) Chooses $d \in [n]$.
Round 5 (P) Opens c_j for $j \neq d$.

Each $\mathbf{t}_i$ is a $K \log T$ term.

SD

- $T = K$;
- Sharing $\mathbf{t}_i$ requires $K \log K$ bits.

RSD

- $T = K/w$;
- Sharing $\mathbf{t}_i$ requires $K \log K/w$ bits.

Round 1 (P)

- $([\![x]\!]_2, [\![\mathbf{r}]\!]_2, [\![\mathbf{r}]\!]_T) = ((\mathbf{x}_1, \cdots, \mathbf{x}_n), (\mathbf{s}_1, \cdots, \mathbf{s}_n), (\mathbf{t}_1, \cdots, \mathbf{t}_n))$.
- $c_i \leftarrow \text{Commit}(\mathbf{x}_i, \mathbf{s}_i, \mathbf{t}_i)$ for $i = 1$ to n.

Round 2 (V) Does something in order to verify the preprocessing phase.
Round 3 (P) Runs the online phase of the MPC in the head.
Round 4 (V) Chooses $d \in [n]$.
Round 5 (P) Opens c_j for $j \neq d$.

Each $\mathbf{t}_i$ is a $K \log T$ term.

SD

- $T = K$;
- Sharing $\mathbf{t}_i$ requires $K \log K$ bits.

RSD

- $T = K/w$;
- Sharing $\mathbf{t}_i$ requires $K \log K/w$ bits.

⚠The higher is the weight, the lower is the cost!

How to share $\mathbf{s} = [\![r]\!]_2$ and $\mathbf{t} = [\![r]\!]_T$

The prover computes the material himself in the preprocessing phase but he has to shuffle it using a uniformly random permutation chosen by the verifier before use it in the online phase of the MPC-in-the-head protocol.

Cheat on preprocessing

How to share $\mathbf{s} = [\![r]\!]_2$ and $\mathbf{t} = [\![r]\!]_T$

The prover computes the material himself in the preprocessing phase but he has to shuffle it using a uniformly random permutation chosen by the verifier before use it in the online phase of the MPC-in-the-head protocol.

$$\begin{array}{c} Hx = y \\ \mathbf{z} = \mathbf{s} \oplus x \\ x' = \mathbf{z} \odot (\mathbf{1} - \mathbf{t}) + (\mathbf{1} - \mathbf{z}) \odot \mathbf{t} \\ \mathrm{HW}(x') = w \end{array} \quad \longrightarrow \quad \begin{array}{c} Hx = y \\ \mathbf{z} = \pi(\mathbf{s}) \oplus x \\ x' = \mathbf{z} \odot (\mathbf{1} - \pi(\mathbf{t})) + (\mathbf{1} - \mathbf{z}) \odot \pi(\mathbf{t}) \\ \mathrm{HW}(x') = w \end{array}$$

Cheat on preprocessing

How to share $\mathbf{s} = [\![r]\!]_2$ and $\mathbf{t} = [\![r]\!]_T$

The prover computes the material himself in the preprocessing phase but he has to shuffle it using a uniformly random permutation chosen by the verifier before use it in the online phase of the MPC-in-the-head protocol.

$$\begin{array}{c} Hx = y \\ \mathbf{z} = \mathbf{s} \oplus x \\ x' = \mathbf{z} \odot (\mathbf{1} - \mathbf{t}) + (\mathbf{1} - \mathbf{z}) \odot \mathbf{t} \\ \mathrm{HW}(x') = w \end{array} \quad \to \quad \begin{array}{c} Hx = y \\ \mathbf{z} = \pi(\mathbf{s}) \oplus x \\ x' = \mathbf{z} \odot (\mathbf{1} - \pi(\mathbf{t})) + (\mathbf{1} - \mathbf{z}) \odot \pi(\mathbf{t}) \\ \mathrm{HW}(x') = w \end{array}$$

Definition

A real $\mathrm{p} \in (0,1)$ is a *combinatorial bound* if for every incorrect witness x, and every pair $(\mathbf{s}, \mathbf{t})$, the probability, over the random choice of π, that x satisfies:

- $x' = \mathbf{z} \odot (\mathbf{1} - \pi(\mathbf{t})) + (\mathbf{1} - \mathbf{z}) \odot \pi(\mathbf{t})$ with $\mathbf{z} = \pi(\mathbf{s}) \oplus x$
- $H \cdot x = y \bmod 2$, $\mathrm{HW}(x) = w \bmod 2$, and $\mathrm{HW}(x') = w \bmod T$

is upper-bounded by p.

