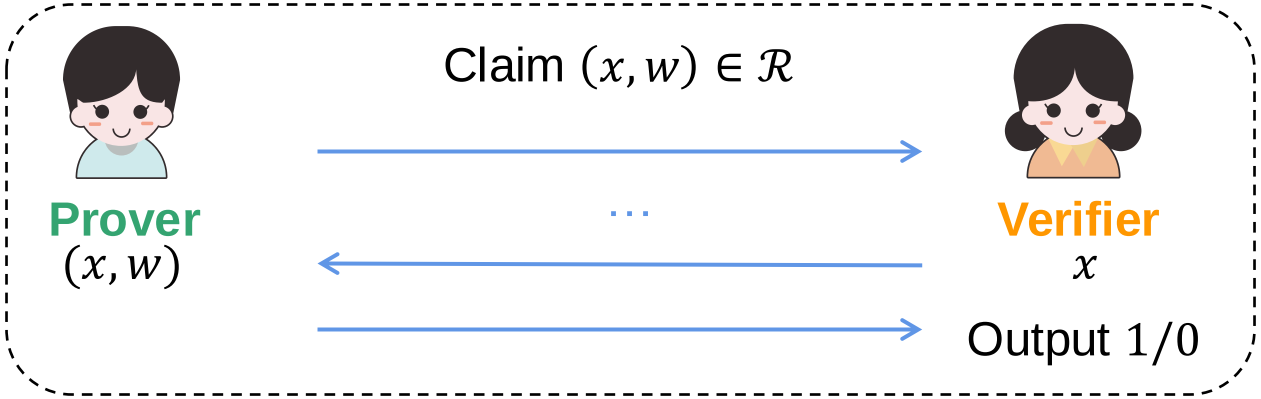
Code-Based Zero-Knowledge from VOLE-in-the-Head and Their Applications: Simpler, Faster, and Smaller

Ying Ouyang   Deng Tang   Yanhong Xu

ASIACRYPT 2024

Background

Zero Knowledge Proofs



- **Completeness**: Verifier always accepts a valid proof.
- **Knowledge Soundness**: If Verifier accepts a proof, then Prover must know a valid witness w.
- **Zero-Knowledge**: Verifier learns nothing about w except $(x, w) \in \mathcal{R}$.

Applications of zero-knowledge proofs

- Privacy-preserving systems such as:
  - Ring signatures (RS)
  - Group signatures (GS)
  - Attribute–based signatures (ABS), …
- Standard signatures

Code-Based Zero Knowledge Protocols

Stern's ZK [Stern96]

- $He = y$
- e has some specific structure

✓ Standard Sig.
✓ Privacy-preserving Sig.

[NTW+19,NNS+21,BGK+23,LNP+24,WCD+24,…]

Large soundness error (2/3)
128-bit security: 219 times
256-bit security: 438 times

MPCitH [IKOS09]

- $C(w) = 1$
- Need a method to share w

VOLEitH [BBG+23]

- $C(w) = 1$ or
- $\begin{cases} f_1(w) = 0 \\ \quad \cdots \\ f_t(w) = 0 \end{cases}$

✓ Standard Sig. [FJR22, CCJ23, MGH+23, MHJ+23, FR23,BCC+24, ARV23,BFG+24,CLY+24,…]

? Privacy-preserving Sig.

Code-Based Zero Knowledge Protocols

Stern's ZK [Stern96]

- $\boldsymbol{He = y}$
- $\boldsymbol{e}$ **has some specific structure**

✓ Standard Sig.
✓ Privacy-preserving Sig.

[NTW+19,NNS+21,BGK+23,LNP+24,WCD+24,…]

Large soundness error (2/3)
128-bit security: 219 times
256-bit security: 438 times

MPCitH [IKOS09]

- $\boldsymbol{C(w) = 1}$
- **Need a method to share** $\boldsymbol{w}$

VOLEitH [BBG+23]

- $\boldsymbol{C(w) = 1}$ **or**
- $\begin{cases} \boldsymbol{f_1(w) = 0} \\ \quad\cdots \\ \boldsymbol{f_t(w) = 0} \end{cases}$

✓ Standard Sig. [FJR22, CCJ23, MGH+23, MHJ+23, FR23,BCC+24, ARV23,BFG+24,CLY+24,…]

? Privacy-preserving Sig.

Can we build code-based privacy-preserving systems from VOLEitH?

Difficulties in Designing Code-Based Privacy-Preserving Systems

- Nguyen et al. [NTW+19AC] built a Merkle-tree accumulator, which employs the following regular encoding.
- Toy Example, it maps c bits to 2^c bits.
  - (00) is encoded to (1000).
  - (01) is encoded to (0100).
  - (10) is encoded to (0010).
  - (11) is encoded to (0001).

  Unit Vectors
- Nguyen et al. designed a dedicated Stern-type ZK for proving the correct regular encoding process.

Difficulties in Designing Code-Based Privacy-Preserving Systems

- Nguyen et al. [NTW+19AC] built a Merkle-tree accumulator, which employs the following regular encoding.
- Toy Example, it maps c bits to 2^c bits.
  - (00) is encoded to (1000).
  - (01) is encoded to (0100).
  - (10) is encoded to (0010).
  - (11) is encoded to (0001).

  Unit Vectors
- Nguyen et al. designed a dedicated Stern-type ZK for proving the correct regular encoding process.

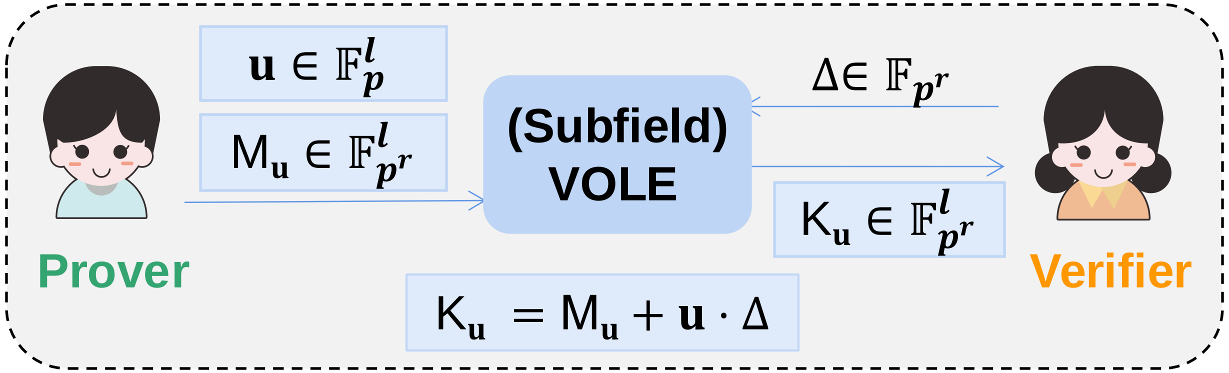
We aim to prove the correct regular encoding process within VOLEitH framework.

