Lattice-Based Zero-Knowledge Proofs (I)

Ngoc Khanh Nguyen

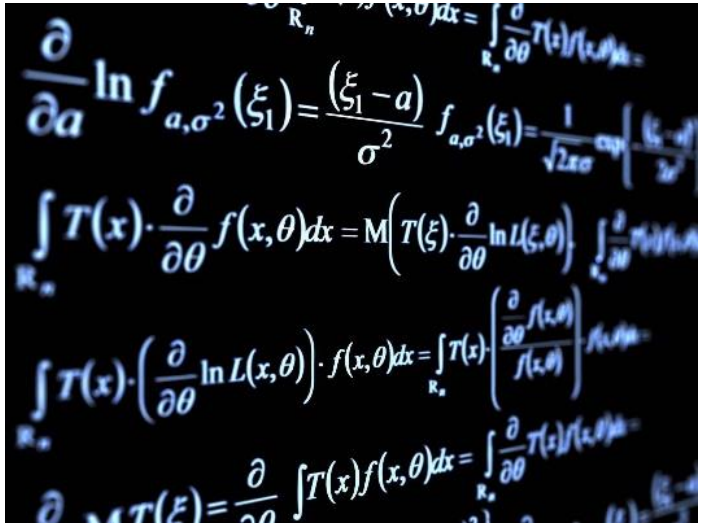
Overview

MOTIVATION ON ZERO-KNOWLEDGE PROOFS (ZKP) AND SNARKS

SIMPLE EXAMPLE

ZKP FOR LATTICE RELATED STATEMENTS

APPLICATIONS TO SIGNATURE SCHEMES (DILITHIUM)



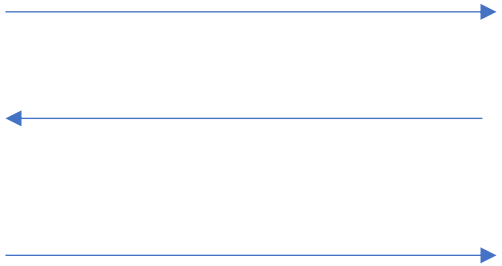
Proof

***a fact or piece of information** that **shows** that something exists or is true*

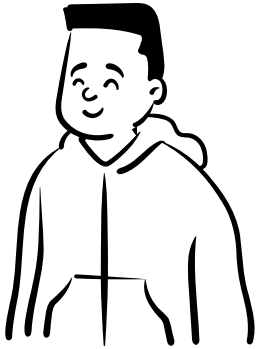
Interactive Proof

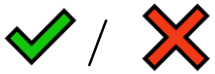
$R(x, w) = 1$

x, w



x





Completeness:
For an honest prover
the verifier accepts

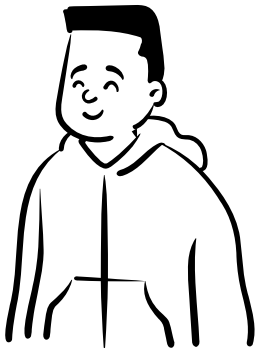
Non-Interactive Proof

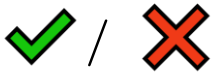
$$R(x, w) = 1$$

x, w

π

x





Completeness:
For an honest prover
the verifier accepts

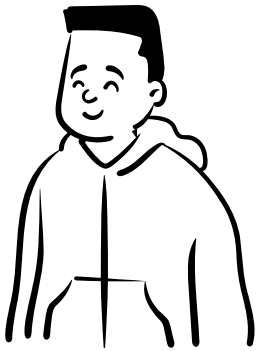
Succinct Non-Interactive Proof

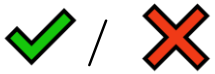
$$R(x, w) = 1$$

x, w

π

x





Completeness:
For an honest prover
the verifier accepts

Succinct: $|\pi| \ll |w|$

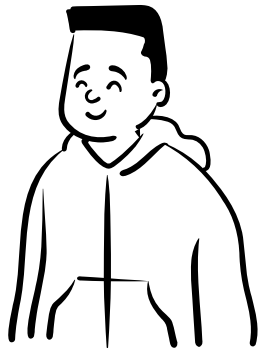
Succinct Non-Interactive Argument of Knowledge

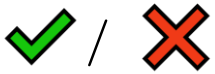
$$R(x, w) = 1$$

x, w

π

x





Succinct: $|\pi| \ll |w|$

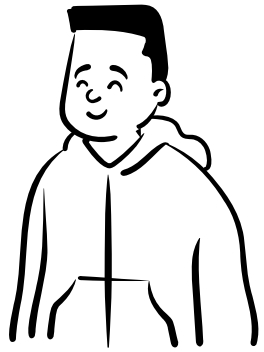
Knowledge soundness: If a prover can convince the verifier with high probability, then it ``must know w''.

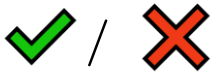
Argument: knowledge soundness holds under a computational assumption.

Statement

Prover

Verifier





Statement

Ha! I know you're **30 years old** and you live **in London**

Prover

Verifier

/

Statement

Ha! I know you're **30 years old** and you live **in London**

Prover

Verifier

/

How can we convince the security that we're over 18 without revealing sensitive data?

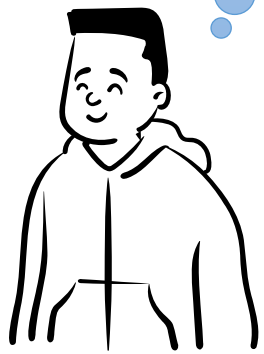
Zero-knowledge proofs

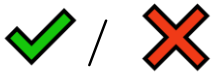
Statement

No secret information learnt apart from the statement

Prover

Verifier





APPLICATIONS OF ZERO-KNOWLEDGE PROOFS

CONFIDENTIAL TRANSACTIONS

E-VOTING

ANONYMOUS CREDENTIALS

A digital ID and personal digital wallet
for EU citizens, residents and businesses

European Commission | EU Digital Identity Wallet

Overview

MOTIVATION ON ZERO-KNOWLEDGE PROOFS (ZKP) AND SNARKS

SIMPLE EXAMPLE

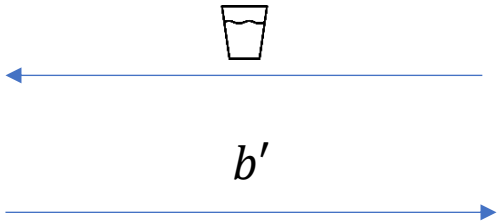
ZKP FOR LATTICE RELATED STATEMENTS

APPLICATIONS TO SIGNATURE SCHEMES (DILITHIUM)

Experiment

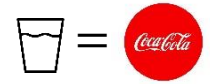
Statement: I know how to distinguish between Pepsi and Coke

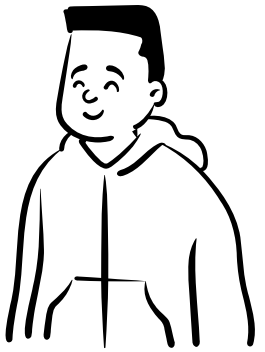
Experiment



$b \leftarrow \{0,1\}$

$If\ b = 0:$ 

$If\ b = 1:$ 



✔ if $b = b'$

✘ otherwise

- Why is it ZK?
- What's the cheating probability?

We want to build ZKP for meaningful statements

- In this presentation, we focus on **lattice-related statements**.
- They can be further adapted to proving **any NP statements**.

Overview

MOTIVATION ON ZERO-KNOWLEDGE PROOFS (ZKP) AND SNARKS

SIMPLE EXAMPLE

ZKP FOR LATTICE RELATED STATEMENTS

APPLICATIONS TO SIGNATURE SCHEMES (DILITHIUM)

Lattice-based cryptography

$$\boldsymbol{As = u}$$

Lattice-based cryptography

$$\boldsymbol{As = u}$$

Equation over ring $\mathbb{Z}_{\boldsymbol{q}}$

Lattice-based cryptography

$$\boldsymbol{As = u}$$

Vector $\boldsymbol{s}$ has *small* coefficients e.g. {-1,0,1}

Equation over ring $\mathbb{Z}_{\boldsymbol{q}}$

Lattice-based cryptography

$$\boldsymbol{As = u}$$

Let us prove knowledge of such $\boldsymbol{s}$!