Cheat on preprocessing

How to share $\mathbf{s} = [\![r]\!]_2$ and $\mathbf{t} = [\![r]\!]_T$

The prover computes the material himself in the preprocessing phase but he has to shuffle it using a uniformly random permutation chosen by the verifier before use it in the online phase of the MPC-in-the-head protocol.

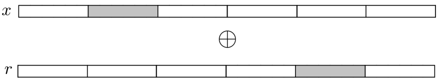
$$\begin{array}{c} Hx = y \\ \mathbf{z} = \mathbf{s} \oplus x \\ x' = \mathbf{z} \odot (\mathbf{1} - \mathbf{t}) + (\mathbf{1} - \mathbf{z}) \odot \mathbf{t} \\ \mathsf{HW}(x') = w \end{array} \quad \longrightarrow \quad \begin{array}{c} Hx = y \\ \mathbf{z} = \pi(\mathbf{s}) \oplus x \\ x' = \mathbf{z} \odot (\mathbf{1} - \pi(\mathbf{t})) + (\mathbf{1} - \mathbf{z}) \odot \pi(\mathbf{t}) \\ \mathsf{HW}(x') = w \end{array}$$

Definition

A real $\mathsf{p} \in (0,1)$ is a *combinatorial bound* if for every incorrect witness x, and every pair $(\mathbf{s}, \mathbf{t})$, the probability, over the random choice of π, that x satisfies:

- $x' = \mathbf{z} \odot (\mathbf{1} - \pi(\mathbf{t})) + (\mathbf{1} - \mathbf{z}) \odot \pi(\mathbf{t})$ with $\mathbf{z} = \pi(\mathbf{s}) \oplus x$
- $H \cdot x = y \bmod 2$, $\mathsf{HW}(x) = w \bmod 2$, and $\mathsf{HW}(x') = w \bmod T$

is upper-bounded by p.



Cheat on preprocessing

How to share $\mathbf{s} = [\![r]\!]_2$ and $\mathbf{t} = [\![r]\!]_T$

The prover computes the material himself in the preprocessing phase but he has to shuffle it using a uniformly random permutation chosen by the verifier before use it in the online phase of the MPC-in-the-head protocol.

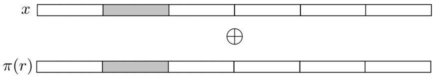
$$\begin{array}{c} Hx = y \\ \mathbf{z} = \mathbf{s} \oplus x \\ x' = \mathbf{z} \odot (\mathbf{1} - \mathbf{t}) + (\mathbf{1} - \mathbf{z}) \odot \mathbf{t} \\ \mathsf{HW}(x') = w \end{array} \quad \rightarrow \quad \begin{array}{c} Hx = y \\ \mathbf{z} = \pi(\mathbf{s}) \oplus x \\ x' = \mathbf{z} \odot (\mathbf{1} - \pi(\mathbf{t})) + (\mathbf{1} - \mathbf{z}) \odot \pi(\mathbf{t}) \\ \mathsf{HW}(x') = w \end{array}$$

Definition

A real $\mathsf{p} \in (0,1)$ is a *combinatorial bound* if for every incorrect witness x, and every pair $(\mathbf{s}, \mathbf{t})$, the probability, over the random choice of π, that x satisfies:

- $x' = \mathbf{z} \odot (\mathbf{1} - \pi(\mathbf{t})) + (\mathbf{1} - \mathbf{z}) \odot \pi(\mathbf{t})$ with $\mathbf{z} = \pi(\mathbf{s}) \oplus x$
- $H \cdot x = y \bmod 2$, $\mathsf{HW}(x) = w \bmod 2$, and $\mathsf{HW}(x') = w \bmod T$

is upper-bounded by p.



f-ALMOST REGULAR SYNDROME DECODING

f-WEAKLY VALID WITNESS

We say that $x \in \mathbb{F}_2^K$ a f-**weakly valid witness** if x is *almost* a regular vector, in the sense that it differs from a regular vector in at most f blocks.

f-WEAKLY VALID WITNESS

We say that $x \in \mathbb{F}_2^K$ a f-**weakly valid witness** if x is *almost* a regular vector, in the sense that it differs from a regular vector in at most f blocks.
Formally, let $(x^j)_{j \leq w}$ be the w length-T blocks of x. Then x is an f-weakly valid witness if

1. $\forall j \leq w$, $\mathrm{HW}(x^j) = 1 \bmod 2$,
2. $|\{j : \mathrm{HW}(x^j) \neq 1 \mod T/2\}| \leq f$.

f-WEAKLY VALID WITNESS

We say that $x \in \mathbb{F}_2^K$ a f-**weakly valid witness** if x is *almost* a regular vector, in the sense that it differs from a regular vector in at most f blocks.
Formally, let $(x^j)_{j \leq w}$ be the w length-T blocks of x. Then x is an f-weakly valid witness if

1. $\forall j \leq w$, $\mathsf{HW}(x^j) = 1 \bmod 2$,
2. $|\{j : \mathsf{HW}(x^j) \neq 1 \mod T/2\}| \leq f$.

This leads to a **gap**: while an honest witness is assumed to be a standard regular vector, the witness extracted from a malicious prover can be an f-almost-regular vector.

f-ALMOST REGULAR SYNDROME DECODING

f-WEAKLY VALID WITNESS

We say that $x \in \mathbb{F}_2^K$ a f-**weakly valid witness** if x is *almost* a regular vector, in the sense that it differs from a regular vector in at most f blocks.
Formally, let $(x^j)_{j \le w}$ be the w length-T blocks of x. Then x is an f-weakly valid witness if

1. $\forall j \le w$, $\mathsf{HW}(x^j) = 1 \bmod 2$,
2. $|\{j : \mathsf{HW}(x^j) \neq 1 \mod T/2\}| \le f$.

This leads to a **gap**: while an honest witness is assumed to be a standard regular vector, the witness extracted from a malicious prover can be an f-almost-regular vector.

⚠We chose parameters in an area s.t. the f-almost regular syndrome decoding is reduced to the standard regular syndrome decoding.

Parameters (K, k, w, T) with $K > k > w$ and $T \leftarrow K/w$.
$H \in \{0,1\}^{k \times K}, y \in \{0,1\}^k$ are public.

Parameters (K, k, w, T) with $K > k > w$ and $T \leftarrow K/w$.
$H \in \{0,1\}^{k \times K}, y \in \{0,1\}^{k}$ are public.
Inputs P,V: (y, w).
P: $x \in \{0,1\}^{K}$ s.t. $Hx = y \bmod 2$ and $\mathsf{HW}(x) = w \mod T$.

Parameters (K, k, w, T) with $K > k > w$ and $T \leftarrow K/w$.
$H \in \{0,1\}^{k \times K}, y \in \{0,1\}^k$ are public.
Inputs P,V: (y, w).
P: $x \in \{0,1\}^K$ s.t. $Hx = y \bmod 2$ and $\mathrm{HW}(x) = w \mod T$.
Round 1 (P)

- $([\![x]\!]_2, [\![\mathbf{r}]\!]_2, [\![\mathbf{r}]\!]_T) = ((\mathbf{x}_1, \cdots, \mathbf{x}_n), (\mathbf{s}_1, \cdots, \mathbf{s}_n), (\mathbf{t}_1, \cdots, \mathbf{t}_n))$.
- $c_i \leftarrow_r \mathrm{Commit}(\mathbf{x}_i, \mathbf{s}_i, \mathbf{t}_i)$ for $i = 1$ to n.