Recap:

VOLEitH Proof System

VOLE-based Zero Knowledge Proof

Linearly homomorphic commitment from VOLE:



We define this VOLE correlation by $[\mathbf{u}]$.

Linear homomorphism:

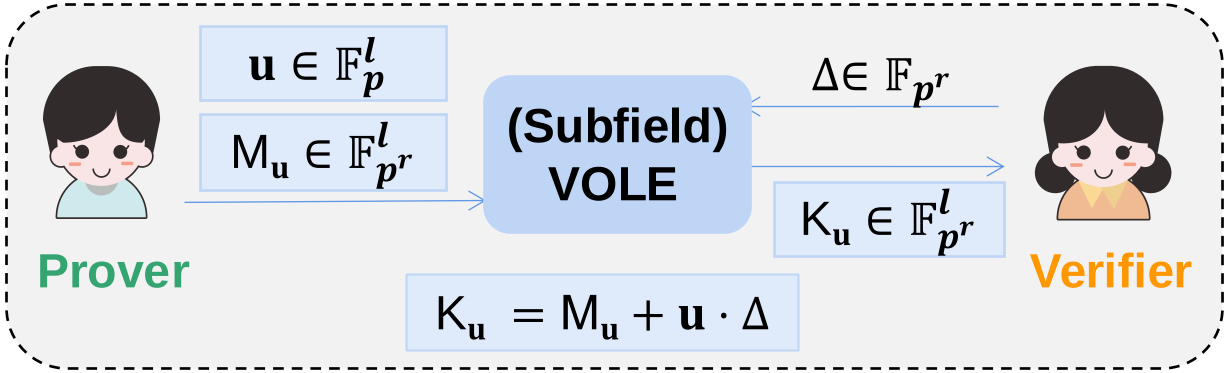
Given $[\mathbf{x}]$, $[\mathbf{y}]$, then $[\boldsymbol{z}] = [a\mathbf{x} + \mathbf{y}]$

is obtained by

$$\underbrace{(a\mathrm{K}_\mathbf{x} + \mathrm{K}_\mathbf{y})}_{\mathrm{K}_\mathbf{z}} = \underbrace{(a\mathrm{M}_\mathbf{x} + \mathrm{M}_\mathbf{y})}_{\mathrm{M}_\mathbf{z}} + \underbrace{(a\mathbf{x} + \mathbf{y})}_{\mathbf{z}} \cdot \Delta$$

VOLE-based Zero Knowledge Proof

Linearly homomorphic commitment from VOLE:



We define this VOLE correlation by $[\mathbf{u}]$.

Linear homomorphism:

Given $[\mathbf{x}]$, $[\mathbf{y}]$, then $[\mathbf{z}] = [a\mathbf{x} + \mathbf{y}]$

is obtained by

$$\underbrace{(a\mathrm{K}_\mathbf{x} + \mathrm{K}_\mathbf{y})}_{\mathrm{K}_\mathbf{z}} = \underbrace{(a\mathrm{M}_\mathbf{x} + \mathrm{M}_\mathbf{y})}_{\mathrm{M}_\mathbf{z}} + \underbrace{(a\mathbf{x} + \mathbf{y})}_{\mathbf{z}} \cdot \Delta$$

Commit to w:

$$\xrightarrow{d = w - u}$$

locally compute $[w]$

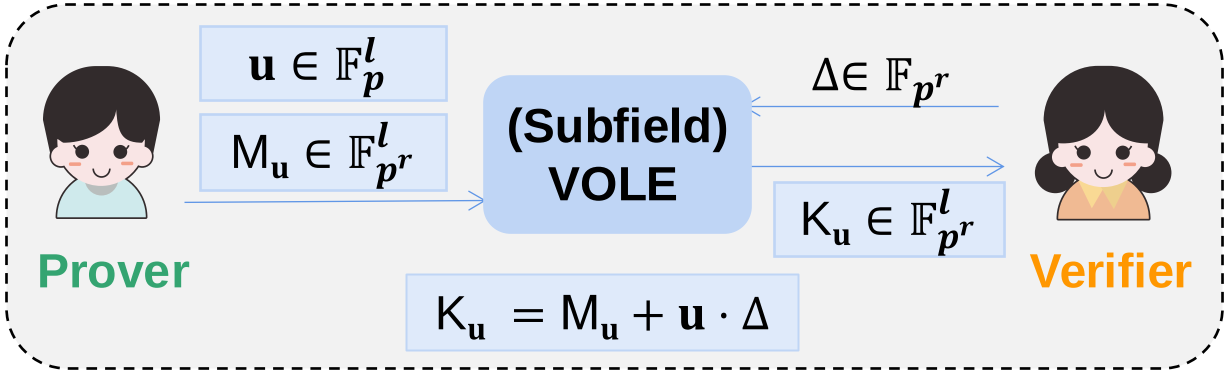
Open:

$$\xrightarrow{w,\ \mathrm{M}_w}$$

verify that $\mathrm{K}_w = \mathrm{M}_w + w \cdot \Delta$

VOLE-based Zero Knowledge Proof

Linearly homomorphic commitment from VOLE:



We define this VOLE correlation by $[\mathbf{u}]$.

