



CROSS

A signature scheme with restricted errors

Violetta Weger

CBCrypto 2025

May 4, 2025



Darmstadt, 2019



CROSS

A signature scheme with restricted errors

Violetta Weger

CBCrypto 2025

May 4, 2025

cbcrypto@googlegroups.com im Auftrag von Violetta Weger <violetta.weger@tum.de>

Fr 31.01.2025 19:19

CROSS

An: pqc-submissions@nist.gov <pqc-submissions@nist.gov>;

Cc: cbcrypto@googlegroups.com <cbcrypto@googlegroups.com>;

Dear NIST PQC team,

We are hereby submitting version 2.0 of CROSS for the round two.

Please find the complete submission package at:

<https://crypto.deib.polimi.it/submission.zip>

Please let us know if you received this email correctly and can open the zip.

Best wishes,

the CROSS team

--

cbcrypto@googlegroups.com im Auftrag von Violetta Weger <violetta.weger@tum.de>

Fr 31.01.2025 19:19

CROSS

An: pqc-submissions@nist.gov <pqc-submissions@nist.gov>;

~~C: cbcrypto@googlegroups.com <cbcrypto@googlegroups.com>;~~

**cross-crypto@googlegroups.com**

Dear NIST PQC team,

We are hereby submitting version 2.0 of CROSS for the round two.

Please find the complete submission package at:

<https://crypto.deib.polimi.it/submission.zip>

Please let us know if you received this email correctly and can open the zip.

Best wishes,

the CROSS team

--

```

// Expanding secret key
1  $\mathbf{e}, \mathbf{e}_G, \mathbf{H}, \mathbf{M} \leftarrow \text{ExpandSK}(\text{Seed}_{sk})$  ;  $\mathbf{e}, \mathbf{H} \leftarrow \text{ExpandSK}(\text{Seed}_{sk})$ 

// Computing the commitments
2  $\text{Seed} \xleftarrow{\$} \{0,1\}^\lambda$ ,  $\text{Salt} \xleftarrow{\$} \{0,1\}^{2\lambda}$ 
3  $(\text{Seed}[1], \dots, \text{Seed}[t]) \leftarrow \text{SeedLeaves}(\text{Seed}, \text{Salt})$ 
// Compute  $\mathbf{v}[i]$  such that  $\mathbf{v}[i] * \mathbf{e}'[i] = \mathbf{e}$ 
4 for  $i$  from 1 to  $t$  do
     $\mathbf{e}'_G[i], \mathbf{u}'[i] \leftarrow \text{CSPRNG}_{\mathbb{F}_2^m \times \mathbb{F}_p^m}(\text{Seed}[i] \parallel \text{Salt} \parallel i + c)$ ;  $\mathbf{e}'[i], \mathbf{u}'[i] \leftarrow \text{CSPRNG}_{\mathbb{F}_2^m \times \mathbb{F}_p^m}(\text{Seed}[i] \parallel \text{Salt} \parallel i + c)$ 

5  $\mathbf{v}_G[i] \leftarrow \mathbf{e}_G - \mathbf{e}'_G[i]$ 
    $\mathbf{e}'[i] \leftarrow \mathbf{e}'_G[i] \mathbf{M}$ 
    $\mathbf{v}[i] \leftarrow \mathbf{e} - \mathbf{e}'[i]$ 
6 for  $j$  from 1 to  $n$  do
7      $\mathbf{v}[i]_j \leftarrow g^{\mathbf{v}[i]_j}$ 
8      $\mathbf{u}[i] \leftarrow \mathbf{v}[i] * \mathbf{u}'[i]$ 
9      $\mathbf{s}'[i] \leftarrow \mathbf{u}[i] \mathbf{H}^T$ 
10  $\text{cmt}_0[i] \leftarrow \text{Hash}(\mathbf{s}'[i] \parallel \mathbf{v}_G[i] \parallel \text{Salt} \parallel i + c)$  ;  $\text{cmt}_0[i] \leftarrow \text{Hash}(\mathbf{s}'[i] \parallel \mathbf{v}[i] \parallel \text{Salt} \parallel i + c)$ 
     $\text{cmt}_1[i] \leftarrow \text{Hash}(\text{Seed}[i] \parallel \text{Salt} \parallel i + c)$ 
11  $\text{digest}_{\text{cmt}_0} \leftarrow \text{TreeRoot}(\text{cmt}_0[1], \dots, \text{cmt}_0[t])$ 
12  $\text{digest}_{\text{cmt}_1} \leftarrow \text{Hash}(\text{cmt}_1[1] \parallel \dots \parallel \text{cmt}_1[t])$ 
13  $\text{digest}_{\text{cmt}} \leftarrow \text{Hash}(\text{digest}_{\text{cmt}_0} \parallel \text{digest}_{\text{cmt}_1})$ 
// Computing first challenge
14  $\text{digest}_{\text{Msg}} \leftarrow \text{Hash}(\text{Msg})$ 
15  $\text{digest}_{\text{chall}_1} \leftarrow \text{Hash}(\text{digest}_{\text{Msg}} \parallel \text{digest}_{\text{cmt}} \parallel \text{Salt})$ 
16  $\text{chall}_1 \leftarrow \text{CSPRNG}_{(\mathbb{F}_p^*)^t}(\text{digest}_{\text{chall}_1} \parallel t + c)$ 
// Computing first response
17 for  $i$  from 1 to  $t$  do
18     for  $j$  from 1 to  $n$  do
19          $\mathbf{e}'[i]_j \leftarrow g^{\mathbf{e}'[i]_j}$ 
20          $\mathbf{y}[i] \leftarrow \mathbf{u}'[i] + \text{chall}_1[i] \mathbf{e}'[i]$ 
// Computing second challenge
21  $\text{digest}_{\text{chall}_2} \leftarrow \text{Hash}(\mathbf{y}[1] \parallel \dots \parallel \mathbf{y}[t] \parallel \text{digest}_{\text{chall}_1})$ 
22  $\text{chall}_2 \leftarrow \text{CSPRNG}_{\mathcal{B}_{(t,n)}}(\text{digest}_{\text{chall}_2} \parallel t + c + 1)$ 
// Computing second response
23  $\text{Proof} \leftarrow \text{TreeProof}(\text{cmt}_0[1], \dots, \text{cmt}_0[t], \text{chall}_2)$ 
24  $\text{Path} \leftarrow \text{SeedPath}(\text{Seed}, \text{Salt}, \text{chall}_2)$ 
25 for  $i$  from 1 to  $t$  do
26     if  $\text{chall}_2[i] = 0$  then
27          $\text{resp}[i]_0 \leftarrow (\mathbf{y}[i], \mathbf{v}_G[i])$  ;  $\text{resp}[i]_0 \leftarrow (\mathbf{y}[i], \mathbf{v}[i])$ 
         $\text{resp}[i]_1 \leftarrow \text{cmt}_1[i]$ 
// Assembling signature
28  $\text{Sgn} \leftarrow (\text{Salt}, \text{digest}_{\text{cmt}}, \text{digest}_{\text{chall}_2}, \text{Path}, \text{Proof}, \text{resp})$ 

```

```

// Expanding secret key
1  $\bar{e}, \bar{e}_G, H, \bar{M} \leftarrow \text{ExpandSK}(\text{Seed}_{sk})$   $\bar{e}, H \leftarrow \text{ExpandSK}(\text{Seed}_{sk})$ 

// Computing the commitments
2  $\text{Seed} \xleftarrow{\$} \{0,1\}^\lambda$ ,  $\text{Salt} \xleftarrow{\$} \{0,1\}^{2\lambda}$ 
3  $(\text{Seed}[1], \dots, \text{Seed}[t]) \leftarrow \text{SeedLeaves}(\text{Seed}, \text{Salt})$ 
// Compute  $v[i]$  such that  $v[i] * e'[i] = e$ 
4 for  $i$  from 1 to  $t$  do
     $\bar{e}_G[i], u'[i] \leftarrow \text{CSRNG}_{\mathbb{F}_p^* \times \mathbb{F}_p^*}(\text{Seed}[i] \parallel \text{Salt} \parallel i + c)$   $\bar{e}'[i], u'[i] \leftarrow \text{CSRNG}_{\mathbb{F}_p^* \times \mathbb{F}_p^*}(\text{Seed}[i] \parallel \text{Salt} \parallel i + c)$ 

5  $v_G[i] \leftarrow \bar{e}_G - \bar{e}_G'[i]$ 
    $\bar{e}'[i] \leftarrow \bar{e}_G'[i] \bar{M}$ 
    $v[i] \leftarrow \bar{e}^{-1} * v_G[i]$ 

6  $\text{cmt}_1 \leftarrow \text{Hash}(\text{cmt}_1[1])$ 
    $\text{est}_{\text{cmt}} \leftarrow \text{Hash}(\text{digest}_{\text{cmt}_0} \parallel \text{cmt}_1)$   $\text{cmt}_0[i] \leftarrow \text{Hash}(s'[i] \parallel v[i] \parallel \text{Salt} \parallel i + c)$ 

// Computing first challenge
 $\text{digest}_{\text{Msg}} \leftarrow \text{Hash}(\text{Msg})$ 
 $\text{digest}_{\text{chall}_1} \leftarrow \text{Hash}(\text{digest}_{\text{Msg}} \parallel \text{digest}_{\text{cmt}_1})$ 
 $\text{chall}_1 \leftarrow \text{CSRNG}_{(\mathbb{F}_p^*)^t}(\text{digest}_{\text{chall}_1})$ 

// Computing first response
for  $i$  from 1 to  $t$  do
    for  $j$  from 1 to  $n$  do
         $e'[i]_j \leftarrow g^{\bar{e}'[i]_j}$ 
         $\text{resp}[i] \leftarrow u'[i] + \text{chall}_1[i] e'[i]$ 

// Computing second challenge
7  $\text{resp}[i]_0 \leftarrow (y[i], v[i])$ 
    $\text{resp}[i]_1 \leftarrow \text{cmt}_1[i]$ 

// Assembling signature
28  $\text{Sgn} \leftarrow (\text{Salt}, \text{digest}_{\text{cmt}}, \text{digest}_{\text{chall}_2}, \text{Path}, \text{Proof}, \text{resp})$ 

```



## CROSS - in a nutshell



### CROSS in a nutshell

- code-based signature scheme
- 2nd round candidate in NIST standardization call
- Zero-Knowledge (ZK) protocol + Fiat-Shamir transform
- well-known ZK protocol based on decoding random code
- tweak: decoding **restricted** errors
- standard optimization techniques



## CROSS - in a nutshell



### CROSS in a nutshell

- code-based signature scheme
- 2nd round candidate in NIST standardization call
- Zero-Knowledge (ZK) protocol + Fiat-Shamir transform
- well-known ZK protocol based on decoding random code
- tweak: decoding **restricted** errors
- standard optimization techniques



### Advertisement

- optimized AVX2
- memory-optimized
- constant worst-case runtime
- available on liboqs
- EUF-CMA security

**fast** < 1 MCycle (NIST cat. I)  
fits on Cortex-M4 microcontroller  
no signature rejection  
and SUPERCOP  
also BUFF



CROSS

-

Who is behind CROSS?



CROSS

-

Who is behind CROSS?



Felice Manganiello



Freeman Slaughter



Antonia Wachter-Zeh



Sebastian Bitzer



Violetta Weger



Alessio Pavoni



Patrick Karl



Jonas Schupp



Marco Gianvecchio



Alessandro Barengi



Michele Battagliola



Edoardo Signorini



Gerardo Pelosi



Paolo Santini



Marco Baldi



CROSS

-

Why “CROSS”?



CROSS

-

Why “CROSS”?



Codes and Restricted Objects Signature Scheme



CROSS

-

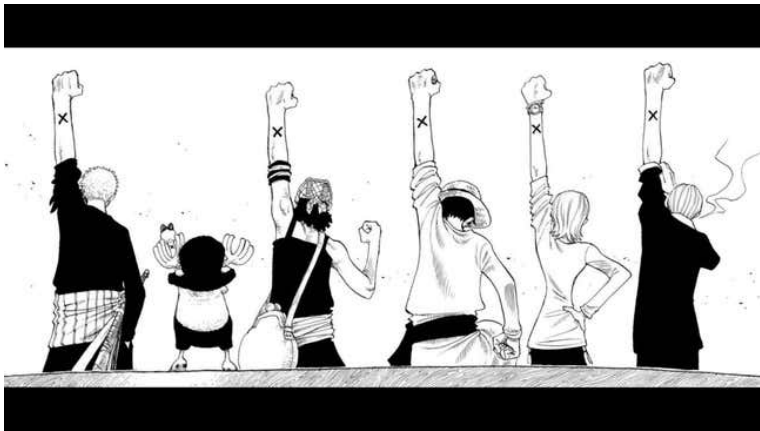
~~Why “CROSS”? Why  $\otimes$ ?~~



CROSS

-

Why "~~CROSS~~"? Why ⊗?





CROSS

-

Why ~~CROSS~~? Why ⊗?





