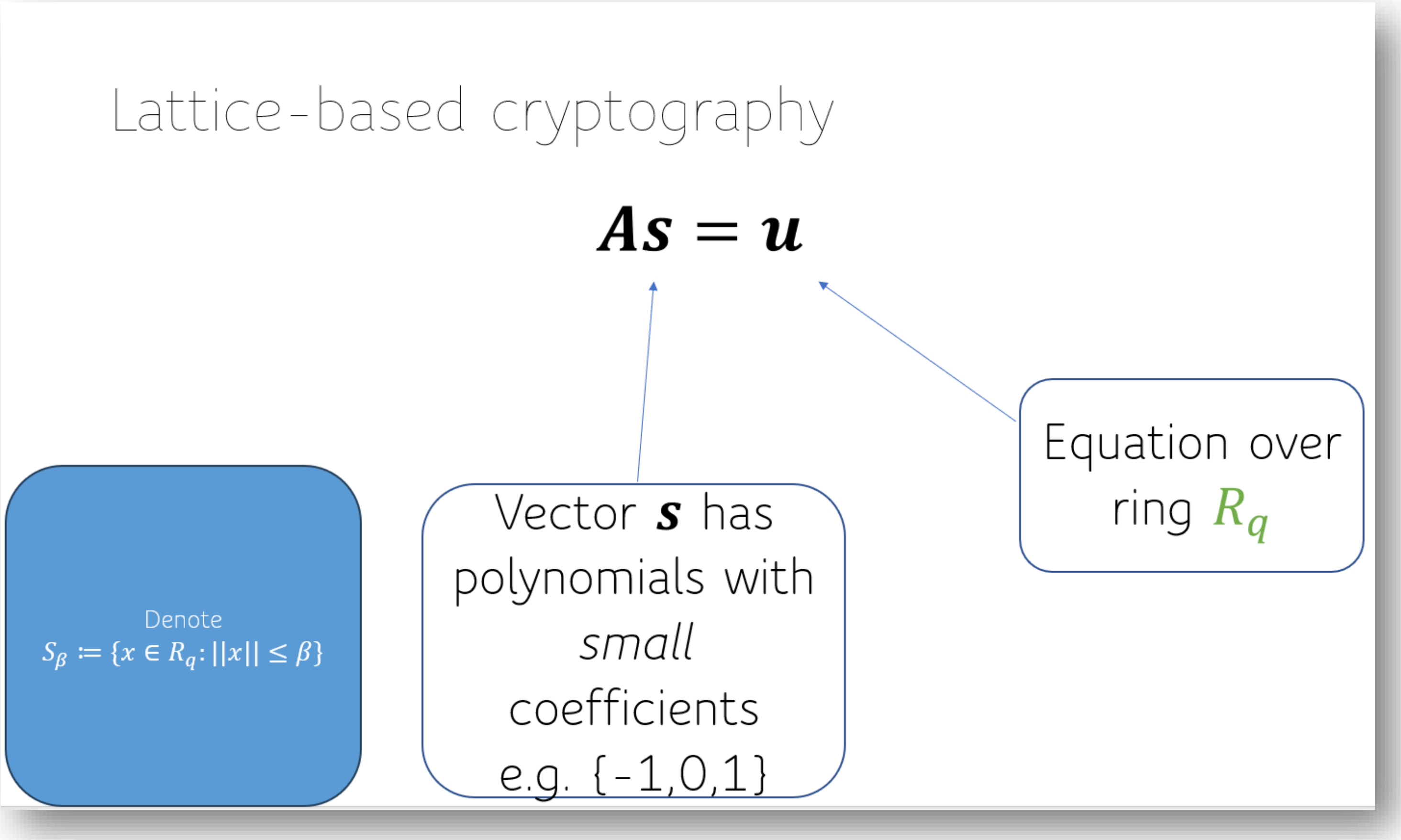
Exact Lattice-Based ZKP

Ngoc Khanh Nguyen

KING'S College LONDON

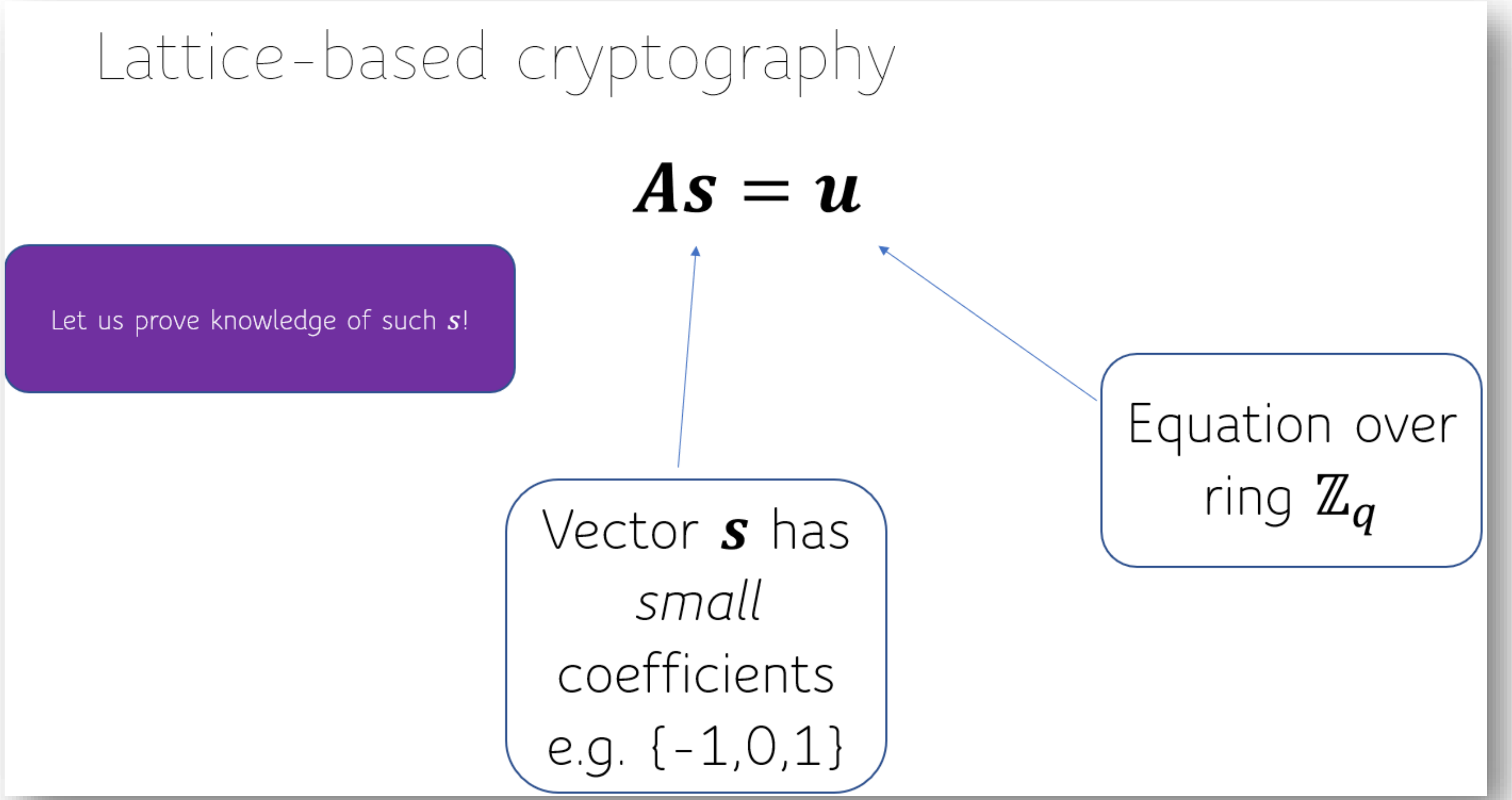
So far…



<u>Approximate [Lyu09,Lyu12]</u>:

- We only prove that we know short $\boldsymbol{s}$ and short c such that $\boldsymbol{As} = c\boldsymbol{u}$.
- This is enough for identification schemes and signatures like CRYSTALS-Dilithium.
- Small proof sizes ($\approx 3KB$).

But we wanted more!



<u>Exact</u>:

- We prove exactly that $\boldsymbol{s}$ is within specified range and $\boldsymbol{As} = \boldsymbol{u}\ (mod\ q)$.
- This is crucial for building more advanced privacy-preserving primitives, e.g. verifiable encryption.
- Much bigger proof sizes.

The main focus of this talk: [LNP22] framework

$$\boldsymbol{As} = \boldsymbol{u}\ (mod\ q) \text{ and } \mathbf{s} \in \{0,1\}^m$$

Equation over ring $\mathbb{Z}_q$

How many people are still following? ☺

Overview

$$\boldsymbol{As} = \boldsymbol{u} \; (\text{mod q})$$

$$\boldsymbol{s} \in \{0,1\}^m$$

Lemma: Let $\boldsymbol{s} \in \mathbb{Z}^{\boldsymbol{m}}$. Then, $\boldsymbol{s} \in \{0,1\}^m$ if and only if $\langle \boldsymbol{s}, \boldsymbol{s} - \mathbf{1} \rangle = 0$.

<u>Proof:</u> Suppose $\langle \boldsymbol{s}, \boldsymbol{s} - \mathbf{1} \rangle = 0$. This means that

$$\sum_{i=1}^{m} s_i(s_i - 1) = 0\,.$$

However, since each s_i is an integer, we have

$$s_i(s_i - 1) \geq 0$$

Hence, the sum is equal to zero if each of the inequalities is an equality, i.e. $s_i \in \{0,1\}$.

Overview

$$\boldsymbol{As} = \boldsymbol{u} \; (\mathrm{mod\, q})$$

$$\langle \boldsymbol{s}, \boldsymbol{s} - \mathbf{1} \rangle = 0.$$

$$\langle \boldsymbol{s}, \boldsymbol{s} - \mathbf{1} \rangle = 0 \; (mod\ q)$$

and

$$||\boldsymbol{s}|| \ll q$$

Overview

$\boldsymbol{As} = \boldsymbol{u}\ (\text{mod q})$

Linear proof

$\langle \boldsymbol{s}, \boldsymbol{s} - \boldsymbol{1} \rangle = 0\ (mod\ q)$

Inner product proof

$||\boldsymbol{s}|| \ll q$

Approximate range proof

How many people are still following? ☺

Overview

- If I take a random short vector $\boldsymbol{b}$, then clearly

$$\langle \boldsymbol{b}, \boldsymbol{s} \rangle$$

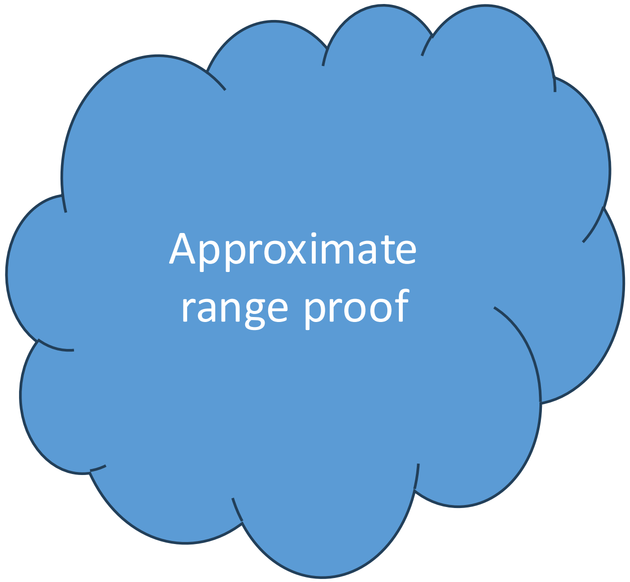
is short.

- But if I am given a large vector $\boldsymbol{s}$, then what's the probability that

$$\langle \boldsymbol{b}, \boldsymbol{s} \rangle$$

is short?

$||\boldsymbol{s}|| \ll q$

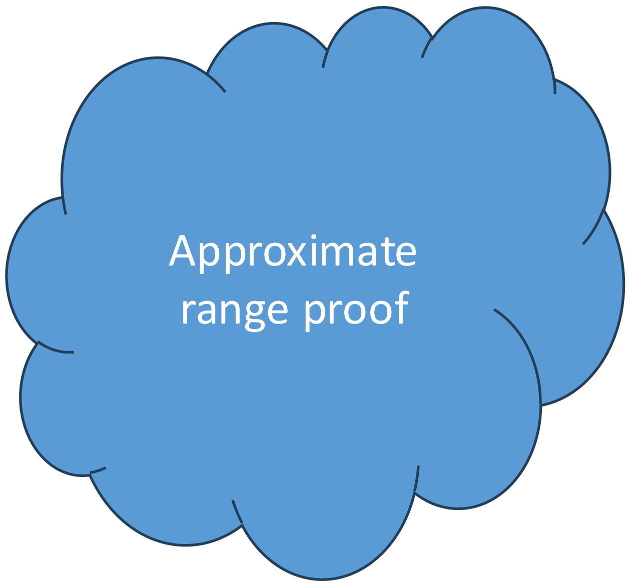


Overview + ZK

$||\boldsymbol{s}|| \ll q$

- If I take a random short vector $\boldsymbol{b}$, add a short mask y then clearly
$$y + \langle \boldsymbol{b}, \boldsymbol{s} \rangle$$
is short.

- But if I am given a large vector $\boldsymbol{s}$ and y, then what's the probability that
$$y + \langle \boldsymbol{b}, \boldsymbol{s} \rangle$$
is short?