Linear homomorphism:

Given $[\mathbf{x}]$, $[\mathbf{y}]$, then $[\mathbf{z}] = [a\mathbf{x} + \mathbf{y}]$ is obtained by

$$\underbrace{(a\mathrm{K}_\mathbf{x} + \mathrm{K}_\mathbf{y})}_{\mathrm{K}_\mathbf{z}} = \underbrace{(a\mathrm{M}_\mathbf{x} + \mathrm{M}_\mathbf{y})}_{\mathrm{M}_\mathbf{z}} + \underbrace{(a\mathbf{x} + \mathbf{y})}_{\mathbf{z}} \cdot \Delta$$

Commit to w:

$$\xrightarrow{d = w - u}$$

locally compute $[w]$

Open:

$$\xrightarrow{w,\ \mathrm{M}_w}$$

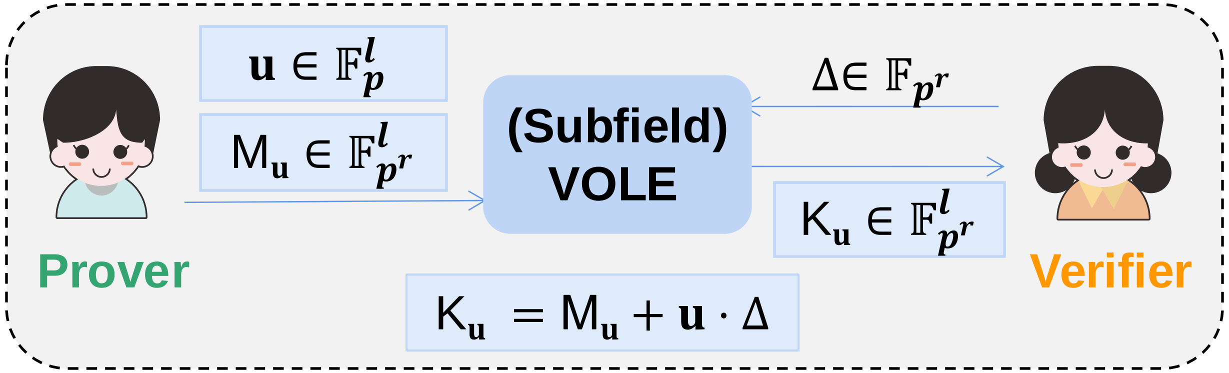
verify that $\mathrm{K}_w = \mathrm{M}_w + w \cdot \Delta$

Proving degree-2 polynomial constraints: $w_1 \cdot w_2 = w_3$

1. The prover commits to w_1, w_2, w_3;
2. How to prove it?

VOLE-based Zero Knowledge Proof

<u>Linearly homomorphic commitment from VOLE:</u>



We define this VOLE correlation by $[\mathbf{u}]$.

Linear homomorphism:

Given $[\mathbf{x}]$, $[\mathbf{y}]$, then $[\mathbf{z}] = [a\mathbf{x} + \mathbf{y}]$

is obtained by

$$\underbrace{(a\mathrm{K}_\mathbf{x} + \mathrm{K}_\mathbf{y})}_{\mathrm{K}_\mathbf{z}} = \underbrace{(a\mathrm{M}_\mathbf{x} + \mathrm{M}_\mathbf{y})}_{\mathrm{M}_\mathbf{z}} + \underbrace{(a\mathbf{x} + \mathbf{y})}_{\mathbf{z}} \cdot \Delta$$

<u>Commit to w:</u>

$$\xrightarrow{d = w - u}$$

locally compute $[w]$

<u>Open:</u>

$$\xrightarrow{w,\ \mathrm{M}_w}$$

verify that $\mathrm{K}_w = \mathrm{M}_w + w \cdot \Delta$

Proving degree-2 polynomial constraints: $w_1 \cdot w_2 = w_3$

1. The prover commits to w_1, w_2, w_3;
2. How to prove it?

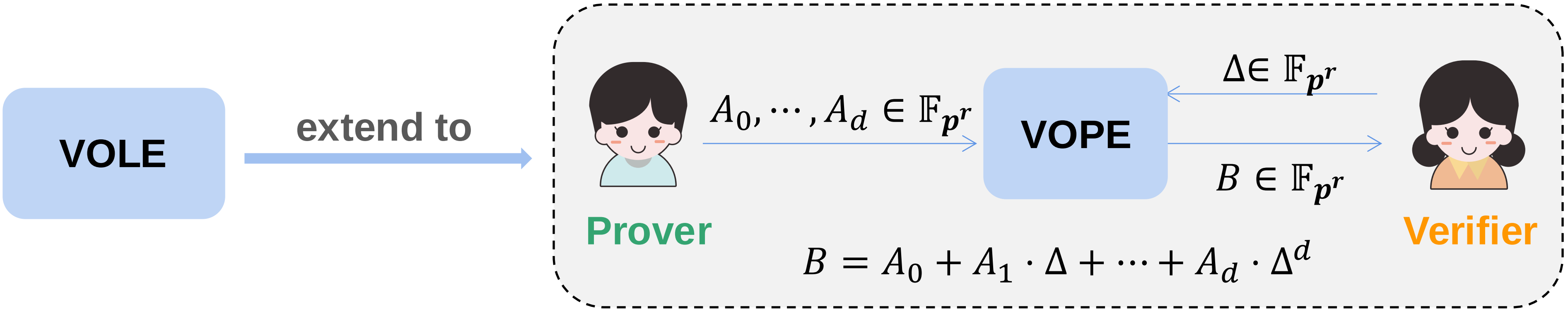
$$B = \underbrace{\mathrm{K}_1 \cdot \mathrm{K}_2 - \mathrm{K}_3 \cdot \Delta}_{\text{know to } \mathcal{V}}$$

$$= \underbrace{\mathrm{M}_1 \cdot \mathrm{M}_2}_{\text{know to } \mathcal{P}} + \underbrace{(\mathrm{M}_2 \cdot w_1 + \mathrm{M}_1 \cdot w_2 - \mathrm{M}_3)}_{\text{know to } \mathcal{P}} \cdot \Delta + \underbrace{(w_1 \cdot w_2 - w_3)}_{0 \text{ if } \mathcal{P} \text{ is honest}} \cdot \Delta^2$$

$$= A_0 + A_1 \cdot \Delta$$

VOLE-based Zero Knowledge Proof

<u>Vector oblivious polynomial evaluation (VOPE):</u>



Extension to prove degree-d polynomial constraints:

degree-separate form: $f(w_1, \cdots, w_l) = \sum_{h \in [0,d]} g_h(w_1, \cdots, w_l) = 0$

$$\mathrm{B} = \underbrace{\sum_{h=0}^{d} g_h(\mathrm{K}_1, \cdots, \mathrm{K}_l) \cdot \Delta^{d-h}}_{\text{know to } \mathcal{V}} = \sum_{h=0}^{d} g_h(\mathrm{M}_1 + w_1 \cdot \Delta, \cdots, \mathrm{M}_l + w_l \cdot \Delta) \cdot \Delta^{d-h}$$

$$= \underbrace{f(w_1, \cdots, w_l)}_{0 \text{ if } \mathcal{P} \text{ is honest}} \cdot \Delta^d + \underbrace{A_0}_{\mathcal{P}} + \underbrace{A_1}_{\mathcal{P}} \cdot \Delta + \cdots + \underbrace{A_{d-1}}_{\text{known to } \mathcal{P}} \cdot \Delta^{d-1}$$