CROSS

-

Why ~~“CROSS”~~? Why ⊗?





CROSS

-

Zero-Knowledge Protocol



CROSS

-

Zero-Knowledge Protocol

Prover



secret key

secret space

Verifier



public key



CROSS

-

Zero-Knowledge Protocol

Prover



secret key

secret space

commitment  $c$

Verifier



public key





CROSS

-

Zero-Knowledge Protocol

Prover



secret key

secret space

commitment  $c$

Verifier



public key



challenge  $z$



CROSS

-

Zero-Knowledge Protocol

Prover



secret key

secret space

commitment  $c$

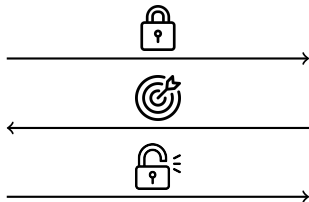
response  $r$

Verifier



public key

challenge  $z$



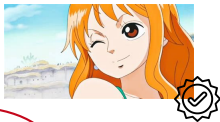


CROSS

-

Zero-Knowledge Protocol

Prover



secret key

secret space

commitment  $c$

response  $r$

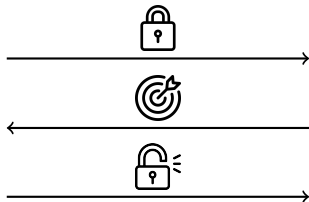
Verifier



public key

challenge  $z$

verify  $\text{key}, z, r \rightarrow c$





CROSS

-

# Zero-Knowledge Protocol

Prover



?

secret space

commitment  $c$

response  $r$

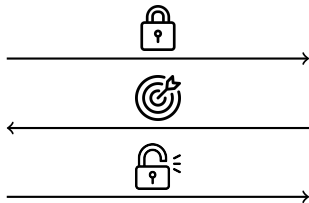
Verifier



$\mathfrak{P}$  public key

challenge  $z$

verify  $\mathfrak{P}, z, r \rightarrow c$





CROSS

-

Zero-Knowledge Protocol

Prover



small cheating probability

commitment  $c$

response  $r$

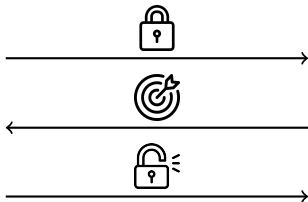
Verifier



$\mathfrak{P}$  public key

challenge  $z$

verify  $\mathfrak{P}, z, r \rightarrow c$



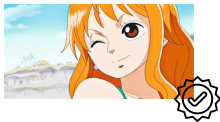


CROSS

-

Zero-Knowledge Protocol

Prover



♀ secret key

Verifier



♀ public key

sec. level  $\lambda$

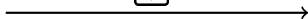
cheat. prob.  $\alpha$

$\rightarrow t$  parallel rounds s.t.  $\alpha^t < 2^{-\lambda}$

commitment  $c$



response  $r$



challenge  $z$

verify ♀,  $z, r \rightarrow c$

$t$  rounds





CROSS

-

Zero-Knowledge Protocol

Signer



$\mathfrak{f}$  secret key

commitment  $c$

$z = \text{Hash}(c, m)$

response  $r$

Fiat-Shamir transform

$m$  message

Verifier



$\mathfrak{g}$  public key

$z = \text{Hash}(c, m)$

verify  $\mathfrak{g}, z, r \rightarrow c$



$\rightarrow$  signature  $= (c_i, r_i)_i$ : transcript of all  $t$  rounds



CROSS

-

Zero-Knowledge Protocol

Prover



SDP: Given  $H, s, t$  find  $e$   
1.  $s = eH^\top$  2.  $\text{wt}(e) = t$

Verifier





CROSS

-

Zero-Knowledge Protocol

Prover



$$\text{wt}(x) = t$$



SDP: Given  $H, s, t$  find  $e$   
1.  $s = eH^\top$  2.  $\text{wt}(e) = t$

Verifier



♀  $H, s, t$



CROSS

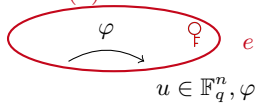
-

Zero-Knowledge Protocol

Prover

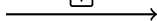


$$\text{wt}(x) = t$$

SDP: Given  $H, s, t$  find  $e$ 1.  $s = eH^\top$  2.  $\text{wt}(e) = t$ 

$$c_1 = \text{Hash}(uH^\top, \varphi)$$

$$c_2 = \text{Hash}(\varphi(u), \varphi(e))$$



Verifier


 $\varphi$   $H, s, t$



CROSS

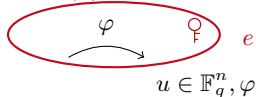
-

Zero-Knowledge Protocol

Prover



$$\text{wt}(x) = t$$

SDP: Given  $H, s, t$  find  $e$ 1.  $s = eH^\top$  2.  $\text{wt}(e) = t$ 

$$c_1 = \text{Hash}(uH^\top, \varphi)$$

$$c_2 = \text{Hash}(\varphi(u), \varphi(e))$$



Verifier

♀  $H, s, t$ 

$$z \in \mathbb{F}_q^*$$



CROSS

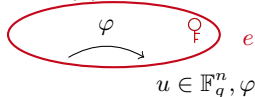
-

Zero-Knowledge Protocol

Prover



$$\text{wt}(x) = t$$



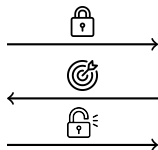
$$c_1 = \text{Hash}(uH^\top, \varphi)$$

$$c_2 = \text{Hash}(\varphi(u), \varphi(e))$$

$$y = \varphi(u + ze)$$

SDP: Given  $H, s, t$  find  $e$ 

1.  $s = eH^\top$
2.  $\text{wt}(e) = t$



Verifier



$$\varphi \quad H, s, t$$

$$z \in \mathbb{F}_q^*$$



CROSS

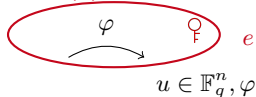
-

Zero-Knowledge Protocol

Prover



$$\text{wt}(x) = t$$



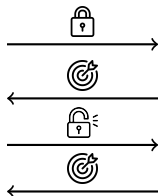
$$c_1 = \text{Hash}(uH^\top, \varphi)$$

$$c_2 = \text{Hash}(\varphi(u), \varphi(e))$$

$$y = \varphi(u + ze)$$

SDP: Given  $H, s, t$  find  $e$ 

1.  $s = eH^\top$
2.  $\text{wt}(e) = t$



Verifier



$$\varphi \quad H, s, t$$

$$z \in \mathbb{F}_q^*$$

$$\beta \in \{1, 2\}$$



CROSS

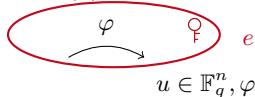
-

Zero-Knowledge Protocol

Prover



$$\text{wt}(x) = t$$



$$c_1 = \text{Hash}(uH^\top, \varphi)$$

$$c_2 = \text{Hash}(\varphi(u), \varphi(e))$$

$$y = \varphi(u + ze)$$

$$r_1 = \varphi$$

$$r_2 = \varphi(e)$$

SDP: Given  $H, s, t$  find  $e$ 

$$1. s = eH^\top \quad 2. \text{wt}(e) = t$$

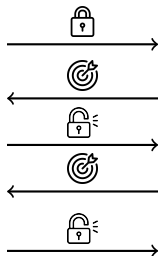
Verifier



$$\varphi \quad H, s, t$$

$$z \in \mathbb{F}_q^*$$

$$\beta \in \{1, 2\}$$





CROSS

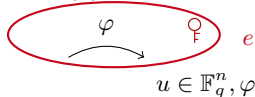
-

Zero-Knowledge Protocol

Prover



$$\text{wt}(x) = t$$



$$y = \varphi(u + ze)$$

$$r_1 = \varphi$$

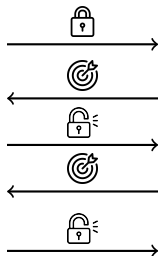
$$r_2 = \varphi(e)$$

SDP: Given  $H, s, t$  find  $e$ 

1.  $s = eH^\top$
2.  $\text{wt}(e) = t$

$$c_1 = \text{Hash}(uH^\top, \varphi)$$

$$c_2 = \text{Hash}(\varphi(u), \varphi(e))$$



Verifier



$$\varphi \quad H, s, t$$

$$z \in \mathbb{F}_q^*$$

$$\beta \in \{1, 2\}$$

$$\varphi^{-1}(y) = u + ze$$



CROSS

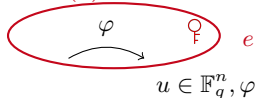
-

Zero-Knowledge Protocol

Prover



$$\text{wt}(x) = t$$



$$y = \varphi(u + ze)$$

$$r_1 = \varphi$$

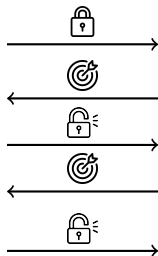
$$r_2 = \varphi(e)$$

SDP: Given  $H, s, t$  find  $e$ 

1.  $s = eH^\top$
2.  $\text{wt}(e) = t$

$$c_1 = \text{Hash}(uH^\top, \varphi)$$

$$c_2 = \text{Hash}(\varphi(u), \varphi(e))$$



Verifier



$$\varphi \quad H, s, t$$

$$z \in \mathbb{F}_q^*$$

$$\beta \in \{1, 2\}$$

$$\varphi^{-1}(y)H^\top = uH^\top + zs$$



CROSS

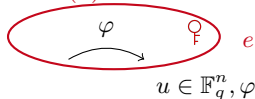
-

Zero-Knowledge Protocol

Prover



$$\text{wt}(x) = t$$



$$c_1 = \text{Hash}(uH^\top, \varphi)$$

$$c_2 = \text{Hash}(\varphi(u), \varphi(e))$$

$$y = \varphi(u + ze)$$

$$r_1 = \varphi$$

$$r_2 = \varphi(e)$$

SDP: Given  $H, s, t$  find  $e$ 

$$1. s = eH^\top \quad 2. \text{wt}(e) = t$$

Verifier

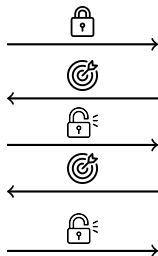


$$\varphi \quad H, s, t$$

$$z \in \mathbb{F}_q^*$$

$$\beta \in \{1, 2\}$$

$$c_1 \stackrel{?}{=} \text{Hash}(uH^\top, \varphi)$$





CROSS

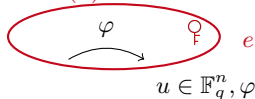
-

Zero-Knowledge Protocol

Prover



$$\text{wt}(x) = t$$



$$c_1 = \text{Hash}(uH^\top, \varphi)$$

$$c_2 = \text{Hash}(\varphi(u), \varphi(e))$$

$$y = \varphi(u + ze)$$

$$r_1 = \varphi$$

$$r_2 = \varphi(e)$$

SDP: Given  $H, s, t$  find  $e$ 

$$1. s = eH^\top \quad 2. \text{wt}(e) = t$$

Verifier



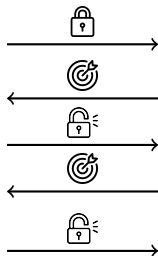
$$\varphi \quad H, s, t$$

$$z \in \mathbb{F}_q^*$$

$$\beta \in \{1, 2\}$$

$$c_1 \stackrel{?}{=} \text{Hash}(uH^\top, \varphi)$$

$$y - z\varphi(e) = \varphi(u)$$





CROSS

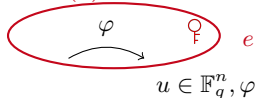
-

Zero-Knowledge Protocol

Prover



$$\text{wt}(x) = t$$



$$c_1 = \text{Hash}(uH^\top, \varphi)$$

$$c_2 = \text{Hash}(\varphi(u), \varphi(e))$$

$$y = \varphi(u + ze)$$

$$r_1 = \varphi$$

$$r_2 = \varphi(e)$$

SDP: Given  $H, s, t$  find  $e$ 

1.  $s = eH^\top$
2.  $\text{wt}(e) = t$

Verifier



$$\varphi \quad H, s, t$$

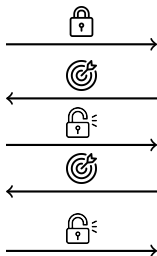
$$z \in \mathbb{F}_q^*$$

$$\beta \in \{1, 2\}$$

$$c_1 \stackrel{?}{=} \text{Hash}(uH^\top, \varphi)$$

$$c_2 \stackrel{?}{=} \text{Hash}(\varphi(u), \varphi(e))$$

$$\text{wt}(\varphi(e)) \stackrel{?}{=} t$$





CROSS

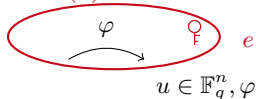
-

Zero-Knowledge Protocol

Prover



$$\text{wt}(x) = t$$



$$c_1 = \text{Hash}(uH^\top, \varphi)$$

$$c_2 = \text{Hash}(\varphi(u), \varphi(e))$$

$$y = \varphi(u + ze)$$

$$r_1 = \varphi$$

$$r_2 = \varphi(e)$$

SDP: Given  $H, s, t$  find  $e$ 

1.  $s = eH^\top$
2.  $\text{wt}(e) = t$

Verifier



$$\varphi \quad H, s, t$$

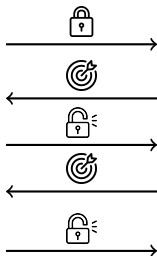
$$z \in \mathbb{F}_q^*$$

$$\beta \in \{1, 2\}$$

$$c_1 \stackrel{?}{=} \text{Hash}(uH^\top, \varphi)$$

$$c_2 \stackrel{?}{=} \text{Hash}(\varphi(u), \varphi(e))$$

$$\text{wt}(\varphi(e)) \stackrel{?}{=} t$$





CROSS

-

Zero-Knowledge Protocol

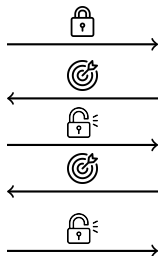
Prover

SDP: Given  $H, s, t$  find  $e$ 1.  $s = eH^\top$  2.  $\text{wt}(e) = t$ probability  $\sim 1/2 \rightarrow t \sim \lambda$ 