Approximate range proof lemma [BL17,LNS21]

$||\boldsymbol{s}|| \ll q$

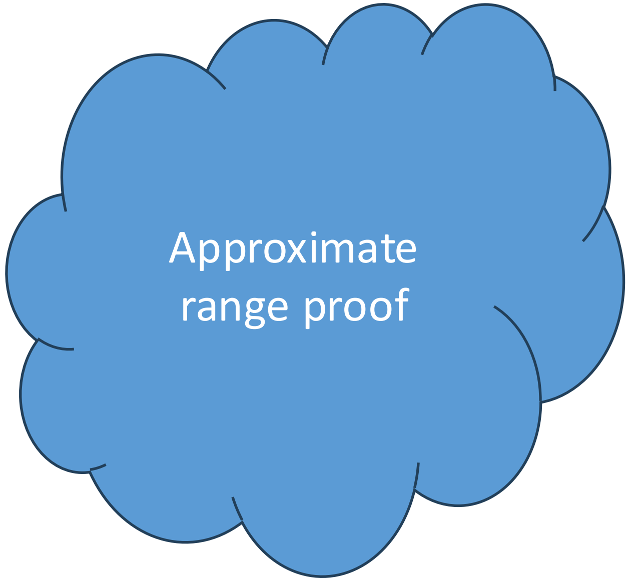
Lemma:

$$\Pr_{\boldsymbol{b} \leftarrow \{0,1\}^m}\left[|\langle \boldsymbol{b}, \boldsymbol{s} \rangle + y| < \frac{1}{2} \cdot ||\boldsymbol{s}||\right] \leq 1/2.$$

Proof: Let $s_i = ||\boldsymbol{s}||$ for some i.
Then, we can write $\langle \boldsymbol{b}, \boldsymbol{s} \rangle + y = b_i s_i + r$.

By the triangle inequality, at least one of $\{r, s_i + r\}$ has to have norm at least $\frac{1}{2} \cdot ||\boldsymbol{s}||$.

The probability of hitting that value is at least ½.



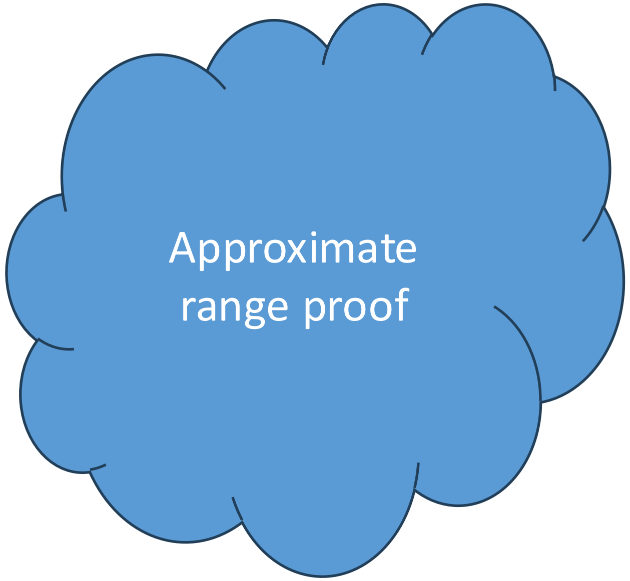
Overview

Lemma:

$$\Pr_{\boldsymbol{B}\leftarrow\{0,1\}^{\lambda\times m}}\left[||\boldsymbol{Bs}+\boldsymbol{y}||<\frac{1}{2}\cdot||\boldsymbol{s}||\right]\leq 1/2^{\lambda}.$$

Proof: By amplification.

$||\boldsymbol{s}|| \ll q$



Intuition

$||\boldsymbol{s}|| \ll q$

$\boldsymbol{s} \in \{0,1\}^m$

Hence, the verifier is convinced that $||\boldsymbol{s}|| \leq 2||\boldsymbol{y} + \boldsymbol{B}\boldsymbol{s}|| \leq 2(\alpha - m)$ (with high probability).

$\boldsymbol{y} \leftarrow [-\alpha, \alpha]^\lambda$

$\boldsymbol{y}, \boldsymbol{s}$ →

$\boldsymbol{B} \leftarrow \{0,1\}^{\lambda \times m}$

← $\boldsymbol{B}$

$\boldsymbol{z}$ →

$\boldsymbol{z} = \boldsymbol{y} + \boldsymbol{B}\boldsymbol{s}$

Check $||\boldsymbol{z}|| \leq \alpha - m$

$\boldsymbol{z} = \boldsymbol{y} + \boldsymbol{B}\boldsymbol{s}$

If $||\boldsymbol{z}|| > \alpha - m$, reject

Commitments

Message m

$t = Com(m; r)$

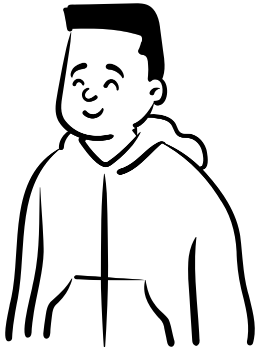
Binding:
It's hard to find two different openings (m, r) and (m', r') such that $Com(m; r) = Com(m'; r')$.

Hiding:
The adversary can't learn any information about (m, r) from t

Attempt 2

$||\boldsymbol{s}|| \ll q$

$\boldsymbol{s} \in \{0,1\}^m$



$\boldsymbol{y} \leftarrow [-\alpha, \alpha]^\lambda$
$\boldsymbol{r} \leftarrow \chi$

$\boldsymbol{t_y} \coloneqq Com(\boldsymbol{y}; \boldsymbol{r}), \boldsymbol{t_s} \coloneqq Com(\boldsymbol{s}; \boldsymbol{r})$ →

$\boldsymbol{B} \leftarrow \{0,1\}^{\lambda \times m}$

← $\boldsymbol{B}$

$\boldsymbol{z} = \boldsymbol{y} + \boldsymbol{B}\boldsymbol{s}$

If $||\boldsymbol{z}|| > \alpha - m$, reject

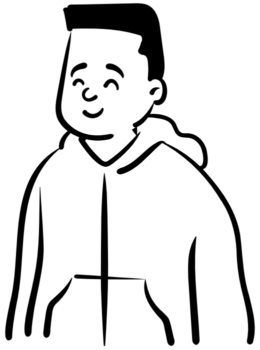
$\boldsymbol{z}, \boldsymbol{y}, \boldsymbol{s}, \boldsymbol{r}$ →

Check $||\boldsymbol{z}|| \leq \alpha - m$
$\boldsymbol{z} = \boldsymbol{y} + \boldsymbol{B}\boldsymbol{s}$
$\boldsymbol{t_y} = Com(\boldsymbol{y}; \boldsymbol{r}), \boldsymbol{t_s} = Com(\boldsymbol{s}; \boldsymbol{r})$

Attempt 2

$||\boldsymbol{s}|| \ll q$

$\boldsymbol{s} \in \{0,1\}^m$



$\boldsymbol{y} \leftarrow [-\alpha, \alpha]^\lambda$

$\boldsymbol{r} \leftarrow \chi$

$\boldsymbol{t_y} \coloneqq Com(\boldsymbol{y}; \boldsymbol{r}), \boldsymbol{t}$

Instead of sending the openings, we prove knowledge of them

$\boldsymbol{B} \leftarrow \{0,1\}^{\lambda \times m}$

$\boldsymbol{B}$

$\boldsymbol{z} = \boldsymbol{y} + \boldsymbol{Bs}$

If $||\boldsymbol{z}|| > \alpha - m$, reject

$\boldsymbol{z}, \boldsymbol{y}, \boldsymbol{s}, \boldsymbol{r}$

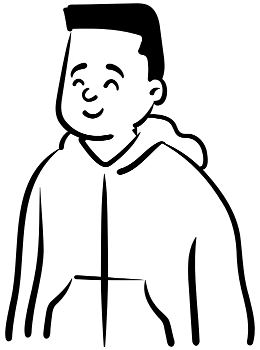
Check $||\boldsymbol{z}|| \leq \alpha - m$

$\boldsymbol{z} = \boldsymbol{y} + \boldsymbol{Bs}$

$\boldsymbol{t_y} = Com(\boldsymbol{y}; \boldsymbol{r}), \boldsymbol{t_s} = Com(\boldsymbol{s}; \boldsymbol{r})$

Approximate range proof

$||\boldsymbol{s}|| \ll q$



$\boldsymbol{s} \in \{0,1\}^m$

$\boldsymbol{y} \leftarrow [-\alpha, \alpha]^\lambda$

$\boldsymbol{r} \leftarrow \chi$

$\boldsymbol{t_y} \coloneqq Com(\boldsymbol{y}; \boldsymbol{r}), \boldsymbol{t_s} \coloneqq Com(\boldsymbol{s}; \boldsymbol{r})$ →

$\boldsymbol{B} \leftarrow \{0,1\}^{\lambda \times m}$

← $\boldsymbol{B}$

$\boldsymbol{z} = \boldsymbol{y} + \boldsymbol{B}\boldsymbol{s}$

If $||\boldsymbol{z}|| > \alpha - m$, reject

$\boldsymbol{z}$ →

Check $||\boldsymbol{z}|| \leq \alpha - m$

Prove knowledge of $\boldsymbol{y}, \boldsymbol{s}, \boldsymbol{r}$ s.t.

$\boldsymbol{t_y} = Com(\boldsymbol{y}; \boldsymbol{r})$

$\boldsymbol{t_s} = Com(\boldsymbol{s}; \boldsymbol{r})$

$\boldsymbol{z} = \boldsymbol{y} + \boldsymbol{B}\boldsymbol{s}$

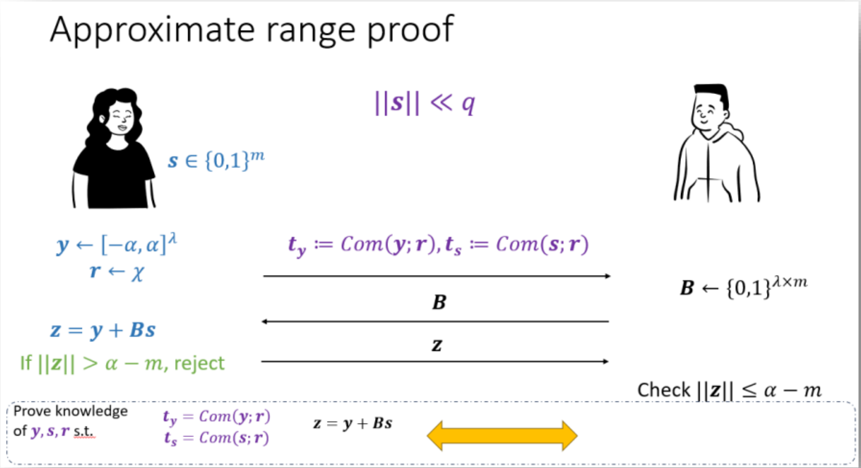
Overview

$$\boldsymbol{As} = \boldsymbol{u} \ (\text{mod q})$$

Linear proof

$$\langle \boldsymbol{s}, \boldsymbol{s} - \mathbf{1} \rangle = 0 \ (mod\ q)$$

Inner product proof



$$\boldsymbol{t_y} = Com(\boldsymbol{y}; \boldsymbol{r})$$
$$\boldsymbol{t_s} = Com(\boldsymbol{s}; \boldsymbol{r})$$

$$\boldsymbol{z} = \boldsymbol{y} + \boldsymbol{Bs}$$

Linear proof

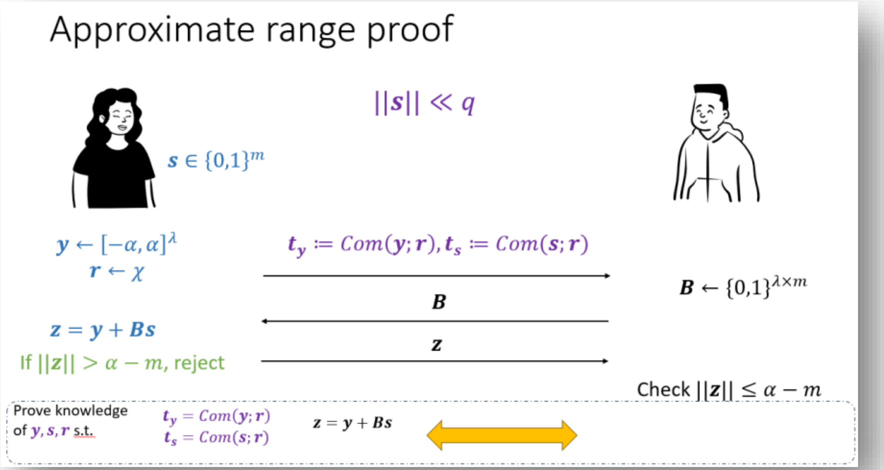
Overview

$$\boldsymbol{As} = \boldsymbol{u} \ (\text{mod q})$$

Inner product proof

$$\langle \boldsymbol{s}, \boldsymbol{s} - \mathbf{1} \rangle = 0 \ (mod\ q)$$

Inner product proof



$$\boldsymbol{t_y} = Com(\boldsymbol{y}; \boldsymbol{r})$$
$$\boldsymbol{t_s} = Com(\boldsymbol{s}; \boldsymbol{r})$$

$$\boldsymbol{z} = \boldsymbol{y} + \boldsymbol{Bs}$$

Inner product proof

How many people are still following? ☺

Next step: inner products over $\mathbb{Z}_q$

- We want to prove inner products (either between two committed messages, or between one secret and one public vector)

- Working natively over integers will result with bad soundness error (see previous lecture)

- We need to translate the inner products into relations over the polynomial ring R_q

$$R_q = \mathbb{Z}_q[X]/(f(X))$$

- For concreteness, set $f(X) := X^d + 1$ for a power-of-two d

$R_q = \mathbb{Z}_q[X]/(f(X))$

- For concreteness, set $f(X) := X^d + 1$ for a power-of-two d
- Let $a = a_0 + a_1 X + \cdots + a_{d-1} X^{d-1} \in \mathrm{R}_q$. Then $||a|| = \max_i |a_i|$.
- <u>Lemma</u>: $||ab|| \leq d \cdot ||a|| \cdot ||b||$.