VOLE-in-the-Head

- **VOLE-ZK**
  - **Pros**: Information-theoretic (after VOLE setup) | Fast prover | Small memory
  - **Cons**: Linear proof size | **Designated verifier**
- **VOLE-in-the-Head**: add **public verifiability**

VOLE-ZK
(designated verifier)

A delayed VOLE → functionality

Public-coin “VOLE-in-the-head” proof system [BBG+23]

↓ Fiat-Shamir

Non-interactive ZK

Our Contributions

Summary of Our Contributions

- A novel ZK protocol for proving the correctness of a regular encoding process
- New ZK protocols for concrete code-based relations
    - ZK arguments of knowledge (ZKAoK) of a valid opening
    - ZKAoK of an accumulated value
    - ZKAoK of a plaintext
- Develop several code-based privacy-preserving primitives
    - Efficient RS, GS, and fully dynamic ABS (FDABS)
    - Achieve signature sizes two to three orders of magnitude smaller than Stern-type constructions
- New standard signature
    - Based on regular syndrome decoding problem
    - With “public key + signature size” 3.05 KB for 128-bit security

ZK for Proving the Correctness of a Regular Encoding Process

Regular Encoding Function

RE: $\{0,1\}^c \longrightarrow \{0,1\}^{2^c}$

Input: c bit binary vector $\mathbf{x} = (x_1, x_2, \cdots, x_c)$

Output: 2^c bit unit vector $\mathbf{y} = (y_1, y_2, \cdots, y_{2^c}) = \mathrm{RE}(\mathbf{x})$

Key Observation

RE: $\{0,1\}^c \longrightarrow \{0,1\}^{2^c}$ can be seen as 2^c c-variate Boolean functions.

$\mathbf{ZK}$ for Proving the Correctness of a Regular Encoding Process

Regular Encoding Function
RE: $\{0,1\}^c \longrightarrow \{0,1\}^{2^c}$

Input: c bit binary vector $\mathbf{x} = (x_1, x_2, \cdots, x_c)$
Output: 2^c bit unit vector $\mathbf{y} = (y_1, y_2, \cdots, y_{2^c}) = \text{RE}(\mathbf{x})$

Key Observation

RE: $\{0,1\}^c \longrightarrow \{0,1\}^{2^c}$ can be seen as 2^c c-variate Boolean functions.

Can we express each y_j explicitly using $(x_1, x_2, \cdots, x_c)$?

ZK for Proving the Correctness of a Regular Encoding Process

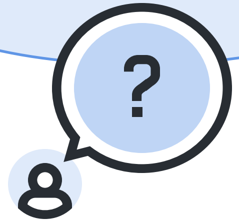
Regular Encoding Function
RE: $\{0,1\}^c \longrightarrow \{0,1\}^{2^c}$

Input: c bit binary vector $\mathbf{x} = (x_1, x_2, \cdots, x_c)$
Output: 2^c bit unit vector $\mathbf{y} = (y_1, y_2, \cdots, y_{2^c}) = \mathrm{RE}(\mathbf{x})$

Key Observation

RE: $\{0,1\}^c \longrightarrow \{0,1\}^{2^c}$ can be seen as 2^c c-variate Boolean functions.

Can we express each y_j explicitly using $(x_1, x_2, \cdots, x_c)$?



YES

ZK for Proving the Correctness of a Regular Encoding Process

Toy Example: $c = 2$

Input	
x_1	x_2
0	0
0	1
1	0
1	1

$\xrightarrow{\text{RE}}$

Output			
y_1	y_2	y_3	y_4
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

ZK for Proving the Correctness of a Regular Encoding Process

Toy Example: $c = 2$

Input	
x_1	x_2
0	0
0	1
1	0
1	1

RE $\rightarrow$

Output			
y_1	y_2	y_3	y_4
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

$$y_1 = f_{(0,0)}(x_1, x_2) = (1 + x_1) \cdot (1 + x_2)$$

$$y_2 = f_{(0,1)}(x_1, x_2) = (1 + x_1) \cdot x_2$$

$$y_3 = f_{(1,0)}(x_1, x_2) = x_1 \cdot (1 + x_2)$$

$$y_4 = f_{(1,1)}(x_1, x_2) = x_1 \cdot x_2$$

ZK for Proving the Correctness of a Regular Encoding Process

Toy Example: $c = 2$

Input	
x_1	x_2
0	0
0	1
1	0
1	1

RE $\rightarrow$

Output			
y_1	y_2	y_3	y_4
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

$$y_1 = f_{(0,0)}(x_1, x_2) = (1 + x_1) \cdot (1 + x_2)$$
$$y_2 = f_{(0,1)}(x_1, x_2) = (1 + x_1) \cdot x_2$$
$$y_3 = f_{(1,0)}(x_1, x_2) = x_1 \cdot (1 + x_2)$$
$$y_4 = f_{(1,1)}(x_1, x_2) = x_1 \cdot x_2$$

$y_j \triangleq f_{(j_1,\cdots,j_c)}(x_1, \cdots, x_c) = \prod_{i=1}^{c}(1 + j_i + x_i)$, where $(j_1, \cdots, j_c) = \text{bin}(j - 1)$

We have transformed the regular encoding process into 2^c degree-c c-variate polynomial relations.

ZK for Proving the Correctness of a Regular Encoding Process

Toy Example: $c = 2$

Input	
x_1	x_2
0	0
0	1
1	0
1	1

RE $\rightarrow$

Output			
y_1	y_2	y_3	y_4
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

$$y_1 = f_{(0,0)}(x_1, x_2) = (1 + x_1) \cdot (1 + x_2)$$

$$y_2 = f_{(0,1)}(x_1, x_2) = (1 + x_1) \cdot x_2$$

$$y_3 = f_{(1,0)}(x_1, x_2) = x_1 \cdot (1 + x_2)$$

$$y_4 = f_{(1,1)}(x_1, x_2) = x_1 \cdot x_2$$

$y_j \triangleq f_{(j_1,\cdots,j_c)}(x_1, \cdots, x_c) = \prod_{i=}^{c}$ … $- 1)$

Thus, can be proven efficiently using VOLEitH proof system.

We have transformed the regular encoding process into 2^c degree-c c-variate polynomial relations.

