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>;

~~Cc: 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

$\mathbf{e}, \mathbf{e}_G, \mathbf{H}, \overline{\mathbf{M}} \leftarrow \mathsf{ExpandSK}(\mathsf{Seed}_{\mathsf{sk}})$: $\mathbf{e}, \mathbf{H} \leftarrow \mathsf{ExpandSK}(\mathsf{Seed}_{\mathsf{sk}})$

// Computing the commitments

$\mathsf{Seed} \xleftarrow{\$} \{0,1\}^{\lambda}, \quad \mathsf{Salt} \xleftarrow{\$} \{0,1\}^{2\lambda}$

$(\mathsf{Seed}[1], \ldots, \mathsf{Seed}[t]) \leftarrow \mathsf{SeedLeaves}(\mathsf{Seed}, \mathsf{Salt})$

// Compute $\mathbf{v}[i]$ such that $\mathbf{v}[i] \star \mathbf{e}'[i] = \mathbf{e}$

for i **from** 1 **to** t **do**

$\mathbf{e}'_G[i], \mathbf{u}'[i] \leftarrow \mathsf{CSPRNG}_{\mathbb{F}_2^m \times \mathbb{F}_p^n}(\mathsf{Seed}[i] \mid \mathsf{Salt} \mid i+c)$: $\mathbf{e}'[i], \mathbf{u}'[i] \leftarrow \mathsf{CSPRNG}_{\mathbb{F}_2^n \times \mathbb{F}_p^n}(\mathsf{Seed}[i] \mid \mathsf{Salt} \mid i+c)$

$\mathbf{v}_G[i] \leftarrow \mathbf{e}_G - \mathbf{e}'_G[i]$

$\mathbf{e}'[i] \leftarrow \mathbf{e}'_G[i]\overline{\mathbf{M}}$

$\mathbf{v}[i] \leftarrow \mathbf{e} - \mathbf{e}'[i]$

  for j **from** 1 **to** n **do**

    $\mathbf{v}[i]_j \leftarrow g^{\mathbf{v}[i]_j}$

  $\mathbf{u}[i] \leftarrow \mathbf{v}[i] \star \mathbf{u}'[i]$

  $\mathbf{s}'[i] \leftarrow \mathbf{u}[i]\mathbf{H}^{\top}$

$\mathsf{cmt}_0[i] \leftarrow \mathsf{Hash}(\mathbf{s}'[i] \mid \mathbf{v}_G[i] \mid \mathsf{Salt} \mid i+c)$: $\mathsf{cmt}_0[i] \leftarrow \mathsf{Hash}(\mathbf{s}'[i] \mid \mathbf{v}[i] \mid \mathsf{Salt} \mid i+c)$

  $\mathsf{cmt}_1[i] \leftarrow \mathsf{Hash}(\mathsf{Seed}[i] \mid \mathsf{Salt} \mid i+c)$

$\mathsf{digest}_{\mathsf{cmt}_0} \leftarrow \mathsf{TreeRoot}(\mathsf{cmt}_0[1], \cdots, \mathsf{cmt}_0[t])$

$\mathsf{digest}_{\mathsf{cmt}_1} \leftarrow \mathsf{Hash}(\mathsf{cmt}_1[1] \mid \cdots \mid \mathsf{cmt}_1[t])$

$\mathsf{digest}_{\mathsf{cmt}} \leftarrow \mathsf{Hash}(\mathsf{digest}_{\mathsf{cmt}_0} \mid \mathsf{digest}_{\mathsf{cmt}_1})$

// Computing first challenge

$\mathsf{digest}_{\mathsf{Msg}} \leftarrow \mathsf{Hash}(\mathsf{Msg})$

$\mathsf{digest}_{\mathsf{chall}_1} \leftarrow \mathsf{Hash}(\mathsf{digest}_{\mathsf{Msg}} \mid \mathsf{digest}_{\mathsf{cmt}} \mid \mathsf{Salt})$

$\mathsf{chall}_1 \leftarrow \mathsf{CSPRNG}_{(\mathbb{F}_p^*)^t}(\mathsf{digest}_{\mathsf{chall}_1} \mid t+c)$

// Computing first response

for i **from** 1 **to** t **do**

  for j **from** 1 **to** n **do**

    $\mathbf{e}'[i]_j \leftarrow g^{\mathbf{e}'[i]_j}$

  $\mathbf{y}[i] \leftarrow \mathbf{u}'[i] + \mathsf{chall}_1[i]\mathbf{e}'[i]$

// Computing second challenge

$\mathsf{digest}_{\mathsf{chall}_2} \leftarrow \mathsf{Hash}(\mathbf{y}[1] \mid \cdots \mid \mathbf{y}[t] \mid \mathsf{digest}_{\mathsf{chall}_1})$

$\mathsf{chall}_2 \leftarrow \mathsf{CSPRNG}_{\mathcal{B}_{(t,w)}}(\mathsf{digest}_{\mathsf{chall}_2} \mid t+c+1)$

// Computing second response

$\mathsf{Proof} \leftarrow \mathsf{TreeProof}(\mathsf{cmt}_0[1], \cdots, \mathsf{cmt}_0[t], \mathsf{chall}_2)$

$\mathsf{Path} \leftarrow \mathsf{SeedPath}(\mathsf{Seed}, \mathsf{Salt}, \mathsf{chall}_2)$

for i **from** 1 **to** t **do**

  if $\mathsf{chall}_2[i] = 0$ **then**

    $\mathsf{resp}[i]_0 \leftarrow (\mathbf{y}[i], \mathbf{v}_G[i])$: $\mathsf{resp}[i]_0 \leftarrow (\mathbf{y}[i], \mathbf{v}[i])$

    $\mathsf{resp}[i]_1 \leftarrow \mathsf{cmt}_1[i]$

// Assembling signature

$\mathsf{Sgn} \leftarrow (\mathsf{Salt}, \mathsf{digest}_{\mathsf{cmt}}, \mathsf{digest}_{\mathsf{chall}_2}, \mathsf{Path}, \mathsf{Proof}, \mathsf{resp})$

// Expanding secret key

1 $\vec{\mathbf{e}}, \vec{\mathbf{e}}_C, \mathbf{H}, \vec{\mathbf{M}} \leftarrow \mathsf{ExpandSK}(\mathsf{Seed}_{\mathsf{sk}})$ | $\vec{\mathbf{e}}, \mathbf{H} \leftarrow \mathsf{ExpandSK}(\mathsf{Seed}_{\mathsf{sk}})$

// Computing the commitments

2 $\mathsf{Seed} \xleftarrow{\$} \{0,1\}^{\lambda}, \quad \mathsf{Salt} \xleftarrow{\$} \{0,1\}^{2\lambda}$

3 $(\mathsf{Seed}[1], \ldots, \mathsf{Seed}[t]) \leftarrow \mathsf{SeedLeaves}(\mathsf{Seed}, \mathsf{Salt})$

// Compute $\mathbf{v}[i]$ such that $\mathbf{v}[i] * \mathbf{e}'[i] = \mathbf{e}$

4 **for** i **from** 1 **to** t **do**

$\vec{\mathbf{e}}'_C[i], \mathbf{u}'[i] \leftarrow \mathsf{CSPRNG}_{\mathbb{F}_p^n \times \mathbb{F}_p^*}(\mathsf{Seed}[i] \,|\, \mathsf{Salt} \,|\, i + c)$ | $\vec{\mathbf{e}}'[i], \mathbf{u}'[i] \leftarrow \mathsf{CSPRNG}_{\mathbb{F}_p^n \times \mathbb{F}_p^*}(\mathsf{Seed}[i] \,|\, \mathsf{Salt} \,|\, i + c)$