Setup

- For $i \in \mathbb{Z}_{2d}^{\times}$, let us denote $\sigma_i\colon R_q \mapsto R_q$ to be the automorphism defined by $\sigma_i(X) = X^i$.
- Let $\sigma \coloneqq \sigma_{-1}$. Seems irrelevant now but it will be useful later!
- For $x \in R_q$, we denote $ct(x) = x_0$ its constant coefficient/term.

The key ingredient

Lemma: Let $\mathrm{u} := \sum_{i=0}^{d-1} u_i X^i$ and $\mathrm{v} := \sum_{i=0}^{d-1} v_i X^i$ be ring elements in R_q. Then, the constant coefficient of the polynomial $u\sigma_{-1}(v) \in R_q$ is $\sum_{i=0}^{d-1} u_i v_i$.

Proof: By definition,

$$u\sigma_{-1}(v) = (\sum_{i=0}^{d-1} u_i X^i)\sigma\left(\sum_{i=0}^{d-1} v_i X^i\right)$$

$$= (\sum_{i=0}^{d-1} u_i X^i)\left(\sum_{i=0}^{d-1} v_i X^{-i}\right) = \sum_{i,j} u_i v_j X^{i-j}.$$

Therefore, the constant term is indeed $u_0 v_0 + u_1 v_1 + \cdots + u_{d-1} v_{d-1}$.

The key ingredient

Lemma: Let $u := \sum_{i=0}^{d-1} u_i X^i$ and $v := \sum_{i=0}^{d-1} v_i X^i$ be ring elements in R_q. Then, the constant coefficient of the polynomial $u\sigma_{-1}(v) \in R_q$ is $\sum_{i=0}^{d-1} u_i v_i$.

As an application of this lemma, we know a vector $\boldsymbol{s} \in \mathbb{Z}^d$ satisfies $\langle \boldsymbol{s}, \boldsymbol{s} - \boldsymbol{1} \rangle = 0 \ (mod\ q)$ **if and only if**

$$ct\left(\left(s - \sum_{i=0}^{d-1} X^i\right) \cdot \sigma(s)\right) = 0$$

where $s := \sum_{i=0}^{d-1} s_i X^i$.

The key ingredient

Lemma: Let $\mathrm{u} := \sum_{i=0}^{d-1} u_i X^i$ and $\mathrm{v} := \sum_{i=0}^{d-1} v_i X^i$ be ring elements in R_q. Then, the constant coefficient of the polynomial $u\sigma_{-1}(v) \in R_q$ is $\sum_{i=0}^{d-1} u_i v_i$.

As an application of this lemma, we know a vector $\boldsymbol{s} = (\boldsymbol{s_1}, \dots, \boldsymbol{s_{m/d}}) \in \mathbb{Z}^m$ satisfies $\langle \boldsymbol{s}, \boldsymbol{s} - \boldsymbol{1} \rangle = 0 \ (mod\ q)$ **if and only if**

$$ct\left(\sum_{j=1}^{m/d}\left(s_j - \sum_{i=0}^{d-1} X^i\right) \cdot \sigma(s_j)\right) = 0$$

where $\mathrm{s_j} := \sum_{i=0}^{d-1} s_{j \cdot d+i} X^i$.

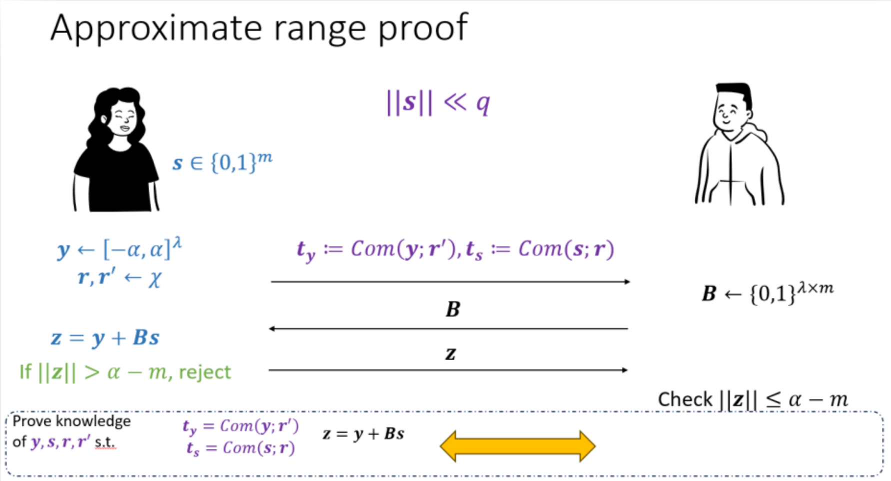
Back to overview

$$\boldsymbol{As} = \boldsymbol{u} \ (\mathrm{mod\ q})$$

$$\forall i, ct(f_i(\boldsymbol{s})) = 0$$

$$\langle \boldsymbol{s}, \boldsymbol{s} - \boldsymbol{1} \rangle = 0 \ (mod\ q)$$

$$ct\left(\sum_{j=1}^{m/d}\left(s_j - \sum_{i=0}^{d-1} X^i\right) \cdot \sigma(s_j)\right) = 0$$



$$\boldsymbol{t_y} = Com(\boldsymbol{y}; \boldsymbol{r}) \qquad \boldsymbol{t_s} = Com(\boldsymbol{s}; \boldsymbol{r}) \qquad \boldsymbol{z} = \boldsymbol{y} + \boldsymbol{Bs}$$

$$\boldsymbol{t_y} = Com(\boldsymbol{y}; \boldsymbol{r}) \qquad \boldsymbol{t_s} = Com(\boldsymbol{s}; \boldsymbol{r}) \qquad \forall i, ct(g_i(\boldsymbol{s}, \boldsymbol{y})) = 0$$

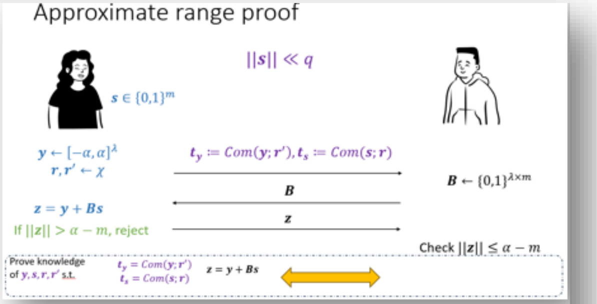
So far so good

$$\boldsymbol{As} = \boldsymbol{u} \ (\text{mod q})$$

$$\forall i, ct(f_i(\boldsymbol{s})) = 0$$

$$\langle \boldsymbol{s}, \boldsymbol{s} - \mathbf{1} \rangle = 0 \ (mod\ q)$$

$$ct\left(\sum_{j=1}^{m/d}\left(s_j - \sum_{i=0}^{d-1} X^i\right) \cdot \sigma(s_j)\right) = 0$$



$$\boldsymbol{t_y} = Com(\boldsymbol{y}; \boldsymbol{r}) \qquad \boldsymbol{z} = \boldsymbol{y} + \boldsymbol{Bs}$$
$$\boldsymbol{t_s} = Com(\boldsymbol{s}; \boldsymbol{r})$$

$$\boldsymbol{t_y} = Com(\boldsymbol{y}; \boldsymbol{r}) \qquad \forall i, ct(g_i(\boldsymbol{s}, \boldsymbol{y})) = 0$$
$$\boldsymbol{t_s} = Com(\boldsymbol{s}; \boldsymbol{r})$$

$$\boldsymbol{t_y} = Com(\boldsymbol{y}; \boldsymbol{r}) \qquad \forall i, ct(f_i(\boldsymbol{s}, \boldsymbol{y})) = 0$$
$$\boldsymbol{t_s} = Com(\boldsymbol{s}; \boldsymbol{r})$$

where f_i are public quadratic functions (with σ)

How many people are still following? ☺

Proving constant coefficients

- We want to prove that $\forall i, ct(f_{\boldsymbol{i}}(\boldsymbol{s}, \boldsymbol{y})) = 0$
- Clearly, for any $\mu_1, \ldots, \mu_k \in \mathbb{Z}_q$ we have

$$ct\left(\sum_{i=1}^{k} \mu_i \cdot f_{\boldsymbol{i}}(\boldsymbol{s}, \boldsymbol{y})\right) = \sum_{i=1}^{k} \mu_i \cdot ct(f_{\boldsymbol{i}}(\boldsymbol{s}, \boldsymbol{y})) = 0\,.$$

Proving constant coefficients

- We want to prove that $\forall i, ct(f_{\boldsymbol{i}}(\boldsymbol{s},\boldsymbol{y})) = 0$

- Clearly, for any $\mu_1, \dots, \mu_k \in \mathbb{Z}_q$ we have

$$ct\left(\sum_{i=1}^{k} \mu_i \cdot f_{\boldsymbol{i}}(\boldsymbol{s},\boldsymbol{y})\right) = \sum_{i=1}^{k} \mu_i \cdot ct(f_{\boldsymbol{i}}(\boldsymbol{s},\boldsymbol{y})) \ = 0\,.$$

But what happens if for some $i, ct(f_{\boldsymbol{i}}(\boldsymbol{s},\boldsymbol{y})) \ \neq 0$?

Then, with prob. $\frac{1}{q}$, we have $ct\left(\sum_{i=1}^{k} \mu_i \cdot f_{\boldsymbol{i}}(\boldsymbol{s},\boldsymbol{y})\right) = 0$. Repeat L times.

Adding zero-knowledge

- $\sum_{i=1}^{k} \mu_i \cdot f_{\boldsymbol{i}}(\boldsymbol{s}, \boldsymbol{y})$ potentially leaks information about $\boldsymbol{s}, \boldsymbol{y}$

Adding zero-knowledge

- $\sum_{i=1}^{k} \mu_i \cdot f_{\boldsymbol{i}}(\boldsymbol{s}, \boldsymbol{y})$ potentially leaks information about $\boldsymbol{s}, \boldsymbol{y}$

- Sample and commit to random polynomials $g_1, \dots, g_L \leftarrow \{x \in R_q : ct(x) = 0\}$.
- Given challenges $\mu_{j,1}, \dots, \mu_{j,k}$ for $j = 1, \dots, L$, compute

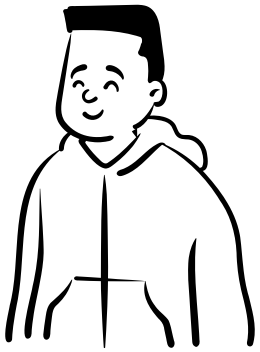
$$h_j := g_j + \sum_{i=1}^{k} \mu_{j,i} \cdot f_{\boldsymbol{i}}(\boldsymbol{s}, \boldsymbol{y})$$

Hence, $ct(h_j) = 0$ and h_j hides info about other coeffs of $\sum_{i=1}^{k} \mu_{j,i} \cdot f_{\boldsymbol{i}}(\boldsymbol{s}, \boldsymbol{y})$

$\boldsymbol{t_y} = Com(\boldsymbol{y}; \boldsymbol{r})$
$\boldsymbol{t_s} = Com(\boldsymbol{s}; \boldsymbol{r})$

$\forall i, ct(f_i(\boldsymbol{s}, \boldsymbol{y})) = 0$

$\boldsymbol{s}, \boldsymbol{y}$



$g_1, \ldots, g_L \leftarrow \{x \in R_q : ct(x) = 0\}$

$\boldsymbol{t_g} \coloneqq Com(\boldsymbol{g}; \boldsymbol{r})$ →

← $(\mu_{j,i})_{j,i}$

$(\mu_{j,i})_{j,i} \leftarrow \mathbb{Z}_q^{L \times k}$

$\forall j, h_j := g_j + \sum_{i=1}^{k} \mu_{j,i} \cdot f_i(\boldsymbol{s}, \boldsymbol{y})$

$h_1, \ldots, h_L$ →

Check $\forall j, ct(h_j) = 0$

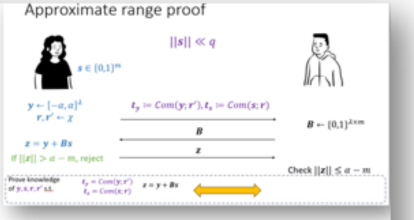
Overview

$\boldsymbol{As} = \boldsymbol{u} \pmod{q}$

$\forall i, ct(f_i(\boldsymbol{s})) = 0$

$\langle \boldsymbol{s}, \boldsymbol{s} - \boldsymbol{1} \rangle = 0 \ (mod\ q)$

$$ct\left(\sum_{j=1}^{m/d}\left(s_j - \sum_{i=0}^{d-1} X^i\right) \cdot \sigma(s_j)\right) = 0$$