ZK Arguments of Knowledge of a Valid Opening

Consider the commitment scheme by Nguyen et al. [NTWZ19AC].

Goal

$\mathcal{P}$ is to prove the knowledge of $\mathbf{x} \in \mathbb{F}_2^L$ and $\mathbf{r} \in \mathbb{F}_2^k$ such that

$$\mathbf{c} = \mathbf{C_0} \cdot \mathrm{RE}(\mathbf{x}) \oplus \mathbf{C_1} \cdot \mathrm{RE}(\mathbf{r}) \qquad (1)$$

$\mathbf{x}$: message
$\mathbf{r}$: opening randomness
$\mathbf{c}$: commitment
$\mathbf{C_0}, \mathbf{C_1}$: public matrices param.

ZK Arguments of Knowledge of a Valid Opening

Goal

Consider the commitment scheme by Nguyen et al. [NTWZ19AC].

$\mathcal{P}$ is to prove the knowledge of $\mathbf{x} \in \mathbb{F}_2^L$ and $\mathbf{r} \in \mathbb{F}_2^k$ such that

$$\mathbf{c} = \mathbf{C_0} \cdot \mathrm{RE}(\mathbf{x}) \oplus \mathbf{C_1} \cdot \mathrm{RE}(\mathbf{r}) \qquad (1)$$

$\mathbf{x}$: message
$\mathbf{r}$: opening randomness
$\mathbf{c}$: commitment
$\mathbf{C_0}, \mathbf{C_1}$: public matrices param.

RE functions are degree-c polynomial relations

Matrix multiplication is a linear operation

Equation (1) is equivalent to some degree-c polynomial constraints

Thus, can be proven efficiently using VOLEitH proof system.

ZK Arguments of Knowledge of an Accumulated Value

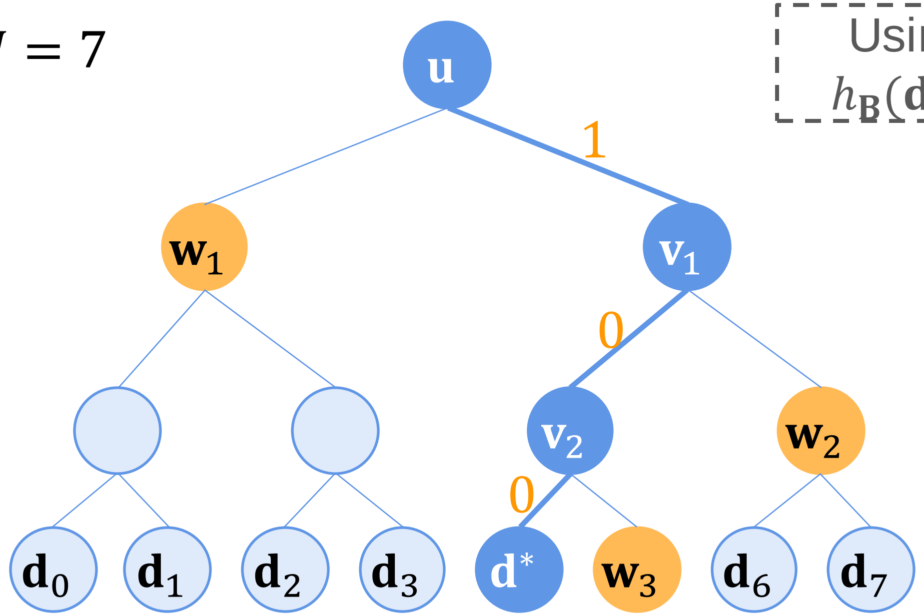
Consider the accumulator by Nguyen et al. [NTWZ19AC].

Goal

$\mathcal{P}$ is to prove the knowledge of $\mathbf{d}^*$ which was accumulated in a value $\mathbf{u}$.

Toy Example: $l = 3, N = 7$

Using the modified AFS hash function:
$h_{\mathbf{B}}(\mathbf{d}_0, \mathbf{d}_1) = \mathbf{B}_0 \cdot \mathrm{RE}(\mathbf{d}_0) \oplus \mathbf{B}_1 \cdot \mathrm{RE}(\mathbf{d}_0)$



$j = 4$

$\mathrm{bin}(j) = (j_1, j_2, j_3) = (100)$

Code-base Merkle-tree Accumulator

ZK Arguments of Knowledge of an Accumulated Value

Consider the accumulator by Nguyen et al. [NTWZ19AC].

Goal

$\mathcal{P}$ is to prove the knowledge of $\mathbf{d}^*$ which was accumulated in a value $\mathbf{u}$.

$\mathcal{P}$ wants to prove knowledge of $j_1, \ldots, j_\ell$, $\mathbf{v}_1, \ldots, \mathbf{v}_\ell = \mathbf{d}^*, \mathbf{w}_1, \ldots, \mathbf{w}_\ell$ such that

$$
\begin{aligned}
\bar{j}_1 \cdot \big(\mathbf{B}_0 \cdot \mathsf{RE}(\mathbf{v}_1) \oplus \mathbf{B}_1 \cdot \mathsf{RE}(\mathbf{w}_1)\big) + j_1 \cdot \big(\mathbf{B}_0 \cdot \mathsf{RE}(\mathbf{w}_1) \oplus \mathbf{B}_1 \cdot \mathsf{RE}(\mathbf{v}_1)\big) &= \mathbf{u}, \\
\bar{j}_2 \cdot \big(\mathbf{B}_0 \cdot \mathsf{RE}(\mathbf{v}_2) \oplus \mathbf{B}_1 \cdot \mathsf{RE}(\mathbf{w}_2)\big) + j_2 \cdot \big(\mathbf{B}_0 \cdot \mathsf{RE}(\mathbf{w}_2) \oplus \mathbf{B}_1 \cdot \mathsf{RE}(\mathbf{v}_2)\big) &= \mathbf{v}_1, \\
&\cdots \\
\bar{j}_l \cdot \big(\mathbf{B}_0 \cdot \mathsf{RE}(\mathbf{v}_\ell) \oplus \mathbf{B}_1 \cdot \mathsf{RE}(\mathbf{w}_\ell)\big) + j_l \cdot \big(\mathbf{B}_0 \cdot \mathsf{RE}(\mathbf{w}_\ell) \oplus \mathbf{B}_1 \cdot \mathsf{RE}(\mathbf{v}_\ell)\big) &= \mathbf{v}_{\ell-1}.
\end{aligned} \tag{2}
$$

ZK Arguments of Knowledge of an Accumulated Value

Goal

Consider the accumulator by Nguyen et al. [NTWZ19AC].