Parameters (K, k, w, T) with $K > k > w$ and $T \leftarrow K/w$.
$H \in \{0,1\}^{k \times K}, y \in \{0,1\}^k$ are public.
Inputs P,V: (y, w).
P: $x \in \{0,1\}^K$ s.t. $Hx = y \bmod 2$ and $\mathrm{HW}(x) = w \mod T$.
Round 1 (P)

- $([\![x]\!]_2, [\![\mathbf{r}]\!]_2, [\![\mathbf{r}]\!]_T) = ((\mathbf{x}_1, \cdots, \mathbf{x}_n), (\mathbf{s}_1, \cdots, \mathbf{s}_n), (\mathbf{t}_1, \cdots, \mathbf{t}_n))$.
- $c_i \leftarrow_r \mathrm{Commit}(\mathbf{x}_i, \mathbf{s}_i, \mathbf{t}_i)$ for $i = 1$ to n.

Round 2 (V) $\pi \leftarrow_r S_K$.

Parameters (K,k,w,T) with $K > k > w$ and $T \leftarrow K/w$.
$H \in \{0,1\}^{k\times K}, y \in \{0,1\}^k$ are public.
Inputs P,V: (y,w).
P: $x \in \{0,1\}^K$ s.t. $Hx = y \bmod 2$ and $\mathsf{HW}(x) = w \mod T$.

Round 1 (P)

- $([\![x]\!]_2, [\![\mathbf{r}]\!]_2, [\![\mathbf{r}]\!]_T) = ((\mathbf{x}_1,\cdots,\mathbf{x}_n),(\mathbf{s}_1,\cdots,\mathbf{s}_n),(\mathbf{t}_1,\cdots,\mathbf{t}_n))$.
- $c_i \leftarrow_r \mathsf{Commit}(\mathbf{x}_i,\mathbf{s}_i,\mathbf{t}_i)$ for $i = 1$ to n.

Round 2 (V) $\pi \leftarrow_r S_K$.
Round 3 (P) runs the online phase of the MPC in the head

Parameters (K, k, w, T) with $K > k > w$ and $T \leftarrow K/w$.
$H \in \{0,1\}^{k \times K}, y \in \{0,1\}^k$ are public.
Inputs P,V: (y, w).
P: $x \in \{0,1\}^K$ s.t. $Hx = y \bmod 2$ and $\mathrm{HW}(x) = w \mod T$.
Round 1 (P)

- $([\![x]\!]_2, [\![\mathbf{r}]\!]_2, [\![\mathbf{r}]\!]_T) = ((\mathbf{x}_1, \cdots, \mathbf{x}_n), (\mathbf{s}_1, \cdots, \mathbf{s}_n), (\mathbf{t}_1, \cdots, \mathbf{t}_n))$.
- $c_i \leftarrow_r$ Commit$(\mathbf{x}_i, \mathbf{s}_i, \mathbf{t}_i)$ for $i = 1$ to n.

Round 2 (V) $\pi \leftarrow_r S_K$.
Round 3 (P) runs the online phase of the MPC in the head

- $[\![\mathbf{y}']\!]_2 = H \cdot [\![x]\!]_2$

Parameters (K,k,w,T) with $K>k>w$ and $T \leftarrow K/w$.
$H \in \{0,1\}^{k\times K}, y \in \{0,1\}^k$ are public.
Inputs P,V: (y,w).
P: $x \in \{0,1\}^K$ s.t. $Hx = y \bmod 2$ and $\mathrm{HW}(x) = w \mod T$.
Round 1 (P)

- $([\![x]\!]_2, [\![\mathbf{r}]\!]_2, [\![\mathbf{r}]\!]_T) = ((\mathbf{x}_1, \cdots, \mathbf{x}_n), (\mathbf{s}_1, \cdots, \mathbf{s}_n), (\mathbf{t}_1, \cdots, \mathbf{t}_n))$.
- $c_i \leftarrow_r \mathrm{Commit}(\mathbf{x}_i, \mathbf{s}_i, \mathbf{t}_i)$ for $i = 1$ to n.