Vector $\boldsymbol{s}$ has

coefficients
e.g. {-1,0,1}

Equation over ring $\mathbb{Z}_q$

Lattices vs discrete log

DLOG

- $x \in \mathbb{Z}_q$ is a secret key
- g^x is a public key
- Given a pk, it is hard to find sk (DLOG assumption)

- $g^x \cdot g^y = g^{x+y}$
- $(g^x)^c = g^{cx}$

Lattices

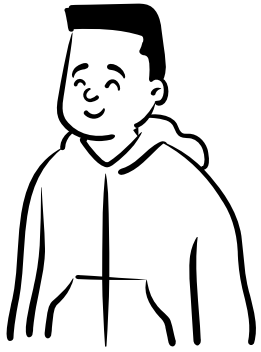
- $\boldsymbol{s} \in \mathbb{Z}_q^m$ is a secret key
- $\boldsymbol{As}$ is a public key
- Given a pk, it is hard to find sk ((I)SIS assumption)

- $\boldsymbol{As} + \boldsymbol{Ay} = \boldsymbol{A}(\boldsymbol{s} + \boldsymbol{y})$
- $c \cdot (\boldsymbol{As}) = \boldsymbol{A}(c\boldsymbol{s})$.

Schnorr ID protocol

$$R := \{(X, x) : g^x = X\}$$

g, x, X



g, X

$y \leftarrow \mathbb{Z}_q$

$Y = g^y$

$\boldsymbol{Y}$

$c \leftarrow \boldsymbol{\mathcal{C}} = \mathbb{Z}_q$

$\boldsymbol{c}$

$z = y + cx$

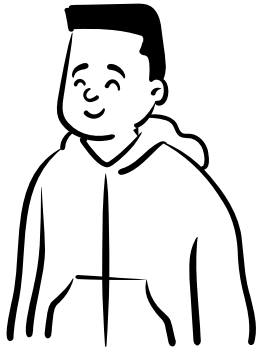
$\boldsymbol{z}$

Check $g^z = Y \cdot X^c$

Schnorr ID protocol

$$R := \{(X, x) : g^x = X\}$$

g, x, X



g, X

$y \leftarrow \mathbb{Z}_q$

$Y = g^y$

$\boldsymbol{Y}$

$c \leftarrow \boldsymbol{c} = \mathbb{Z}_q$

$\boldsymbol{c}$

$z = y + cx$

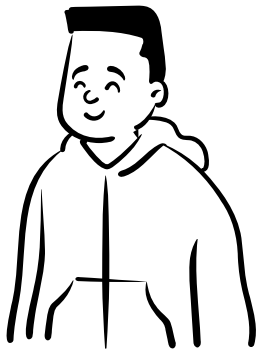
$\boldsymbol{z}$

Check $g^z = Y \cdot X^c$

Schnorr in the lattice world

$$\boldsymbol{As = u}\ (mod\ q) \text{ and } ||\boldsymbol{s}|| \leq \beta$$

$A, \boldsymbol{s}, \boldsymbol{u}$



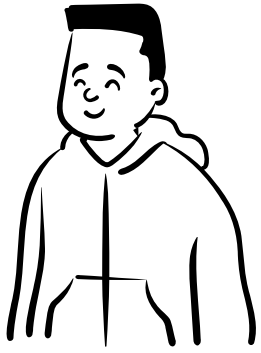
$A, \boldsymbol{u}$

Say hi and discuss with your neighbour how to translate Schnorr protocol in the lattice setting (2 minutes)!

Schnorr in the lattice world [Lyu09,Lyu12]

$$\boldsymbol{As = u}\ (mod\ q) \text{ and } \boldsymbol{||s|| \leq \beta}$$

$A, \boldsymbol{s}, \boldsymbol{u}$



$A, \boldsymbol{u}$

$\boldsymbol{y} \leftarrow \mathbb{Z}_q^m$

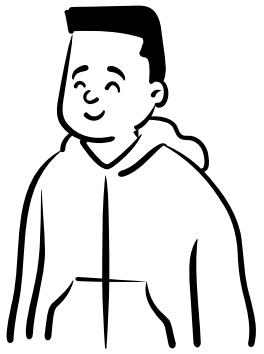
$\boldsymbol{w = Ay}$

$\boldsymbol{w}$

Schnorr in the lattice world [Lyu09,Lyu12]

$$\boldsymbol{As = u}\ (mod\ q) \text{ and } \boldsymbol{||s|| \leq \beta}$$

$A, \boldsymbol{s}, \boldsymbol{u}$



$A, \boldsymbol{u}$

$\boldsymbol{y} \leftarrow \mathbb{Z}_q^m$

$\boldsymbol{w = Ay}$

$\boldsymbol{w}$

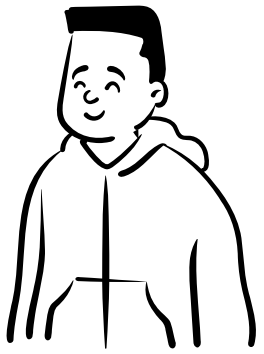
$\boldsymbol{c}$

$c \leftarrow \boldsymbol{\mathcal{C}} = \mathbb{Z}_q$

Schnorr in the lattice world [Lyu09,Lyu12]

$$\boldsymbol{As = u}\ (mod\ q) \text{ and } \boldsymbol{||s|| \leq \beta}$$

$A, \boldsymbol{s}, \boldsymbol{u}$



$A, \boldsymbol{u}$

$\boldsymbol{y} \leftarrow \mathbb{Z}_q^m$

$\boldsymbol{w = Ay}$

$\boldsymbol{w}$ →

$c \leftarrow \boldsymbol{\mathcal{C}} = \mathbb{Z}_q$

← $\boldsymbol{c}$

$\boldsymbol{z = y + cs}$

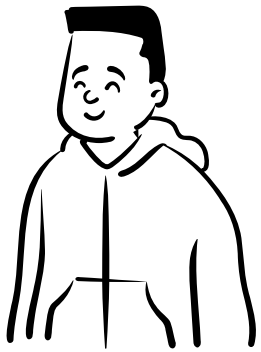
$\boldsymbol{z}$ →

Check $\boldsymbol{Az = w + cu}$

What about malicious provers? Attempt 1

$$\boldsymbol{As = u}\ (mod\ q) \text{ and } \boldsymbol{||s||} \leq \beta$$

$A, \boldsymbol{u}$



$A, \boldsymbol{u}$

We need norm bound checks!

$c \leftarrow \boldsymbol{C} = \mathbb{Z}_q$

$\boldsymbol{z}$

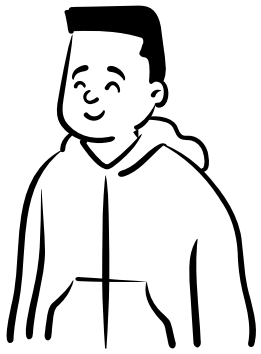
Using linear algebra,
find $\boldsymbol{z}$ s.t. $\boldsymbol{Az = w + cu}$