5

$\mathbf{v}_C[i] \leftarrow \mathbf{e}_C - \mathbf{e}'_C[i]$

$\mathbf{e}'[i] \leftarrow \vec{\mathbf{e}}'_C[i]\mathbf{M}$

$\mathbf{v}[i] \leftarrow$

$\mathsf{cmt}_1 \leftarrow \mathsf{Hash}(\mathsf{cmt}_1[1]$

$\mathsf{digest}_{\mathsf{cmt}} \leftarrow \mathsf{Hash}(\mathsf{digest}_{\mathsf{cmt}_0}$ | $\mathsf{cmt}_0[i] \leftarrow \mathsf{Hash}(\mathbf{s}'[i] \,|\, \mathbf{v}[i] \,|\, \mathsf{Salt} \,|\, i + c)$

// Computing first challenge

$\mathsf{digest}_{\mathsf{Msg}} \leftarrow \mathsf{Hash}(\mathsf{Msg})$

$\mathsf{digest}_{\mathsf{chall}_1} \leftarrow \mathsf{Hash}(\mathsf{digest}_{\mathsf{Msg}} \mid \mathsf{d}$

$\mathsf{chall}_1 \leftarrow \mathsf{CSPRNG}_{(\mathbb{F}_p^*)^t}(\mathsf{digest}_{\mathsf{ch}}$

// Computing first response

for i **from** 1 **to** t **do**

for j **from** 1 **to** n **do**

$\mathbf{e}'[i]_j \leftarrow g^{\vec{\mathbf{e}}'[i]_j}$

$[i] \leftarrow \mathbf{u}'[i] + \mathsf{chall}_1[i]\mathbf{e}'[i]$

ting second cha

27 $\mathsf{resp}[i]_0 \leftarrow (\mathbf{y}[i], \mathbf{v}[i])$

$\mathsf{resp}[i]_1 \leftarrow \mathsf{cmt}_1[i]$

// Assembling signature

28 $\mathsf{Sgn} \leftarrow (\mathsf{Salt}, \mathsf{digest}_{\mathsf{cmt}}, \mathsf{digest}_{\mathsf{chall}_2}, \mathsf{Path}, \mathsf{Proof}, \mathsf{resp})$

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 — fast < 1 MCycle (NIST cat. I)
- memory-optimized — fits on Cortex-M4 microcontroller
- constant worst-case runtime — no signature rejection
- available on liboqs — and SUPERCOP
- EUF-CMA security — 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 Barenghi


Michele Battagliola


Edoardo Signorini


Gerardo Pelosi


Paolo Santini


Marco Baldi

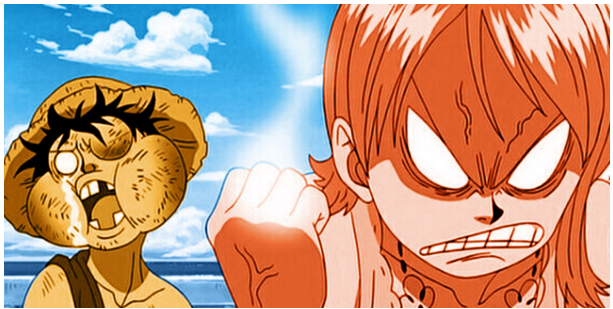
CROSS - Why “CROSS”?

Codes and Restricted Objects Signature Scheme

CROSS - ~~Why "CROSS"?~~ Why $\otimes$?

CROSS - ~~Why "CROSS"?~~ Why $\otimes$?

CROSS - ~~Why "CROSS"?~~ Why $\otimes$?



CROSS - Zero-Knowledge Protocol

CROSS - Zero-Knowledge Protocol

Prover

secret key

secret space

Verifier

public key

Prover

secret key

secret space

commitment c

Verifier

public key

Prover

secret key

secret space

commitment c

Verifier

public key

challenge z

CROSS - Zero-Knowledge Protocol

Prover

secret key

secret space

Verifier

public key

commitment c

challenge z

response r

CROSS - Zero-Knowledge Protocol

Prover

secret key

secret space

commitment c

response r

Verifier

public key

challenge z

verify , z, $r \rightarrow c$

Prover

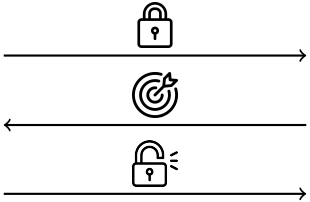
?

secret space

Verifier

public key

commitment c



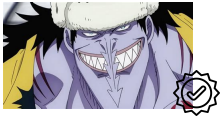
challenge z

response r

verify , z, $r \rightarrow c$

CROSS - Zero-Knowledge Protocol

Prover



small cheating probability

Verifier

public key

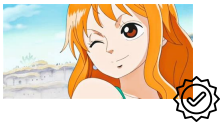
commitment c

challenge z

response r

verify , z, $r \rightarrow c$

Prover



Verifier

sec. level λ

cheat. prob. α

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

secret key

public key

commitment c

challenge z

response r

verify , z, $r \rightarrow c$

t rounds

CROSS - Zero-Knowledge Protocol

Signer

Fiat-Shamir transform

Verifier

secret key

m message

public key

commitment c

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

response r

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

verify , z, $r \rightarrow c$

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

Prover

♀

♡

SDP: Given H, s, t find e

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

Verifier

♀

Prover

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

e

SDP: Given H, s, t find e

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

Verifier

H, s, t

Prover

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

φ e

$u \in \mathbb{F}_q^n, \varphi$

SDP: Given H, s, t find e

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

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

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

Verifier

H, s, t

Prover

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

φ e

$u \in \mathbb{F}_q^n, \varphi$

SDP: Given H, s, t find e

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

Verifier

H, s, t

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

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

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

Prover

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

φ e

$u \in \mathbb{F}_q^n, \varphi$

SDP: Given H, s, t find e

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

Verifier

H, s, t

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

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

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

$y = \varphi(u + ze)$

Prover

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

φ e

$u \in \mathbb{F}_q^n, \varphi$

SDP: Given H, s, t find e

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

Verifier

H, s, t

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

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

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

$y = \varphi(u + ze)$

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

Prover

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

φ e

$u \in \mathbb{F}_q^n, \varphi$

SDP: Given H, s, t find e

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

Verifier

H, s, t

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

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

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

$y = \varphi(u + ze)$

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

$r_1 = \varphi$

$r_2 = \varphi(e)$

Prover

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

φ $\quad$ e

$u \in \mathbb{F}_q^n, \varphi$

SDP: Given H, s, t find e

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

Verifier

H, s, t

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

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

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

$y = \varphi(u + ze)$

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

$r_1 = \varphi$

$r_2 = \varphi(e)$

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

Prover

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

φ e

$u \in \mathbb{F}_q^n, \varphi$

SDP: Given H, s, t find e

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

Verifier

H, s, t

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

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

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

$y = \varphi(u + ze)$

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

$r_1 = \varphi$

$r_2 = \varphi(e)$

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

Prover

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

φ $\quad$ e

$u \in \mathbb{F}_q^n, \varphi$

SDP: Given H, s, t find e

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

Verifier

H, s, t

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

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

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

$y = \varphi(u + ze)$

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

$r_1 = \varphi$

$r_2 = \varphi(e)$

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

Prover

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

φ $\quad$ e

$u \in \mathbb{F}_q^n, \varphi$

SDP: Given H, s, t find e

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

Verifier

H, s, t

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

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

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

$y = \varphi(u + ze)$

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

$r_1 = \varphi$

$r_2 = \varphi(e)$

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

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

Prover

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

φ e

$u \in \mathbb{F}_q^n, \varphi$

SDP: Given H, s, t find e

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

Verifier

H, s, t

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

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

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

$y = \varphi(u + ze)$

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

$r_1 = \varphi$

$r_2 = \varphi(e)$

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

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

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

Prover

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

φ $\quad$ e

$u \in \mathbb{F}_q^n, \varphi$

SDP: Given H, s, t find e

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

Verifier

H, s, t

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

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

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

$y = \varphi(u + ze)$

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

$r_1 = \varphi$

$r_2 = \varphi(e)$

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

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

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

Prover

Verifier

SDP: Given H, s, t find e

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

probability $\sim 1/2 \to t \sim \lambda$

H, s, t

$u \in \mathbb{F}_q^n, \varphi$

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

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

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

$y = \varphi(u + ze)$

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

$r_1 = \varphi$

$r_2 = \varphi(e)$

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

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

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

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 ~ 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 ~ 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 → many rounds → large signatures ∼ 43 kB for $\lambda = 128$