Round 2 (V) $\pi \leftarrow_r S_K$.
Round 3 (P) runs the online phase of the MPC in the head

- $[\![\mathbf{y}']\!]_2 = H \cdot [\![x]\!]_2$
- $[\![\mathbf{z}]\!]_2 = [\![\pi(\mathbf{r})]\!]_2 + [\![x]\!]_2$

Parameters (K, k, w, T) with $K > k > w$ and $T \leftarrow K/w$.
$H \in \{0,1\}^{k \times K}, y \in \{0,1\}^k$ are public.
Inputs P,V: (y, w).
P: $x \in \{0,1\}^K$ s.t. $Hx = y \bmod 2$ and $\mathrm{HW}(x) = w \mod T$.
Round 1 (P)

- $([\![x]\!]_2, [\![\mathbf{r}]\!]_2, [\![\mathbf{r}]\!]_T) = ((\mathbf{x}_1, \cdots, \mathbf{x}_n), (\mathbf{s}_1, \cdots, \mathbf{s}_n), (\mathbf{t}_1, \cdots, \mathbf{t}_n))$.
- $c_i \leftarrow_r \mathrm{Commit}(\mathbf{x}_i, \mathbf{s}_i, \mathbf{t}_i)$ for $i = 1$ to n.

Round 2 (V) $\pi \leftarrow_r S_K$.
Round 3 (P) runs the online phase of the MPC in the head

- $[\![\mathbf{y}']\!]_2 = H \cdot [\![x]\!]_2$
- $[\![\mathbf{z}]\!]_2 = [\![\pi(\mathbf{r})]\!]_2 + [\![x]\!]_2$
- $[\![\mathbf{w}']\!]_T \leftarrow \langle 1, (\mathbf{z} \odot [\![\mathbf{1} - \pi(\mathbf{r})]\!]_T + (\mathbf{1} - \mathbf{z}) \odot [\![\pi(\mathbf{r})]\!]_T) \rangle$.

Parameters (K,k,w,T) with $K>k>w$ and $T \leftarrow K/w$.
$H \in \{0,1\}^{k\times K}, y \in \{0,1\}^k$ are public.
Inputs P,V: (y,w).
P: $x \in \{0,1\}^K$ s.t. $Hx = y \bmod 2$ and $\mathrm{HW}(x) = w \mod T$.
Round 1 (P)

- $([\![x]\!]_2, [\![\mathbf{r}]\!]_2, [\![\mathbf{r}]\!]_T) = ((\mathbf{x}_1,\cdots,\mathbf{x}_n),(\mathbf{s}_1,\cdots,\mathbf{s}_n),(\mathbf{t}_1,\cdots,\mathbf{t}_n))$.
- $c_i \leftarrow_r \mathrm{Commit}(\mathbf{x}_i,\mathbf{s}_i,\mathbf{t}_i)$ for $i=1$ to n.

Round 2 (V) $\pi \leftarrow_r S_K$.
Round 3 (P) runs the online phase of the MPC in the head

- $[\![\mathbf{y}']\!]_2 = H \cdot [\![x]\!]_2$
- $[\![\mathbf{z}]\!]_2 = [\![\pi(\mathbf{r})]\!]_2 + [\![x]\!]_2$
- $[\![\mathbf{w}']\!]_T \leftarrow \langle 1, (\mathbf{z} \odot [\![\mathbf{1}-\pi(\mathbf{r})]\!]_T + (\mathbf{1}-\mathbf{z}) \odot [\![\pi(\mathbf{r})]\!]_T)\rangle$.