$\mathcal{P}$ is to prove the knowledge of $\mathbf{d}^*$ which was accumulated in a value $\mathbf{u}$.

$\mathcal{P}$ wants to prove knowledge of $j_1, \ldots, j_\ell, \mathbf{v}_1, \ldots, \mathbf{v}_\ell = \mathbf{d}^*, \mathbf{w}_1, \ldots, \mathbf{w}_\ell$ such that

$$
\begin{aligned}
\bar{j_1} \cdot \left(\mathbf{B}_0 \cdot \mathsf{RE}(\mathbf{v}_1) \oplus \mathbf{B}_1 \cdot \mathsf{RE}(\mathbf{w}_1)\right) + j_1 \cdot \left(\mathbf{B}_0 \cdot \mathsf{RE}(\mathbf{w}_1) \oplus \mathbf{B}_1 \cdot \mathsf{RE}(\mathbf{v}_1)\right) &= \mathbf{u}, \\
\bar{j_2} \cdot \left(\mathbf{B}_0 \cdot \mathsf{RE}(\mathbf{v}_2) \oplus \mathbf{B}_1 \cdot \mathsf{RE}(\mathbf{w}_2)\right) + j_2 \cdot \left(\mathbf{B}_0 \cdot \mathsf{RE}(\mathbf{w}_2) \oplus \mathbf{B}_1 \cdot \mathsf{RE}(\mathbf{v}_2)\right) &= \mathbf{v}_1, \\
&\cdots \\
\bar{j_l} \cdot \left(\mathbf{B}_0 \cdot \mathsf{RE}(\mathbf{v}_\ell) \oplus \mathbf{B}_1 \cdot \mathsf{RE}(\mathbf{w}_\ell)\right) + j_l \cdot \left(\mathbf{B}_0 \cdot \mathsf{RE}(\mathbf{w}_\ell) \oplus \mathbf{B}_1 \cdot \mathsf{RE}(\mathbf{v}_\ell)\right) &= \mathbf{v}_{\ell-1}.
\end{aligned}
\tag{2}
$$

RE functions are degree-c polynomial relations

Matrix multiplication is a linear operation

$j_1, \ldots, j_\ell$ are also witnesses

→

Equation (2) are equivalent to some degree-$(c+1)$ polynomial constraints

ZK Arguments of Knowledge of an Accumulated Value

Goal

Consider the accumulator by Nguyen et al. [NTWZ19AC].

$\mathcal{P}$ is to prove the knowledge of $\mathbf{d}^*$ which was accumulated in a value $\mathbf{u}$.

$\mathcal{P}$ wants to prove knowledge of $j_1, \ldots, j_\ell$, $\mathbf{v}_1, \ldots, \mathbf{v}_\ell = \mathbf{d}^*, \mathbf{w}_1, \ldots, \mathbf{w}_\ell$ such that

$$\bar{j}_1 \cdot \left(\mathbf{B}_0 \cdot \mathrm{RE}(\mathbf{v}_1) \oplus \mathbf{B}_1 \cdot \mathrm{RE}(\mathbf{w}_1)\right) + j_1 \cdot \left(\mathbf{B}_0 \cdot \mathrm{RE}(\mathbf{w}_1) \oplus \mathbf{B}_1 \cdot \mathrm{RE}(\mathbf{v}_1)\right)$$

$$\bar{j}_2 \cdot \left(\mathbf{B}_0 \cdot \mathrm{RE}(\mathbf{v}_2) \oplus \mathbf{B}_1 \cdot \mathrm{RE}(\mathbf{w}_2)\right) + j_2 \cdot \left(\mathbf{B}_0 \cdot \mathrm{RE}(\mathbf{w}_2) \oplus \right.$$

$$\cdots$$

$$\bar{j}_l \cdot \left(\mathbf{B}_0 \cdot \mathrm{RE}(\mathbf{v}_\ell) \oplus \mathbf{B}_1 \cdot \mathrm{RE}(\mathbf{w}_\ell)\right) + j_l \cdot \left(\mathbf{B}_0 \cdot \mathrm{RE}(\mathbf{w}_\ell) \oplus \right.$$

Thus, can be proven efficiently using VOLEitH proof system.

RE functions are degree-c polynomial relations

Matrix multiplication is a linear operation

$j_1, \ldots, j_\ell$ are also witnesses

Equation (2) are equivalent to some degree-$(c+1)$ polynomial constraints

ZK Arguments of Knowledge of a Plaintext

Consider a variant of McEliece encryption scheme.

Goal

$\mathcal{P}$ wants to prove knowledge of $\mathbf{u}, \mathbf{m}$ and $\mathbf{e}$ such that

$$\mathbf{c} = \mathbf{G} \cdot \binom{\mathbf{u}}{\mathbf{m}} \oplus \mathsf{RE}(\mathbf{e}) \qquad (3)$$

$\mathbf{u}$: randomness
$\mathbf{m}$: message
$\mathbf{e}$: noise
$\mathbf{c}$: ciphertext
$\mathbf{G}$: pk

ZK Arguments of Knowledge of a Plaintext

Goal

Consider a variant of McEliece encryption scheme.

$\mathcal{P}$ wants to prove knowledge of $\mathbf{u}, \mathbf{m}$ and $\mathbf{e}$ such that

$$\mathbf{c} = \mathbf{G} \cdot \binom{\mathbf{u}}{\mathbf{m}} \oplus \mathrm{RE}(\mathbf{e}) \qquad (3)$$

$\mathbf{u}$: randomness
$\mathbf{m}$: message
$\mathbf{e}$: noise
$\mathbf{c}$: ciphertext
$\mathbf{G}$: pk

Thus, can be proven efficiently using VOLEitH proof system.