$\boldsymbol{t_y} = Com(\boldsymbol{y}; \boldsymbol{r})$
$\boldsymbol{t_s} = Com(\boldsymbol{s}; \boldsymbol{r})$ $\quad \boldsymbol{z} = \boldsymbol{y} + \boldsymbol{Bs}$

$\boldsymbol{t_y} = Com(\boldsymbol{y}; \boldsymbol{r})$
$\boldsymbol{t_s} = Com(\boldsymbol{s}; \boldsymbol{r})$ $\quad \forall i, ct(g_i(\boldsymbol{s}, \boldsymbol{y})) = 0$

$\boldsymbol{t_y} = Com(\boldsymbol{y}; \boldsymbol{r})$
$\boldsymbol{t_s} = Com(\boldsymbol{s}; \boldsymbol{r})$ $\quad \forall i, ct(f_i(\boldsymbol{s}, \boldsymbol{y})) = 0$

where f_i are public quadratic functions (with σ)

$\boldsymbol{t_y} = Com(\boldsymbol{y}; \boldsymbol{r})$
$\boldsymbol{t_s} = Com(\boldsymbol{s}; \boldsymbol{r})$

$\boldsymbol{t_g} := Com(\boldsymbol{g}; \boldsymbol{r})$

$$\forall j, h_j = g_j + \sum_{i=1}^{k} \mu_{j,i} \cdot f_i(\boldsymbol{s}, \boldsymbol{y})$$

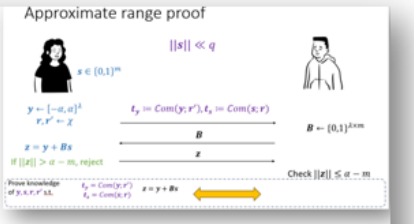
In other words

$$\boldsymbol{As} = \boldsymbol{u} \text{ (mod q)}$$

$$\forall i, ct(f_i(\boldsymbol{s})) = 0$$

$$\langle \boldsymbol{s}, \boldsymbol{s} - \boldsymbol{1} \rangle = 0 \ (mod\ q)$$

$$ct\left(\sum_{j=1}^{m/d}\left(s_j - \sum_{i=0}^{d-1} X^i\right) \cdot \sigma(s_j)\right) = 0$$



$$\boldsymbol{t_y} = Com(\boldsymbol{y}; \boldsymbol{r}) \quad \boldsymbol{t_s} = Com(\boldsymbol{s}; \boldsymbol{r}) \qquad \boldsymbol{z} = \boldsymbol{y} + \boldsymbol{Bs}$$

$$\boldsymbol{t_y} = Com(\boldsymbol{y}; \boldsymbol{r}) \quad \boldsymbol{t_s} = Com(\boldsymbol{s}; \boldsymbol{r}) \qquad \forall i, ct(g_i(\boldsymbol{s}, \boldsymbol{y})) = 0$$

$$\boldsymbol{t_y} = Com(\boldsymbol{y}; \boldsymbol{r}) \quad \boldsymbol{t_s} = Com(\boldsymbol{s}; \boldsymbol{r}) \qquad \forall i, ct(f_i(\boldsymbol{s}, \boldsymbol{y})) = 0$$

where f_i are public quadratic functions (with σ)

$$\boldsymbol{t_y} = Com(\boldsymbol{y}; \boldsymbol{r}) \quad \boldsymbol{t_s} = Com(\boldsymbol{s}; \boldsymbol{r})$$

$$\boldsymbol{t_g} \coloneqq Com(\boldsymbol{g}; \boldsymbol{r})$$

$$\forall j, P_j(\boldsymbol{s}, \boldsymbol{y}, \boldsymbol{g}) = 0$$

Public quadratic function (with σ)

How many people are still following? ☺

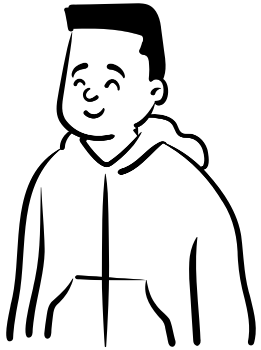
Simple amortisation

$t_y = Com(y; r)$

$t_s = Com(s; r)$ $\quad \forall j, P_j(s, y, g) = 0$

$t_g := Com(g; r)$

s, y



$\eta_1, \ldots, \eta_L$

$\eta_i \leftarrow R_q^L$

Prove that:

$$\sum_{j=1}^{L} \eta_j \cdot P_j(s, y, g) = 0$$

Soundness analysis

- What's the probability that $\sum_{j=1}^{L} \eta_j \cdot P_j(\boldsymbol{s}, \boldsymbol{y}, \boldsymbol{g}) = 0$ if for some j, $P_j(\boldsymbol{s}, \boldsymbol{y}, \boldsymbol{g}) \neq 0$?

- Consider the standard polynomial ring $R_q = \mathbb{Z}_q[X]/(X^d + 1)$ where d is a power-of-two and $q = 5 \ (mod\ 8)$.

Soundness analysis

- What's the probability that $\sum_{j=1}^{L} \eta_j \cdot P_j(\boldsymbol{s}, \boldsymbol{y}, \boldsymbol{g}) = 0$ if for some j, $P_j(\boldsymbol{s}, \boldsymbol{y}, \boldsymbol{g}) \neq 0$?

- Consider the standard polynomial ring $R_q = \mathbb{Z}_q[X]/(X^d + 1)$ where d is a power-of-two and $q = 5\ (mod\ 8)$.

- Then, $X^d + 1 = \left(X^{\frac{d}{2}} - r\right)\left(X^{\frac{d}{2}} + r\right)$ factors into two irreducible polynomials modulo q.

- By CRT, R_q is isomorphic to $\frac{\mathbb{Z}[X]}{\left(X^{\frac{d}{2}} - r, q\right)} \times \frac{\mathbb{Z}[X]}{\left(X^{\frac{d}{2}} - r, q\right)}$.

Soundness analysis

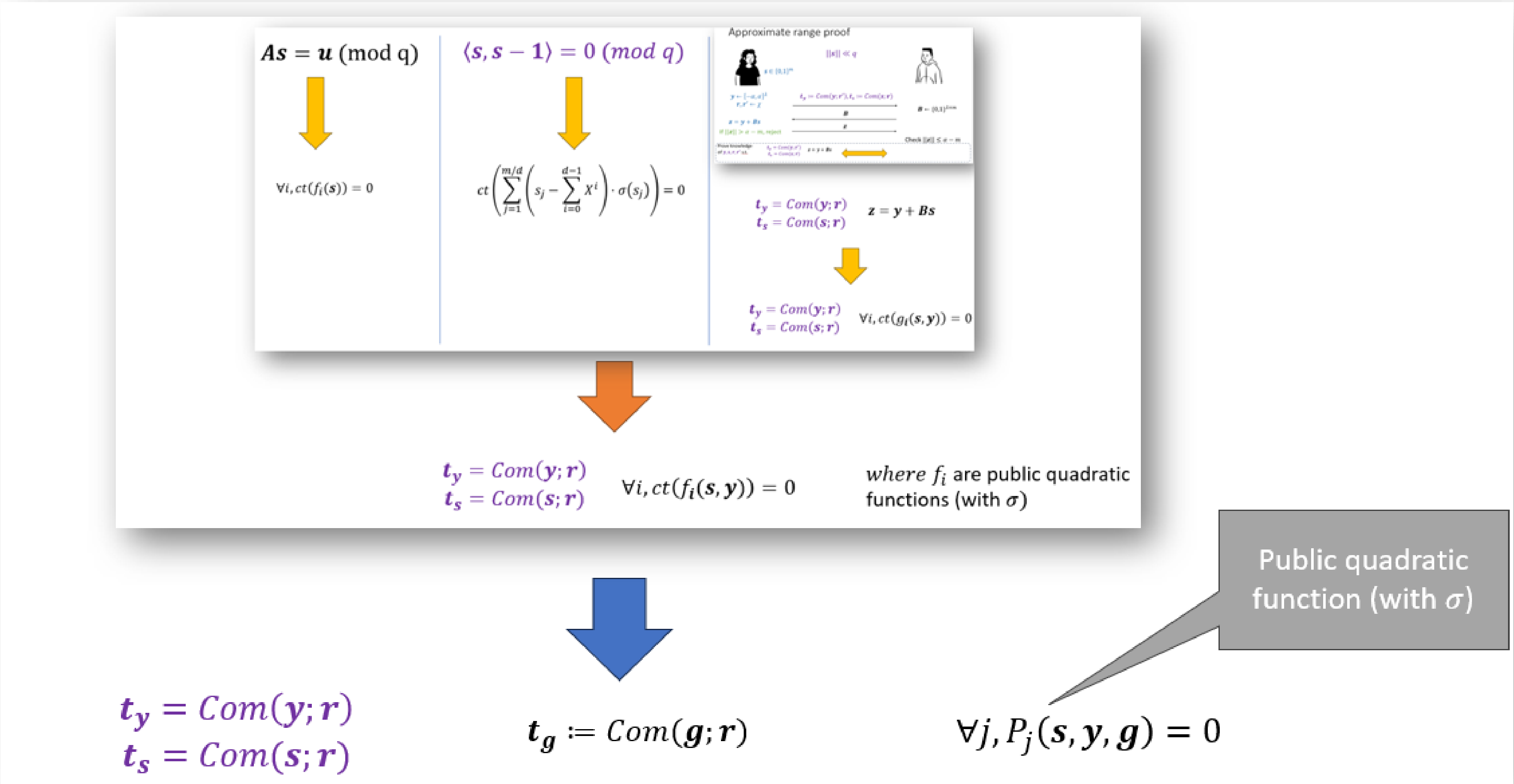
- What's the probability that $\sum_{j=1}^{L} \eta_j \cdot P_j(\boldsymbol{s}, \boldsymbol{y}, \boldsymbol{g}) = 0$ if for some j, $P_j(\boldsymbol{s}, \boldsymbol{y}, \boldsymbol{g}) \neq 0$?

- Consider the standard polynomial ring $R_q = \mathbb{Z}_q[X]/(X^d + 1)$ where d is a power-of-two and $q = 5\ (mod\ 8)$.

- Then, $X^d + 1 = \left(X^{\frac{d}{2}} - r\right)\left(X^{\frac{d}{2}} + r\right)$ factors into two irreducible polynomials modulo q.

- By CRT, R_q is isomorphic to $\frac{\mathbb{Z}[X]}{\left(X^{\frac{d}{2}} - r, q\right)} \times \frac{\mathbb{Z}[X]}{\left(X^{\frac{d}{2}} - r, q\right)}$.

- Hence the probability that $\eta_j \cdot P_j(\boldsymbol{s}, \boldsymbol{y}, \boldsymbol{g}) = x$ is at most $2q^{-d/2}$.



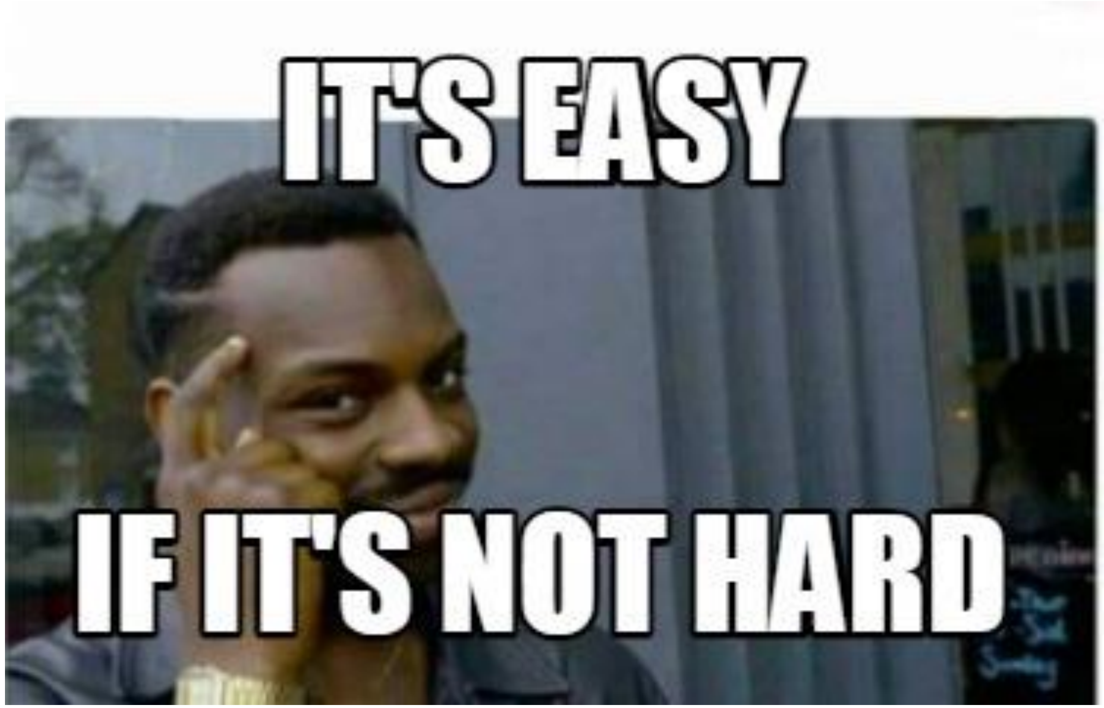
$$\boldsymbol{t_y} = Com(\boldsymbol{y}; \boldsymbol{r})$$
$$\boldsymbol{t_s} = Com(\boldsymbol{s}; \boldsymbol{r})$$

$$\boldsymbol{t_g} \coloneqq Com(\boldsymbol{g}; \boldsymbol{r})$$

$$Q(\boldsymbol{s}, \boldsymbol{y}, \boldsymbol{g}) = 0$$

How many people are still following? ☺

I can only do handwaving thus far



ABDLOP commitment (= [Ajt96] + [BDLOP18])

- Suppose we want to commit to a polynomial vector $(\boldsymbol{s_1}, \boldsymbol{m}) \in R_q^{m_1+l}$ where $\boldsymbol{s_1}$ has small norm (but not necessarily $\boldsymbol{m}$).