Check $\boldsymbol{Az = w + cu}$

Completeness is destroyed

$$\boldsymbol{As = u}\ (mod\ q) \text{ and } ||\boldsymbol{s}|| \leq \beta$$

$A, \boldsymbol{u}$



$A, \boldsymbol{u}$

$\boldsymbol{w}$

$\boldsymbol{c}$

$\boldsymbol{z}$

$c \leftarrow \boldsymbol{C} = \mathbb{Z}_q$

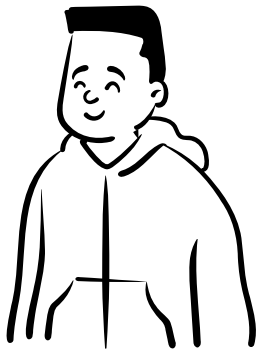
Check $\boldsymbol{Az = w + cu}$

Check $||\boldsymbol{z}|| \leq \gamma$

Fixing completeness

$$\boldsymbol{As = u}\ (mod\ q) \text{ and } ||\boldsymbol{s}|| \leq \beta$$

$A, \boldsymbol{s}, \boldsymbol{u}$



$A, \boldsymbol{u}$

short

$\boldsymbol{y} \leftarrow \mathbb{Z}_q^m$

$\boldsymbol{w = Ay}$

$\boldsymbol{w}$

$\boldsymbol{c}$

$\boldsymbol{z}$

short

$c \leftarrow \mathcal{C} = \mathbb{Z}_q$

$\boldsymbol{z = y + cs}$

short

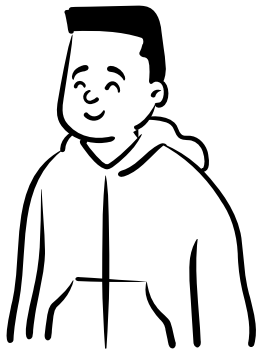
Check $\boldsymbol{Az = w + cu}$

Check $||\boldsymbol{z}|| \leq \gamma$

Fixing completeness

$$\boldsymbol{As = u}\ (mod\ q) \text{ and } ||\boldsymbol{s}|| \leq \beta$$

$A, \boldsymbol{s}, \boldsymbol{u}$



$A, \boldsymbol{u}$

short

$\boldsymbol{y} \leftarrow D^m$

$\boldsymbol{w = Ay}$

$\boldsymbol{w}$

short

$c \leftarrow \mathcal{C} \subseteq \mathbb{Z}_q$

c

$\boldsymbol{z = y + cs}$

short

$\boldsymbol{z}$

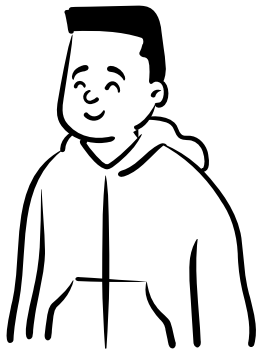
Check $\boldsymbol{Az = w + cu}$

Check $||\boldsymbol{z}|| \leq \gamma$

Fixing completeness – Attempt 1

$$\boldsymbol{As = u}\ (mod\ q) \text{ and } ||\boldsymbol{s}|| \leq \beta$$

$A, \boldsymbol{s}, \boldsymbol{u}$



$A, \boldsymbol{u}$

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

$\boldsymbol{w = Ay}$

$\boldsymbol{w}$ →

$\leftarrow c$

$\boldsymbol{z}$ →

$c \leftarrow [-\delta, \delta]$

$\boldsymbol{z = y + cs}$

short

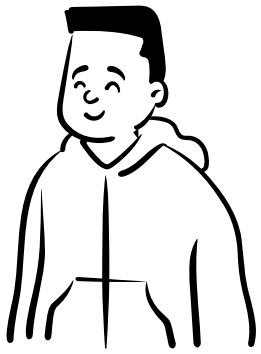
Check $\boldsymbol{Az = w + cu}$

Check $||\boldsymbol{z}|| \leq \alpha + \beta\delta$

Fixing completeness – Attempt 1

$$\boldsymbol{As = u}\ (mod\ q) \text{ and } \boldsymbol{||s|| \leq \beta}$$

$A, \boldsymbol{u}$



$A, \boldsymbol{u}$

Guess $c' \leftarrow [-\delta, \delta]$

Pick any short enough $\boldsymbol{z}$

$\boldsymbol{w = Az - c'u}$

If $\boldsymbol{c = c'}$, send $\boldsymbol{z}$

$\boldsymbol{w}$ →

← $\boldsymbol{c}$

$\boldsymbol{z}$ →

$c \leftarrow [-\delta, \delta]$

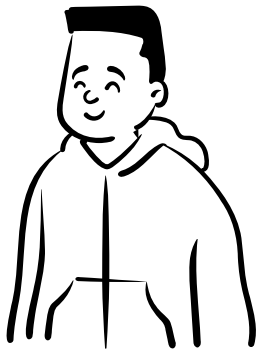
Check $\boldsymbol{Az = w + cu}$

Check $\boldsymbol{||z|| \leq \alpha + \beta\delta}$

mpleteness – Attempt 1

$$\boldsymbol{As = u}\ (mod\ q) \text{ and } \boldsymbol{||s|| \leq \beta}$$

$A, \boldsymbol{u}$



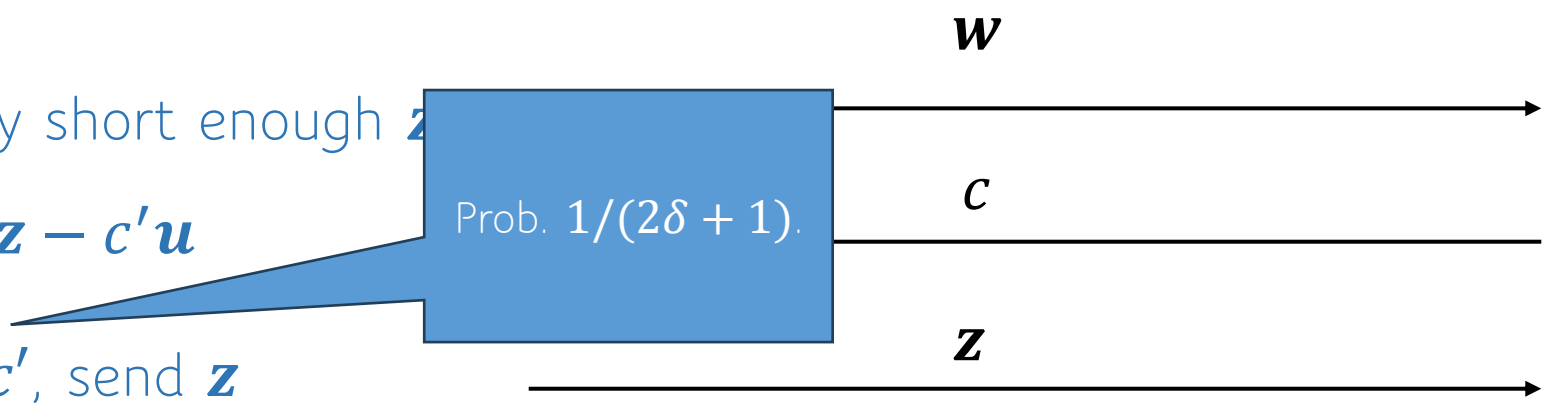
$A, \boldsymbol{u}$

Guess $c' \leftarrow [-\delta, \delta]$

Pick any short enough $\boldsymbol{z}$

$\boldsymbol{w = Az - c'u}$

If $\boldsymbol{c = c'}$, send $\boldsymbol{z}$



$c \leftarrow [-\delta, \delta]$

Check $\boldsymbol{Az = w + cu}$

Check $\boldsymbol{||z|| \leq \alpha + \beta\delta}$

Issues with soundness

- To achieve negligible (knowledge) soundness, one needs $\delta = \exp(\lambda)$.
- But since $\delta \ll q$, the modulus q has to be exponential as well!

- To overcome this limitation, we use polynomial rings (large challenge space of ``short'' elements)
- $R_q = \mathbb{Z}_q[X]/(f(X))$

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

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

Exercise:

Let $q = 7$ and $d = 4$. Compute $(X^3 + 4X) \cdot (3X^2 - 1)$ over R_q .

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

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

Exercise:

Let $q = 7$ and $d = 4$. Compute $(X^3 + 4X) \cdot (3X^2 - 1)$ over R_q .

$$(X^3 + 4X) \cdot (3X^2 - 1) = 3X^5 - X^3 + 12X^3 - 4X$$
$$= -3X - 11X^3 - 4X \ (mod\ X^4 + 1) = 3X^3\big(mod\ (X^4 + 1, 7)\big)$$

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

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

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

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

$$\left(\sum_{i=0}^{d-1} a_i X^i\right)\left(\sum_{j=0}^{d-1} b_j X^j\right) = \sum_{i,j} a_i b_j X^{i+j}$$

Hence, the k-th coefficient of ab has absolute value

$$\left|\sum_{i+j=k(mod\ d)} \pm a_i b_j\right| \leq \sqrt{\left(\sum_{i=0}^{d-1} a_i^2\right)\left(\sum_{j=0}^{d-1} b_j^2\right)} \leq \sqrt{d^2 ||a||^2 \cdot ||b||^2}.$$

Lattice-based cryptography

$$\boldsymbol{As = u}$$

Equation over ring R_q

Vector $\boldsymbol{s}$ has polynomials with

coefficients e.g. {-1,0,1}

Denote
$S_\beta := \{x \in R_q : ||x|| \leq \beta\}$

Schnorr in the polynomial ring setting

$$\boldsymbol{As = u}\ (mod\ q) \text{ and } \boldsymbol{s} \in S_\beta$$

$A, \boldsymbol{s},$

We manage to defend from simple cheating strategies. But how to prove knowledge soundness?

Size of the challenge set is $(2\delta + 1)^d$. For $d = O(\lambda)$, that's exponential!