$$u \in \mathbb{F}_q^n, \varphi \quad \begin{aligned} c_1 &= \text{Hash}(uH^\top, \varphi) \\ c_2 &= \text{Hash}(\varphi(u), \varphi(e)) \end{aligned}$$

$$y = \varphi(u + ze)$$

$$\begin{aligned} r_1 &= \varphi \\ r_2 &= \varphi(e) \end{aligned}$$



Verifier

♀  $H, s, t$ 

$$z \in \mathbb{F}_q^*$$

$$\beta \in \{1, 2\}$$

$$\begin{aligned} c_1 &\stackrel{?}{=} \text{Hash}(uH^\top, \varphi) \\ c_2 &\stackrel{?}{=} \text{Hash}(\varphi(u), \varphi(e)) \\ \text{wt}(\varphi(e)) &\stackrel{?}{=} t \end{aligned}$$



CROSS

-

Why not submitted?

well-studied ☺



J. Stern. “A new identification scheme based on syndrome decoding”, Annual Int. Cryptology Conf., 1993.



P.-L. Cayrel, P. Véron, S. El Yousfi Alaoui. “A zero-knowledge identification scheme based on the  $q$ -ary syndrome decoding problem”, Int. Workshop on Selected Areas in Cryptography, 2011.



CROSS

-

Why not submitted?

well-studied ☺



J. Stern. “A new identification scheme based on syndrome decoding”, Annual Int. Cryptology Conf., 1993.



P.-L. Cayrel, P. Véron, S. El Yousfi Alaoui. “A zero-knowledge identification scheme based on the  $q$ -ary syndrome decoding problem”, Int. Workshop on Selected Areas in Cryptography, 2011.

too large ☹

large cheating probability  $\rightarrow$  many rounds  $\rightarrow$  large signatures  $\sim 43$  kB for  $\lambda = 128$



CROSS

-

Why not submitted?

well-studied ☺



J. Stern. “A new identification scheme based on syndrome decoding”, Annual Int. Cryptology Conf., 1993.



P.-L. Cayrel, P. Véron, S. El Yousfi Alaoui. “A zero-knowledge identification scheme based on the  $q$ -ary syndrome decoding problem”, Int. Workshop on Selected Areas in Cryptography, 2011.

too large ☹

large cheating probability  $\rightarrow$  many rounds  $\rightarrow$  large signatures  $\sim 43$  kB for  $\lambda = 128$

solution 💡

change protocol: use MPC to split error  $\rightarrow$  smaller cheating probability



CROSS

-

Why not submitted?

well-studied 😊



J. Stern. “A new identification scheme based on syndrome decoding”, Annual Int. Cryptology Conf., 1993.



P.-L. Cayrel, P. Véron, S. El Yousfi Alaoui. “A zero-knowledge identification scheme based on the  $q$ -ary syndrome decoding problem”, Int. Workshop on Selected Areas in Cryptography, 2011.

too large 😞

large cheating probability  $\rightarrow$  many rounds  $\rightarrow$  large signatures  $\sim 43$  kB for  $\lambda = 128$

solution 💡

change protocol: use MPC to split error  $\rightarrow$  smaller cheating probability

$\rightarrow$  used in

- SDitH (regular)
- RYDE (rank)
- Mirath (matrix rank)
- PERK (PKP)



CROSS

-

what is MPCitH?

Prover

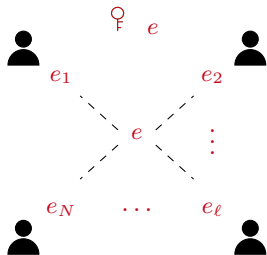


SDP: Given  $H, s, t$  find  $e$   
1.  $s = eH^\top$  2.  $\text{wt}(e) = t$

Verifier



$\text{♀ } H, s, t$





CROSS

-

what is MPCitH?

Prover



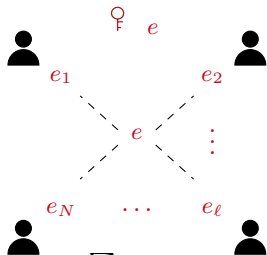
SDP: Given  $H, s, t$  find  $e$

1.  $s = eH^\top$  2.  $\text{wt}(e) = t$

Verifier



$H, s, t$



$$\sum e_i = e$$

1.  $s_i = e_i H^\top \rightarrow \sum s_i = s$

2.  $\alpha_i \rightarrow \text{wt}(e) = t$



CROSS

-

what is MPCitH?

Prover

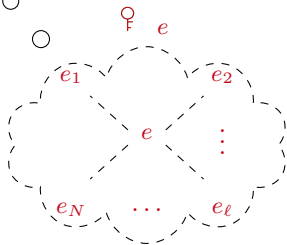


SDP: Given  $H, s, t$  find  $e$   
1.  $s = eH^\top$  2.  $\text{wt}(e) = t$

Verifier



$H, s, t$



$$\sum e_i = e$$

1.  $s_i = e_i H^\top \rightarrow \sum s_i = s$

2.  $\alpha_i \rightarrow \text{wt}(e) = t$

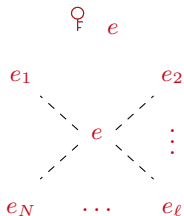


CROSS

-

what is MPCitH?

Prover

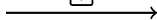


- $$\sum e_i = e$$
1.  $s_i = e_i H^\top \rightarrow \sum s_i = s$
  2.  $\alpha_i \rightarrow \text{wt}(e) = t$



SDP: Given  $H, s, t$  find  $e$

1.  $s = eH^\top$
2.  $\text{wt}(e) = t$

 $c_i = (s_i, \alpha_i)$ 


Verifier


 $\text{♀ } H, s, t$

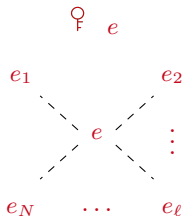


CROSS

-

what is MPCitH?

Prover



- $$\sum e_i = e$$
1.  $s_i = e_i H^\top \rightarrow \sum s_i = s$
  2.  $\alpha_i \rightarrow \text{wt}(e) = t$



SDP: Given  $H, s, t$  find  $e$   
1.  $s = eH^\top$  2.  $\text{wt}(e) = t$

$c_i = (s_i, \alpha_i)$



Verifier



$\text{♀ } H, s, t$

$z \in \{1, \dots, N\}$

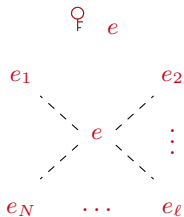


CROSS

-

what is MPCitH?

Prover



- $$\sum e_i = e$$
1.  $s_i = e_i H^\top \rightarrow \sum s_i = s$
  2.  $\alpha_i \rightarrow \text{wt}(e) = t$



SDP: Given  $H, s, t$  find  $e$

1.  $s = eH^\top$
2.  $\text{wt}(e) = t$

$$c_i = (s_i, \alpha_i)$$

$$r_z = (e_i)_{i \neq z}$$



Verifier



$$\text{♀ } H, s, t$$

$$z \in \{1, \dots, N\}$$

$$\text{check } (c_i)_{i \neq z}$$



CROSS

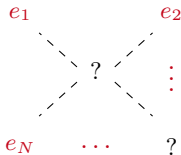
-

what is MPCitH?

Prover



???



$$\sum e_i = e$$

$$1. s_i = e_i H^\top \rightarrow \sum s_i = s$$

$$2. \alpha_i \rightarrow \text{wt}(e) = t$$

SDP: Given  $H, s, t$  find  $e$ 

$$1. s = eH^\top \quad 2. \text{wt}(e) = t$$

$$c_i = (s_i, \alpha_i)$$



$$r_z = (e_i)_{i \neq z}$$

Verifier



$$\text{? } H, s, t$$

$$z \in \{1, \dots, N\}$$

check  $(c_i)_{i \neq z}$



CROSS

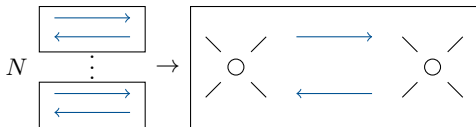
-

what is MPCitH?

Prover

SDP: Given  $H, s, t$  find  $e$ 1.  $s = eH^\top$  2.  $\text{wt}(e) = t$ 

Verifier

cheating probability  $\sim 1/N$  $c_i = (s_i, \alpha_i)$  $\forall H, s, t$  $z \in \{1, \dots, N\}$ check  $(c_i)_{i \neq z}$ 😊 smaller cheating prob  $\rightarrow$  smaller signatures



CROSS

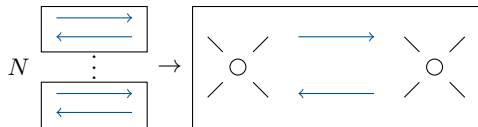
-

what is MPCitH?

Prover



Verifier

SDP: Given  $H, s, t$  find  $e$ 1.  $s = eH^\top$  2.  $\text{wt}(e) = t$ cheating probability  $\sim 1/N$  $c_i = (s_i, \alpha_i)$  $\forall H, s, t$  $z \in \{1, \dots, N\}$ check  $(c_i)_{i \neq z}$ 😊 smaller cheating prob  $\rightarrow$  smaller signatures😞 more computations  $\rightarrow$  slow



CROSS

-

Why not submitted?

well-studied ☺



J. Stern. “A new identification scheme based on syndrome decoding”, Annual Int. Cryptology Conf., 1993.



P.-L. Cayrel, P. Véron, S. El Yousfi Alaoui. “A zero-knowledge identification scheme based on the  $q$ -ary syndrome decoding problem”, Int. Workshop on Selected Areas in Cryptography, 2011.

too large ☹

large cheating probability  $\rightarrow$  many rounds  $\rightarrow$  large signatures  $\sim 43$  kB for  $\lambda = 128$



CROSS

-

Why not submitted?

well-studied ☺



J. Stern. “A new identification scheme based on syndrome decoding”, Annual Int. Cryptology Conf., 1993.



P.-L. Cayrel, P. Véron, S. El Yousfi Alaoui. “A zero-knowledge identification scheme based on the  $q$ -ary syndrome decoding problem”, Int. Workshop on Selected Areas in Cryptography, 2011.

too large ☹

large communication cost in 1 round



CROSS

-

Why not submitted?

well-studied ☺



J. Stern. “A new identification scheme based on syndrome decoding”, Annual Int. Cryptology Conf., 1993.

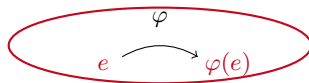


P.-L. Cayrel, P. Véron, S. El Yousfi Alaoui. “A zero-knowledge identification scheme based on the  $q$ -ary syndrome decoding problem”, Int. Workshop on Selected Areas in Cryptography, 2011.

too large ☹

large communication cost in 1 round

secret space



$\text{wt}(x) = t$

errors  $|e| = t(\log(n) + \log(q - 1))$

maps  $|\varphi| = n(\log(n) + \log(q - 1))$



CROSS

-

Why not submitted?

well-studied ☺



J. Stern. “A new identification scheme based on syndrome decoding”, Annual Int. Cryptology Conf., 1993.



P.-L. Cayrel, P. Véron, S. El Yousfi Alaoui. “A zero-knowledge identification scheme based on the  $q$ -ary syndrome decoding problem”, Int. Workshop on Selected Areas in Cryptography, 2011.

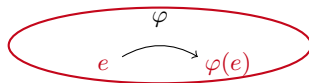
too large ☹

large communication cost in 1 round

solution 💡

change problem: use restricted errors

secret space



$\text{wt}(x) = t$

errors  $|e| = t(\log(n) + \log(q - 1))$

maps  $|\varphi| = n(\log(n) + \log(q - 1))$



CROSS

-

What are restricted errors?



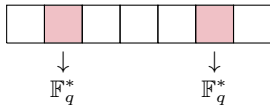
CROSS

-

What are restricted errors?