We could treat $\boldsymbol{s_1} \coloneqq \boldsymbol{s}$ and $\boldsymbol{m} \coloneqq (\boldsymbol{y}, \boldsymbol{g})$.

ABDLOP commitment (= [Ajt96] + [BDLOP18])

- Suppose we want to commit to a polynomial vector $(\boldsymbol{s_1}, \boldsymbol{m}) \in R_q^{m_1+l}$ where $\boldsymbol{s_1}$ has small norm (but not necessarily $\boldsymbol{m}$).

- The ABDLOP commitment under randomness $\boldsymbol{s}_2 \in R_q^{m_2}$ is defined as:

$$\begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{s_1} + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \end{bmatrix} \boldsymbol{s_2} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{m} \end{bmatrix}.$$

ABDLOP commitment (= [Ajt96] + [BDLOP18])

- Suppose we want to commit to a polynomial vector $(\boldsymbol{s_1}, \boldsymbol{m}) \in R_q^{m_1+l}$ where $\boldsymbol{s_1}$ has small norm (but not necessarily $\boldsymbol{m}$).

- The ABDLOP commitment under randomness $\boldsymbol{s}_2 \in R_q^{m_2}$ is defined as:

$$\begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{s_1} + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \end{bmatrix} \boldsymbol{s_2} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{m} \end{bmatrix}.$$

If $l = 0$ then ABDLOP = Ajtai commitment.

If $m_1 = 0$ then ABDLOP = BDLOP commitment.

ABDLOP commitment (= [Ajt96] + [BDLOP18])

- Suppose we want to commit to a polynomial vector $(\boldsymbol{s_1}, \boldsymbol{m}) \in R_q^{m_1+l}$ where $\boldsymbol{s_1}$ has small norm (but not necessarily $\boldsymbol{m}$).

- The ABDLOP commitment under randomness $\boldsymbol{s}_2 \in R_q^{m_2}$ is defined as:

$$\begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{s_1} + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \end{bmatrix} \boldsymbol{s_2} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{m} \end{bmatrix}.$$

Security:

Breaking binding implies finding a MSIS solution to $[\boldsymbol{A}_1 \;\; \boldsymbol{A}_2]$.

ABDLOP commitment (= [Ajt96] + [BDLOP18])

- Suppose we want to commit to a polynomial vector $(\boldsymbol{s_1}, \boldsymbol{m}) \in R_q^{m_1+l}$ where $\boldsymbol{s_1}$ has small norm (but not necessarily $\boldsymbol{m}$).

- The ABDLOP commitment under randomness $\boldsymbol{s}_2 \in R_q^{m_2}$ is defined as:

$$\begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{s_1} + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \end{bmatrix} \boldsymbol{s_2} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{m} \end{bmatrix}.$$

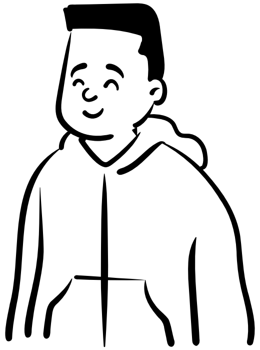
Security:

Hiding follows from MLWE since $\begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \end{bmatrix} \boldsymbol{s_2}$ looks uniformly random (for long enough randomness)

ABDLOP opening proof

$$\begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{s_1} + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \end{bmatrix} \boldsymbol{s_2} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{m} \end{bmatrix}$$ and $\boldsymbol{s}_1, \boldsymbol{s}_2$ have small coefficients

$(\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{B}, \boldsymbol{t}_A, \boldsymbol{t}_B), (\boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{m})$



$(\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{B}, \boldsymbol{t}_A, \boldsymbol{t}_B)$

$\boldsymbol{y}_i \leftarrow D^{m_i}$

$\boldsymbol{w} = \boldsymbol{A}_1\boldsymbol{y}_1 + \boldsymbol{A}_2\boldsymbol{y}_2$ $\xrightarrow{\boldsymbol{w}}$

$\xleftarrow{c}$ $c \leftarrow \boldsymbol{\mathcal{C}}$

$\boldsymbol{z}_i = \boldsymbol{y}_i + c\boldsymbol{s}_i$ $\xrightarrow{\boldsymbol{z}_1, \boldsymbol{z}_2}$

Check: i) $\boldsymbol{z}_1, \boldsymbol{z}_2$ are small
ii) $\boldsymbol{A}_1\boldsymbol{z}_1 + \boldsymbol{A}_2\boldsymbol{z}_2 = \boldsymbol{w} + c\boldsymbol{t}_A$

How many people are still following? ☺

Quadratic equations

$$\begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{s}_1 + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \end{bmatrix} \boldsymbol{s}_2 + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{m} \end{bmatrix}$$

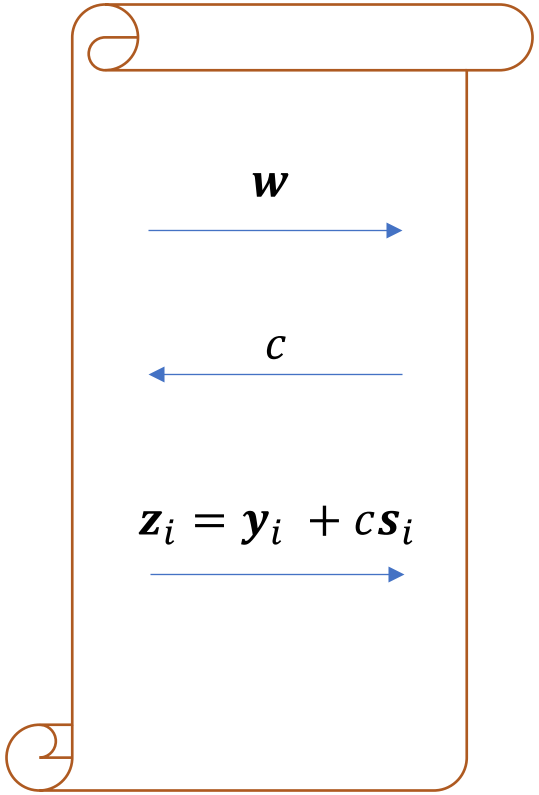
- Suppose we want to prove $\boldsymbol{s}_1^T \boldsymbol{s}_1 + \boldsymbol{m}^T \boldsymbol{m} = \boldsymbol{0}$.

Quadratic equations

$$\begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{s}_1 + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \end{bmatrix} \boldsymbol{s}_2 + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{m} \end{bmatrix}$$

- Suppose we want to prove $\boldsymbol{s}_1^T \boldsymbol{s}_1 + \boldsymbol{m}^T \boldsymbol{m} = \boldsymbol{0}$.

ABDLOP opening proof

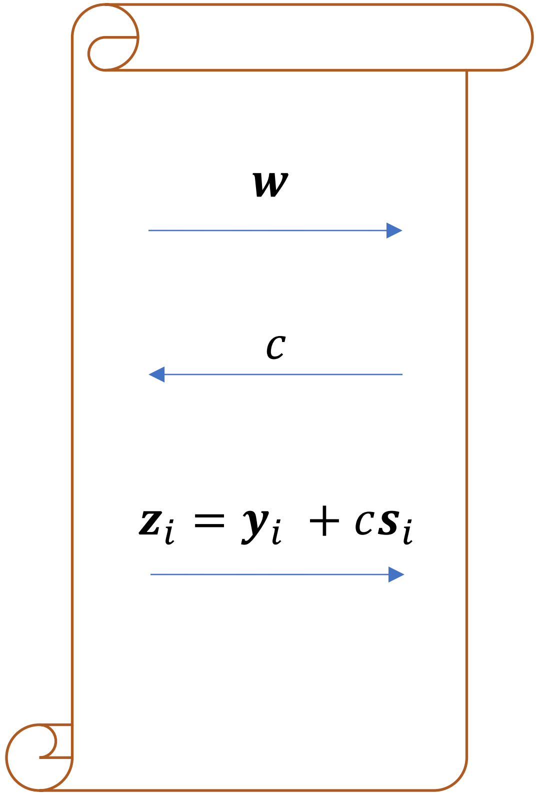


Quadratic equations

$$\begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{s}_1 + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \end{bmatrix} \boldsymbol{s}_2 + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{m} \end{bmatrix}$$

- Suppose we want to prove $\boldsymbol{s}_1^T \boldsymbol{s}_1 + \boldsymbol{m}^T \boldsymbol{m} = \boldsymbol{0}$.

ABDLOP opening proof



Note that the verifier can compute

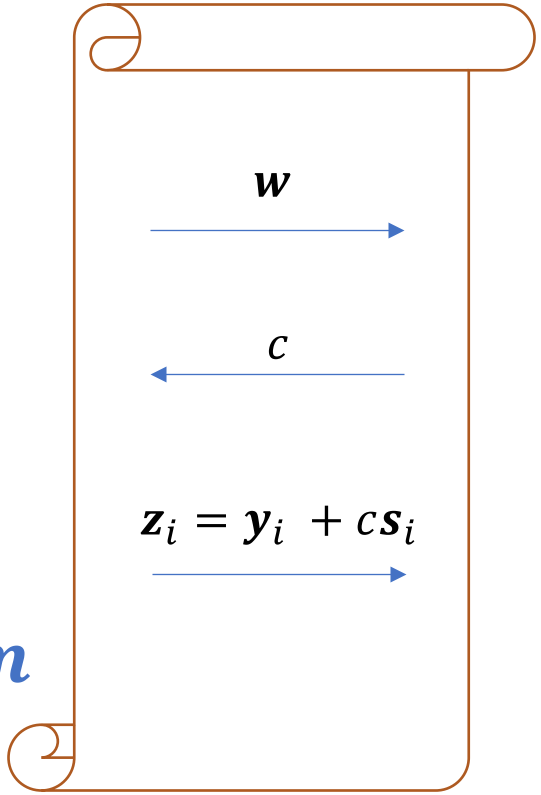
$$\boldsymbol{z}_1^T \boldsymbol{z}_1 = \boldsymbol{y}_1^T \boldsymbol{y}_1 + 2c\boldsymbol{y}_1^T \boldsymbol{s}_1 + c^2 \boldsymbol{s}_1^T \boldsymbol{s}_1$$

Quadratic equations

$$\begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_B \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \end{bmatrix} \mathbf{s}_1 + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B} \end{bmatrix} \mathbf{s}_2 + \begin{bmatrix} \mathbf{0} \\ \mathbf{m} \end{bmatrix}$$

- Suppose we want to prove $\mathbf{s}_1^T \mathbf{s}_1 + \mathbf{m}^T \mathbf{m} = \mathbf{0}$.

ABDLOP opening proof



Note that the verifier can compute

$$\mathbf{z}_1^T \mathbf{z}_1 = \mathbf{y}_1^T \mathbf{y}_1 + 2c\mathbf{y}_1^T \mathbf{s}_1 + c^2 \mathbf{s}_1^T \mathbf{s}_1$$

Moreover, we know $c\mathbf{t}_B - \mathbf{B}\mathbf{z}_2 = -\mathbf{B}\mathbf{y}_2 + c\mathbf{m}$.

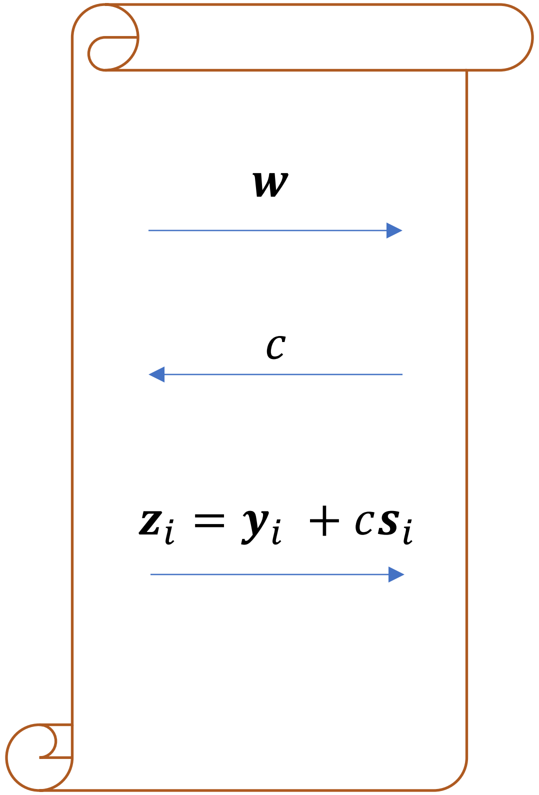
Thus:

$$(c\mathbf{t}_B - \mathbf{B}\mathbf{z}_2)^T (c\mathbf{t}_B - \mathbf{B}\mathbf{z}_2) = (\mathbf{B}\mathbf{y}_2)^T \mathbf{B}\mathbf{y}_2 - 2c(\mathbf{B}\mathbf{y}_2)^T \mathbf{m} + c^2 \mathbf{m}^T \mathbf{m}$$

Quadratic equations

$$\begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{s}_1 + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \end{bmatrix} \boldsymbol{s}_2 + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{m} \end{bmatrix}$$

ABDLOP opening proof



- Suppose we want to prove $\boldsymbol{s}_1^T \boldsymbol{s}_1 + \boldsymbol{m}^T \boldsymbol{m} = \boldsymbol{0}$.