$\boldsymbol{y} \leftarrow S_\alpha^m$

$\boldsymbol{w = Ay}$

$\boldsymbol{w}$ →

$\boldsymbol{c}$ ←

$c \leftarrow S_\delta$

$\boldsymbol{z = y + cs}$

short

$\boldsymbol{z}$ →

Check $\boldsymbol{Az = w + cu}$

Check $||\boldsymbol{z}|| \leq \alpha + d\beta\delta$

Security proof

Knowledge Soundness

- Parameter δ

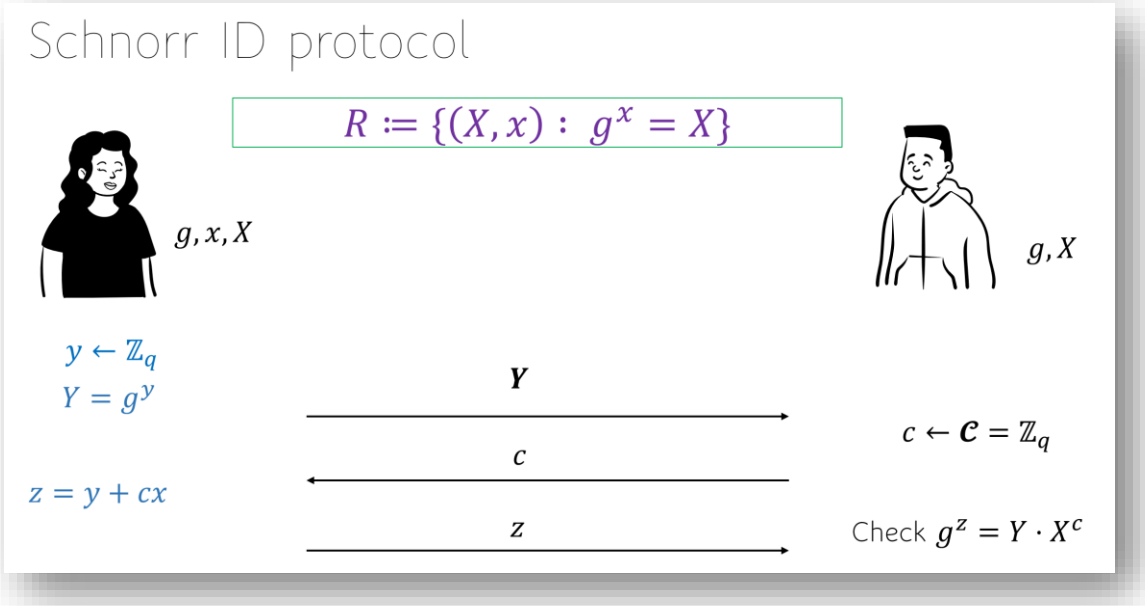
(Honest-Verifier) Zero-Knowledge

- Parameter α

Towards knowledge soundness

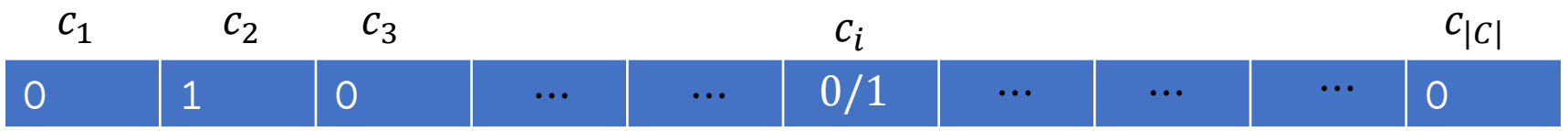
- Special-soundness: given two valid transcripts (Y, c, z) and (Y, c', z') for $c \neq c'$, one can extract x^* s.t. $(X, x^*) \in R$.

- Indeed, we know
  - $g^z = Y \cdot X^c$ and $g^{z'} = Y \cdot X^{c'}$
  - Thus $g^{\frac{z-z'}{c-c'}} = X$. Set $x^* = \frac{z-z'}{c-c'}$.



Why is special-soundness cool?

Suppose this (deterministic) cheating prover can convince the verifier with probability $\epsilon > 1/|C|$.



- Extraction strategy:
  - Sample a random $c_i \leftarrow C$
  - If the adversary fails for c_i, abort
  - Let (Y, c_i, z_i) be a valid transcript. Do:
    - sample a random $c_j \leftarrow C \backslash \{c_i\}$

    While (Y, c_j, z_j) is not a valid transcript
  - Return (Y, c_i, z_i) and (Y, c_j, z_j)

We write 1 if the adversary succeeds on challenge c_i and 0 otherwise

Why is special-soundness cool?

Suppose this (deterministic) cheating prover can convince the verifier with probability $\epsilon > 1/|C|$.

c_1	c_2	c_3			c_i				$c_{\|C\|}$
0	1	0	…	…	0/1	…	…	…	0

- Extraction strategy:
  - Sample a random $c_i \leftarrow C$
  - If the adversary fails for c_i, abort
  - Let (Y, c_i, z_i) be a valid transcript. Do:
    - sample a random $c_j \leftarrow C \backslash \{c_i\}$

    While (Y, c_j, z_j) is not a valid transcript
  - Return (Y, c_i, z_i) and (Y, c_j, z_j)

Expected running time T:

$$\mathrm{E}[T] = E[T|success] \cdot \Pr[success] + E[T|\neg success] \cdot \Pr[\neg success]$$

$$\leq \left(1 + \frac{|C| - 1}{\epsilon|C| - 1}\right) \cdot \epsilon + 1 \cdot (1 - \epsilon)$$

$$\leq 1 + \left(\frac{|C| - 1}{\epsilon|C| - 1}\right) \cdot \epsilon \leq 2$$

Why is special-soundness cool?

Suppose this (deterministic) cheating prover can convince the verifier with probability $\epsilon > 1/|C|$.

c_1	c_2	c_3			c_i				$c_{\|C\|}$
0	1	0	…	…	0/1	…	…	…	0

- Extraction strategy:
  - Sample a random $c_i \leftarrow C$
  - If the adversary fails for c_i, abort
  - Let (Y, c_i, z_i) be a valid transcript. Do:
    - sample a random $c_j \leftarrow C \backslash \{c_i\}$

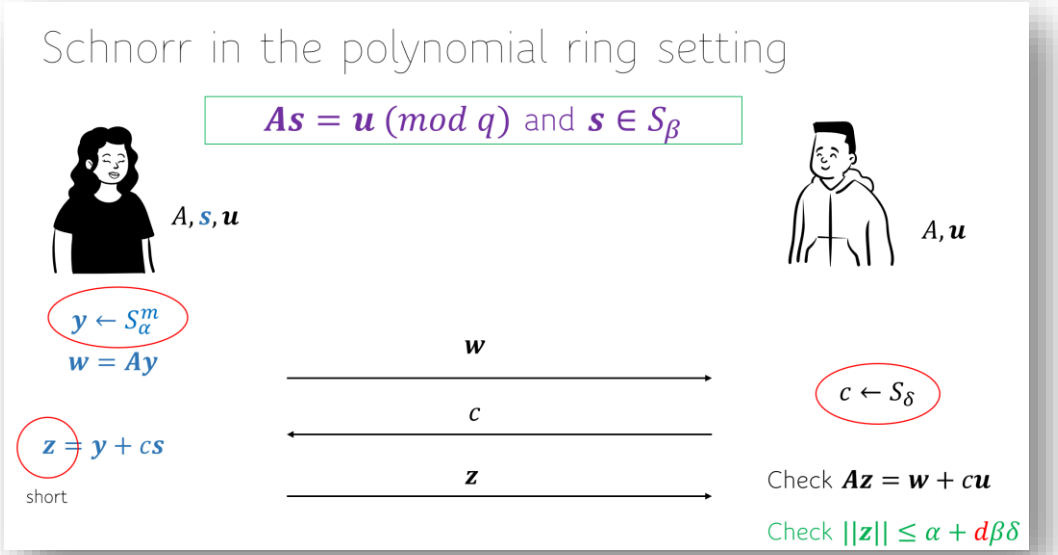
    While (Y, c_j, z_j) is not a valid transcript
  - Return (Y, c_i, z_i) and (Y, c_j, z_j)

Success probability:
If $\epsilon > 1/|C|$, there must be at least two 1s. So:

$$\Pr[E] = \epsilon$$

Special-soundness in the lattice setting

- Special-soundness: given two valid transcripts (w, c, z) and (w, c', z') for $c \neq c'$, one can extract s^* s.t. $((A, u), s^*) \in R$.



- Let's try:

$= w + cu$

The witness is NOT short!

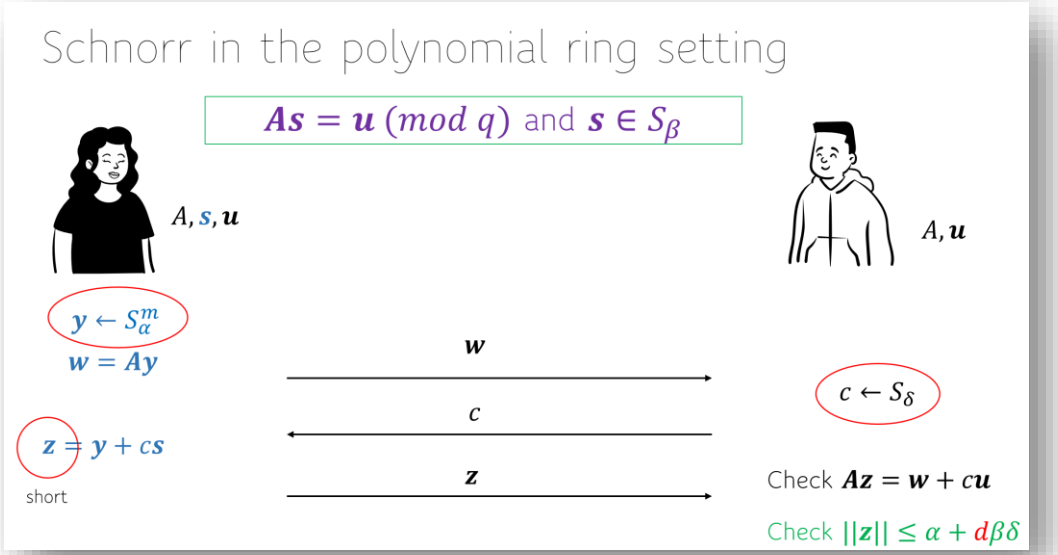
$$A\left(\frac{z - z'}{c - c'}\right) = u$$

- $Az' = w + c'u$

Division may not be defined

Special-soundness in the lattice setting

- Special-soundness: given two valid transcripts (w, c, z) and (w, c', z) for $c \neq c'$, one can extract s^* s.t. $((A, u), s^*) \in R$.