SDP: Given  $H, s, t$  find  $e$ : 1.  $s = eH^\top$ , 2.  $\text{wt}(e) = t$

 $e$ 

$$\varphi \in (\mathbb{F}_q^*)^n \rtimes S_n$$



CROSS

-

What are restricted errors?



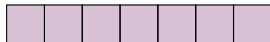
SDP: Given  $H, s, t$  find  $e$ : 1.  $s = eH^\top$ , 2.  $\text{wt}(e) = t$

 $e$  $\downarrow$   
 $\mathbb{F}_q^*$  $\downarrow$   
 $\mathbb{F}_q^*$ 

$$\varphi \in (\mathbb{F}_q^*)^n \rtimes S_n$$



R-SDP: Given  $H, s, \mathbb{E}$  find  $e$ : 1.  $s = eH^\top$ , 2.  $e \in \mathbb{E}^n$

 $e$  $\downarrow$   
 $\mathbb{E}$  $\dots$  $\downarrow$   
 $\mathbb{E}$ 

$$\varphi : \mathbb{E}^n \rightarrow \mathbb{E}^n ?$$



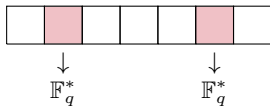
CROSS

-

What are restricted errors?



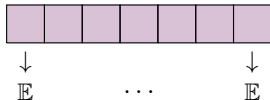
SDP: Given  $H, s, t$  find  $e$ : 1.  $s = eH^\top$ , 2.  $\text{wt}(e) = t$

 $e$ 

$$\varphi \in (\mathbb{F}_q^*)^n \rtimes S_n$$



R-SDP: Given  $H, s, \mathbb{E}$  find  $e$ : 1.  $s = eH^\top$ , 2.  $e \in \mathbb{E}^n$

 $e$ 

$$\varphi : \mathbb{E}^n \rightarrow \mathbb{E}^n ?$$

$$\mathbb{E} < \mathbb{F}_q^*$$

$$\mathbb{E} = \{g^i \mid i \in \{1, \dots, z\}\}$$

$$\text{ord}(g) = z \text{ prime}$$



CROSS

-

What are restricted errors?



R-SDP: Given  $H, s, \mathbb{E}$  find  $e$ : 1.  $s = eH^\top$ , 2.  $e \in \mathbb{E}^n$

$$\mathbb{E}^n = \{g^i\}^n \subset \mathbb{F}_q^n$$

$e$

|           |         |           |
|-----------|---------|-----------|
| $g^{i_1}$ | $\dots$ | $g^{i_n}$ |
|-----------|---------|-----------|



CROSS

-

What are restricted errors?



R-SDP: Given  $H, s, \mathbb{E}$  find  $e$ : 1.  $s = eH^\top$ , 2.  $e \in \mathbb{E}^n$

$$\mathbb{E}^n = \{g^i\}^n \subset \mathbb{F}_q^n$$

 $e$ 

|           |         |           |
|-----------|---------|-----------|
| $g^{i_1}$ | $\dots$ | $g^{i_n}$ |
|-----------|---------|-----------|

 $\ell(e)$ 

|       |         |       |
|-------|---------|-------|
| $i_1$ | $\dots$ | $i_n$ |
|-------|---------|-------|

 $\mathbb{F}_z^n$



CROSS

-

What are restricted errors?



R-SDP: Given  $H, s, \mathbb{E}$  find  $e$ : 1.  $s = eH^\top$ , 2.  $e \in \mathbb{E}^n$

$$\mathbb{E}^n = \{g^i\}^n \subset \mathbb{F}_q^n$$

$$e \quad \boxed{g^{i_1} \mid \cdots \mid g^{i_n}}$$



$$\ell(e) \quad \boxed{i_1 \mid \cdots \mid i_n}$$

$$\mathbb{F}_z^n$$

$$\rightarrow |e| = n \log(z)$$



CROSS

-

What are restricted errors?



R-SDP: Given  $H, s, \mathbb{E}$  find  $e$ : 1.  $s = eH^\top$ , 2.  $e \in \mathbb{E}^n$

$$\mathbb{E}^n = \{g^i\}^n \subset \mathbb{F}_q^n$$

$$e \quad \boxed{g^{i_1} \mid \cdots \mid g^{i_n}} \quad \xrightarrow{\varphi} \quad e' \quad \boxed{g^{j_1} \mid \cdots \mid g^{j_n}}$$



$$\ell(e) \quad \boxed{i_1 \mid \cdots \mid i_n}$$

$$\mathbb{F}_z^n$$

$$\rightarrow |e| = n \log(z)$$



CROSS

-

What are restricted errors?



R-SDP: Given  $H, s, \mathbb{E}$  find  $e$ : 1.  $s = eH^\top$ , 2.  $e \in \mathbb{E}^n$

$$\mathbb{E}^n = \{g^i\}^n \subset \mathbb{F}_q^n$$

$$e \quad \boxed{g^{i_1} \mid \cdots \mid g^{i_n}} \quad \star \quad \boxed{g^{j_1-i_1} \mid \cdots \mid g^{j_n-i_n}} \quad e' \quad \boxed{g^{j_1} \mid \cdots \mid g^{j_n}}$$



$$\ell(e) \quad \boxed{i_1 \mid \cdots \mid i_n}$$

 $\mathbb{F}_z^n$ 

$$\rightarrow |e| = n \log(z)$$



CROSS

-

What are restricted errors?



R-SDP: Given  $H, s, \mathbb{E}$  find  $e$ : 1.  $s = eH^\top$ , 2.  $e \in \mathbb{E}^n$

$$\mathbb{E}^n = \{g^i\}^n \subset \mathbb{F}_q^n$$

$$e \quad \boxed{g^{i_1} \mid \cdots \mid g^{i_n}} \quad \star \quad \boxed{g^{j_1 - i_1} \mid \cdots \mid g^{j_n - i_n}} \quad e' \quad \boxed{g^{j_1} \mid \cdots \mid g^{j_n}}$$



$$\ell(e) \quad \boxed{i_1 \mid \cdots \mid i_n} \quad \boxed{j_1 - i_1 \mid \cdots \mid j_n - i_n} \quad \ell(e') \quad \boxed{j_1 \mid \cdots \mid j_n}$$

 $\mathbb{F}_z^n$ 

$$\rightarrow |e| = n \log(z)$$

$$\rightarrow |\varphi| = n \log(z)$$



CROSS

-

What are restricted errors?



R-SDP: Given  $H, s, \mathbb{E}$  find  $e$ : 1.  $s = eH^\top$ , 2.  $e \in \mathbb{E}^n$

$$\mathbb{E}^n = \{g^i\}^n \subset \mathbb{F}_q^n$$

$$e \quad \boxed{g^{i_1} \mid \cdots \mid g^{i_n}} \quad \star \quad \boxed{g^{j_1-i_1} \mid \cdots \mid g^{j_n-i_n}} \quad e' \quad \boxed{g^{j_1} \mid \cdots \mid g^{j_n}}$$



$$\ell(e) \quad \boxed{i_1 \mid \cdots \mid i_n} \quad + \quad \boxed{j_1 - i_1 \mid \cdots \mid j_n - i_n} \quad \ell(e') \quad \boxed{j_1 \mid \cdots \mid j_n}$$

 $\mathbb{F}_z^n$ 

$$\rightarrow |e| = n \log(z)$$

$$\rightarrow |\varphi| = n \log(z)$$



CROSS

-

What are restricted errors?



R-SDP: Given  $H, s, \mathbb{E}$  find  $e$ : 1.  $s = eH^\top$ , 2.  $e \in \mathbb{E}^n$

$$\mathbb{E}^n = \{g^i\}^n \subset \mathbb{F}_q^n$$

$$e \quad \boxed{g^{i_1} \mid \cdots \mid g^{i_n}} \quad \star \quad \boxed{g^{j_1-i_1} \mid \cdots \mid g^{j_n-i_n}} \quad e' \quad \boxed{g^{j_1} \mid \cdots \mid g^{j_n}}$$



$$\ell(e) \quad \boxed{i_1 \mid \cdots \mid i_n} \quad + \quad \boxed{j_1 - i_1 \mid \cdots \mid j_n - i_n} \quad \ell(e') \quad \boxed{j_1 \mid \cdots \mid j_n}$$

 $\mathbb{F}_z^n$ 

$$\rightarrow |e| = n \log(z)$$

$$\rightarrow |\varphi| = n \log(z)$$

$$\rightarrow (\mathbb{E}^n, \star) \cong (\mathbb{F}_z^n, +)$$



CROSS

-

What are restricted errors?



R-SDP: Given  $H, s, \mathbb{E}$  find  $e$ : 1.  $s = eH^\top$ , 2.  $e \in \mathbb{E}^n$

$$\mathbb{E}^n = \{g^i\}^n \subset \mathbb{F}_q^n$$

$$e \quad \boxed{g^{i_1} \mid \cdots \mid g^{i_n}} \quad \star \quad \boxed{g^{j_1-i_1} \mid \cdots \mid g^{j_n-i_n}} \quad e' \quad \boxed{g^{j_1} \mid \cdots \mid g^{j_n}}$$



$$\ell(e) \quad \boxed{i_1 \mid \cdots \mid i_n} \quad + \quad \boxed{j_1 - i_1 \mid \cdots \mid j_n - i_n} \quad \ell(e') \quad \boxed{j_1 \mid \cdots \mid j_n}$$

 $\mathbb{F}_z^n$ 

$$\rightarrow |e| = n \log(z)$$

$$\rightarrow |\varphi| = n \log(z)$$

$$\rightarrow (\mathbb{E}^n, \star) \cong (\mathbb{F}_z^n, +)$$

☺ smaller sizes

☺ faster arithmetic



CROSS

-

Toy example

$$\mathbb{E} = \{3^0, 3^1, 3^2\} = \{1, 3, 9\} < \mathbb{F}_{13}$$

$$e = (1, 9, 3, 3)$$

$$e' = (3, 1, 1, 3)$$



CROSS

-

Toy example

$$\mathbb{E} = \{3^0, 3^1, 3^2\} = \{1, 3, 9\} < \mathbb{F}_{13}$$

$$e = (1, 9, 3, 3)$$

$$\star(3, 3, 9, 1)$$

$$e' = (3, 1, 1, 3)$$



CROSS

-

Toy example

$$\mathbb{E} = \{3^0, 3^1, 3^2\} = \{1, 3, 9\} < \mathbb{F}_{13}$$

$$e = (1, 9, 3, 3)$$

$$\star(3, 3, 9, 1)$$

$$e' = (3, 1, 1, 3)$$



$$\ell(e) = (0, 2, 1, 1)$$

$$+(1, 1, 2, 0)$$

$$\ell(e') = (1, 0, 0, 1)$$

$$(\mathbb{F}_3^n, +)$$



CROSS

-

Toy example

$$\mathbb{E} = \{3^0, 3^1, 3^2\} = \{1, 3, 9\} < \mathbb{F}_{13}$$

$$e = (1, 9, 3, 3)$$

$$\star(3, 3, 9, 1)$$

$$e' = (3, 1, 1, 3)$$



$$\ell(e) = (0, 2, 1, 1)$$

$$+(1, 1, 2, 0)$$

$$\ell(e') = (1, 0, 0, 1)$$

$$(\mathbb{F}_3^n, +)$$

allowing any exponent in  $\mathbb{F}_z^n$ only allow exponents in  $\mathcal{C} \subset \mathbb{F}_z^n$



CROSS

-

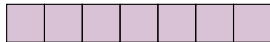
What are restricted errors?



SDP: Given  $H, s, t$  find  $e$ : 1.  $s = eH^\top$ , 2.  $\text{wt}(e) = t$

 $e$  $\in \mathbb{F}_q^*$ 

R-SDP: Given  $H, s, \mathbb{E}$  find  $e$ : 1.  $s = eH^\top$ , 2.  $e \in \mathbb{E}^n$

 $e$  $\in \mathbb{E} < \mathbb{F}_q^*$  $\ell(\mathbb{E}^n) \cong \mathbb{F}_z^n$



CROSS

-

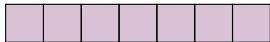
What are restricted errors?



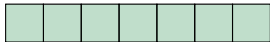
SDP: Given  $H, s, t$  find  $e$ : 1.  $s = eH^\top$ , 2.  $\text{wt}(e) = t$

 $e$  $\in \mathbb{F}_q^*$ 

R-SDP: Given  $H, s, \mathbb{E}$  find  $e$ : 1.  $s = eH^\top$ , 2.  $e \in \mathbb{E}^n$

 $e$  $\in \mathbb{E} < \mathbb{F}_q^*$  $\ell(\mathbb{E}^n) \cong \mathbb{F}_z^n$ 

R-SDP( $G$ ): Given  $H, s, G < \mathbb{E}^n$  find  $e$ : 1.  $s = eH^\top$ , 2.  $e \in G$

 $e$  $\in G < \mathbb{E}^n$  $\ell(G) \cong \mathcal{C} \subset \mathbb{F}_z^n$



CROSS

-

What are restricted errors?