Then,

$$\boldsymbol{z}_1^T \boldsymbol{z}_1 + (c\boldsymbol{t}_B - \boldsymbol{B}\boldsymbol{z}_2)^T (c\boldsymbol{t}_B - \boldsymbol{B}\boldsymbol{z}_2)$$
$$= g_0 + c g_1 + c^2 (\boldsymbol{s}_1^T \boldsymbol{s}_1 + \boldsymbol{m}^T \boldsymbol{m})$$

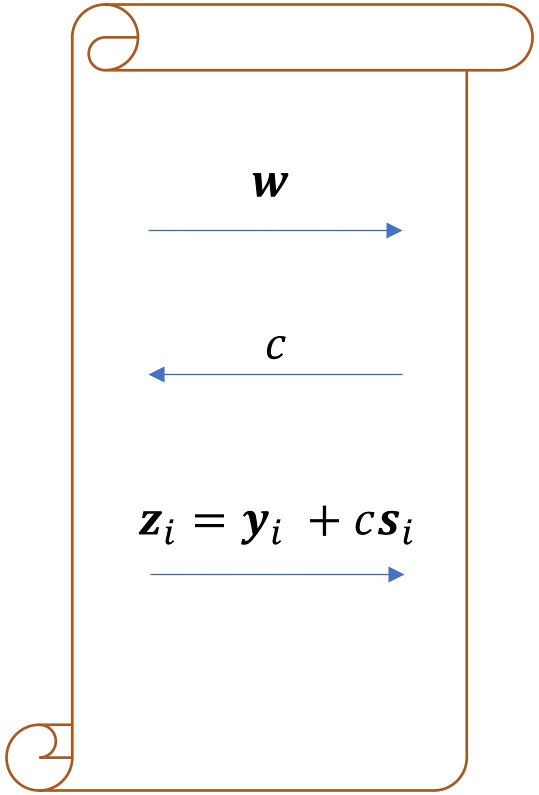
where

$$g_0 = \boldsymbol{y}_1^T \boldsymbol{y}_1 + (\boldsymbol{B}\boldsymbol{y}_2)^T \boldsymbol{B}\boldsymbol{y}_2$$
$$g_1 = 2\boldsymbol{y}_1^T \boldsymbol{s}_1 - 2(\boldsymbol{B}\boldsymbol{y}_2)^T \boldsymbol{m}.$$

Quadratic equations

$$\begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \\ t_1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{s}_1 + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \\ \boldsymbol{b}_1^T \end{bmatrix} \boldsymbol{s}_2 + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{m} \\ g_1 \end{bmatrix}$$

ABDLOP opening proof



- Suppose we want to prove $\boldsymbol{s}_1^T \boldsymbol{s}_1 + \boldsymbol{m}^T \boldsymbol{m} = \boldsymbol{0}$.

Then,

$$\boldsymbol{z}_1^T \boldsymbol{z}_1 + (c\boldsymbol{t}_B - \boldsymbol{B}\boldsymbol{z}_2)^T (c\boldsymbol{t}_B - \boldsymbol{B}\boldsymbol{z}_2)$$
$$= g_0 + c g_1 + c^2 (\boldsymbol{s}_1^T \boldsymbol{s}_1 + \boldsymbol{m}^T \boldsymbol{m})$$

where

$$g_0 = \boldsymbol{y}_1^T \boldsymbol{y}_1 + (\boldsymbol{B}\boldsymbol{y}_2)^T \boldsymbol{B}\boldsymbol{y}_2$$
$$g_1 = 2\boldsymbol{y}_1^T \boldsymbol{s}_1 - 2(\boldsymbol{B}\boldsymbol{y}_2)^T \boldsymbol{m}.$$

Hence, commit to $t_1 := \boldsymbol{b}_0^T \boldsymbol{s}_2 + g_1$.

$$\begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \\ t_1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{s}_1 + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \\ \boldsymbol{b}_1^T \end{bmatrix} \boldsymbol{s}_2 + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{m} \\ g_1 \end{bmatrix}$$

Quadratic equations

ABDLOP opening proof

- Suppose we ... $\boldsymbol{m}^T\boldsymbol{m} = \boldsymbol{0}$.

Appending the ABDLOP commitment

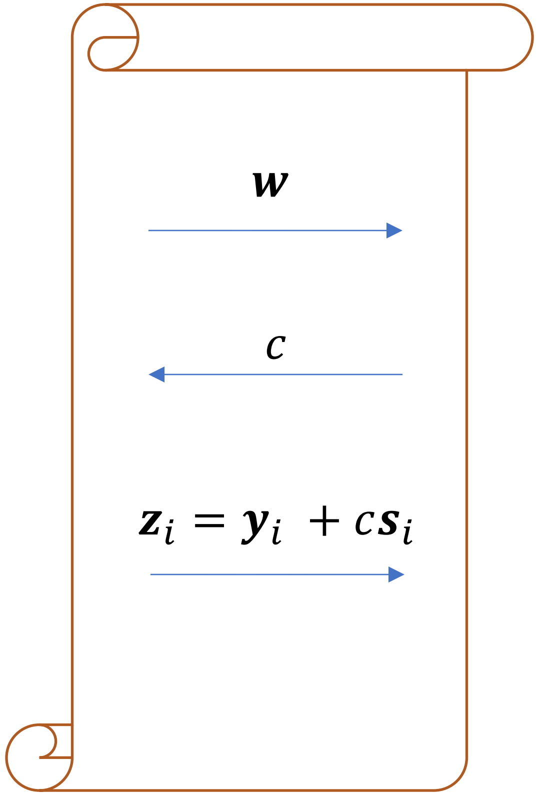
Then,

$$\boldsymbol{z}_1^T\boldsymbol{z}_1 + (c\boldsymbol{t}_B - \boldsymbol{B}\boldsymbol{z}_2)^T(c\boldsymbol{t}_B - \boldsymbol{B}\boldsymbol{z}_2)$$
$$= g_0 + cg_1 + c^2(\boldsymbol{s}_1^T\boldsymbol{s}_1 + \boldsymbol{m}^T\boldsymbol{m})$$

where

$$g_0 = \boldsymbol{y}_1^T\boldsymbol{y}_1 + (\boldsymbol{B}\boldsymbol{y}_2)^T\boldsymbol{B}\boldsymbol{y}_2$$
$$g_1 = 2\boldsymbol{y}_1^T\boldsymbol{s}_1 - 2(\boldsymbol{B}\boldsymbol{y}_2)^T\boldsymbol{m}.$$

Hence, commit to $t_1 := \boldsymbol{b}_0^T\boldsymbol{s}_2 + g_1$.



Quadratic equations

$$\begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \\ t_1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{s}_1 + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \\ \boldsymbol{b}_1^T \end{bmatrix} \boldsymbol{s}_2 + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{m} \\ g_1 \end{bmatrix}$$

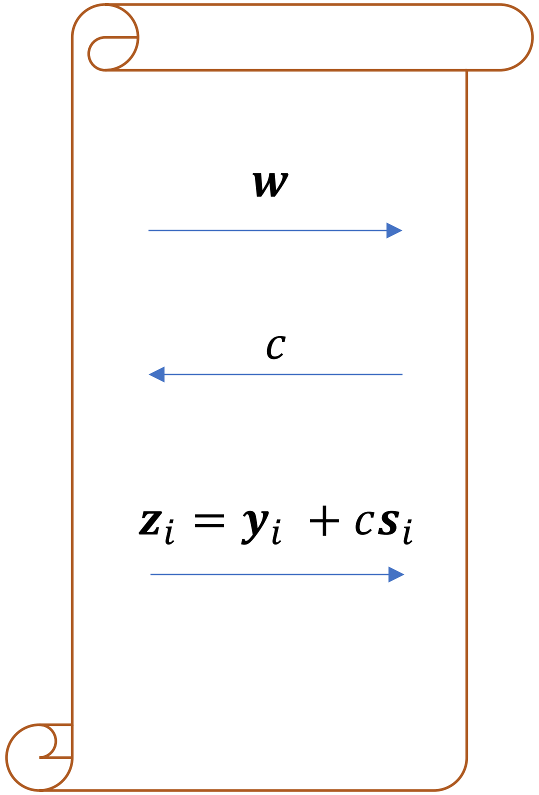
- Suppose we want to prove $\boldsymbol{s}_1^T \boldsymbol{s}_1 + \boldsymbol{m}^T \boldsymbol{m} = \boldsymbol{0}$.
- $\boldsymbol{z}_1^T \boldsymbol{z}_1 + (c\boldsymbol{t}_B - \boldsymbol{B}\boldsymbol{z}_2)^T (c\boldsymbol{t}_B - \boldsymbol{B}\boldsymbol{z}_2) - (ct_1 - \boldsymbol{b}_1^T \boldsymbol{z}_2)$

$$= g_0 + cg_1 - (ct_1 - \boldsymbol{b}_1^T \boldsymbol{z}_2)$$

$$= g_0 + \boldsymbol{b}_1^T \boldsymbol{y}_2$$

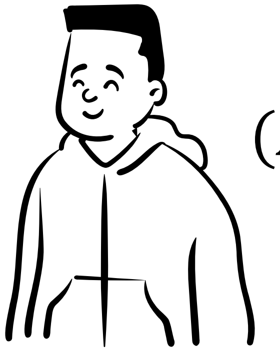
where the right-hand side does not depend on c.

ABDLOP opening proof



Proving $\boldsymbol{s}_1^T\boldsymbol{s}_1 + \boldsymbol{m}^T\boldsymbol{m} = \boldsymbol{0}$.

$(\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{B}, \boldsymbol{t}_A, \boldsymbol{t}_B), (\boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{m})$



$(\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{B}, \boldsymbol{t}_A, \boldsymbol{t}_B)$

$\boldsymbol{y}_i \leftarrow D^{m_i}$

$\boldsymbol{w} = \boldsymbol{A}_1\boldsymbol{y}_1 + \boldsymbol{A}_2\boldsymbol{y}_2$

$g_1 = 2\boldsymbol{y}_1^T\boldsymbol{s}_1 - 2(\boldsymbol{B}\boldsymbol{y}_2)^T\boldsymbol{m}$

$t_1 := \boldsymbol{b}_1^T\boldsymbol{s}_2 + g_1$

$v := \boldsymbol{y}_1^T\boldsymbol{y}_1 + (\boldsymbol{B}\boldsymbol{y}_2)^T\boldsymbol{B}\boldsymbol{y}_2 + \boldsymbol{b}_1^T\boldsymbol{y}_2$

$\boldsymbol{w}, t_1, v$ →

← c

$c \leftarrow \mathcal{C}$

$\boldsymbol{z}_1, \boldsymbol{z}_2$ →

$\boldsymbol{z}_i = \boldsymbol{y}_i + c\boldsymbol{s}_i$

Check: - $\boldsymbol{z}_1, \boldsymbol{z}_2$ are small

- $\boldsymbol{A}_1\boldsymbol{z}_1 + \boldsymbol{A}_2\boldsymbol{z}_2 = \boldsymbol{w} + c\boldsymbol{t}_A$

- and:

$\boldsymbol{z}_1^T\boldsymbol{z}_1 + (c\boldsymbol{t}_B - \boldsymbol{B}\boldsymbol{z}_2)^T(c\boldsymbol{t}_B - \boldsymbol{B}\boldsymbol{z}_2) - (ct_1 - \boldsymbol{b}_1^T\boldsymbol{z}_2) = v$

How many people are still following? ☺

Quadratic equations with automorphism

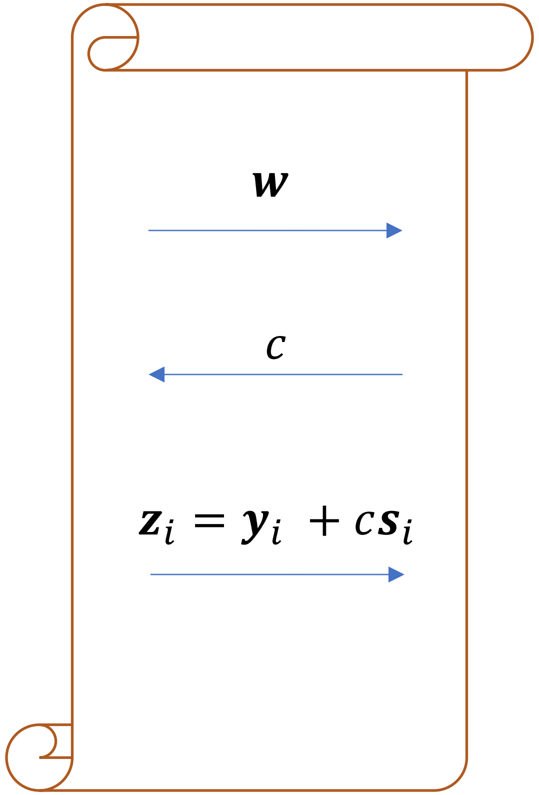
$$\begin{bmatrix} t_A \\ t_B \end{bmatrix} = \begin{bmatrix} A_1 \\ \mathbf{0} \end{bmatrix} \boldsymbol{s}_1 + \begin{bmatrix} A_2 \\ \boldsymbol{B} \end{bmatrix} \boldsymbol{s}_2 + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{m} \end{bmatrix}$$

- Suppose we want to mix quadratic equations with automorphisms, e.g.

$$\boldsymbol{s}_1^T \sigma(\boldsymbol{s}_1) + \boldsymbol{m}^T \sigma(\boldsymbol{m}) = \mathbf{0}.$$

If we assume that each challenge $c \in C$ is stable under the σ automorphism, then one can prove the statement as before!