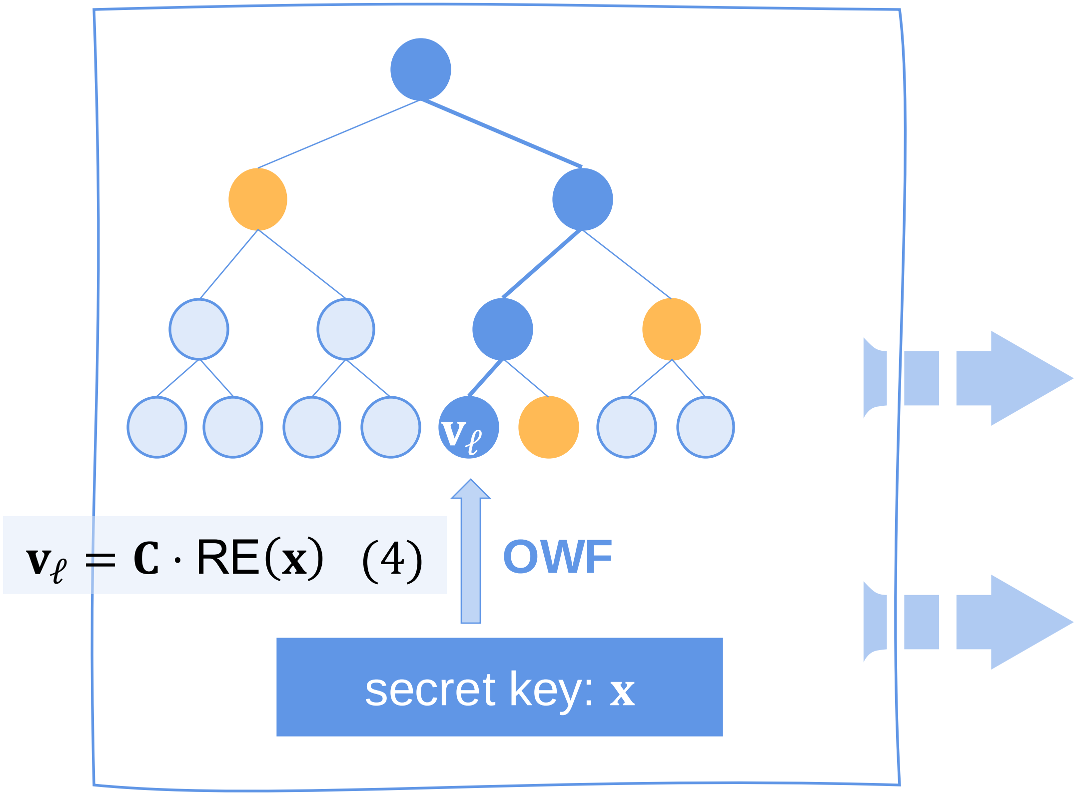
RE can be transformed into degree-c polynomial relations

Matrix multiplication is a linear operation

Equation (3) is equivalent to some degree-c polynomial constraints

Applications

Ring Signatures [NTWZ19AC]



Underlying ZK protocols for the ring signature

ZK of an accumulated value

+

ZK of the knowledge of $\mathbf{x}$ that satisfies (4)

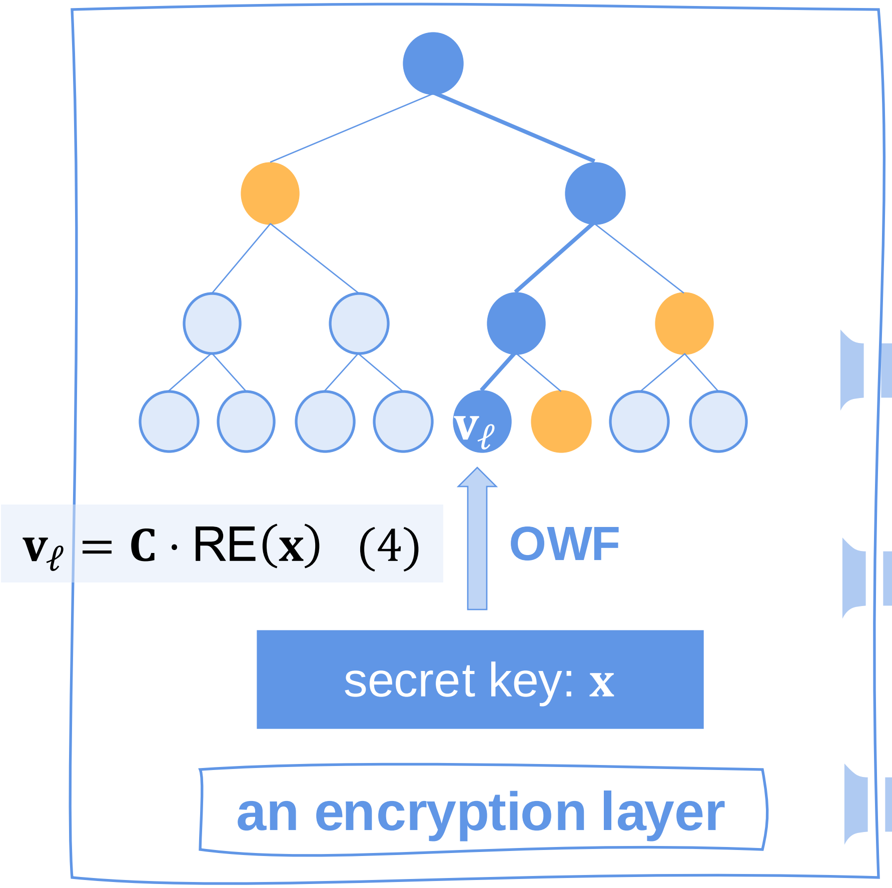
Our Ring Signatures

We replace the stern-like ZK for the ring signature [NTWZ19AC] with our ZK, obtaining a new RS with much smaller signature sizes.

Ring size	128-bit security		256-bit security	
	This paper (KB)	Stern-type (MB)	This paper (KB)	Stern-type (MB)
2^5	35.12	32.26	140.24	129.04
2^7	45.12	43.93	180.25	175.74
2^{10}	60.13	61.44	240.26	245.78
2^{15}	85.14	90.63	340.28	362.51
2^{20}	110.15	119.81	440.30	479.25
2^{30}	160.17	178.18	640.34	712.72

$934\times \sim 1140\times$

Group Signatures [NTWZ19AC]



Underlying ZK protocols for the group signature

ZK of an accumulated value

+

ZK of the knowledge of $\mathbf{x}$ that satisfies(4)