R-SDP( $G$ ): Given  $H, s, G < \mathbb{E}^n$  find  $e$ : 1.  $s = eH^\top$ , 2.  $e \in G$

$G \subset \mathbb{F}_q^n$

$$(G, \star) < (\mathbb{F}_q^n, \star)$$

$$G = \langle x_1, \dots, x_m \rangle_\star$$

$$e = x_1^{u_1} \star \dots \star x_m^{u_m}$$



CROSS

-

What are restricted errors?



R-SDP( $G$ ): Given  $H, s, G < \mathbb{E}^n$  find  $e$ : 1.  $s = eH^\top$ , 2.  $e \in G$

$G \subset \mathbb{F}_q^n$

$(G, \star) < (\mathbb{E}^n, \star)$

$G = \langle x_1, \dots, x_m \rangle_\star$

$e = x_1^{u_1} \star \dots \star x_m^{u_m}$



$M = \begin{pmatrix} \ell(x_1) \\ \vdots \\ \ell(x_m) \end{pmatrix} \in \mathbb{F}_z^{m \times n}$

$\ell(e) = uM \in \langle M \rangle$

$\rightarrow$  only need  $u \in \mathbb{F}_z^m$

$\mathbb{F}_z^n$



CROSS

-

What are restricted errors?



R-SDP( $G$ ): Given  $H, s, G < \mathbb{E}^n$  find  $e$ : 1.  $s = eH^\top$ , 2.  $e \in G$

$$G \subset \mathbb{F}_q^n$$

$$(G, \star) < (\mathbb{E}^n, \star)$$

$$G = \langle x_1, \dots, x_m \rangle_\star$$

$$e = x_1^{u_1} \star \dots \star x_m^{u_m}$$



$$M = \begin{pmatrix} \ell(x_1) \\ \vdots \\ \ell(x_m) \end{pmatrix} \in \mathbb{F}_z^{m \times n}$$

$$\ell(e) = uM \in \langle M \rangle$$

$\rightarrow$  only need  $u \in \mathbb{F}_z^m$

$$\begin{array}{ccc} \mathbb{F}_z^m & \xrightarrow{M} & \mathbb{F}_z^n \\ u = (u_1, \dots, u_m) & & \end{array}$$

$$\ell(e) = uM = (i_1, \dots, i_n)$$

$$\begin{array}{ccc} & \xrightarrow{g} & \mathbb{E}^n \subset \mathbb{F}_q^n \\ & & e = (g^{i_1}, \dots, g^{i_n}) \end{array}$$



CROSS

-

What are restricted errors?



R-SDP( $G$ ): Given  $H, s, G < \mathbb{E}^n$  find  $e$ : 1.  $s = eH^\top$ , 2.  $e \in G$

$G \subset \mathbb{F}_q^n$

$(G, \star) < (\mathbb{E}^n, \star)$

$G = \langle x_1, \dots, x_m \rangle_\star$

$e = x_1^{u_1} \star \dots \star x_m^{u_m}$



$M = \begin{pmatrix} \ell(x_1) \\ \vdots \\ \ell(x_m) \end{pmatrix} \in \mathbb{F}_z^{m \times n}$

$\ell(e) = uM \in \langle M \rangle$

$\rightarrow$  only need  $u \in \mathbb{F}_z^m$

$\rightarrow |e| = |\varphi| = m \log(z)$

☺ even smaller



CROSS

-

Toy example

$$G < \mathbb{E}^n = \{1, 3, 9\}^n \subset \mathbb{F}_{13}^n$$

$$G = \langle x_1, x_2 \rangle$$

$$x_1 = (3, 1, 1, 3), x_2 = (1, 3, 9, 1)$$

$$e = x_1^2 \star x_2^1 = (9, 3, 9, 9)$$



CROSS

-

Toy example

$$G < \mathbb{E}^n = \{1, 3, 9\}^n \subset \mathbb{F}_{13}^n$$

$$G = \langle x_1, x_2 \rangle$$

$$x_1 = (3, 1, 1, 3), x_2 = (1, 3, 9, 1)$$

$$e = x_1^2 \star x_2^1 = (9, 3, 9, 9)$$



$$\ell(G) = \langle M \rangle \subset \mathbb{F}_3^n$$

$$M = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 2 & 0 \end{pmatrix}$$

$$\ell(e) = (2, 1)M = (2, 1, 1, 2)$$

$$\langle M \rangle \subset \mathbb{F}_3^n$$



CROSS

-

Toy example

$$G < \mathbb{E}^n = \{1, 3, 9\}^n \subset \mathbb{F}_{13}^n$$

$$G = \langle x_1, x_2 \rangle$$

$$x_1 = (3, 1, 1, 3), x_2 = (1, 3, 9, 1)$$

$$e = x_1^2 \star x_2^1 = (9, 3, 9, 9)$$



$$\ell(G) = \langle M \rangle \subset \mathbb{F}_3^n$$

$$M = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 2 & 0 \end{pmatrix}$$

$$\ell(e) = (2, 1)M = (2, 1, 1, 2)$$

$$\langle M \rangle \subset \mathbb{F}_3^n$$

previous  $e = (1, 9, 3, 3) \notin G$



CROSS

-

How hard is decoding?

- $\mathbb{E}, G$  have **multiplicative** structure

$$e = (g^{i_1}, \dots, g^{i_n})$$

- $s = eH^\top$  has **additive** structure

$$s_j = \sum_{\ell=1}^n h_{j,\ell} g^{i_\ell} \text{ for } j \in \{1, \dots, n-k\}$$



CROSS

-

How hard is decoding?

- $\mathbb{E}, G$  have **multiplicative** structure

$$e = (g^{i_1}, \dots, g^{i_n})$$

not DLP

- $s = eH^\top$  has **additive** structure

$$s_j = \sum_{\ell=1}^n h_{j,\ell} g^{i_\ell} \text{ for } j \in \{1, \dots, n-k\}$$



CROSS

-

How hard is decoding?

- $\mathbb{E}, G$  have **multiplicative** structure

$$e = (g^{i_1}, \dots, g^{i_n})$$

not DLP

- similar to SDP  $\rightarrow$  NP-hard

- $s = eH^\top$  has **additive** structure

$$s_j = \sum_{\ell=1}^n h_{j,\ell} g^{i_\ell} \text{ for } j \in \{1, \dots, n-k\}$$

- adapted ISD algorithms



## CROSS

-

## How hard is decoding?

- $\mathbb{E}, G$  have **multiplicative** structure

$$e = (g^{i_1}, \dots, g^{i_n})$$

not DLP

- similar to SDP  $\rightarrow$  NP-hard
- generalize to  $e \in (\mathbb{E} \cup \{0\})^n$ ,  $\text{wt}(e) = t$
- choose  $t$  s.t. unique solution ( $q = 251$ )

- $s = eH^\top$  has **additive** structure

$$s_j = \sum_{\ell=1}^n h_{j,\ell} g^{i_\ell} \text{ for } j \in \{1, \dots, n-k\}$$

- adapted ISD algorithms



- $\mathbb{E}, G$  have **multiplicative** structure

$$e = (g^{i_1}, \dots, g^{i_n}) \quad \boxed{\text{not DLP}}$$

- similar to SDP  $\rightarrow$  NP-hard

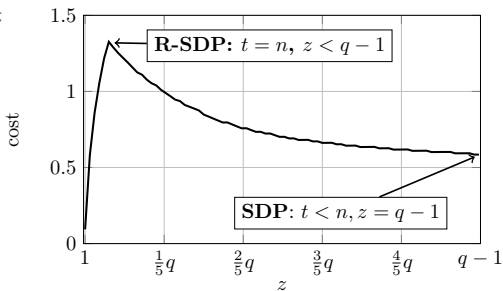
- generalize to  $e \in (\mathbb{E} \cup \{0\})^n$ ,  $\text{wt}(e) = t$

- choose  $t$  s.t. unique solution ( $q = 251$ )

- $s = eH^\top$  has **additive** structure

$$s_j = \sum_{\ell=1}^n h_{j,\ell} g^{i_\ell} \text{ for } j \in \{1, \dots, n-k\}$$

- adapted ISD algorithms





## CROSS

-

## How hard is decoding?

- $\mathbb{E}, G$  have **multiplicative** structure

$$e = (g^{i_1}, \dots, g^{i_n})$$

not DLP

- similar to SDP  $\rightarrow$  NP-hard
- Take  $\mathbb{E}$  with **no** additive structure

- $s = eH^\top$  has **additive** structure

$$s_j = \sum_{\ell=1}^n h_{j,\ell} g^{i_\ell} \text{ for } j \in \{1, \dots, n-k\}$$

- adapted ISD algorithms
- $(\mathbb{E} + \mathbb{E}) \cap (\mathbb{E} \cup \{0\})$  small



## CROSS

-

## How hard is decoding?

- $\mathbb{E}, G$  have **multiplicative** structure

$$e = (g^{i_1}, \dots, g^{i_n})$$

not DLP

- similar to SDP  $\rightarrow$  NP-hard
- Take  $\mathbb{E}$  with **no** additive structure

- $s = eH^\top$  has **additive** structure

$$s_j = \sum_{\ell=1}^n h_{j,\ell} g^{i_\ell} \text{ for } j \in \{1, \dots, n-k\}$$

- adapted ISD algorithms

○ **good**:  $q = 13, g = 3, \mathbb{E} = \{1, 3, 9\}$

○ **bad**:  $q = 13, g = 5, \mathbb{E} = \{\pm 1, \pm 5\}$



- $\mathbb{E}, G$  have **multiplicative** structure

$$e = (g^{i_1}, \dots, g^{i_n})$$

not DLP

- similar to SDP  $\rightarrow$  NP-hard
- Take  $\mathbb{E}$  with **no** additive structure
- Also no shift  $\mathbb{E} + x$  has additive structure

- $s = eH^\top$  has **additive** structure

$$s_j = \sum_{\ell=1}^n h_{j,\ell} g^{i_\ell} \text{ for } j \in \{1, \dots, n-k\}$$

- adapted ISD algorithms
- **good**:  $q = 13, g = 3, \mathbb{E} = \{1, 3, 9\}$
- **bad**:  $q = 13, g = 5, \mathbb{E} = \{\pm 1, \pm 5\}$



- $\mathbb{E}, G$  have **multiplicative** structure

$$e = (g^{i_1}, \dots, g^{i_n})$$

not DLP

- similar to SDP  $\rightarrow$  NP-hard
- Take  $\mathbb{E}$  with **no** additive structure
- Also no shift  $\mathbb{E} + x$  has additive structure

- $s = eH^\top$  has **additive** structure

$$s_j = \sum_{\ell=1}^n h_{j,\ell} g^{i_\ell} \text{ for } j \in \{1, \dots, n-k\}$$

- adapted ISD algorithms

○ **good**:  $q = 13, g = 3, \mathbb{E} = \{1, 3, 9\}$

○ **bad**:  $q = 13, g = 5, \mathbb{E} = \{\pm 1, \pm 5\}$

$$\rightarrow q = 127, z = 7$$



## CROSS

-

## How hard is decoding?

- $\mathbb{E}, G$  have **multiplicative** structure

$$e = (g^{i_1}, \dots, g^{i_n})$$

not DLP

- similar to SDP  $\rightarrow$  NP-hard
- Take  $\mathbb{E}$  with **no** additive structure
- Also no shift  $\mathbb{E} + x$  has additive structure

- combinatorial solvers:

ISD algos

- $s = eH^\top$  has **additive** structure

$$s_j = \sum_{\ell=1}^n h_{j,\ell} g^{i_\ell} \text{ for } j \in \{1, \dots, n-k\}$$

- adapted ISD algorithms

◦ **good**:  $q = 13, g = 3, \mathbb{E} = \{1, 3, 9\}$

◦ **bad**:  $q = 13, g = 5, \mathbb{E} = \{\pm 1, \pm 5\}$

$$\rightarrow q = 127, z = 7$$



S. Bitzer, A. Pavoni, V. Weger, P. Santini, M. Baldi, and A. Wachter-Zeh. “**Generic Decoding of Restricted Errors**”, ISIT, 2023.



M. Baldi, S. Bitzer, A. Pavoni, P. Santini, A. Wachter-Zeh, and V. Weger. “**Zero knowledge protocols and signatures from the restricted syndrome decoding problem**”, PKC, 2024.



## CROSS

-

## How hard is decoding?

- $\mathbb{E}, G$  have **multiplicative** structure

$$e = (g^{i_1}, \dots, g^{i_n})$$

not DLP

- similar to SDP  $\rightarrow$  NP-hard
- Take  $\mathbb{E}$  with **no** additive structure
- Also no shift  $\mathbb{E} + x$  has additive structure

- $s = eH^\top$  has **additive** structure

$$s_j = \sum_{\ell=1}^n h_{j,\ell} g^{i_\ell} \text{ for } j \in \{1, \dots, n-k\}$$

- adapted ISD algorithms
- **good**:  $q = 13, g = 3, \mathbb{E} = \{1, 3, 9\}$
- **bad**:  $q = 13, g = 5, \mathbb{E} = \{\pm 1, \pm 5\}$

$$\rightarrow q = 127, z = 7$$

- combinatorial solvers:

ISD algos



S. Bitzer, A. Pavoni, V. Weger, P. Santini, M. Baldi, and A. Wachter-Zeh. “[Generic Decoding of Restricted Errors](#)”, ISIT, 2023.