$\mathrm{msg}_i = (\mathbf{y}'_i, \mathbf{z}_i, \mathbf{w}'_i)$

Parameters (K, k, w, T) with $K > k > w$ and $T \leftarrow K/w$.
$H \in \{0,1\}^{k\times K}, y \in \{0,1\}^k$ are public.
Inputs P,V: (y, w).
P: $x \in \{0,1\}^K$ s.t. $Hx = y \bmod 2$ and $\mathsf{HW}(x) = w \mod T$.
Round 1 (P)

- $([\![x]\!]_2, [\![\mathbf{r}]\!]_2, [\![\mathbf{r}]\!]_T) = ((\mathbf{x}_1, \cdots, \mathbf{x}_n), (\mathbf{s}_1, \cdots, \mathbf{s}_n), (\mathbf{t}_1, \cdots, \mathbf{t}_n))$.
- $c_i \leftarrow_r \mathsf{Commit}(\mathbf{x}_i, \mathbf{s}_i, \mathbf{t}_i)$ for $i = 1$ to n.

Round 2 (V) $\pi \leftarrow_r S_K$.
Round 3 (P) runs the online phase of the MPC in the head

- $[\![\mathbf{y}']\!]_2 = H \cdot [\![x]\!]_2$
- $[\![\mathbf{z}]\!]_2 = [\![\pi(\mathbf{r})]\!]_2 + [\![x]\!]_2$
- $[\![\mathbf{w}']\!]_T \leftarrow \langle 1, (\mathbf{z} \odot [\![\mathbf{1} - \pi(\mathbf{r})]\!]_T + (\mathbf{1} - \mathbf{z}) \odot [\![\pi(\mathbf{r})]\!]_T) \rangle$.

$\mathsf{msg}_i = (\mathbf{y}'_i, \mathbf{z}_i, \mathbf{w}'_i)$
Round 4 (V) $d \in [n]$.

Parameters (K, k, w, T) with $K > k > w$ and $T \leftarrow K/w$.
$H \in \{0,1\}^{k \times K}, y \in \{0,1\}^k$ are public.
Inputs P,V: (y, w).
P: $x \in \{0,1\}^K$ s.t. $Hx = y \bmod 2$ and $\mathrm{HW}(x) = w \mod T$.
Round 1 (P)

- $([\![x]\!]_2, [\![\mathbf{r}]\!]_2, [\![\mathbf{r}]\!]_T) = ((\mathbf{x}_1, \cdots, \mathbf{x}_n), (\mathbf{s}_1, \cdots, \mathbf{s}_n), (\mathbf{t}_1, \cdots, \mathbf{t}_n))$.
- $c_i \leftarrow_r \mathrm{Commit}(\mathbf{x}_i, \mathbf{s}_i, \mathbf{t}_i)$ for $i = 1$ to n.

Round 2 (V) $\pi \leftarrow_r S_K$.
Round 3 (P) runs the online phase of the MPC in the head

- $[\![\mathbf{y}']\!]_2 = H \cdot [\![x]\!]_2$
- $[\![\mathbf{z}]\!]_2 = [\![\pi(\mathbf{r})]\!]_2 + [\![x]\!]_2$
- $[\![\mathbf{w}']\!]_T \leftarrow \langle 1, (\mathbf{z} \odot [\![\mathbf{1} - \pi(\mathbf{r})]\!]_T + (\mathbf{1} - \mathbf{z}) \odot [\![\pi(\mathbf{r})]\!]_T) \rangle$.

$\mathrm{msg}_i = (\mathbf{y}'_i, \mathbf{z}_i, \mathbf{w}'_i)$
Round 4 (V) $d \in [n]$.
Round 5 (P) opens c_j for $j \neq d$.

Parameters (K,k,w,T) with $K>k>w$ and $T \leftarrow K/w$.
$H \in \{0,1\}^{k\times K}, y \in \{0,1\}^k$ are public.
Inputs P,V: (y,w).
P: $x \in \{0,1\}^K$ s.t. $Hx = y \bmod 2$ and $\mathrm{HW}(x) = w \mod T$.
Round 1 (P)

- $([\![x]\!]_2, [\![\mathbf{r}]\!]_2, [\![\mathbf{r}]\!]_T) = ((\mathbf{x}_1,\cdots,\mathbf{x}_n),(\mathbf{s}_1,\cdots,\mathbf{s}_n),(\mathbf{t}_1,\cdots,\mathbf{t}_n))$.
- $c_i \leftarrow_r \mathrm{Commit}(\mathbf{x}_i,\mathbf{s}_i,\mathbf{t}_i)$ for $i=1$ to n.