ABDLOP opening proof



Quadratic equations with automorphism

$$\begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{s}_1 + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \end{bmatrix} \boldsymbol{s}_2 + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{m} \end{bmatrix}$$

- Suppose we want to mix quadratic equations with automorphisms, e.g.

$$\boldsymbol{s}_1^T \sigma(\boldsymbol{s}_1) + \boldsymbol{m}^T \sigma(\boldsymbol{m}) = \boldsymbol{0}.$$

Then,

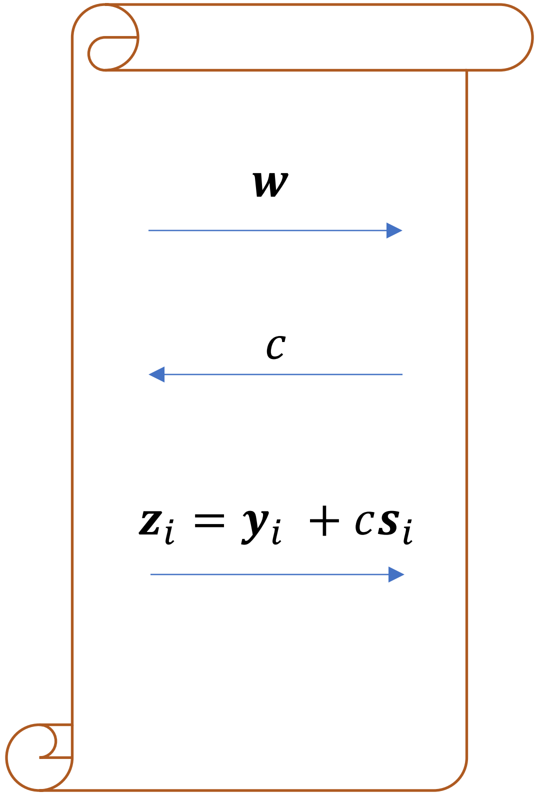
$$\boldsymbol{z}_1^T \sigma(\boldsymbol{z}_1) + (c\boldsymbol{t}_B - \boldsymbol{B}\boldsymbol{z}_2)^T \sigma(c\boldsymbol{t}_B - \boldsymbol{B}\boldsymbol{z}_2) = g_0 + c g_1 + c^2(\boldsymbol{s}_1^T \sigma(\boldsymbol{s}_1) + \boldsymbol{m}^T \boldsymbol{\sigma}(\boldsymbol{m}))$$

where

$$g_0 = \boldsymbol{y}_1^T \sigma(\boldsymbol{y}_1) + (\boldsymbol{B}\boldsymbol{y}_2)^T \sigma(\boldsymbol{B}\boldsymbol{y}_2)$$

$$g_1 = \boldsymbol{y}_1^T \sigma(\boldsymbol{s}_1) + \sigma(\boldsymbol{y}_1^T)\boldsymbol{s}_1 - \sigma(\boldsymbol{B}\boldsymbol{y}_2)^T \boldsymbol{m} - (\boldsymbol{B}\boldsymbol{y}_2)^T \sigma(\boldsymbol{m}).$$

ABDLOP opening proof



Quadratic equations with automorphism

$$\begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{s}_1 + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \end{bmatrix} \boldsymbol{s}_2 + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{m} \end{bmatrix}$$

- Suppose we want to mix quadratic equations with automorphisms, e.g.

$$\boldsymbol{s}_1^T \sigma(\boldsymbol{s}_1) + \boldsymbol{m}^T \sigma(\boldsymbol{m})$$

Then,

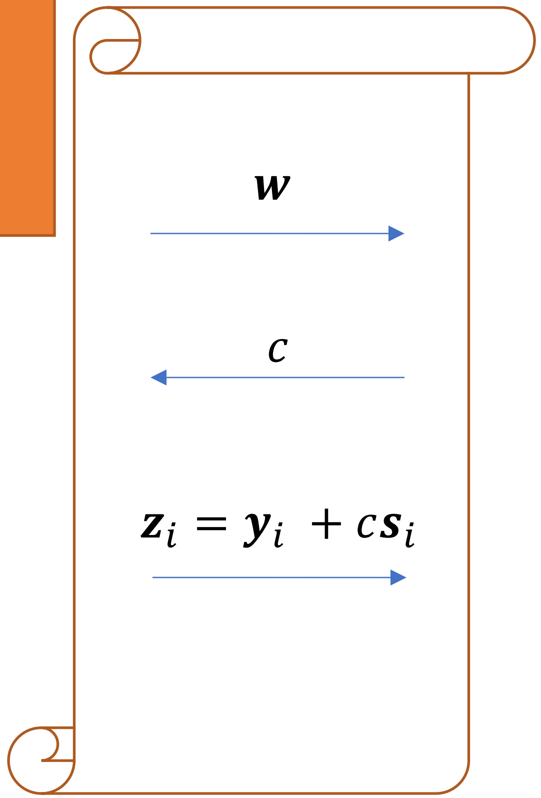
$$\boldsymbol{z}_1^T \sigma(\boldsymbol{z}_1) + (c\boldsymbol{t}_B - \boldsymbol{B}\boldsymbol{z}_2)^T \sigma(c\boldsymbol{t}_B - \boldsymbol{B}\boldsymbol{z}_2)$$
$$= g_0 + c g_1 + c^2 (\boldsymbol{s}_1^T \sigma(\boldsymbol{s}_1) + \boldsymbol{m}^T \boldsymbol{\sigma}(\boldsymbol{m}))$$

We assumed $\sigma(c) = c$.

where

$$g_0 = \boldsymbol{y}_1^T \sigma(\boldsymbol{y}_1) + (\boldsymbol{B}\boldsymbol{y}_2)^T \sigma(\boldsymbol{B}\boldsymbol{y}_2)$$
$$g_1 = \boldsymbol{y}_1^T \sigma(\boldsymbol{s}_1) + \sigma(\boldsymbol{y}_1^T) \boldsymbol{s}_1 - \sigma(\boldsymbol{B}\boldsymbol{y}_2)^T \boldsymbol{m} - (\boldsymbol{B}\boldsymbol{y}_2)^T \sigma(\boldsymbol{m}).$$

ABDLOP opening proof



Challenge space

- We need exponentially large challenge space C.
- We want $\sigma(c) = c$ for any $c \in C$.
- We want the difference of any distinct $c, c' \in C$ to be invertible over R_q.

Challenge space

- We need exponentially large challenge space C.
- We want $\sigma(c) = c$ for any $c \in C$.
- We want the difference of any distinct $c, c' \in C$ to be invertible over R_q.

Let us pick:

$$C = \{c_0 + c_1 X + \cdots + c_{\frac{d}{2}-1} X^{\frac{d}{2}-1} - c_{\frac{d}{2}-1} X^{\frac{d}{2}+1} - \cdots - c_1 X^{d-1} : c_i \in [-\kappa, \kappa]\}.$$

Challenge space

- We need exponentially large challenge space C.
- We want $\sigma(c) = c$ for any $c \in C$.
- We want the difference of any distinct $c, c' \in C$ to be invertible over R_q.

Let us pick:

$$C = \{c_0 + c_1 X + \cdots + c_{\frac{d}{2}-1} X^{\frac{d}{2}-1} - c_{\frac{d}{2}-1} X^{\frac{d}{2}+1} - \cdots - c_1 X^{d-1} : c_i \in [-\kappa, \kappa]\}.$$

$|C| = (2\kappa + 1)^{d/2}$.

Challenge space

- We need exponentially large challenge space C.
- We want $\sigma(c) = c$ for any $c \in C$.
- We want the difference of any distinct $c, c' \in C$ to be invertible over R_q.

Let us pick:

$$C = \{c_0 + c_1 X + \cdots + c_{\frac{d}{2}-1} X^{\frac{d}{2}-1} - c_{\frac{d}{2}-1} X^{\frac{d}{2}+1} - \cdots - c_1 X^{d-1} : c_i \in [-\kappa, \kappa]\}.$$

Challenge space

- We need exponentially large challenge space C.
- We want $\sigma(c) = c$ for any $c \in C$.
- We want the difference of any distinct $c, c' \in C$ to be invertible over R_q.

Let us pick:

$$C = \{c_0 + c_1 X + \cdots + c_{\frac{d}{2}-1} X^{\frac{d}{2}-1} - c_{\frac{d}{2}-1} X^{\frac{d}{2}+1} - \cdots - c_1 X^{d-1} : c_i \in [-\kappa, \kappa]\}.$$

Lemma: Suppose $q \equiv 5 \ (mod\ 8)$. If $\sigma_{-1}(c) = c$ and c is non-zero, then c is invertible over R_q.

How many people are still following? ☺

Soundness analysis

- Since the verification equation is a ``quadratic equation'', we actually need to extract three transcripts $(\boldsymbol{w}, c, \boldsymbol{z}), (\boldsymbol{w}, c', \boldsymbol{z}'), (\boldsymbol{w}, c'', \boldsymbol{z}'')$ with pairwise different $c, c', c'' \in C$.

- (Relaxed) Binding from SIS

- Interpolation approach to prove quadratic equations

We only extract $(\boldsymbol{s}_1^*, \boldsymbol{s}_2^*, c^*)$ s.t. $\boldsymbol{A_1}\boldsymbol{s}_1^* + \boldsymbol{A_2}\boldsymbol{s}_2^* = c^*\boldsymbol{u}\ (mod\ q)$, $\boldsymbol{s}_1^*, \boldsymbol{s}_2^*, c^*$- short.

<u>Lemma:</u> Suppose there are two $(\boldsymbol{s}_1^*, \boldsymbol{s}_2^*, c^*)$ and $(\boldsymbol{s}_1', \boldsymbol{s}_2', c')$ which satisfy the above. Then, under the Module-SIS assumption,

$$\boldsymbol{s}_1 := \frac{\boldsymbol{s}_1^*}{c^*} = \frac{\boldsymbol{s}_1'}{c'} \text{ and } \boldsymbol{s}_2 := \frac{\boldsymbol{s}_2^*}{c^*} = \frac{\boldsymbol{s}_2'}{c'}$$

Candidate witness

Proof sketch:

$$\mathbf{0} = c^*c'\boldsymbol{u} - c'c^*\boldsymbol{u} = \boldsymbol{A_1}(c^*\,\boldsymbol{s}_1' - c'\boldsymbol{s}_1^*) + \boldsymbol{A_2}(c^*\,\boldsymbol{s}_2' - c'\boldsymbol{s}_2^*)$$

Short!

Soundness analysis

- Since the verification equation is a ``quadratic equation'', we actually need to extract three transcripts $(\boldsymbol{w}, c, \boldsymbol{z}), (\boldsymbol{w}, c', \boldsymbol{z}'), (\boldsymbol{w}, c'', \boldsymbol{z}'')$ with pairwise different $c, c', c'' \in C$.

- (Relaxed) Binding from SIS

- Interpolation approach to prove quadratic equations

- We extract a candidate witness $\boldsymbol{s}_i := \boldsymbol{s}_i^* / c^*$ (division of two short elements) and $\boldsymbol{m}$, s.t. $\boldsymbol{A}_1 \boldsymbol{s}_1 + \boldsymbol{A}_2 \boldsymbol{s}_2 = \boldsymbol{t}_A$ and $\boldsymbol{B} \boldsymbol{s}_2 + \boldsymbol{m} = \boldsymbol{t}_B$.

Extraction - meaning

- From the opening proof, we obtain a candidate witness $\boldsymbol{s}$, it could be large (but relaxed binding holds)

Extraction - meaning

- From the opening proof, we obtain a candidate witness $\boldsymbol{s}$, it could be large (but relaxed binding holds)

- quadratic equations/proving constant terms make sure that

$$\boldsymbol{A}\boldsymbol{s} = \boldsymbol{u} \pmod{q} \qquad \langle \boldsymbol{s}, \boldsymbol{s} - \boldsymbol{1} \rangle = 0 \ (mod\ q)$$

Extraction - meaning

- From the opening proof, we obtain a candidate witness $\boldsymbol{s}$, it could be large (but relaxed binding holds)

- quadratic equations/proving constant terms make sure that

$$\boldsymbol{A}\boldsymbol{s} = \boldsymbol{u} \pmod{q} \qquad \langle \boldsymbol{s}, \boldsymbol{s} - \mathbf{1} \rangle = 0 \ (mod\ q)$$

- Approximate range proof makes sure that $||\boldsymbol{s}|| \ll q$, and we are done.