M. Baldi, S. Bitzer, A. Pavoni, P. Santini, A. Wachter-Zeh, and V. Weger. “[Zero knowledge protocols and signatures from the restricted syndrome decoding problem](#)”, PKC, 2024.

- algebraic attacks:

$$e_i^z = 1 \text{ Gröbner basis}$$



M. Baldi, et al. “[CROSS security guide](#)”, NIST PQC round 2, 2025.



W. Beullens, P. Briaud, M. Øygarden. “[A Security Analysis of Restricted Syndrome Decoding Problems](#)”, 2024.



CROSS

-

How to decode?

◦  $q = 127, z = 7$

→  $\mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$

unique sol.  $\frac{z^n}{q^{n-k}} < 1$



CROSS

-

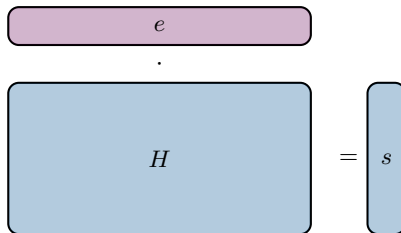
How to decode?

◦  $q = 127, z = 7$

→  $\mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$

unique sol.  $\frac{z^n}{q^{n-k}} < 1$

### 1. Reduction





CROSS

-

How to decode?

◦  $q = 127, z = 7$

→  $\mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$

unique sol.  $\frac{z^n}{q^{n-k}} < 1$

### 1. Reduction

$$\begin{array}{|c|c|} \hline e' & e'' \\ \hline \end{array} \cdot \begin{array}{|c|c|} \hline \text{Id} & H' \\ \hline & H'' \\ \hline \end{array} = \begin{array}{|c|} \hline s' \\ \hline s'' \\ \hline \end{array}$$



CROSS

-

How to decode?

◦  $q = 127, z = 7$

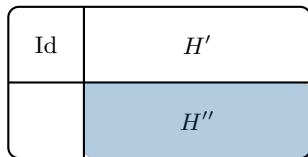
→  $\mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$

unique sol.  $\frac{z^n}{q^{n-k}} < 1$

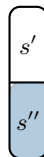
### 1. Reduction



.



=



$$e' = s' - e''H'^{\top} \in \mathbb{E}?$$



CROSS

-

How to decode?

◦  $q = 127, z = 7 \rightarrow \mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$

unique sol.  $\frac{z^n}{q^{n-k}} < 1$

## 2. Meet-in-the-Middle

$$\begin{array}{c} e'' \\ \cdot \\ H'' \end{array} = s''$$



CROSS

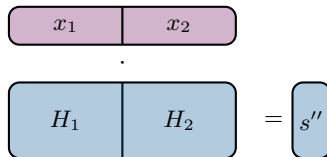
-

How to decode?

◦  $q = 127, z = 7 \rightarrow \mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$

unique sol.  $\frac{z^n}{q^{n-k}} < 1$

## 2. Meet-in-the-Middle





CROSS

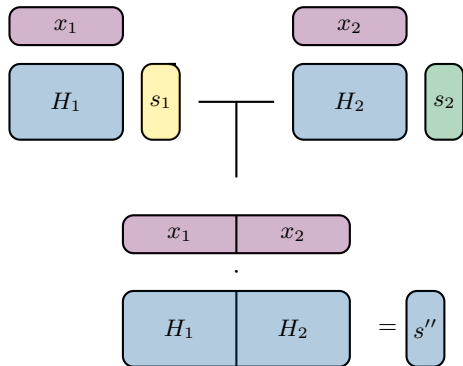
-

How to decode?

◦  $q = 127, z = 7 \rightarrow \mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$

unique sol.  $\frac{z^n}{q^{n-k}} < 1$

## 2. Meet-in-the-Middle



$\rightarrow$  Stern-like solver  
for R-SDP, R-SDP( $G$ )



CROSS

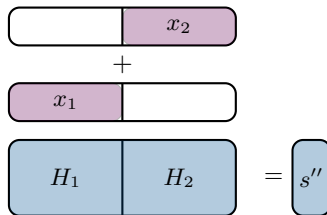
-

How to decode?

◦  $q = 127, z = 7 \rightarrow \mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$

unique sol.  $\frac{z^n}{q^{n-k}} < 1$

### 3. Representation technique





CROSS

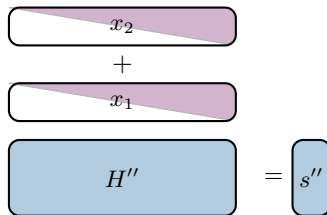
-

How to decode?

◦  $q = 127, z = 7 \rightarrow \mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$

unique sol.  $\frac{z^n}{q^{n-k}} < 1$

### 3. Representation technique





CROSS

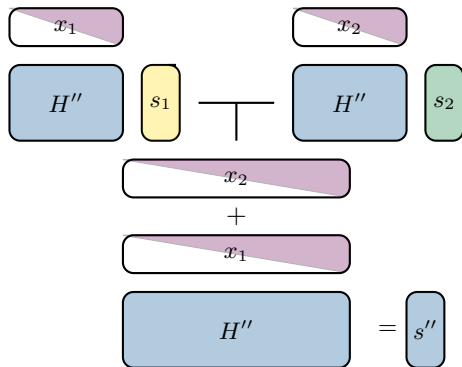
-

How to decode?

◦  $q = 127, z = 7 \rightarrow \mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$

unique sol.  $\frac{z^n}{q^{n-k}} < 1$

### 3. Representation technique



$\rightarrow$  BJMM-like solver  
for R-SDP



CROSS

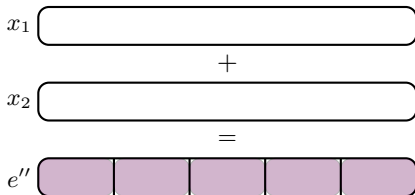
-

How to decode?

◦  $q = 127, z = 7 \rightarrow \mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$

unique sol.  $\frac{z^n}{q^{n-k}} < 1$

### 3. Representation technique



many repr.  $(x_1, x_2) : e'' = x_1 + x_2$   
small search space for  $(x_1, x_2)$



CROSS

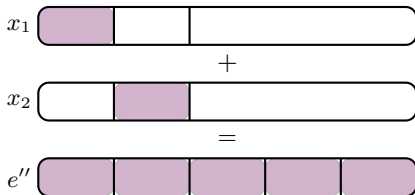
-

How to decode?

$$\circ q = 127, z = 7 \rightarrow \mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$$

unique sol.  $\frac{z^n}{q^{n-k}} < 1$

### 3. Representation technique



$$\text{purple box} + \text{white box} = \text{purple box}$$



many repr.  $(x_1, x_2) : e'' = x_1 + x_2$

small search space for  $(x_1, x_2)$



CROSS

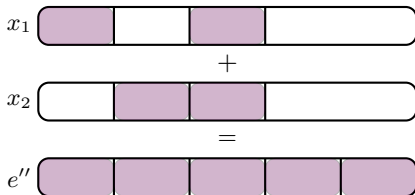
-

How to decode?

$$\circ q = 127, z = 7 \rightarrow \mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$$

unique sol.  $\frac{z^n}{q^{n-k}} < 1$

### 3. Representation technique



$$\text{purple square} + \text{white square} = \text{purple square}$$

$$\text{purple square} + \text{purple square} = \text{purple square}$$



many repr.  $(x_1, x_2) : e'' = x_1 + x_2$   
 small search space for  $(x_1, x_2)$



want  $(\mathbb{E} + \mathbb{E}) \cap \mathbb{E}$  small



CROSS

-

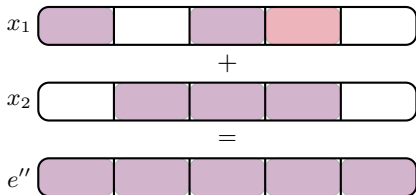
How to decode?

$$\circ q = 127, z = 7 \rightarrow \mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$$

$$\text{unique sol. } \frac{z^n}{q^{n-k}} < 1$$

### 3. Representation technique

$$\mathbb{D} = \mathbb{E} - \mathbb{E} = \{a - b \mid a, b \in \mathbb{E}\}$$



$$\text{purple} + \text{white} = \text{purple}$$

$$\text{purple} + \text{purple} = \text{purple}$$

$$\text{red} + \text{purple} = \text{purple}$$



many repr.  $(x_1, x_2) : e'' = x_1 + x_2$   
 small search space for  $(x_1, x_2)$



want  $(\mathbb{E} + \mathbb{E}) \cap \mathbb{E}$  small  
 want  $\mathbb{D}$  big



CROSS

-

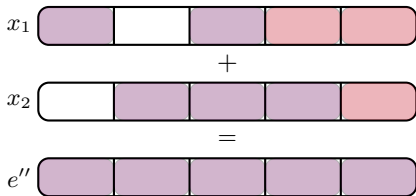
How to decode?

$$\circ q = 127, z = 7 \rightarrow \mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$$

$$\text{unique sol. } \frac{z^n}{q^{n-k}} < 1$$

### 3. Representation technique

$$\mathbb{D} = \mathbb{E} - \mathbb{E} = \{a - b \mid a, b \in \mathbb{E}\}$$



$$\text{purple} + \text{white} = \text{purple}$$

$$\text{purple} + \text{purple} = \text{purple}$$

$$\text{red} + \text{purple} = \text{purple}$$

$$\text{red} + \text{red} = \text{purple}$$



many repr.  $(x_1, x_2) : e'' = x_1 + x_2$   
 small search space for  $(x_1, x_2)$



want  $(\mathbb{E} + \mathbb{E}) \cap \mathbb{E}$  small  
 want  $\mathbb{D}$  big



CROSS

-

Is this now CROSS?



CROSS

-

Is this now CROSS?

**not**  $t$  parallel repetitions

$$\rightarrow (\beta[1], \dots, \beta[t]) = \text{Hash}(m, c_1[1], c_2[1], \dots, c_1[t], c_2[t])$$



CROSS

-

Is this now CROSS?

**not**  $t$  parallel repetitions

$$\rightarrow (\beta[1], \dots, \beta[t]) = \text{Hash}(m, c_1[1], c_2[1], \dots, c_1[t], c_2[t])$$

weighted challenges

$\varphi \leftarrow \text{Seed}$

$\varphi(e) \leftarrow \text{not from Seed}$



CROSS

-

Is this now CROSS?

**not**  $t$  parallel repetitions  $\rightarrow (\beta[1], \dots, \beta[t]) = \text{Hash}(m, c_1[1], c_2[1], \dots, c_1[t], c_2[t])$

weighted challenges

$\varphi \leftarrow \text{Seed} \rightarrow \text{small}$

$\varphi(e) \leftarrow \text{not from Seed}$



CROSS

-

Is this now CROSS?

**not**  $t$  parallel repetitions  $\rightarrow (\beta[1], \dots, \beta[t]) = \text{Hash}(m, c_1[1], c_2[1], \dots, c_1[t], c_2[t])$

weighted challenges

$r_1 = \varphi \leftarrow \text{Seed} \rightarrow \text{small}$

$r_2 = \varphi(e) \leftarrow \text{not from Seed}$



CROSS

-

Is this now CROSS?

not  $t$  parallel repetitions  $\rightarrow (\beta[1], \dots, \beta[t]) = \text{Hash}(m, c_1[1], c_2[1], \dots, c_1[t], c_2[t])$

weighted challenges

$r_1 = \varphi \leftarrow \text{Seed} \rightarrow \text{small} \rightarrow \text{unbalanced response size}$

$r_2 = \varphi(e) \leftarrow \text{not from Seed}$



CROSS

-

Is this now CROSS?