- So what do we have?
  - $A(z - z') = (c - c')u$

short short

Relaxed relation: $R^* := \{((A, u), (s^*, c^*)) : As^* = c^* u \ (mod\ q),\ s^* \in S_{2(\alpha + d\beta\delta)}, c^* \in S_{2\delta}\}$

Special soundness summary

- We don't manage to extract the exact witness $\boldsymbol{s} \in S_\beta$
- Instead, we only get $(\boldsymbol{s}^*, \boldsymbol{c}^*)$ s.t. $\boldsymbol{A}\boldsymbol{s}^* = c^*\boldsymbol{u} \ (mod\ q)$, $\boldsymbol{s}^* \in S_{2(\alpha+d\beta\delta)}$, $c^* \in S_{2\delta}$
- Actually, this relaxation is fine for signatures!

Instead, we only get $(\boldsymbol{s}^*, c^*)$ s.t. $\boldsymbol{A}\boldsymbol{s}^* = c^*\boldsymbol{u}\ (mod\ q)$, $\boldsymbol{s}^* \in S_{2(\alpha+d\beta\delta)}$, $c^* \in S_{2\delta}$

So, intuitively we want to say that our candidate witness is $\mathbf{s} \coloneqq \boldsymbol{s}^*/c^*$.

But first, is it well-defined?

For $q \equiv 5(mod\ 8)$, a non-zero element $c \in S_{\sqrt{q/2}}$ is invertible over R_q.

But $\mathbf{s}$ is not short, right?

But maybe there is still something meaningful…

Instead, we only get $(\boldsymbol{s}^*, c^*)$ s.t. $\boldsymbol{A}\boldsymbol{s}^* = c^*\boldsymbol{u}\ (mod\ q)$, $\boldsymbol{s}^* \in S_{2(\alpha+d\beta\delta)}$, $c^* \in S_{2\delta}$

Lemma: Suppose there are two $(\boldsymbol{s}_0^*, c_0^*)$ and $(\boldsymbol{s}_1^*, c_1^*)$ which satisfy the above. Then, under the Module-SIS assumption,

$$\boldsymbol{s} := \frac{\boldsymbol{s}_0^*}{c_0^*} = \frac{\boldsymbol{s}_1^*}{c_1^*}$$

Proof sketch: $\boldsymbol{0} = c_0^* c_1^* \boldsymbol{u} - c_1^* c_0^* \boldsymbol{u} = \boldsymbol{A}(c_0^* \boldsymbol{s}_1^* - c_1^* \boldsymbol{s}_0^*)$

Short!

Security proof

Knowledge Soundness

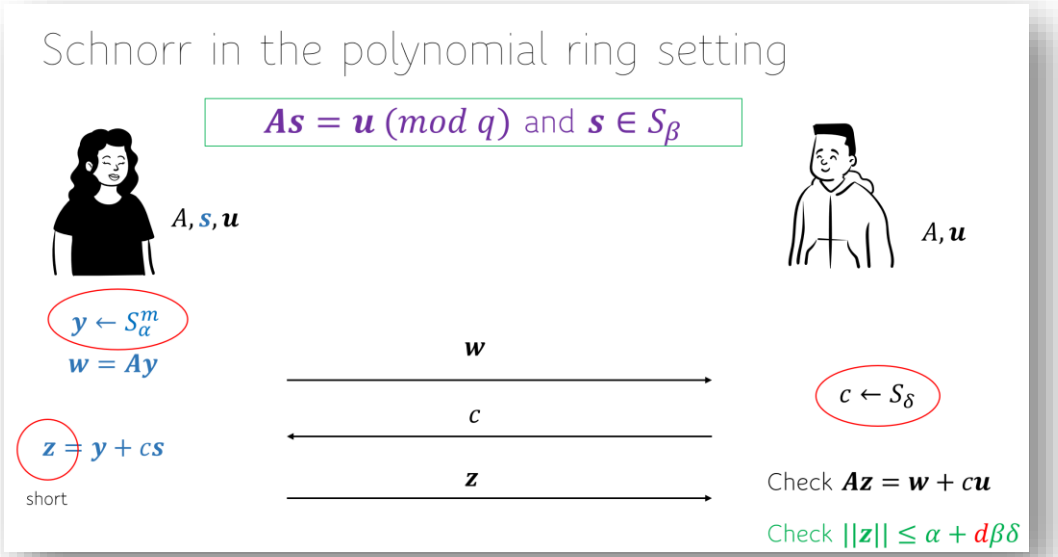
- Parameter δ

(Honest-Verifier) Zero-Knowledge

- Parameter α

Honest-verifier zero-knowledge

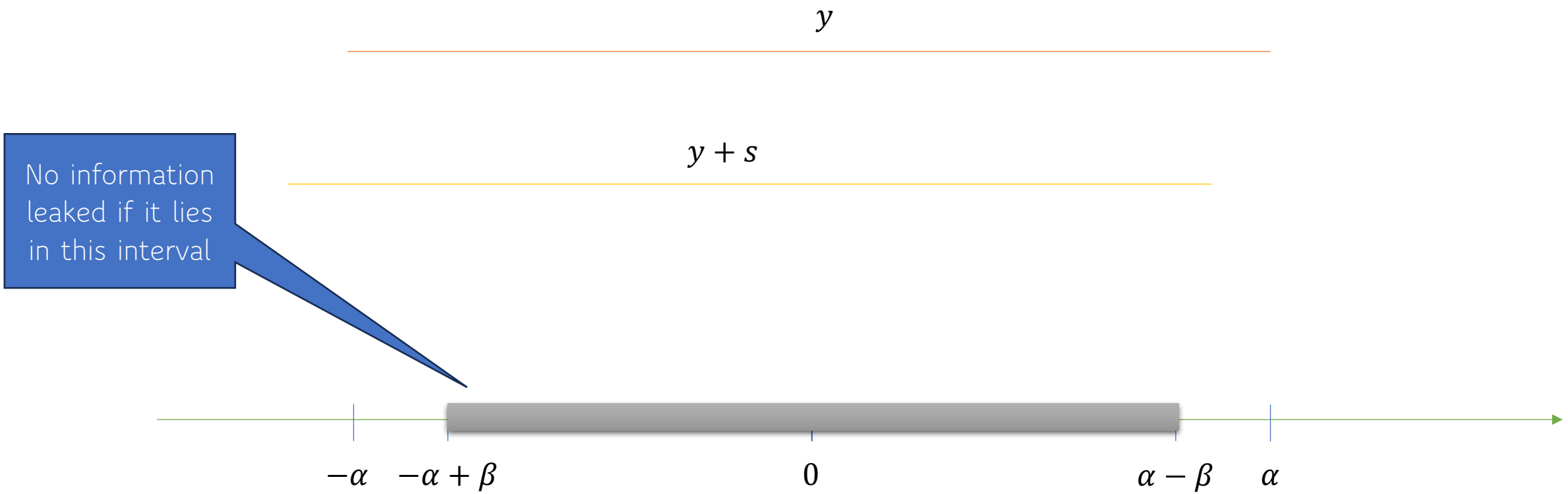
- Zero-knowledge: no information about $\boldsymbol{s}$ is leaked.
- Is it the case here?



- Simple exercise.
- Let $s \in \{-1,0,1\}$ and $y \in [-100,100]$ be hidden. I reveal $z = y + s$.
- If $z = 101$, what can we deduce?
- If $z = 100$, what can we deduce?
- If $z = 99$, what can we deduce?