solution

change protocol: use MPC to split error → smaller cheating probability

→ used in

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

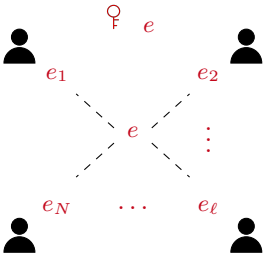
Prover

SDP: Given H, s, t find e

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

Verifier

H, s, t



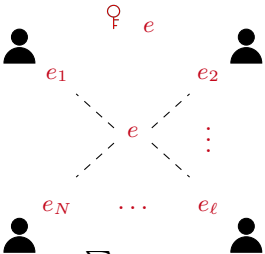
Prover

SDP: Given H, s, t find e

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

Verifier

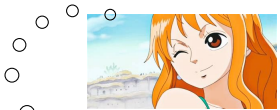
H, s, t

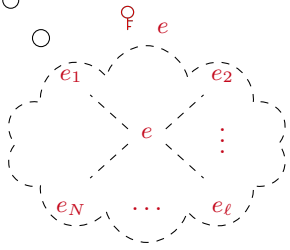


$$\sum e_i = e$$

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

Prover





$$\sum e_i = e$$

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

SDP: Given H, s, t find e

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

Verifier

H, s, t

Prover

Verifier

SDP: Given H, s, t find e

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

♀ e

e_1 e_2 e e_N $\cdots$ e_ℓ

$\sum e_i = e$

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

$c_i = (s_i, \alpha_i)$ →

♀ H, s, t

Prover

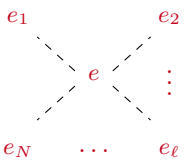
Verifier

SDP: Given H, s, t find e

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

P e

V H, s, t



$c_i = (s_i, \alpha_i)$

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

$$\sum e_i = e$$

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

Prover

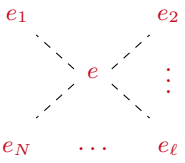
Verifier

SDP: Given H, s, t find e

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

H, s, t

e



$c_i = (s_i, \alpha_i)$

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

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

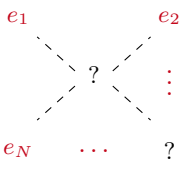
check $(c_i)_{i \neq z}$

$$\sum e_i = e$$

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

Prover

???



$$\sum e_i = e$$

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

SDP: Given H, s, t find e

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

$c_i = (s_i, \alpha_i)$

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

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

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

Verifier

H, s, t

Prover

Verifier

SDP: Given H, s, t find e

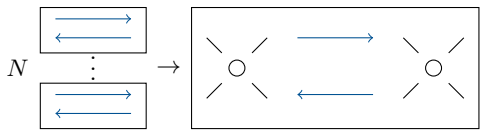
1. $s = eH^\top$ 2. $\mathrm{wt}(e) = t$

cheating probability $\sim 1/N$

e_1 e_2 ? e_N $\cdots$?

$c_i = (s_i, \alpha_i)$

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



H, s, t

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

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

☺ smaller cheating prob → smaller signatures

Prover

Verifier

SDP: Given H, s, t find e

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

cheating probability $\sim 1/N$

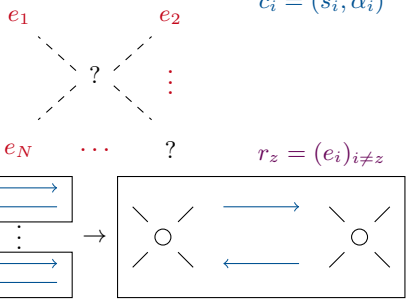
H, s, t

$c_i = (s_i, \alpha_i)$

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

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

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



☺ smaller cheating prob → smaller signatures

☹ more computations → 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 ~ 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?

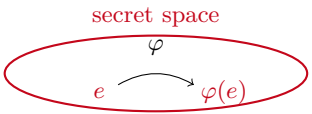
well-studied ☺

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



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.

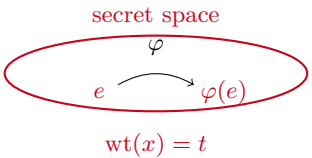
P.-L. Cayrel, P. Véron, S. El Yousfi Alaoui. "A zero-knowledge identification scheme based on the q-ary syndrome decoding problem", Int. Workshop on Selected Areas in Cryptography, 2011.

too large ☹

large communication cost in 1 round

solution

change problem: use restricted errors

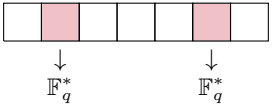


errors $|e| = t(\log(n) + \log(q-1))$

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

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

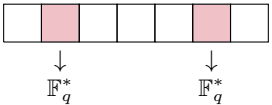
e



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

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

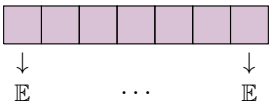
e



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

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

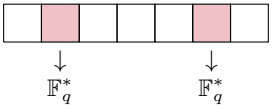
e



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

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

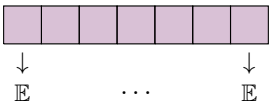
e



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

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

e



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

$\mathbb{E} < \mathbb{F}_q^*$ $\quad \mathbb{E} = \{g^i \mid i \in \{1, \ldots, z\}\}$ $\quad \mathrm{ord}(g) = z$ prime

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} | $\cdots$ | g^{i_n} |

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} | $\cdots$ | g^{i_n} |

↓

$\ell(e)$ | i_1 | $\cdots$ | i_n |

$\mathbb{F}_z^n$

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} | $\cdots$ | g^{i_n} |

↓

$\ell(e)$ | i_1 | $\cdots$ | i_n |

$\mathbb{F}_z^n$

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

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} | $\cdots$ | g^{i_n} $\xrightarrow{\varphi}$ e' | g^{j_1} | $\cdots$ | g^{j_n}

$\downarrow$

$\ell(e)$ | i_1 | $\cdots$ | i_n

$\mathbb{F}_z^n$

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

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} | $\cdots$ | g^{i_n} $\star$ | $g^{j_1 - i_1}$ | $\cdots$ | $g^{j_n - i_n}$ e' | g^{j_1} | $\cdots$ | g^{j_n}

$\downarrow$

$\ell(e)$ | i_1 | $\cdots$ | i_n

$\mathbb{F}_z^n$

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

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} | $\cdots$ | g^{i_n} | $\star$ | $g^{j_1-i_1}$ | $\cdots$ | $g^{j_n-i_n}$ | e' | g^{j_1} | $\cdots$ | g^{j_n}

$\downarrow$

$\ell(e)$ | i_1 | $\cdots$ | i_n | $j_1 - i_1$ | $\cdots$ | $j_n - i_n$ | $\ell(e')$ | j_1 | $\cdots$ | j_n

$\mathbb{F}_z^n$

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

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} | $\cdots$ | g^{i_n} | $\star$ | $g^{j_1-i_1}$ | $\cdots$ | $g^{j_n-i_n}$ | e' | g^{j_1} | $\cdots$ | g^{j_n}

$\downarrow$

$\ell(e)$ | i_1 | $\cdots$ | i_n | $+$ | $j_1 - i_1$ | $\cdots$ | $j_n - i_n$ | $\ell(e')$ | j_1 | $\cdots$ | j_n