Which d to pick - tradeoff

- We want d to be large enough, so that the challenge space is exponential-size
- We want d to be as small as possible, since sending ring elements will be costly

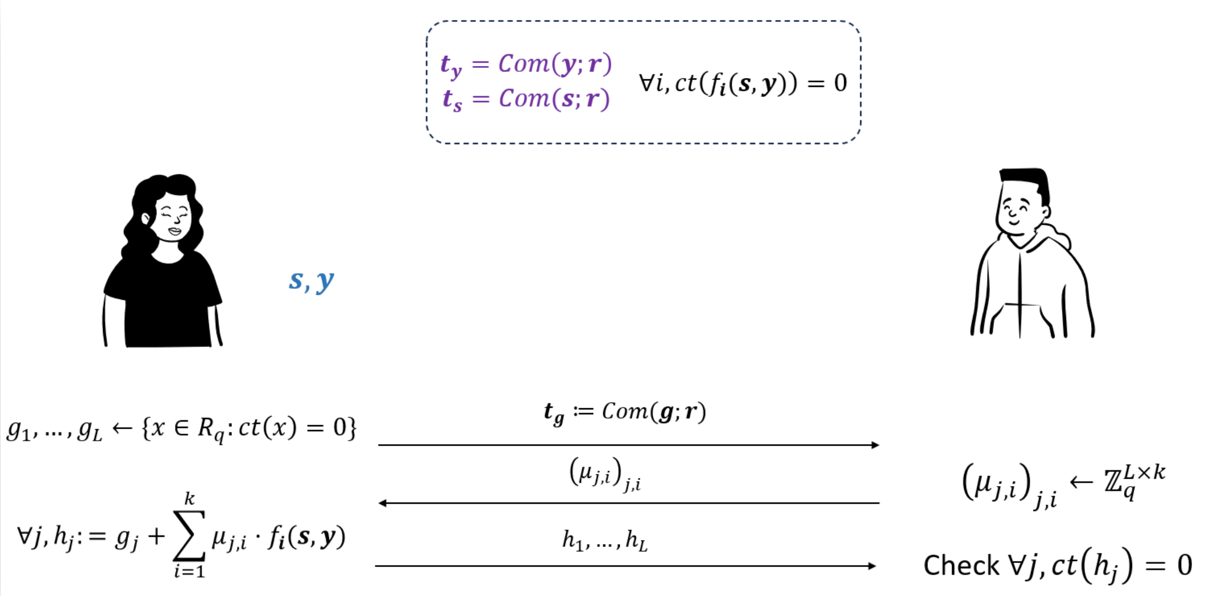
Challenge space

- We need exponentially large challenge space C.
- We want $\sigma(c) = c$ for any $c \in C$.
- We want the difference of any distinct $c, c' \in C$ to be invertible over R_q.

Let us pick:

$$C = \{c_0 + c_1 X + \cdots + c_{\frac{d}{2}-1} X^{\frac{d}{2}-1} - c_{\frac{d}{2}-1} X^{\frac{d}{2}+1} - \cdots - c_1 X^{d-1} : c_i \in [-\kappa, \kappa]\}.$$

$|C| = (2\kappa + 1)^{d/2}$.



How many people are still following? ☺

Efficiency and applications

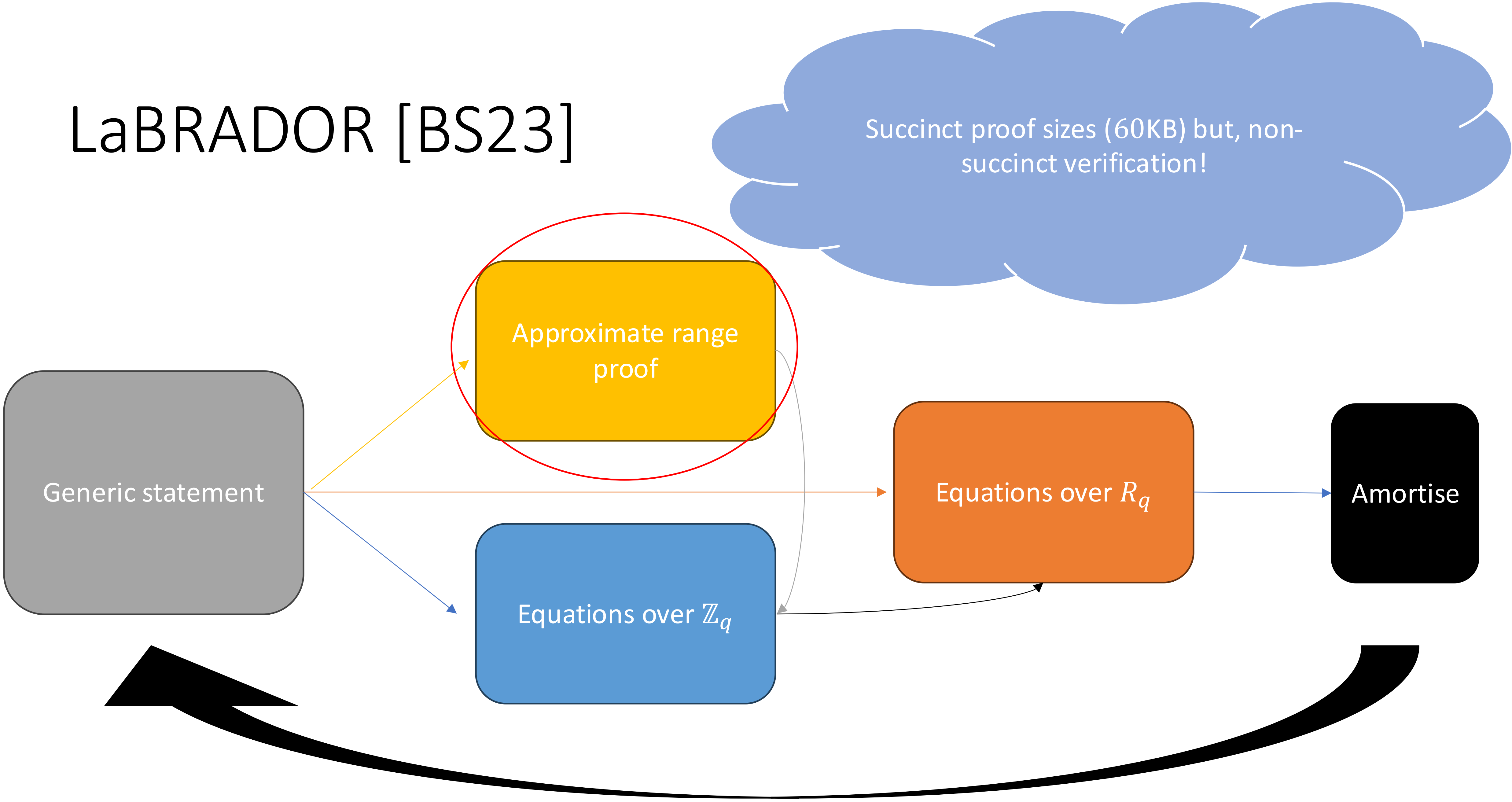
Applications

- Proving knowledge of short $\boldsymbol{s}, \boldsymbol{e}$ s.t. $\boldsymbol{A}\boldsymbol{s} + \boldsymbol{e} = \boldsymbol{u}$.

Scheme	Proof size
Stern proofs (e.g. [Ste93,LNSW13])	3MB
[Beu20]	233KB
[BLS19,YAZ+19]	384KB
Ligero [AHIV17]	157KB
Aurora [BCR+19,BCOS20]	72KB
MPC-in-the-head approach [FR23]	22-60KB
[ALS20,E**N**S20]	47KB
[L**N**S21]	33KB
[L**N**P22]	14KB

What about SNARKs?

LaBRADOR [BS23]



Approximate range proof

$\boldsymbol{s} \in \{0,1\}^m$

$||\boldsymbol{s}|| \ll q$

Linear-sized matrix $\boldsymbol{B}$

$\boldsymbol{y} \leftarrow [-\alpha, \alpha]^\lambda$
$\boldsymbol{r} \leftarrow \chi$

$\boldsymbol{t}_{\boldsymbol{y}} \coloneqq Com(\boldsymbol{y}; \boldsymbol{r}), \boldsymbol{t}_{\boldsymbol{s}} \coloneqq Com(\boldsymbol{s}; \boldsymbol{r})$ →

$\boldsymbol{B} \leftarrow \{0,1\}^{\lambda \times m}$

← $\boldsymbol{B}$

$\boldsymbol{z} = \boldsymbol{y} + \boldsymbol{B}\boldsymbol{s}$

If $||\boldsymbol{z}|| > \alpha - m$, reject

$\boldsymbol{z}$ →

Check $||\boldsymbol{z}|| \leq \alpha - m$

Prove knowledge of $\boldsymbol{y}, \boldsymbol{s}, \boldsymbol{r}$ s.t. $\boldsymbol{t}_{\boldsymbol{y}} = Com(\boldsymbol{y}; \boldsymbol{r})$, $\boldsymbol{t}_{\boldsymbol{s}} = Com(\boldsymbol{s}; \boldsymbol{r})$, $\boldsymbol{z} = \boldsymbol{y} + \boldsymbol{B}\boldsymbol{s}$ ⟺

How to achieve sublinear verification with ARP

- Use a structured tensor-type matrix $\boldsymbol{B}$ [CM**N**W24]

- Use LaBRADOR as a subroutine [**N**S24]

- Just don’t use ARP (and deal with its consequences – next talk)

¯_(ツ)_/¯

Summary

- Linear-sized efficient ``exact'' ZKP from lattices
    - ➢ Under standard assumptions: MSIS and MLWE
    - ➢ Transparent setup
    - ➢ Sizes: $\approx$ 15KB
    - ➢ Can be made non-interactive via Fiat-Shamir transformation
- ``Approximate'' proofs more efficient and have some applications

https://eprint.iacr.org/2022/284

Thank you!

References

[Ajt96] Miklós Ajtai. Generating Hard Instances of Lattice Problems (Extended Abstract).In STOC 1996.

[ACLMT22] Martin R. Albrecht, Valerio Cini, Russell W. F. Lai, Giulio Malavolta, Sri AravindaKrishnan Thyagarajan. Lattice-Based SNARKs: Publicly Verifiable, Preprocessing, and Recursively Composable. In CRYPTO 2022.

[AL21] Martin R. Albrecht and Russell W. F. Lai. Subtractive Sets over Cyclotomic Rings: Limits of Schnorr-like Arguments over Lattices. In CRYPTO 2021.

[AHIV17] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. In ACM CCS 2017.

[ACK21] A Compressed Σ-Protocol Theory for Lattices. Thomas Attema, Ronald Cramer, and Lisa Kohl. In CRYPTO 2021.

[AKSY21] Shweta Agrawal and Elena Kirshanova and Damien Stehle and Anshu Yadav. Practical, Round-Optimal Lattice-Based Blind Signatures. IACR Cryptol. ePrint Arch., 2021:1565

[BCOS20] Cecilia Boschini, Jan Camenisch, Max Ovsiankin, and Nicholas Spooner . Efficient Post-Quantum SNARKs for RSIS and RLWE and their Applications to Privacy. In PQCrypto 2020.

[BCR+19] Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P. Ward. Aurora: Transparent succinct arguments for R1CS. In EUROCRYPT 2019.

[Beu20] Ward Beullens. Sigma protocols for mq, PKP and sis, and fishy signature schemes. In EUROCRYPT 2020.

[BLNS20] Jonathan Bootle, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. A non-PCP approach to succinct quantum-safe zero-knowledge. In CRYPTO 2020.

[BLS19] Jonathan Bootle, Vadim Lyubashevsky, and Gregor Seiler. Algebraic techniques for short(er) exact lattice- based zero-knowledge proofs. In CRYPTO 2019.

[COS20] Alessandro Chiesa, Dev Ojha, and Nicholas Spooner. Fractal: Post-quantum and transparent recursive proofs from holography. In EUROCRYPT 2020.

[ISW21] Yuval Ishai and Hang Su and David J. Wu. Shorter and Faster Post-Quantum Designated-Verifier zkSNARKs from Lattices. In ACM CCS 2021.

[LN22] Vadim Lyubashevsky and Ngoc Khanh Nguyen. BLOOM: Bimodal Lattice One-Out-of-Many Proofs and Applications. In submission.

[LNP22] Vadim Lyubashevsky Ngoc Khanh Nguyen and Maxime Plancon. Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General. In CRYPTO 2022.

[LNSW13] San Ling, Khoa Nguyen, Damien Stehle, and Huaxiong Wang. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In PKC 2013.

[Lyu09] Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In ASIACRYPT 2009.

[Lyu12] Vadim Lyubashevsky. Lattice signatures without trapdoors. In EUROCRYPT 2012.

[NS22] Ngoc Khanh Nguyen and Gregor Seiler. Practical Sublinear Proofs for R1CS from Lattices. In CRYPTO 2022.

[Sta21] StarkWare Team. ethSTARK documentation. IACR Cryptol. ePrint Arch., 2021:582, 2021

[Ste93] Jacques Stern. A new identification scheme based on syndrome decoding. In CRYPTO 1993.

[YAZ+19] Rupeng Yang, Man Ho Au, Zhenfei Zhang, Qiuliang Xu, Zuoxia Yu, and William Whyte. Efficient lattice- based zero-knowledge arguments with standard soundness: Construction and applications. In CRYPTO 2019.