Rejection sampling [Lyu09]

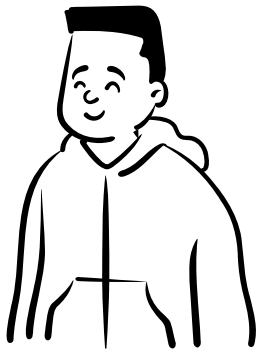
- $y \in [-\alpha, \alpha]$
- $s \in [-\beta, \beta]$



Rejection sampling

$$\boldsymbol{As = u}\ (mod\ q) \text{ and } \boldsymbol{s} \in S_\beta$$

$A, \boldsymbol{s}, \boldsymbol{u}$



$A, \boldsymbol{u}$

$\boldsymbol{y} \leftarrow S_\alpha^m$

$\boldsymbol{w} = \boldsymbol{Ay}$

$\boldsymbol{w}$ →

$c \leftarrow S_\delta$

← c

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

If $||\boldsymbol{z}|| > \alpha - d\beta\delta$, reject

$\boldsymbol{z}$ →

Check $\boldsymbol{Az} = \boldsymbol{w} + c\boldsymbol{u}$

Check $||\boldsymbol{z}|| \leq \alpha - d\beta\delta$

What's the probability of not rejecting?

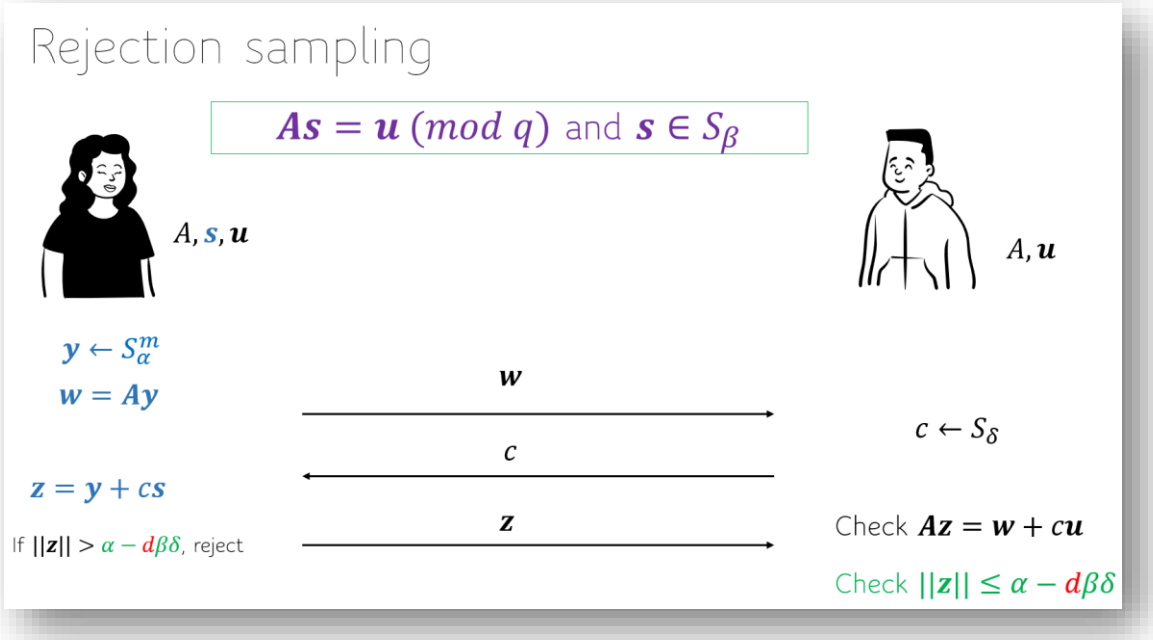
If $||\boldsymbol{y}|| \leq \alpha - 2d\beta\delta$, we're safe for sure.

$$\left(\frac{2(\alpha - 2d\beta\delta) - 1}{2\alpha - 1}\right)^{md} = \left(1 - \frac{4d\beta\delta}{2\alpha - 1}\right)^{md}$$

$$= \left(1 - \frac{2d\beta\delta}{\alpha - 1/2}\right)^{\frac{\alpha - 1/2}{2d\beta\delta}\cdot\frac{2d\beta\delta}{\alpha - 1/2}md}$$

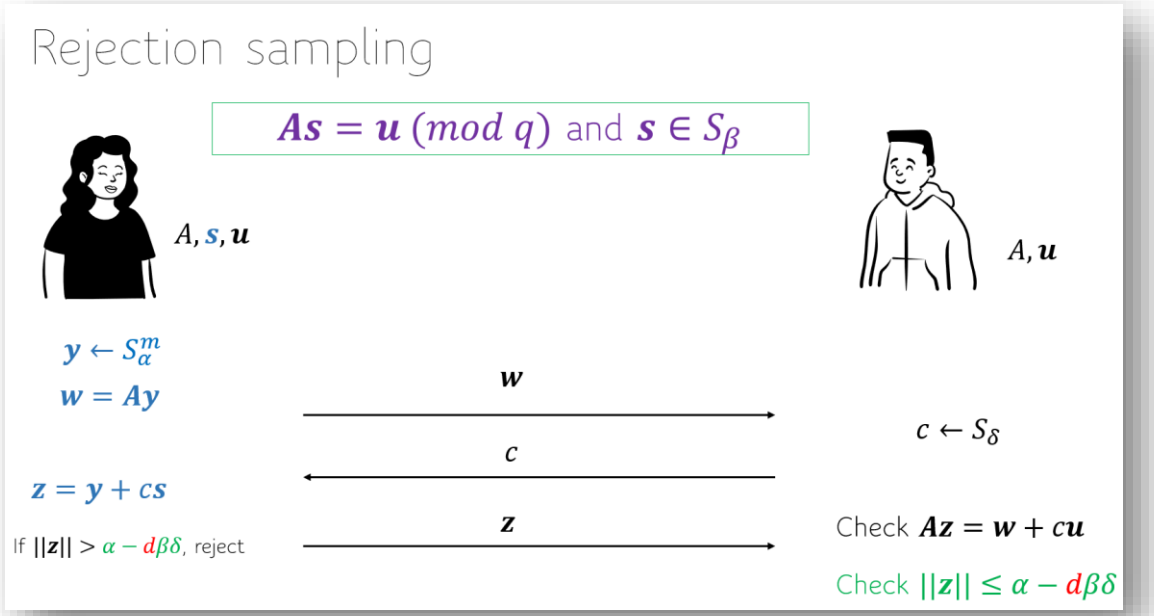
$$\approx \exp\left(-\frac{2d\beta\delta md}{\alpha}\right)$$

$$\approx \exp(-1) \qquad \text{for } \alpha \geq 2d^2\beta\delta m.$$



(Non-abort)Honest-verifier zero-knowledge

- Zero-knowledge: no information about $\boldsymbol{s}$ is leaked. In other words, one can simulate a valid (non-aborting) transcript.



Lemma: Distribution of $\boldsymbol{z}$ is uniform.

Proof: Note that for any value $\boldsymbol{x} \in \boldsymbol{S}_{\alpha - d\beta\delta}$ we have

$$\Pr[\boldsymbol{z} = \boldsymbol{x}] = \Pr[\boldsymbol{y} = \boldsymbol{x} - c\boldsymbol{s}] = \left(\frac{1}{2\alpha - 1}\right)^{md}.$$

We simulate the valid non-aborting transcript as follows.

1. $\boldsymbol{z} \leftarrow [\alpha - d\beta\delta, \alpha + d\beta\delta]$
2. $c \leftarrow S_\delta$
3. $\boldsymbol{w} := \boldsymbol{Az} - c\boldsymbol{u}$.
4. Output $(\boldsymbol{w}, c, \boldsymbol{z})$.