$\mathbb{F}_z^n$

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

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} | $\cdots$ | g^{i_n} | $\star$ | $g^{j_1-i_1}$ | $\cdots$ | $g^{j_n-i_n}$ | e' | g^{j_1} | $\cdots$ | g^{j_n}

$\downarrow$

$\ell(e)$ | i_1 | $\cdots$ | i_n | $+$ | $j_1 - i_1$ | $\cdots$ | $j_n - i_n$ | $\ell(e')$ | j_1 | $\cdots$ | j_n

$\mathbb{F}_z^n$

$\rightarrow |e| = n\log(z)$ $\rightarrow |\varphi| = n\log(z)$ $\rightarrow (\mathbb{E}^n, \star) \cong (\mathbb{F}_z^n, +)$

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} | $\cdots$ | g^{i_n} | $\star$ | $g^{j_1-i_1}$ | $\cdots$ | $g^{j_n-i_n}$ | e' | g^{j_1} | $\cdots$ | g^{j_n}

$\downarrow$

$\ell(e)$ | i_1 | $\cdots$ | i_n | $+$ | j_1-i_1 | $\cdots$ | j_n-i_n | $\ell(e')$ | j_1 | $\cdots$ | 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

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

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

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

$e = (1, 9, 3, 3)$ $\star(3, 3, 9, 1)$ $e' = (3, 1, 1, 3)$

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

$e = (1, 9, 3, 3)$ $\star(3, 3, 9, 1)$ $e' = (3, 1, 1, 3)$

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

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

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

$e = (1, 9, 3, 3)$ $\star(3, 3, 9, 1)$ $e' = (3, 1, 1, 3)$

$\downarrow$

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

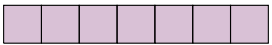
$(\mathbb{F}_3^n, +)$

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

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

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

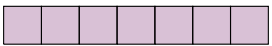
R-SDP: Given $H, s, \mathbb{E}$ find e: 1. $s = eH^\top$, 2. $e \in \mathbb{E}^n$

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

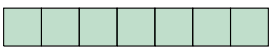
SDP: Given H, s, t find e: 1. $s = eH^\top$, 2. $\mathrm{wt}(e) = t$

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. $e \in G$

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

$(G, \star) < (\mathbb{E}^n, \star)$ $\quad G = \langle x_1, \ldots, x_m \rangle_\star$ $\quad e = x_1^{u_1} \star \cdots \star x_m^{u_m}$

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

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

$(G, \star) < (\mathbb{E}^n, \star)$ $\quad G = \langle x_1, \ldots, x_m \rangle_\star$ $\quad e = x_1^{u_1} \star \cdots \star x_m^{u_m}$

$\downarrow$

$$M = \begin{pmatrix} \ell(x_1) \\ \vdots \\ \ell(x_m) \end{pmatrix} \in \mathbb{F}_z^{m \times n} \qquad \ell(e) = uM \in \langle M \rangle \qquad \rightarrow \text{only need } u \in \mathbb{F}_z^m$$

$\mathbb{F}_z^n$

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

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

$(G, \star) < (\mathbb{E}^n, \star)$ $\quad G = \langle x_1, \ldots, x_m \rangle_\star$ $\quad e = x_1^{u_1} \star \cdots \star x_m^{u_m}$

$\downarrow$

$$M = \begin{pmatrix} \ell(x_1) \\ \vdots \\ \ell(x_m) \end{pmatrix} \in \mathbb{F}_z^{m \times n} \qquad \ell(e) = uM \in \langle M \rangle \qquad \rightarrow \text{only need } u \in \mathbb{F}_z^m$$

$\mathbb{F}_z^n$

$$\underset{u = (u_1, \ldots, u_m)}{\mathbb{F}_z^m} \xrightarrow{M} \underset{\ell(e) = uM = (i_1, \ldots, i_n)}{\mathbb{F}_z^n} \xrightarrow{g} \underset{e = (g^{i_1}, \ldots, g^{i_n})}{\mathbb{E}^n \subset \mathbb{F}_q^n}$$

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

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

$(G, \star) < (\mathbb{E}^n, \star)$ $\quad G = \langle x_1, \ldots, x_m \rangle_\star$ $\quad e = x_1^{u_1} \star \cdots \star x_m^{u_m}$

$\downarrow$

$$M = \begin{pmatrix} \ell(x_1) \\ \vdots \\ \ell(x_m) \end{pmatrix} \in \mathbb{F}_z^{m \times n} \qquad \ell(e) = uM \in \langle M \rangle \qquad \rightarrow \text{only need } u \in \mathbb{F}_z^m$$

$\mathbb{F}_z^n$

$\rightarrow |e| = |\varphi| = m \log(z)$ $\quad$ ☺ even smaller

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

$G = \langle x_1, x_2 \rangle$ $x_1 = (3, 1, 1, 3), x_2 = (1, 3, 9, 1)$ $e = {x_1}^2 \star {x_2}^1 = (9, 3, 9, 9)$

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

$G = \langle x_1, x_2 \rangle \qquad x_1 = (3, 1, 1, 3), x_2 = (1, 3, 9, 1) \qquad e = {x_1}^2 \star {x_2}^1 = (9, 3, 9, 9)$

$\downarrow$

$\ell(G) = \langle M \rangle \subset \mathbb{F}_3^n \qquad M = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 2 & 0 \end{pmatrix} \qquad \ell(e) = (2, 1)M = (2, 1, 1, 2)$

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

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

$G = \langle x_1, x_2 \rangle$ $\quad x_1 = (3, 1, 1, 3), x_2 = (1, 3, 9, 1)$ $\quad e = {x_1}^2 \star {x_2}^1 = (9, 3, 9, 9)$

$\downarrow$

$\ell(G) = \langle M \rangle \subset \mathbb{F}_3^n$ $\quad M = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 2 & 0 \end{pmatrix}$ $\quad \ell(e) = (2, 1)M = (2, 1, 1, 2)$

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

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

- $\mathbb{E}, G$ have multiplicative structure
  $e = (g^{i_1}, \ldots, g^{i_n})$
- $s = eH^\top$ has additive structure
  $s_j = \sum_{\ell=1}^{n} h_{j,\ell} g^{i_\ell}$ for $j \in \{1, \ldots, n-k\}$

- $\mathbb{E}, G$ have multiplicative structure
  $e = (g^{i_1}, \ldots, g^{i_n})$ not DLP
- $s = eH^\top$ has additive structure
  $s_j = \sum_{\ell=1}^{n} h_{j,\ell} g^{i_\ell}$ for $j \in \{1, \ldots, n-k\}$

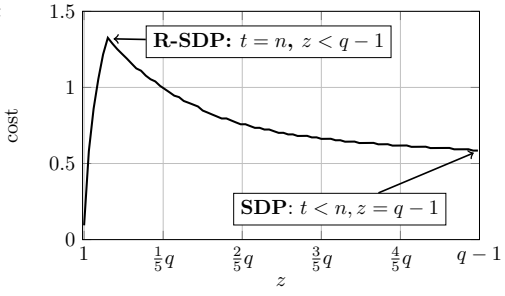
- $\mathbb{E}, G$ have multiplicative structure
  $e = (g^{i_1}, \ldots, g^{i_n})$ not DLP
- similar to SDP $\rightarrow$ NP-hard
- $s = eH^\top$ has additive structure
  $s_j = \sum_{\ell=1}^{n} h_{j,\ell} g^{i_\ell}$ for $j \in \{1, \ldots, n-k\}$
- adapted ISD algorithms

- $\mathbb{E}, G$ have multiplicative structure
  $e = (g^{i_1}, \ldots, g^{i_n})$ not DLP
- similar to SDP $\rightarrow$ NP-hard
- generalize to $e \in (\mathbb{E} \cup \{0\})^n$, $\mathrm{wt}(e) = t$
- choose t s.t. unique solution ($q = 251$)