**not**  $t$  parallel repetitions  $\rightarrow (\beta[1], \dots, \beta[t]) = \text{Hash}(m, c_1[1], c_2[1], \dots, c_1[t], c_2[t])$

weighted challenges

$r_1 = \varphi \leftarrow \text{Seed} \rightarrow \text{small}$

$\rightarrow$  unbalanced response size

$r_2 = \varphi(e) \leftarrow \text{not from Seed}$

$\rightarrow$  weighted challenges



CROSS

-

Is this now CROSS?

not  $t$  parallel repetitions

$$\rightarrow (\beta[1], \dots, \beta[t]) = \text{Hash}(m, c_1[1], c_2[1], \dots, c_1[t], c_2[t])$$

weighted challenges

$$r_1 = \varphi \leftarrow \text{Seed}$$

$\rightarrow$  small

$\rightarrow$  unbalanced response size

$$r_2 = \varphi(e) \leftarrow \text{not from Seed}$$

$\rightarrow$  weighted challenges



want  $\beta[i] = 1$   $w$  times



CROSS

-

Is this now CROSS?

not  $t$  parallel repetitions

$$\rightarrow (\beta[1], \dots, \beta[t]) = \text{Hash}(m, c_1[1], c_2[1], \dots, c_1[t], c_2[t])$$

weighted challenges

$$r_1 = \varphi \leftarrow \text{Seed}$$

$\rightarrow$  small

$\rightarrow$  unbalanced response size

$$r_2 = \varphi(e) \leftarrow \text{not from Seed}$$

$\rightarrow$  weighted challenges



want  $\beta[i] = 1$   $w$  times

$\rightarrow$  adapt security analysis



CROSS

-

Is this now CROSS?

not  $t$  parallel repetitions

$$\rightarrow (\beta[1], \dots, \beta[t]) = \text{Hash}(m, c_1[1], c_2[1], \dots, c_1[t], c_2[t])$$

weighted challenges

$$r_1 = \varphi \leftarrow \text{Seed}$$

 $\rightarrow$  small $\rightarrow$  unbalanced response size

$$r_2 = \varphi(e) \leftarrow \text{not from Seed}$$

 $\rightarrow$  weighted challengeswant  $\beta[i] = 1$   $w$  times $\rightarrow$  adapt security analysis

$$O\left(\min_{t^* \in \{0, \dots, t\}} \left\{ \frac{1}{P_1(t, t^*, p)} + \frac{1}{P_2(t, t^*, w, p)} \right\}\right),$$

where

$$P_1(t, t^*, p) = \sum_{j=t^*}^t \binom{t}{j} \left(\frac{1}{p-1}\right)^j \left(1 - \frac{1}{p-1}\right)^{t-j},$$

$$P_2(t, t^*, w, p) = \max_{\alpha \in \{w, \dots, t\}} \sum_{j=t^*}^t \frac{\binom{t}{j} \left(\frac{1}{p-1}\right)^j \left(1 - \frac{1}{p-1}\right)^{t-j}}{P_1(t, t^*, p)} \sum_{w^* = \max\{0, \alpha - j\}}^{\min\{t-j, w\}} \frac{\binom{t-j}{w^*} \binom{j}{\alpha - w^*} \binom{j}{w - w^*}}{\binom{t}{\alpha} \binom{t}{w}}.$$



CROSS

-

Is this now CROSS?

not  $t$  parallel repetitions

$$\rightarrow (\beta[1], \dots, \beta[t]) = \text{Hash}(m, c_1[1], c_2[1], \dots, c_1[t], c_2[t])$$

weighted challenges

$$r_1 = \varphi \leftarrow \text{Seed}$$

 $\rightarrow$  small $\rightarrow$  unbalanced response size

$$r_2 = \varphi(e) \leftarrow \text{not from Seed}$$

 $\rightarrow$  weighted challengeswant  $\beta[i] = 1$   $w$  times $\rightarrow$  adapt security analysis

$$O\left(\min_{t^* \in \{0, \dots, t\}} \left\{ \frac{1}{P_1(t, t^*, p)} + \frac{1}{P_2(t, t^*, w, p)} \right\}\right),$$

$$w > t/2 \quad \rightarrow \text{small}$$

$$w \sim t/2 \quad \rightarrow \text{fast}$$

where

$$P_1(t, t^*, p) = \sum_{j=t^*}^t \binom{t}{j} \left(\frac{1}{p-1}\right)^j \left(1 - \frac{1}{p-1}\right)^{t-j},$$

$$P_2(t, t^*, w, p) = \max_{\alpha \in \{w, \dots, t\}} \sum_{j=t^*}^t \frac{\binom{t}{j} \left(\frac{1}{p-1}\right)^j \left(1 - \frac{1}{p-1}\right)^{t-j}}{P_1(t, t^*, p)} \sum_{w^* = \max\{0, \alpha - j\}}^{\min\{t-j, w\}} \frac{\binom{t-j}{w^*} \binom{j}{\alpha - w^*} \binom{j}{w - w^*}}{\binom{t}{\alpha} \binom{t}{w}}.$$



CROSS

-

Is this now CROSS?

**not**  $t$  parallel repetitions  $\rightarrow (\beta[1], \dots, \beta[t]) = \text{Hash}(m, c_1[1], c_2[1], \dots, c_1[t], c_2[t])$

weighted challenges:  $\beta[i] = 1$   $w$  many times



CROSS

-

Is this now CROSS?

**not**  $t$  parallel repetitions  $\rightarrow (\beta[1], \dots, \beta[t]) = \text{Hash}(m, c_1[1], c_2[1], \dots, c_1[t], c_2[t])$

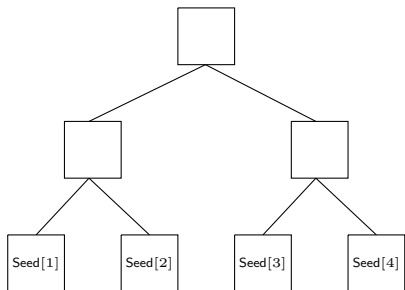
weighted challenges:  $\beta[i] = 1$   $w$  many times

Seed path

round  $i$ :  $\varphi[i] \leftarrow \text{Seed}[i]$

to send all  $w$  requested  $\varphi[i]$

$\rightarrow$  can send Path





CROSS

-

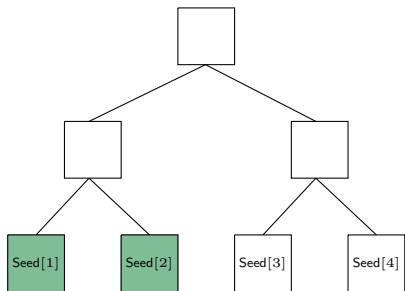
Is this now CROSS?

**not**  $t$  parallel repetitions  $\rightarrow (\beta[1], \dots, \beta[t]) = \text{Hash}(m, c_1[1], c_2[1], \dots, c_1[t], c_2[t])$

weighted challenges:  $\beta[i] = 1$   $w$  many times

Seed path

round  $i$ :  $\varphi[i] \leftarrow \text{Seed}[i]$   
to send all  $w$  requested  $\varphi[i]$   
 $\rightarrow$  can send Path





CROSS

-

Is this now CROSS?

**not**  $t$  parallel repetitions  $\rightarrow (\beta[1], \dots, \beta[t]) = \text{Hash}(m, c_1[1], c_2[1], \dots, c_1[t], c_2[t])$

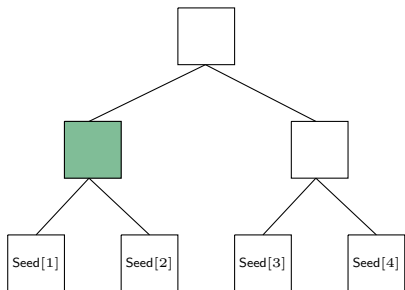
weighted challenges:  $\beta[i] = 1$   $w$  many times

Seed path

round  $i$ :  $\varphi[i] \leftarrow \text{Seed}[i]$

to send all  $w$  requested  $\varphi[i]$

$\rightarrow$  can send Path





CROSS

-

Is this now CROSS?

**not**  $t$  parallel repetitions  $\rightarrow (\beta[1], \dots, \beta[t]) = \text{Hash}(m, c_1[1], c_2[1], \dots, c_1[t], c_2[t])$

weighted challenges:  $\beta[i] = 1$   $w$  many times

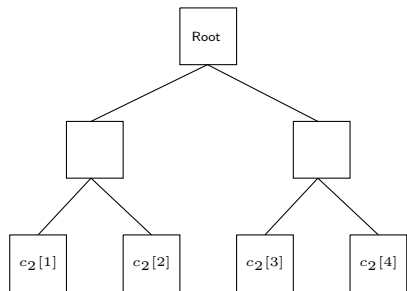
for round  $i$ : if challenged with  $\beta$ , put response  $r_\beta[i]$  and  $c_{\beta'}[i]$   $\rightarrow$  recover both  $c_\beta[i], c_{\beta'}[i]$

### Merkle proof

include  $w$  many  $c_2[i]$  in response

use  $\text{Hash}(\text{Root})$  instead of  $\text{Hash}(c_2[1], \dots, c_2[t])$

$\rightarrow$  send Proof to recompute Root





CROSS

-

Is this now CROSS?

**not**  $t$  parallel repetitions  $\rightarrow (\beta[1], \dots, \beta[t]) = \text{Hash}(m, c_1[1], c_2[1], \dots, c_1[t], c_2[t])$

weighted challenges:  $\beta[i] = 1$   $w$  many times

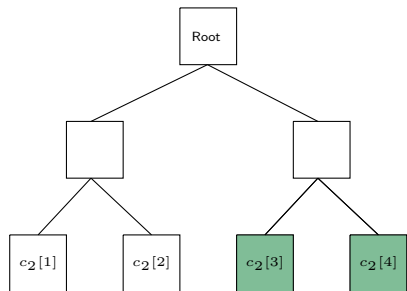
for round  $i$ : if challenged with  $\beta$ , put response  $r_\beta[i]$  and  $c_{\beta'}[i]$   $\rightarrow$  recover both  $c_\beta[i], c_{\beta'}[i]$

## Merkle proof

include  $w$  many  $c_2[i]$  in response

use  $\text{Hash}(\text{Root})$  instead of  $\text{Hash}(c_2[1], \dots, c_2[t])$

$\rightarrow$  send Proof to recompute Root





CROSS

-

Is this now CROSS?

**not**  $t$  parallel repetitions  $\rightarrow (\beta[1], \dots, \beta[t]) = \text{Hash}(m, c_1[1], c_2[1], \dots, c_1[t], c_2[t])$

weighted challenges:  $\beta[i] = 1$   $w$  many times

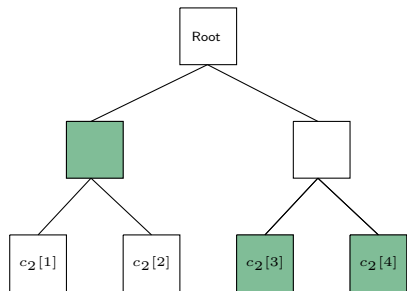
for round  $i$ : if challenged with  $\beta$ , put response  $r_\beta[i]$  and  $c_{\beta'}[i]$   $\rightarrow$  recover both  $c_\beta[i], c_{\beta'}[i]$

## Merkle proof

include  $w$  many  $c_2[i]$  in response

use  $\text{Hash}(\text{Root})$  instead of  $\text{Hash}(c_2[1], \dots, c_2[t])$

$\rightarrow$  send Proof to recompute Root





CROSS

-

How good is CROSS?

NIST cat. I

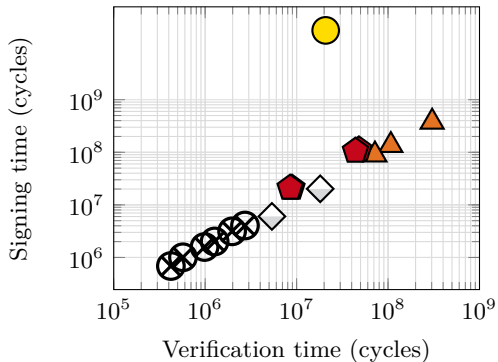
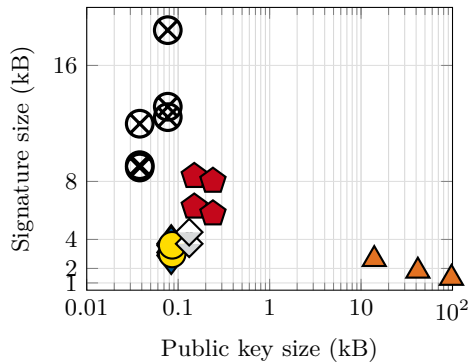
| Problem      | $q, z$     | Type     | $(n, k, m)$  | $(t, w)$   | Sign.<br>(kB) | Sign<br>(MCycles) | Verify<br>(MCycles) |
|--------------|------------|----------|--------------|------------|---------------|-------------------|---------------------|
| R-SDP        | (127, 7)   | fast     | (127, 76, -) | (157, 82)  | 18.4          | 1.01              | 0.57                |
|              |            | balanced |              | (256, 215) | 13.1          | 2.01              | 1.27                |
|              |            | short    |              | (520, 488) | 12.4          | 4.05              | 2.73                |
| R-SDP( $G$ ) | (509, 127) | fast     | (55, 36, 25) | (147, 76)  | 11.9          | 0.68              | 0.42                |
|              |            | balanced |              | (256, 220) | 9.1           | 1.58              | 0.99                |
|              |            | short    |              | (512, 484) | 8.9           | 3.14              | 1.97                |

private and public keys &lt; 0.1 kB

key gen. &lt; 0.1 MCycle

1

vs. other code-based?





CROSS

-

Questions?

What's next?