Security proof

Knowledge Soundness

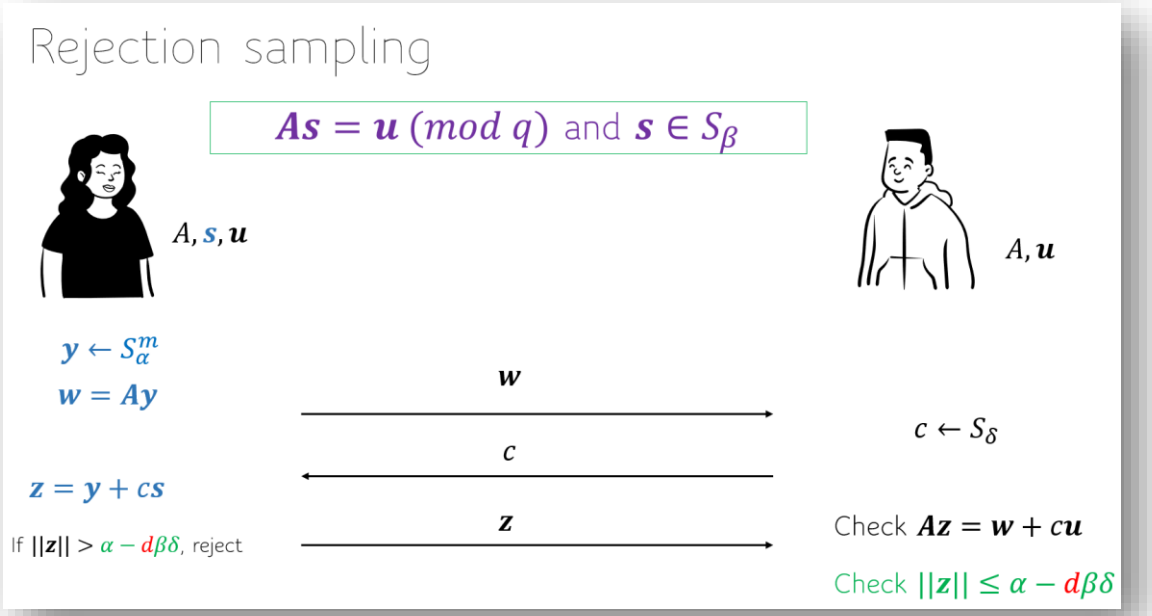
- Parameter δ to make sure $|C| = (2\delta + 1)^d$ is exponential

(Honest-Verifier) Zero-Knowledge

- Parameter $\alpha \geq 2d^2\beta\delta m$ to ensure the probability of non-rejection to be 1/3

Fiat-Shamir transformation

- Let $H: \{0,1\}^* \to S_\delta$ be a hash function.
- We obtain a *non-interactive proof* as follows.



1. $y \leftarrow S_\alpha^m$
2. $w = Ay$
3. $c = H((A, u), w)$
4. $z = y + cs$
5. If $||z|| > \alpha - d\beta\delta$, restart
6. Output $\pi = (w, z)$.

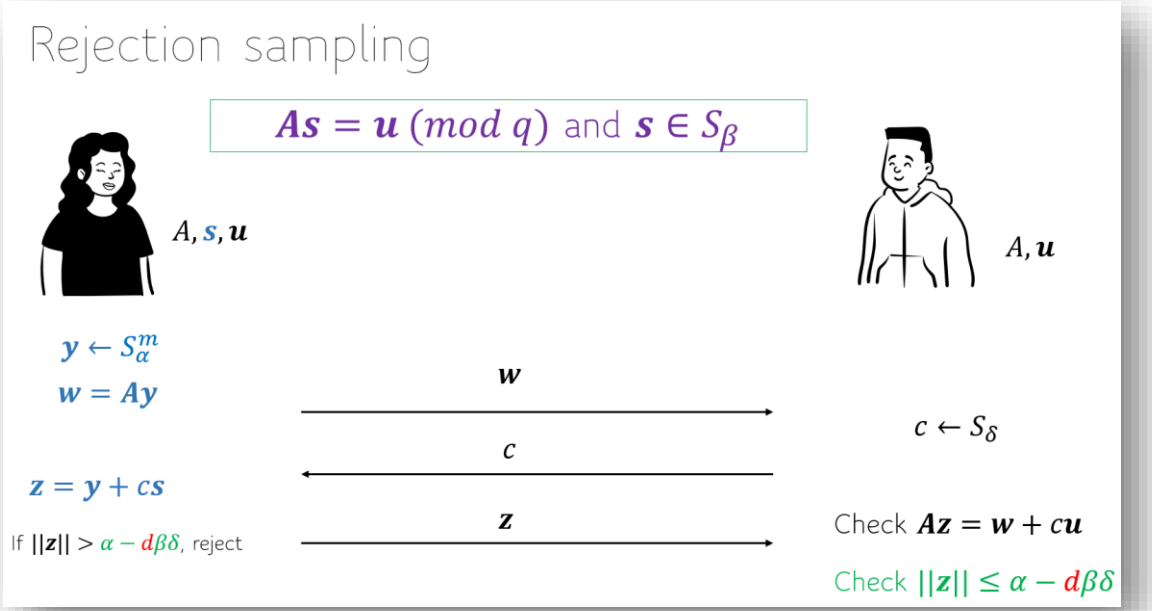
To verify $\pi = (w, z)$, check:

1. $||z|| \leq \alpha - d\beta\delta$ and $Az = w + cu$ where $c = H((A, u), w)$

Proof size: $n + m$ ring elements

Fiat-Shamir transformation

- Let $H: \{0,1\}^* \to S_\delta$ be a hash function.
- Optimisation:



1. $y \leftarrow S_\alpha^m$
2. $w = Ay$
3. $c = H((A, u), w)$
4. $z = y + cs$
5. If $||z|| > \alpha - d\beta\delta$, restart
6. Output $\pi = (c, z)$.

To verify $\pi = (c, z)$, check:
1. $||z|| \leq \alpha - d\beta\delta$ and $c = H((A, u), Az - cu)$.

Proof size: $1 + m$ ring elements

Zero-knowledge in ROM

- No efficient adversary can distinguish between valid proofs and simulated proofs

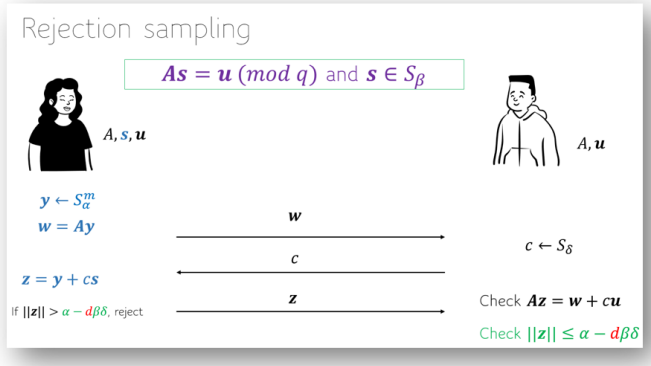
Simulate:

1. $\boldsymbol{z} \leftarrow [\alpha - d\beta\delta, \alpha + d\beta\delta]$
2. $c \leftarrow S_\delta$
3. $\boldsymbol{w} \coloneqq \boldsymbol{Az} - c\boldsymbol{u}$.
4. Program $H((\boldsymbol{A}, \boldsymbol{u}), \boldsymbol{w}) \coloneqq c$
5. Output $\pi \coloneqq (c, \boldsymbol{z})$.

Simple entropy argument to show that we will never ``overwrite'' the random oracle

Fiat-Shamir transformation

- Let $H: \{0,1\}^* \rightarrow S_\delta$ be a hash function.
- Optimisation:



1. $\boldsymbol{y} \leftarrow S_\alpha^m$
2. $\boldsymbol{w} = \boldsymbol{Ay}$
3. $c = H((\boldsymbol{A}, \boldsymbol{u}), \boldsymbol{w})$
4. $\boldsymbol{z} = \boldsymbol{y} + c\boldsymbol{s}$
5. If $||\boldsymbol{z}|| > \alpha - d\beta\delta$, restart
6. Output $\pi = (c, \boldsymbol{z})$.

To verify $\pi = (c, \boldsymbol{z})$, check:
1. $||\boldsymbol{z}|| \leq \alpha - d\beta\delta$ and $c = H((\boldsymbol{A}, \boldsymbol{u}), \boldsymbol{Az} - c\boldsymbol{u})$.

Proof size: $1 + m$ ring elements

Overview

MOTIVATION ON ZERO-KNOWLEDGE PROOFS (ZKP) AND SNARKS

SIMPLE EXAMPLE

ZKP FOR LATTICE RELATED STATEMENTS

APPLICATIONS TO SIGNATURE SCHEMES (DILITHIUM)

From an ID-protocol to a FSwA signature

KeyGen:
sk:= $\boldsymbol{s} \leftarrow S_\beta$
pk:= $\mathbf{u} = \boldsymbol{As}$

Sign(sk,m):
1. $\boldsymbol{y} \leftarrow S_\alpha^m$
2. $\boldsymbol{w} = \boldsymbol{Ay}$
3. $c = H((\boldsymbol{A}, \boldsymbol{u}), \boldsymbol{w}, \text{m})$
4. $\boldsymbol{z} = \boldsymbol{y} + c\boldsymbol{s}$
5. If $||\boldsymbol{z}|| > \alpha - d\beta\delta$, restart
6. Output $\pi = (\boldsymbol{c}, \boldsymbol{z})$.

Verify(pk,m,$\pi = (c, \boldsymbol{z})$):
Return 1 if all the following hold:
1. $||\boldsymbol{z}|| \leq \alpha - d\beta\delta$
2. $c = H((\boldsymbol{A}, \boldsymbol{u}), \boldsymbol{Az} - c\boldsymbol{u}, \text{m})$.