- $s = eH^\top$ has additive structure
  $s_j = \sum_{\ell=1}^{n} h_{j,\ell} g^{i_\ell}$ for $j \in \{1, \ldots, n-k\}$
- adapted ISD algorithms

- $\mathbb{E}, G$ have multiplicative structure
  $e = (g^{i_1}, \ldots, g^{i_n})$ not DLP
- similar to SDP $\rightarrow$ NP-hard
- generalize to $e \in (\mathbb{E} \cup \{0\})^n$, $\mathrm{wt}(e) = t$
- choose t s.t. unique solution ($q = 251$)

- $s = eH^\top$ has additive structure
  $s_j = \sum_{\ell=1}^{n} h_{j,\ell} g^{i_\ell}$ for $j \in \{1, \ldots, n-k\}$
- adapted ISD algorithms



- $\mathbb{E}, G$ have multiplicative structure
  $e = (g^{i_1}, \ldots, g^{i_n})$ not DLP
- similar to SDP $\rightarrow$ NP-hard
- Take $\mathbb{E}$ with no additive structure
- $s = eH^\top$ has additive structure
  $s_j = \sum_{\ell=1}^{n} h_{j,\ell} g^{i_\ell}$ for $j \in \{1, \ldots, n-k\}$
- adapted ISD algorithms
- $(\mathbb{E} + \mathbb{E}) \cap (\mathbb{E} \cup \{0\})$ small

- $\mathbb{E}, G$ have multiplicative structure
  $e = (g^{i_1}, \ldots, g^{i_n})$ not DLP
- similar to SDP $\rightarrow$ NP-hard
- Take $\mathbb{E}$ with no additive structure

- $s = eH^\top$ has additive structure
  $s_j = \sum_{\ell=1}^{n} h_{j,\ell} g^{i_\ell}$ for $j \in \{1, \ldots, n-k\}$
- 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
  $e = (g^{i_1}, \ldots, g^{i_n})$ 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
  $s_j = \sum_{\ell=1}^{n} h_{j,\ell} g^{i_\ell}$ for $j \in \{1, \ldots, n-k\}$
- 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
  $e = (g^{i_1}, \ldots, g^{i_n})$ 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
  $s_j = \sum_{\ell=1}^{n} h_{j,\ell} g^{i_\ell}$ for $j \in \{1, \ldots, n-k\}$
- adapted ISD algorithms
  - good: $q = 13, g = 3, \mathbb{E} = \{1, 3, 9\}$
  - bad: $q = 13, g = 5, \mathbb{E} = \{\pm 1, \pm 5\}$

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

- $\mathbb{E}, G$ have multiplicative structure
  $e = (g^{i_1}, \ldots, g^{i_n})$ 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
  $s_j = \sum_{\ell=1}^{n} h_{j,\ell} g^{i_\ell}$ for $j \in \{1, \ldots, n-k\}$
- adapted ISD algorithms
  - good: $q = 13, g = 3, \mathbb{E} = \{1, 3, 9\}$
  - bad: $q = 13, g = 5, \mathbb{E} = \{\pm 1, \pm 5\}$

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

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.

- $\mathbb{E}, G$ have multiplicative structure
  $e = (g^{i_1}, \ldots, g^{i_n})$ 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
- algebraic attacks:
  $e_i^z = 1$ Gröbner basis

- $s = eH^\top$ has additive structure
  $s_j = \sum_{\ell=1}^{n} h_{j,\ell} g^{i_\ell}$ for $j \in \{1, \ldots, n-k\}$
- adapted ISD algorithms
  - good: $q = 13, g = 3, \mathbb{E} = \{1, 3, 9\}$
  - bad: $q = 13, g = 5, \mathbb{E} = \{\pm 1, \pm 5\}$

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

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.

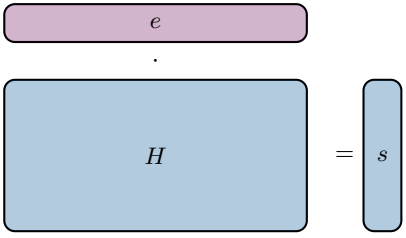
M. Baldi, et al. "CROSS security guide", NIST PQC round 2, 2025.

W. Beullens, P. Briaud, M. Øygarden. "A Security Analysis of Restricted Syndrome Decoding Problems", 2024.

- $q = 127, z = 7 \quad \rightarrow \quad \mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$ unique sol. $\frac{z^n}{q^{n-k}} < 1$

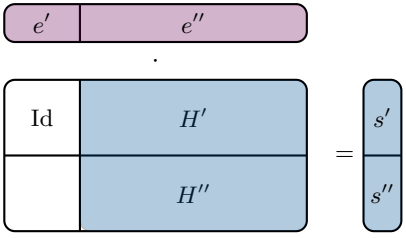
- $q = 127, z = 7$ $\rightarrow$ $\mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$ unique sol. $\frac{z^n}{q^{n-k}} < 1$

1. Reduction



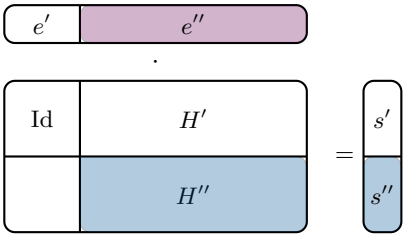
- $q = 127, z = 7 \quad \rightarrow \quad \mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$ unique sol. $\frac{z^n}{q^{n-k}} < 1$

1. Reduction



- $q = 127, z = 7$ $\rightarrow$ $\mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$ unique sol. $\frac{z^n}{q^{n-k}} < 1$

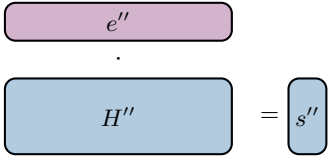
1. Reduction



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

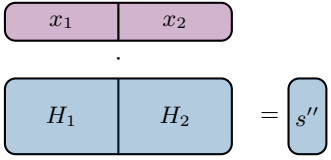
- $q = 127, z = 7 \quad \rightarrow \quad \mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$ unique sol. $\frac{z^n}{q^{n-k}} < 1$

2. Meet-in-the-Middle



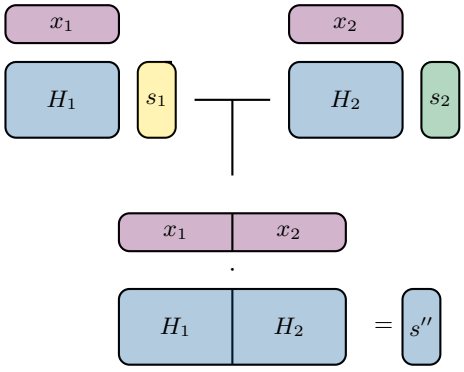
- $q = 127, z = 7 \quad \rightarrow \quad \mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$ unique sol. $\frac{z^n}{q^{n-k}} < 1$

2. Meet-in-the-Middle



- $q = 127, z = 7 \quad \rightarrow \quad \mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$ unique sol. $\frac{z^n}{q^{n-k}} < 1$

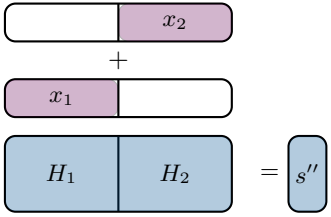
2. Meet-in-the-Middle



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

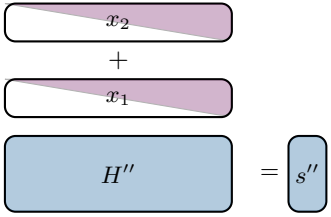
- $q = 127, z = 7 \rightarrow \mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$ unique sol. $\frac{z^n}{q^{n-k}} < 1$