- Hardware implementation
- VOLE  $\rightarrow$  smaller signatures



Slides



### What's next?

- Hardware implementation
- VOLE  $\rightarrow$  smaller signatures



Slides

### Announcement



Talk: Sebastian Bitzer

Title: VOLEitH-based Signatures  
from Restricted Decoding Problems

Time: 11:50



### What's next?

- Hardware implementation
- VOLE → smaller signatures



Slides

### NIST timeline

- 2022 Announcement on ramp call
- 2023 1st round candidates
- 2024 2nd round candidates
- 2026 3rd round candidates (finalists?)
- 2035 RSA, ECDSA disallowed



CROSS

-

Questions?

What's next?

- Hardware implementation
- VOLE → smaller signatures



Slides



Website



CROSS

Codes & Restricted Objects Signature Scheme

<https://cross-crypto.com>

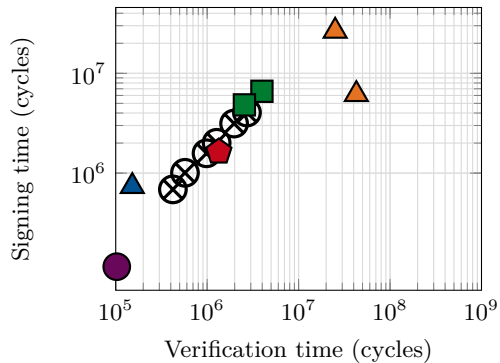
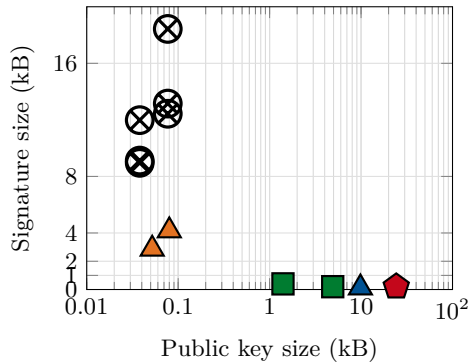
Thank you!

TO BE  
CONTINUED

## CROSS

**1**

vs. Multivariate?

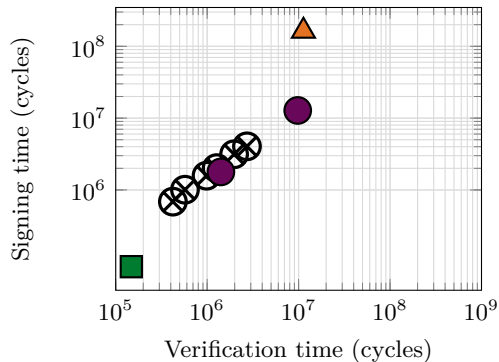
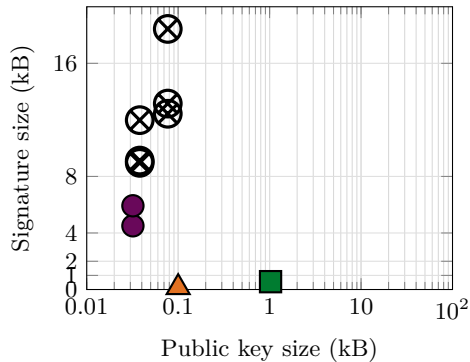




CROSS

-

vs. rest?



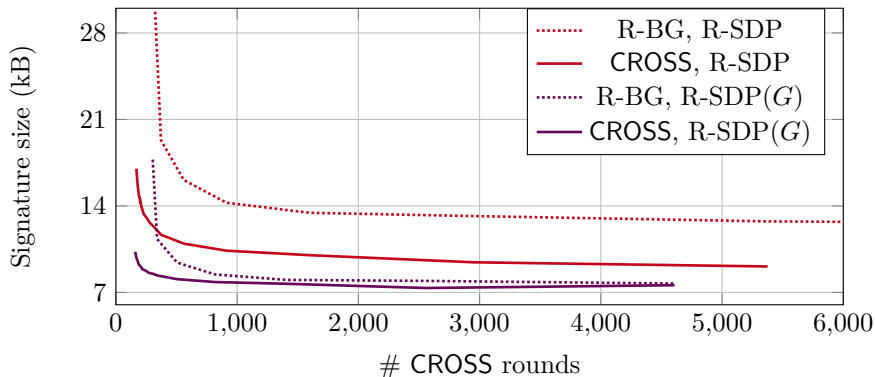


CROSS

Why such a simple ZK  
protocol?

Baldi et al., 2023: R-BG protocol, soundness error  $\varepsilon \approx \max \left\{ \frac{1}{N}; \frac{1}{q-1} \right\}$

Computational cost: one round of R-BG is  $\approx N$  rounds of CROSS





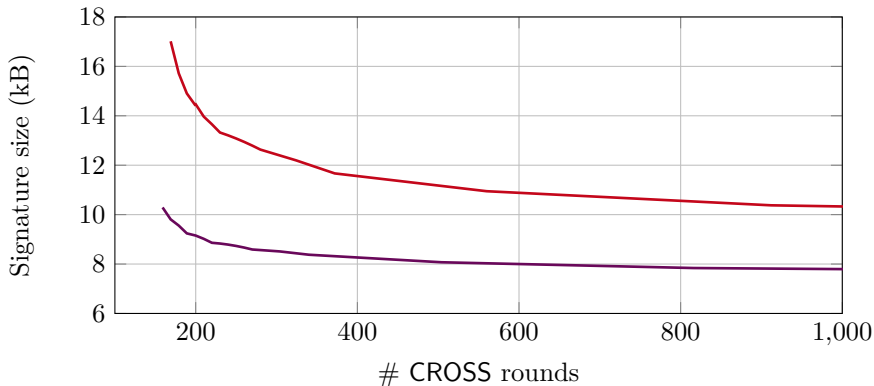
CROSS

-

Why such a simple ZK  
protocol?

Baldi et al., 2023: R-BG protocol, soundness error  $\varepsilon \approx \max \left\{ \frac{1}{N}; \frac{1}{q-1} \right\}$

Computational cost: one round of R-BG is  $\approx N$  rounds of CROSS



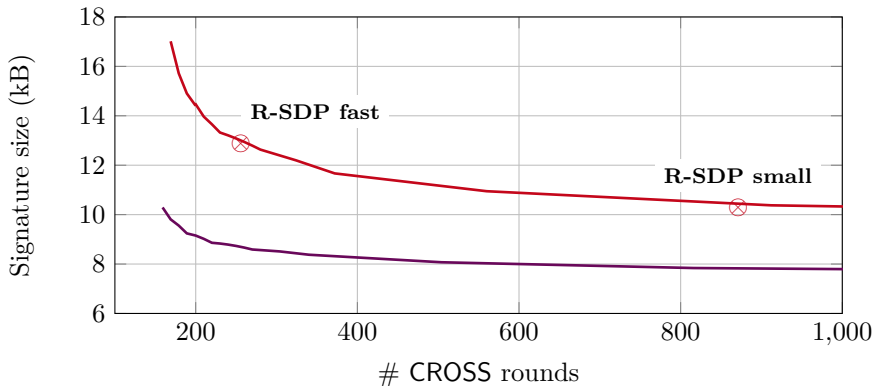


CROSS

Why such a simple ZK  
protocol?

Baldi et al., 2023: R-BG protocol, soundness error  $\varepsilon \approx \max \left\{ \frac{1}{N}; \frac{1}{q-1} \right\}$

Computational cost: one round of R-BG is  $\approx N$  rounds of CROSS



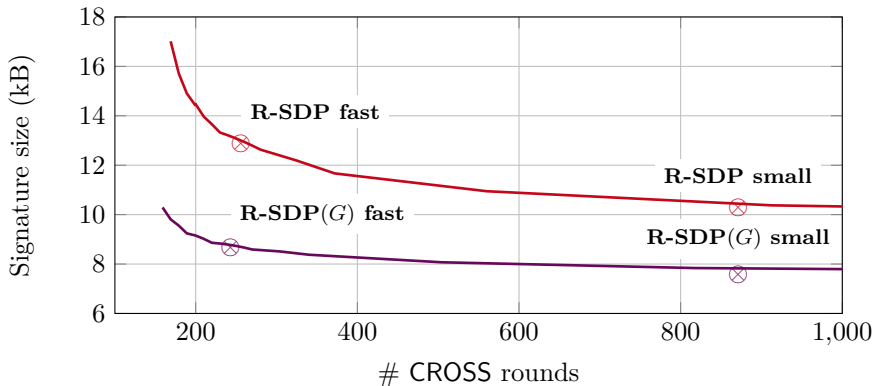


CROSS

Why such a simple ZK  
protocol?

Baldi et al., 2023: R-BG protocol, soundness error  $\varepsilon \approx \max \left\{ \frac{1}{N}; \frac{1}{q-1} \right\}$

Computational cost: one round of R-BG is  $\approx N$  rounds of CROSS



# VOLE

## Vector Oblivious Linear Transfer

Prover

Ⓢ secret  $s$

$v$  random

$$f(x) = sx + v$$

## ZK Protocol

Verifier

Ⓢ public

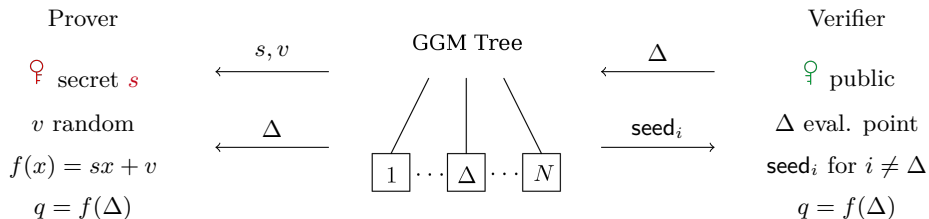
$\Delta$  eval. point

$$q = f(\Delta)$$

# VOLE

## Vector Oblivious Linear Transfer

### ZK Protocol



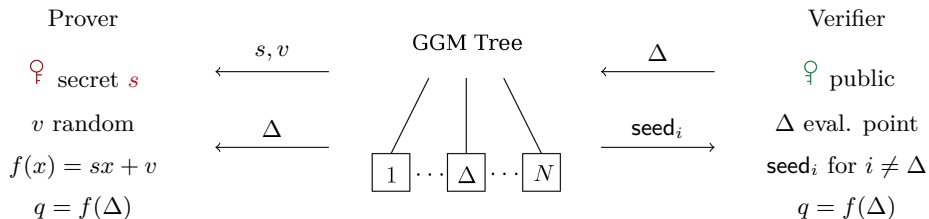
VOLE correlation  $q = s\Delta + v = f(\Delta)$

dishonest prover needs to guess  $\Delta$  before committing to GGM tree:  $\mathbb{P} \sim 1/p$

# VOLE

## Vector Oblivious Linear Transfer

### ZK Protocol



#### MPC

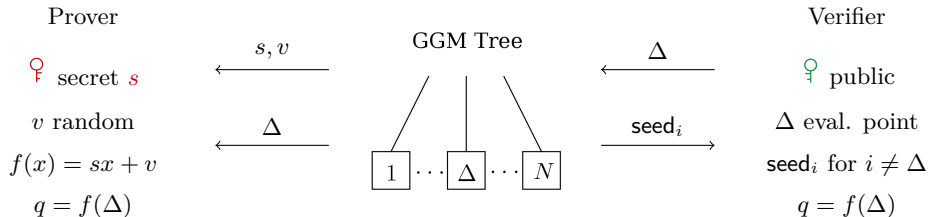
$$s = \sum s_i \quad \text{MPC} \xleftarrow{\ell} N - 1 \text{ views}$$

#### VOLE

$$s = \sum s_i \quad \text{GGM} \xleftarrow{\Delta} N - 1 \text{ seeds}$$
$$v = \sum i s_i \quad q = \sum s_i (\Delta - i) = s\Delta + v$$

## Vector Oblivious Linear Transfer

### ZK Protocol



$$f(x) = \sum_{i=0}^d f_i x^i,$$

$$s = f_d$$

$$f_1(x), f_2(x)$$

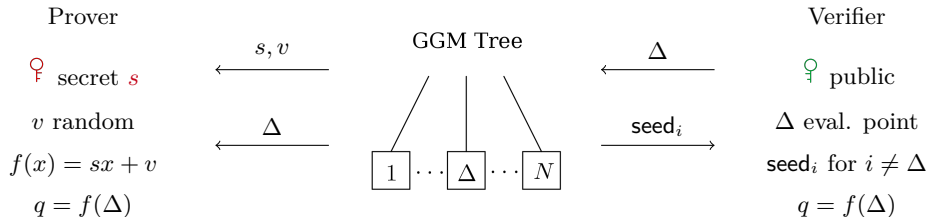
$$f_1(\Delta) + f_2(\Delta) = (f_1 + f_2)(\Delta)$$

$$f_1(\Delta) f_2(\Delta) = (f_1 f_2)(\Delta)$$

# VOLE

## Vector Oblivious Linear Transfer

### ZK Protocol



Disadvantages: slow

Advantages: small sizes