Round 2 (V) $\pi \leftarrow_r S_K$.
Round 3 (P) runs the online phase of the MPC in the head

- $[\![\mathbf{y}']\!]_2 = H\cdot[\![x]\!]_2$
- $[\![\mathbf{z}]\!]_2 = [\![\pi(\mathbf{r})]\!]_2 + [\![x]\!]_2$
- $[\![\mathbf{w}']\!]_T \leftarrow \langle 1, (\mathbf{z}\odot[\![\mathbf{1}-\pi(\mathbf{r})]\!]_T + (\mathbf{1}-\mathbf{z})\odot[\![\pi(\mathbf{r})]\!]_T)\rangle$.

$\mathrm{msg}_i = (\mathbf{y}'_i, \mathbf{z}_i, \mathbf{w}'_i)$
Round 4 (V) $d \in [n]$.
Round 5 (P) opens c_j for $j \neq d$.
Verification (V) checks

- all commitments were opened correctly;
- $\bigoplus_i y'_i = y$ and $\sum_i w'_i = w \mod T$;
- msg_j is consistent with $(\mathbf{x}_j,\mathbf{s}_j,\mathbf{t}_j)$.

THEOREM

Let

- Commit *be a non-interactive commitment scheme,*
- H *be collision-resistant hash function,*
- p *be the combinatorial bound previously discussed.*

Theorem

Let

- Commit *be a non-interactive commitment scheme,*
- H *be collision-resistant hash function,*
- p *be the combinatorial bound previously discussed.*

Then our protocol is a gap honest-verifier zero-knowledge argument of knowledge for the relation $\mathscr{R}$ *such that*

$$((H,y),x) \in \mathscr{R} \text{ if } H \cdot x = y \bmod 2 \text{ and } x \text{ is a regular vector of weight } w$$

The gap relation $\mathscr{R}'$ *is such that*

$$((H,y),x) \in \mathscr{R}' \text{ if } H \cdot x = y \bmod 2 \text{ and } x \text{ is an } f\text{-weakly valid witness}$$

Soundness

Theorem

Let

- Commit *be a non-interactive commitment scheme,*
- H *be collision-resistant hash function,*
- p *be the combinatorial bound previously discussed.*

Then our protocol is a gap honest-verifier zero-knowledge argument of knowledge for the relation $\mathscr{R}$ *such that*

$$((H,y),x) \in \mathscr{R} \text{ if } H \cdot x = y \bmod 2 \text{ and } x \text{ is a regular vector of weight } w$$

The gap relation $\mathscr{R}'$ *is such that*

$$((H,y),x) \in \mathscr{R}' \text{ if } H \cdot x = y \bmod 2 \text{ and } x \text{ is an } f\text{-weakly valid witness}$$

The soundness error of the proof is at most $\varepsilon = \mathsf{p} + 1/n - \mathsf{p}/n$.

Communication cost

Expected communication

$$4\lambda + \tau \cdot \left(\lambda(\log n + 1) + \left(\frac{2n-1}{n}\right) \frac{T-1}{T} (K-k) + \left(\frac{n-1}{n}\right) K \log_2 T/2 \right) \text{bits}$$

HOW DEFINE A SIGNATURE USING FIAT-SHAMIR TRANSORM?

The outputs of the first four round of our 5-round protocol are computed as follows:

- $h_1 = H(m, \mathsf{salt}, h)$,
- $\pi \leftarrow \mathsf{PRG}(h_1)$,
- $h_2 = H(m, \mathsf{salt}, h, h^{'})$,
- $d \leftarrow \mathsf{PRG}(h_2)$.