3. Representation technique



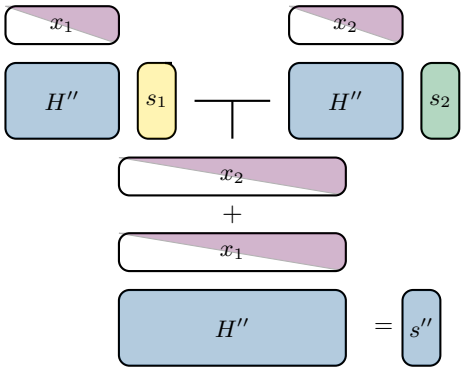
- $q = 127, z = 7 \quad \rightarrow \quad \mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$ unique sol. $\frac{z^n}{q^{n-k}} < 1$

3. Representation technique



- $q = 127, z = 7$ $\rightarrow$ $\mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$ unique sol. $\frac{z^n}{q^{n-k}} < 1$

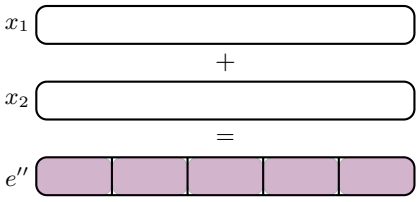
3. Representation technique



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

- $q = 127, z = 7 \quad \rightarrow \quad \mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$ unique sol. $\frac{z^n}{q^{n-k}} < 1$

3. Representation technique

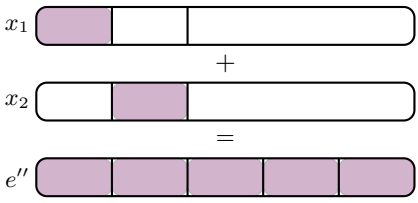


many repr. $(x_1, x_2) : e'' = x_1 + x_2$

small search space for (x_1, x_2)

- $q = 127, z = 7 \rightarrow \mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$ unique sol. $\frac{z^n}{q^{n-k}} < 1$

3. Representation technique

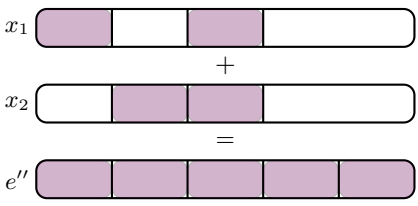


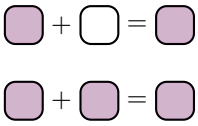
many repr. $(x_1, x_2) : e'' = x_1 + x_2$

small search space for (x_1, x_2)

- $q = 127, z = 7 \quad \rightarrow \quad \mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$ unique sol. $\frac{z^n}{q^{n-k}} < 1$

3. Representation technique





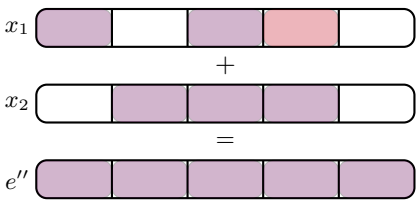
many repr. $(x_1, x_2) : e'' = x_1 + x_2$
small search space for (x_1, x_2)

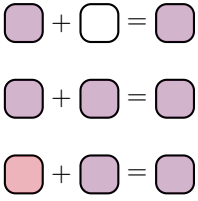
want $(\mathbb{E} + \mathbb{E}) \cap \mathbb{E}$ small

- $q = 127, z = 7 \quad \rightarrow \quad \mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$ unique sol. $\frac{z^n}{q^{n-k}} < 1$

3. Representation technique

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





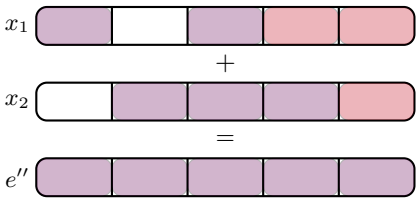
many repr. $(x_1, x_2) : e'' = x_1 + x_2$
small search space for (x_1, x_2)

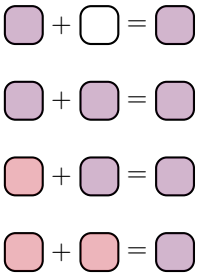
want $(\mathbb{E} + \mathbb{E}) \cap \mathbb{E}$ small
want $\mathbb{D}$ big

- $q = 127, z = 7 \rightarrow \mathbb{E} = \{1, 2, 4, 8, 16, 32, 64\}$ unique sol. $\frac{z^n}{q^{n-k}} < 1$

3. Representation technique

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





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

CROSS - Is this now CROSS?

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

weighted challenges

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

CROSS - Is this now CROSS?

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

weighted challenges

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

$\varphi(e) \leftarrow$ not from Seed

CROSS - Is this now CROSS?

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

weighted challenges

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

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

CROSS - Is this now CROSS?

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

weighted challenges

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

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

CROSS - Is this now CROSS?

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

weighted challenges

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

$r_2 = \varphi(e) \leftarrow$ not from Seed $\rightarrow$ weighted challenges

CROSS - Is this now CROSS?

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

weighted challenges

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

$r_2 = \varphi(e) \leftarrow$ not from Seed $\rightarrow$ weighted challenges

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

CROSS - Is this now CROSS?

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

weighted challenges

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

$r_2 = \varphi(e) \leftarrow$ 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 $\rightarrow (\beta[1], \ldots, \beta[t]) = \mathsf{Hash}(m, c_1[1], c_2[1], \ldots, c_1[t], c_2[t])$

weighted challenges

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

$r_2 = \varphi(e) \leftarrow$ not from Seed $\rightarrow$ weighted challenges

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

$\rightarrow$ adapt security analysis

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

where

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

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

CROSS - Is this now CROSS?

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

weighted challenges

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

$r_2 = \varphi(e) \leftarrow$ not from Seed $\rightarrow$ weighted challenges

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

$\rightarrow$ adapt security analysis

$w > t/2$ $\rightarrow$ small

$w \sim t/2$ $\rightarrow$ fast

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

where

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

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

CROSS - Is this now CROSS?

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

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

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

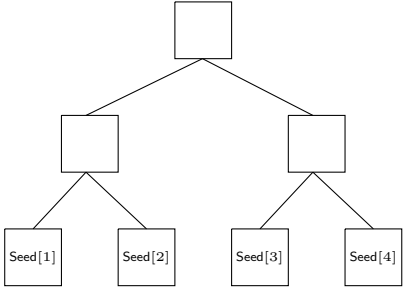
weighted challenges: $\beta[i] = 1$ w many times

Seed path

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

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

$\rightarrow$ can send Path



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

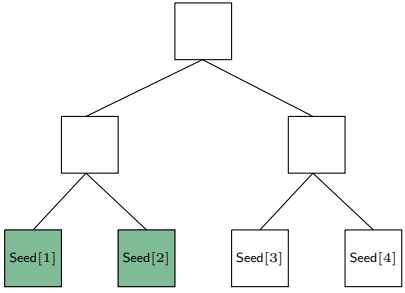
weighted challenges: $\beta[i] = 1$ w many times

Seed path

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

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

$\rightarrow$ can send Path



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

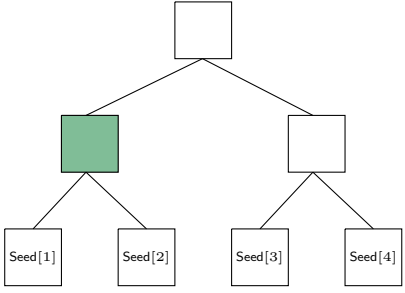
weighted challenges: $\beta[i] = 1$ w many times

Seed path

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

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

$\rightarrow$ can send Path



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

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

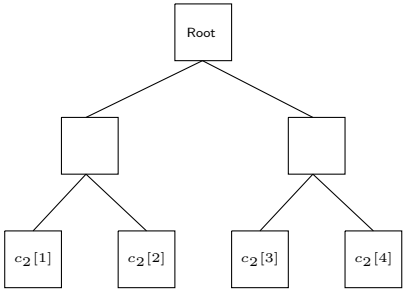
for round i : if challenged with β, put response $r_\beta[i]$ and $c_{\beta'}[i]$ $\rightarrow$ recover both $c_\beta[i], c_{\beta'}[i]$