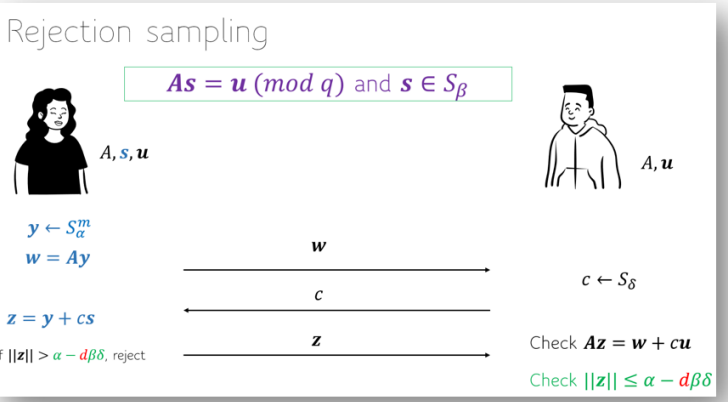
Fiat-Shamir transformation

- Let $H: \{0,1\}^* \rightarrow S_\delta$ be a hash function.
- Optimisation:

1. $\boldsymbol{y} \leftarrow S_\alpha^m$
2. $\boldsymbol{w} = \boldsymbol{Ay}$
3. $c = H((\boldsymbol{A}, \boldsymbol{u}), \boldsymbol{w})$
4. $\boldsymbol{z} = \boldsymbol{y} + c\boldsymbol{s}$
5. If $||\boldsymbol{z}|| > \alpha - d\beta\delta$, restart
6. Output $\pi = (\boldsymbol{c}, \boldsymbol{z})$.

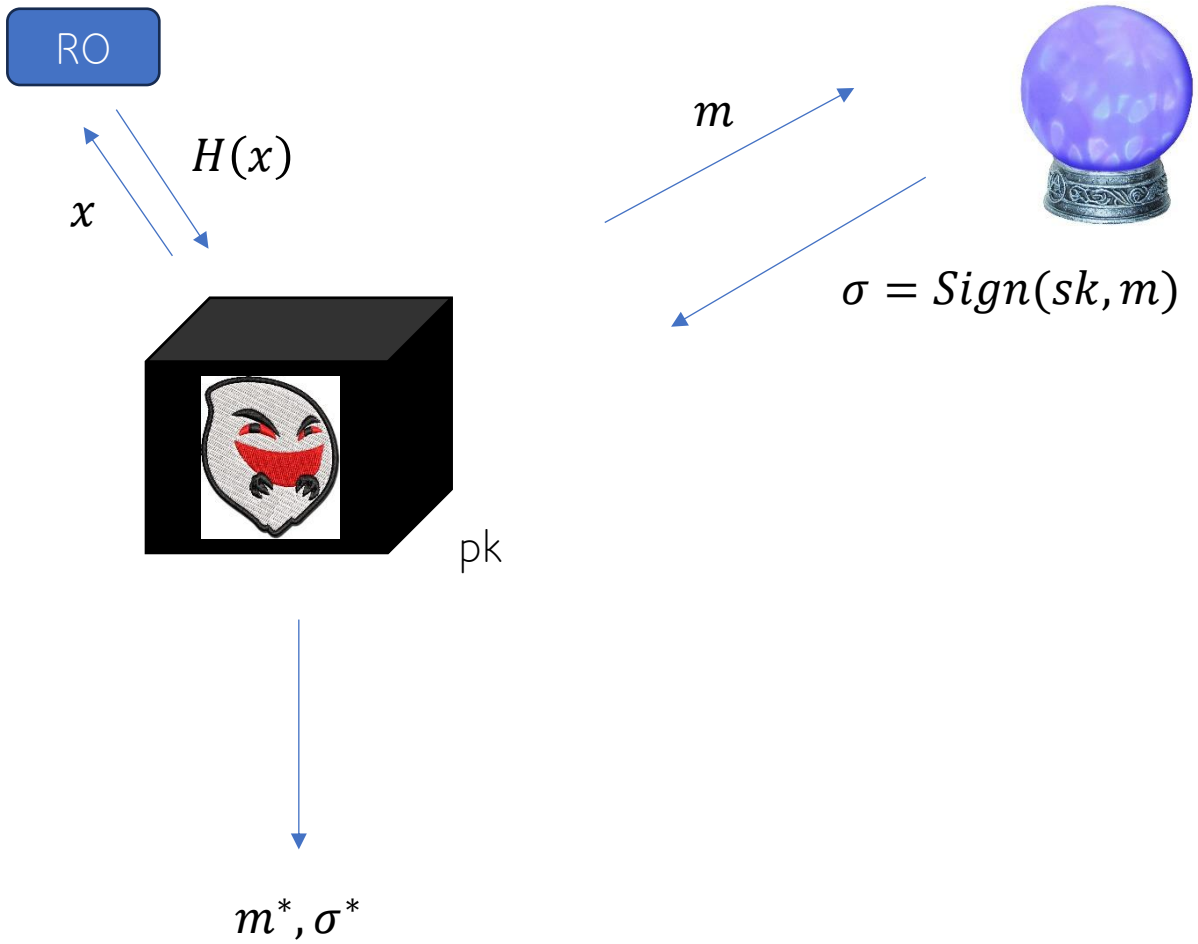
To verify $\pi = (\boldsymbol{c}, \boldsymbol{z})$, check:
1. $||\boldsymbol{z}|| \leq \alpha - d\beta\delta$ and $c = H((\boldsymbol{A}, \boldsymbol{u}), \boldsymbol{Az} - c\boldsymbol{u})$.

Rejection sampling



Proof size: $1 + m$ ring elements

sEUF-CMA security in the ROM



Assumptions:

- Adversary makes q_H RO queries (no duplicates)
- Adv makes q_S signing queries

Adversary wins if
$Ver(pk, m^*, \sigma^*) = 1$
and (m^*, σ^*) does not belong to the query/answer set

Unforgeability proof

- Step 1: simulate the signing oracle via HVZK proof (no sk is used)
- Step 2: apply the forking lemma [BN06]
- Step 3: reduce to the special soundness case.

Unforgeability proof

- Step 1: simulate the signing oracle via HVZK proof (no sk is used)
- Step 2: apply the forking lemma [BN06]
- Step 3: reduce to the special soundness case.

$$\boldsymbol{A}(\boldsymbol{z}-\boldsymbol{z}') = (c-c')\boldsymbol{u} = \boldsymbol{A}(c-c')\boldsymbol{s}$$

We need to argue
$(\boldsymbol{z}-\boldsymbol{z}') \neq (\boldsymbol{c}-c')\boldsymbol{s}$

The only information known to the adversary about $\boldsymbol{s}$ is $\boldsymbol{u} = \boldsymbol{As}$

Lemma: For any $\boldsymbol{A} \in R_q^{n \times m}$, where $m \geq 2n \log q / \log(2\beta + 1)$, for $\boldsymbol{s} \leftarrow S_\beta^m$, with probability $1 - 2^{-d}$ there exists another $\boldsymbol{s'} \in S_\beta^m$ s.t. $\boldsymbol{As} = \boldsymbol{As'}$.

Proof: $\boldsymbol{A}: R_q^m \to R_q^n$ can be thought of as a linear transformation whose range has size q^{nd}.

Thus, there are at most q^{nd} elements in S_β^m which do not collide with any other element in S_β^m by the pigeonhole principle.

Hence, the probability of $\boldsymbol{s}$ being such an element is bounded by

$$\frac{q^{nd}}{(2\beta+1)^{md}} = \left(\frac{q^n}{(2\beta+1)^m}\right)^d \leq 2^{-d}.$$

Unforgeability proof

- Step 1: simulate the signing oracle via HVZK proof (no sk is used)
- Step 2: apply the forking lemma [BN06]
- Step 3: reduce to the special soundness case.

$$\boldsymbol{A}(\boldsymbol{z}-\boldsymbol{z}') = (c-c')\boldsymbol{u} = \boldsymbol{A}(c-c')\boldsymbol{s}$$

- Step 4: deduce that with non-negligible probability $(\boldsymbol{z}-\boldsymbol{z}') - (c-c')\boldsymbol{s} \neq \boldsymbol{0}$

Towards Dilithium [DKL+18]

- CRYSTALS-Dilithium is a ``Fiat-Shamir with Aborts'' signature scheme standardised by NIST
- The scheme applies various (low-order bit) compression
- No forking required, relies on a tailored assumption, SelfTargetMSIS
- Smarter parameter selection and analysis

Further Discussion

- Rejection sampling using Discrete Gaussians [Lyu12]
- Removing rejection sampling (to avoid side-channel attacks), Raccoon [KRPP24], G+G [DPS23]
- Rejection sampling for FSwA – technical issues [BBDD+23, DFPS23]

Thank you!