+

ZK of the knowledge of plaintexts

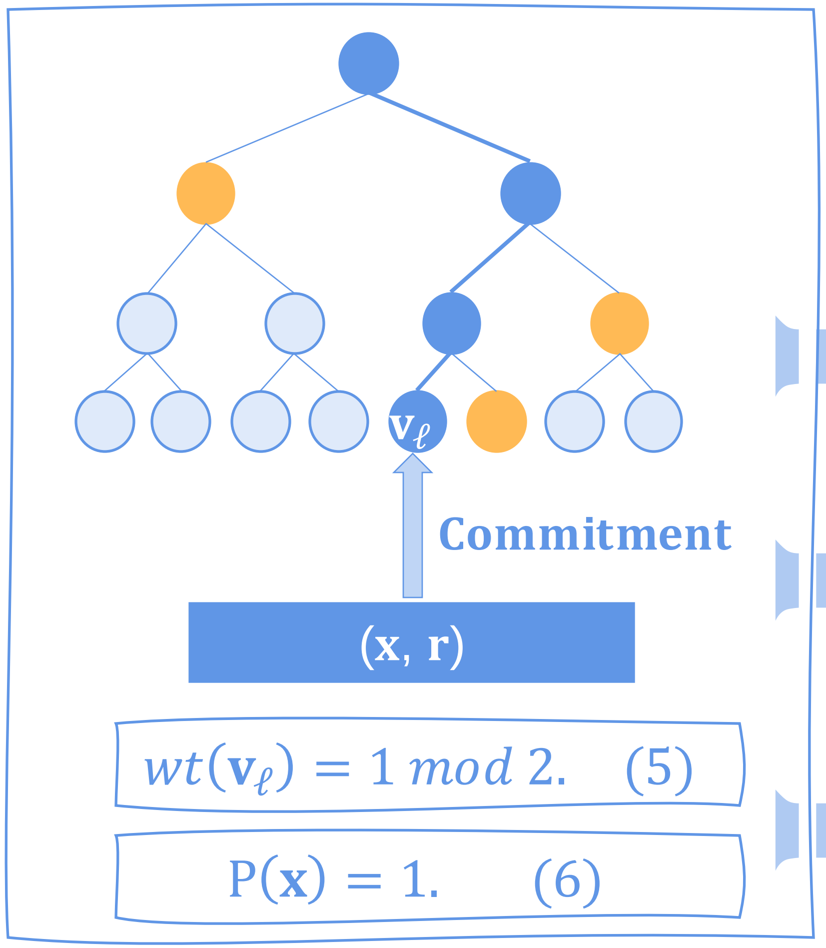
Our Group Signatures

We replace the stern-like ZK for the group signature scheme [NTWZ19AC] using our ZK, obtaining a new GS with much smaller signature sizes.

Group size	128-bit security		256-bit security	
	This paper (KB)	Stern-type (MB)	This paper (KB)	Stern-type (MB)
2^5	49.60	33.27	197.19	133.02
2^7	59.60	44.94	237.19	179.72
2^{10}	74.59	52.45	297.18	249.76
2^{15}	99.58	91.63	397.16	366.50
2^{20}	124.57	120.82	497.14	483.23
2^{30}	174.55	179.18	687.10	716.70

$683\times \sim 1053\times$

Fully Dynamic Attribute-Based Signatures [LNP+24PKC]



Underlying ZK protocols for the FDABS scheme

ZK of an accumulated value

+

ZK of a valid opening

+

ZK protocols for (5)

ZK protocols for (6)

Our Fully Dynamic Attribute-Based Signatures

We replace the stern-like ZK for the FDABS scheme [LNP+24PKC] using our ZK, obtaining a new FDABS with much smaller signature sizes.

$\boldsymbol{2^l}$ denotes the maximum number of attributes; $\boldsymbol{K}$ denotes the size of the circuit P.

$(2^l, K)$	128-bit security		256-bit security	
	This paper (KB)	**Stern-type (MB)**	**This paper (KB)**	**Stern-type (MB)**
$(2^{10},2^{9})$	59.38	45.41	234.76	181.29
$(2^{10},2^{16})$	186.38	52.20	488.76	194.87
$(2^{15},2^{9})$	84.39	67.30	334.78	268.85
$(2^{15},2^{16})$	211.39	74.08	588.78	282.43
$(2^{20},2^{9})$	109.40	89.18	434.80	356.40
$(2^{20},2^{16})$	236.40	95.97	688.80	369.98

$\mathbf{783\times \sim 839\times}$

Comparison with Other Post-Quantum Constructions

- Focus on 128-bit security and ring/group size 2^{10}.
- For FDABS, choose $2^{l} = 2^{10}$, $K = 29$.

Schemes	Code-based			Hash-based [KKW18]	Lattice-based [LN22]
	This paper	Stern-type			
RS	60 KB	61 MB [NTW+19]	61 KB [LW24]	388 KB (240 KB)	13 KB
GS	75 KB	63 MB [NTW+19]	121 KB* [LW24]	418 KB** (297 KB)	18 KB*
FDABS	62 KB	46 MB [LNP+24]	-	-	-

* : They only achieve CPA-anonymity.
**: It only achieves selfless anonymity

A Standard Signature from VOLEitH

<u>A canonical paradigm in signatures</u>

a public coin ZK proof for one-way functions $\xrightarrow{\text{Fiat-Shamir}}$ signatures

- Choose regular syndrome decoding problem

  Let $m = \frac{n}{c} \cdot 2^c$.

  - Verification key: $\mathbf{B} \in \{0,1\}^{n \times m}$ and $\mathbf{y} \in \{0,1\}^n$.
  - Secret Key: $\mathbf{x} \in \{0,1\}^m$ such that

$$\mathbf{B} \cdot \mathsf{RE}(\mathbf{x}) = \mathbf{y}\,. \qquad (4)$$

  - To sign a message: the signer proves knowledge of $\mathbf{x}$ that satisfies (4). This can be achieved using our ZK technique.

Comparison with the Scheme [CLY+24]

- [CLY+24] is also based on RSD problem.
- Different method to prove that a given vector is regular within VOLEitH framework.
- Note: they do not involve proving the regular encoding process.

Scheme parameters		Signature sizes in bytes			
		CLY+24	$c = 2$	$c = 3$	$c = 4$
$\tau = 14$	$T_{open} = -$	4082	4572(+12.0%)	4026(−1.4%)	4040(−1.0%)
	$T_{open} = 112$	3826	4316(+12.9%)	3770(−1.5%)	3784(−1.1%)
$\tau = 10$	$T_{open} = -$	3510	3860(+10.0%)	3470(−1.1%)	3480(−0.9%)
	$T_{open} = 102$	3094	3444(+11.3%)	3054(−1.3%)	3064(−1.0%)

- Adapt optimizations from Baum et al. [BBM+24]: set the same value of T_{open}.
- When $c = 3$, slightly smaller signature size.

Thank you

Q & A

ouyang_ying@sjtu.edu.cn