Merkle proof

include w many $c_2[i]$ in response

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

$\rightarrow$ send Proof to recompute Root



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

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

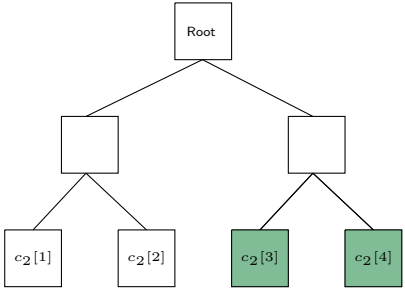
for round i : if challenged with β, put response $r_\beta[i]$ and $c_{\beta'}[i]$ $\rightarrow$ recover both $c_\beta[i], c_{\beta'}[i]$

Merkle proof

include w many $c_2[i]$ in response

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

$\rightarrow$ send Proof to recompute Root



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

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

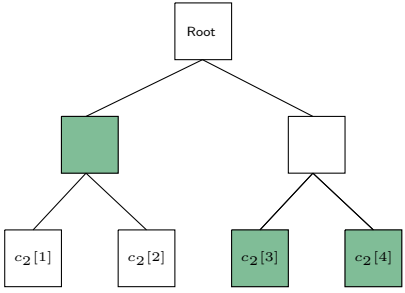
for round i : if challenged with β, put response $r_\beta[i]$ and $c_{\beta'}[i]$ $\rightarrow$ recover both $c_\beta[i], c_{\beta'}[i]$

Merkle proof

include w many $c_2[i]$ in response

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

$\rightarrow$ send Proof to recompute Root

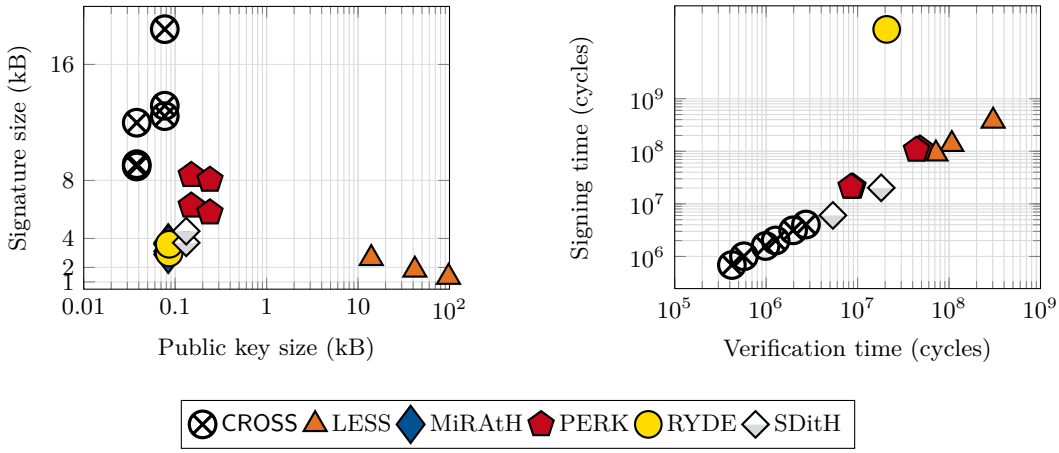


NIST cat. I

Problem	q, z	**Type**	(n, k, m)	(t, w)	|**Sign.**| (kB)	**Sign** (MCycles)	**Verify** (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 < 0.1 kB key gen. < 0.1 MCycle

CROSS - vs. other code-based?



What's next?

- Hardware implementation
- VOLE → smaller signatures

Slides

What's next?

- Hardware implementation
- VOLE → 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

NIST timeline

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

Slides

What's next?

- Hardware implementation
- VOLE → smaller signatures

Slides

Website

CROSS

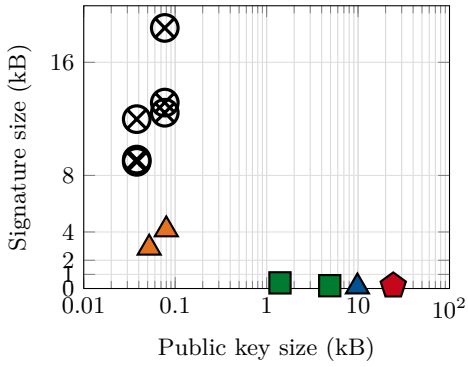
Codes & Restricted Objects Signature Scheme

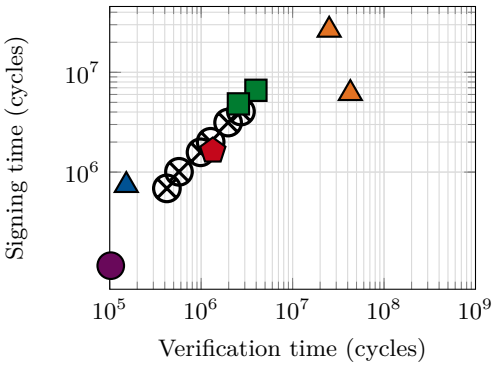
https://cross-crypto.com

Thank you!


TO BE
CONTINUED

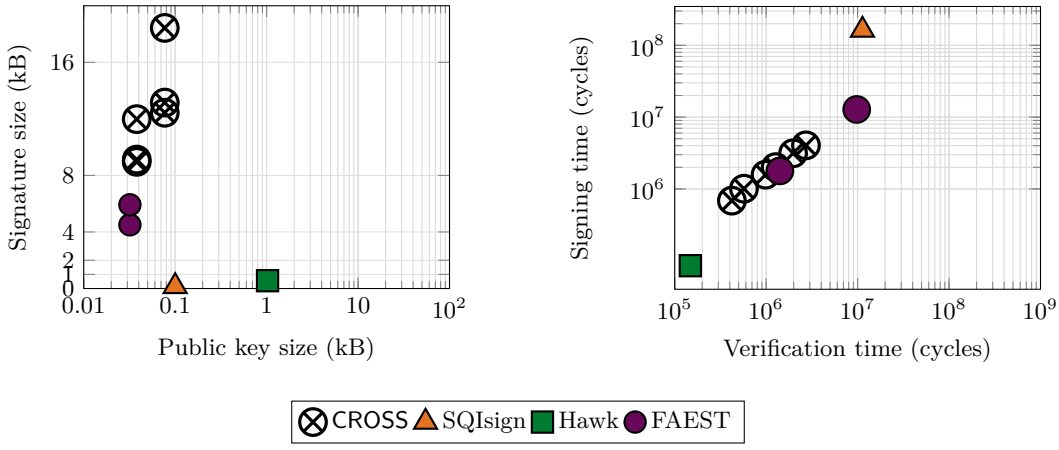
CROSS - vs. Multivariate?





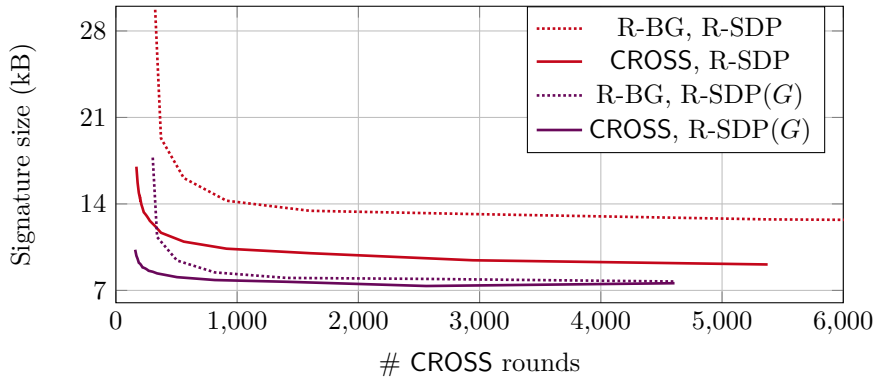
CROSS MQOM MAYO QR-UOV SNOVA UOV

CROSS - vs. rest?