Final results

$f = 12, K = 1842,\ k = 1017, w = 307, \lambda = 128.$

Setting 1 – fast signature (RSD-F)

$\tau = 18,\ n = 193.$ Signature size $= 12.52$ KB. Runtime 2.7ms.

Setting 2 – medium signature 1 (RSD-M1)

$\tau = 13,\ n = 1723.$ Signature size $= 9.69$ KB. Runtime 17ms.

Setting 3 – medium signature 2 (RSD-M2)

$\tau = 12,\ n = 3391.$ Signature size $= 9.13$ KB. Runtime 31ms.

Setting 4 – short signature (RSD-S)

$\tau = 11,\ n = 7644.$ Signature size $= 8.55$ KB. Runtime 65ms.

Thank you for your attention!

Other developments in the paper:

- Combinatorial Analysis of the Construction
- Uniqueness Bound for Regular Syndrome Decoding
- Relation between SD, RSD and almost RSD
- Improvement of already known attacks against RSD
- Definition of a new attack based on an approximate birthday paradox

Comparison

Scheme	|sgn|	|pk|	t_{sgn}	Assumption
Wave	1.07 KB	3.2 MB	300	large-weight SD over $\mathbb{F}_3$, $(U, U+V)$-codes indist.
Durandal - I	3.97 KB	14.9 KB	4	Rank SD over $\mathbb{F}_{2^m}$
Durandal - II	4.90 KB	18.2 KB	5	Rank SD over $\mathbb{F}_{2^m}$
LESS-FM - I	9.77 KB	15.2 KB	-	Linear Code Equivalence
LESS-FM - II	206 KB	5.25 KB	-	Perm. Code Equivalence
LESS-FM - III	11.57 KB	10.39 KB	-	Perm. Code Equivalence
GPS - 256	24.0 KB	0.11 KB	-	SD over $\mathbb{F}_{256}$
GPS - 256	19.8 KB	0.12 KB	-	SD over $\mathbb{F}_{1024}$
FJR (fast)	22.6 KB	0.09 KB	13	SD over $\mathbb{F}_2$
FJR (short)	16.0 KB	0.09 KB	62	SD over $\mathbb{F}_2$
BGKM Sig1	23.7 KB	0.1 KB	-	SD over $\mathbb{F}_2$
BGKM Sig2	20.6 KB	0.2 KB	-	(QC)SD over $\mathbb{F}_2$
FJR - Var1f	15.6 KB	0.09 KB	-	SD over $\mathbb{F}_2$
FJR - Var1s	10.9 KB	0.09 KB	-	SD over $\mathbb{F}_2$
FJR - Var2f	17.0 KB	0.09 KB	13	SD over $\mathbb{F}_2$
FJR - Var2s	11.8 KB	0.09 KB	64	SD over $\mathbb{F}_2$
FJR - Var3f	11.5 KB	0.14 KB	6	SD over $\mathbb{F}_{256}$
FJR - Var3s	8.26 KB	0.14 KB	30	SD over $\mathbb{F}_{256}$
Our scheme - rsd-f	12.52 KB	0.09 KB	2.8*	RSD over $\mathbb{F}_2$
Our scheme - rsd-m1	9.69 KB	0.09 KB	17*	RSD over $\mathbb{F}_2$
Our scheme - rsd-m2	9.13 KB	0.09 KB	31*	RSD over $\mathbb{F}_2$
Our scheme - rsd-s	8.55 KB	0.09 KB	65*	RSD over $\mathbb{F}_2$
Our scheme - arsd-f	11.25 KB	0.09 KB	2.4*	f-almost-RSD over $\mathbb{F}_2$
Our scheme - arsd-m1	8.76 KB	0.09 KB	15*	f-almost-RSD over $\mathbb{F}_2$
Our scheme - arsd-m2	8.28 KB	0.09 KB	28*	f-almost-RSD over $\mathbb{F}_2$
Our scheme - arsd-s	7.77 KB	0.09 KB	57*	f-almost-RSD over $\mathbb{F}_2$