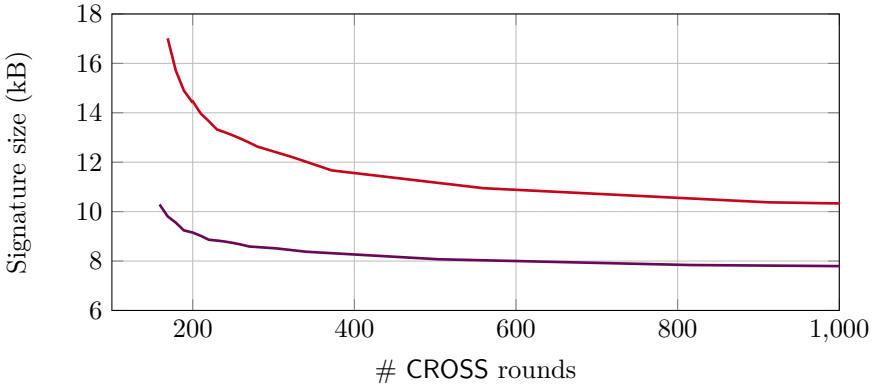
Baldi et al., 2023: R-BG protocol, soundness error $\varepsilon \approx \max\left\{\frac{1}{N}; \frac{1}{q-1}\right\}$

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



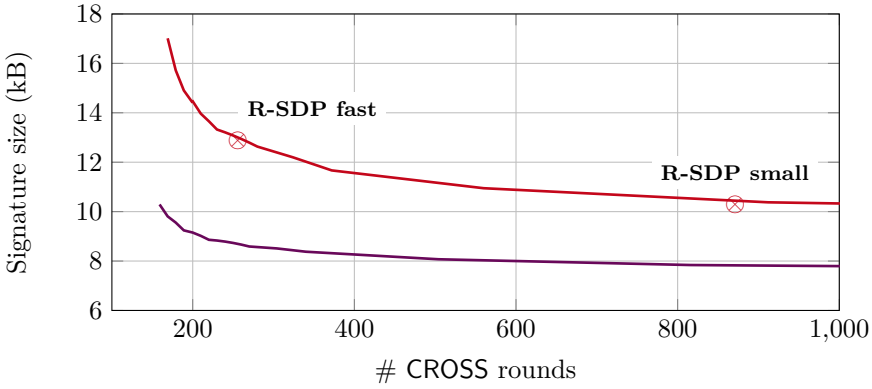
Baldi et al., 2023: R-BG protocol, soundness error $\varepsilon \approx \max\left\{\frac{1}{N}; \frac{1}{q-1}\right\}$

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



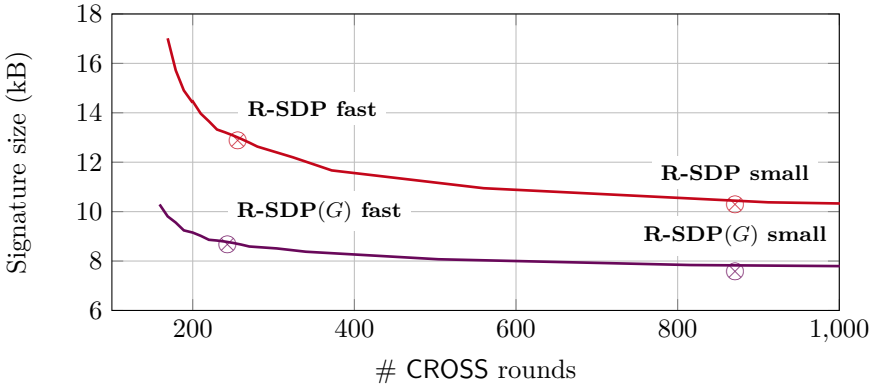
Baldi et al., 2023: R-BG protocol, soundness error $\varepsilon \approx \max\left\{\frac{1}{N}; \frac{1}{q-1}\right\}$

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



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

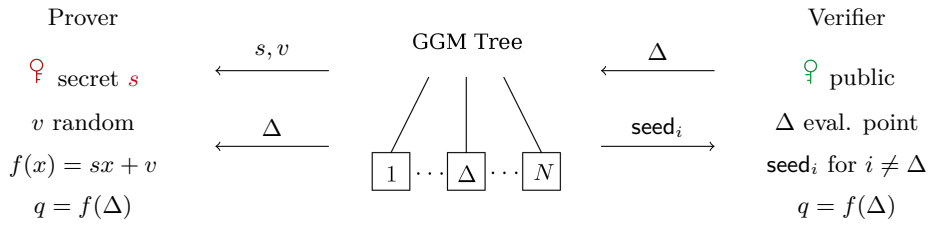
ZK Protocol

Prover	Verifier
secret s	public
v random	Δ eval. point
$f(x) = sx + v$	
	$q = f(\Delta)$

VOLE

Vector Oblivious Linear Transfer

ZK Protocol



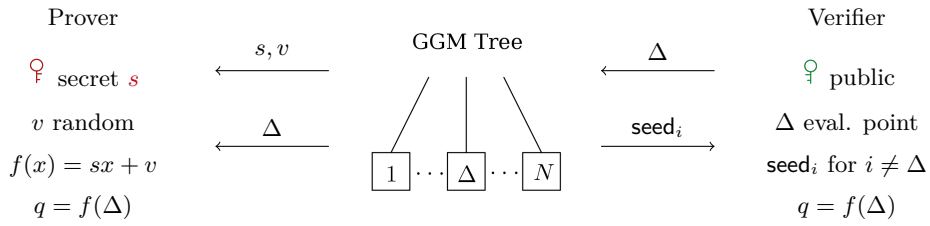
VOLE correlation $q = s\Delta + v = f(\Delta)$

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

VOLE

Vector Oblivious Linear Transfer

ZK Protocol



MPC

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

VOLE

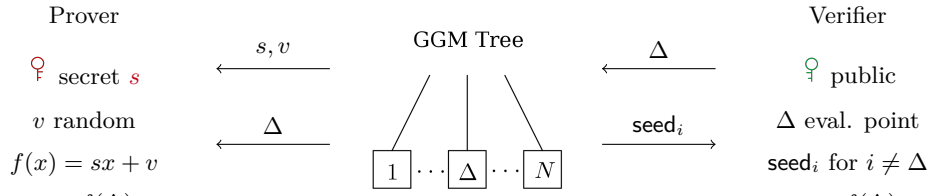
$s = \sum s_i$ GGM $\xleftarrow{\Delta}$ $N-1$ seeds

$v = \sum i s_i$ $q = \sum s_i(\Delta - i) = s\Delta + v$

VOLE

Vector Oblivious Linear Transfer

ZK Protocol



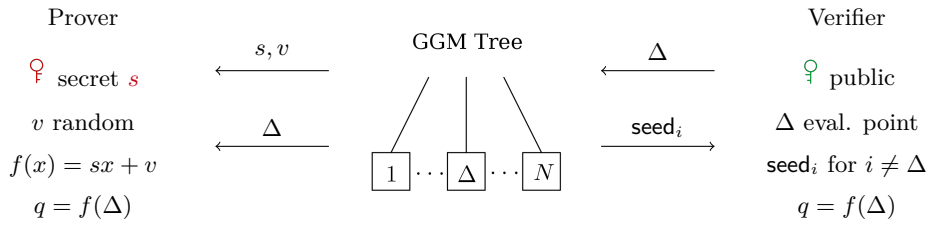
$$f(x) = \sum_{i=0}^{d} f_i x^i,$$
$$s = f_d$$
$$f_1(x), f_2(x)$$

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

VOLE

Vector Oblivious Linear Transfer

ZK Protocol



Disadvantages: slow

Advantages: small sizes