Threshold Raccoon: Practical Threshold Signatures from Standard Lattice Assumptions

Rafael del Pino, Shuichi Katsumata, Mary Maller, Fabrice Mouhartem, Thomas Prest, Markku-Juhani O. Saarinen

PQShield, AIST, ethereum foundation, CryptPad, Tampere University

Threshold Raccoon in Short

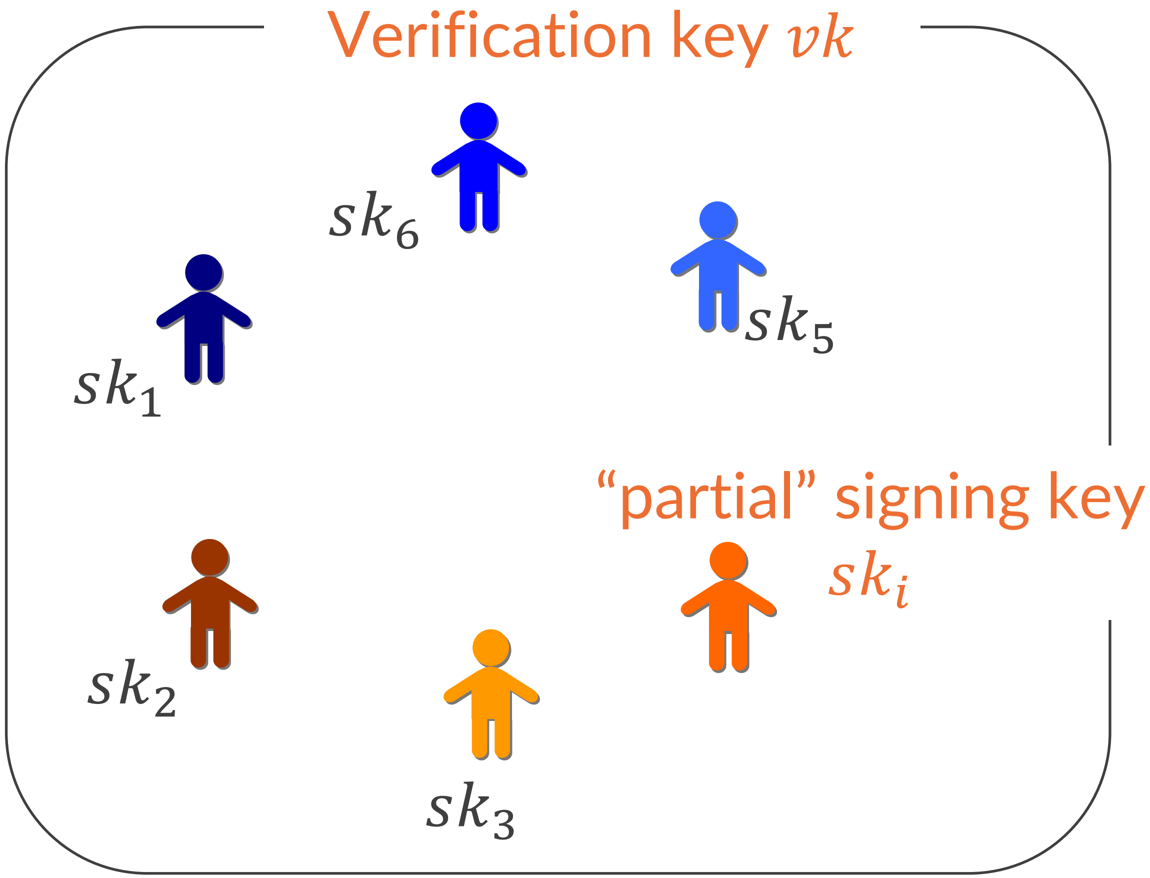
A practical 3 round
lattice-based Threshold Signature

- The **first** scheme w/o heavy tools (e.g., FHE, hom. TDF)
- Scales *gracefully* up to 1024 signers with:
  - **Signature size ~ 13KB**
  - Communication cost ~ 40KB
- Compatible with **Raccoon**@NIST Additional PQ Sig.
- Implementations too ☺

1. Background

What are (T-out-of-N) Threshold Signatures?

⇒ An interactive signing protocol to **"distribute trust"**.

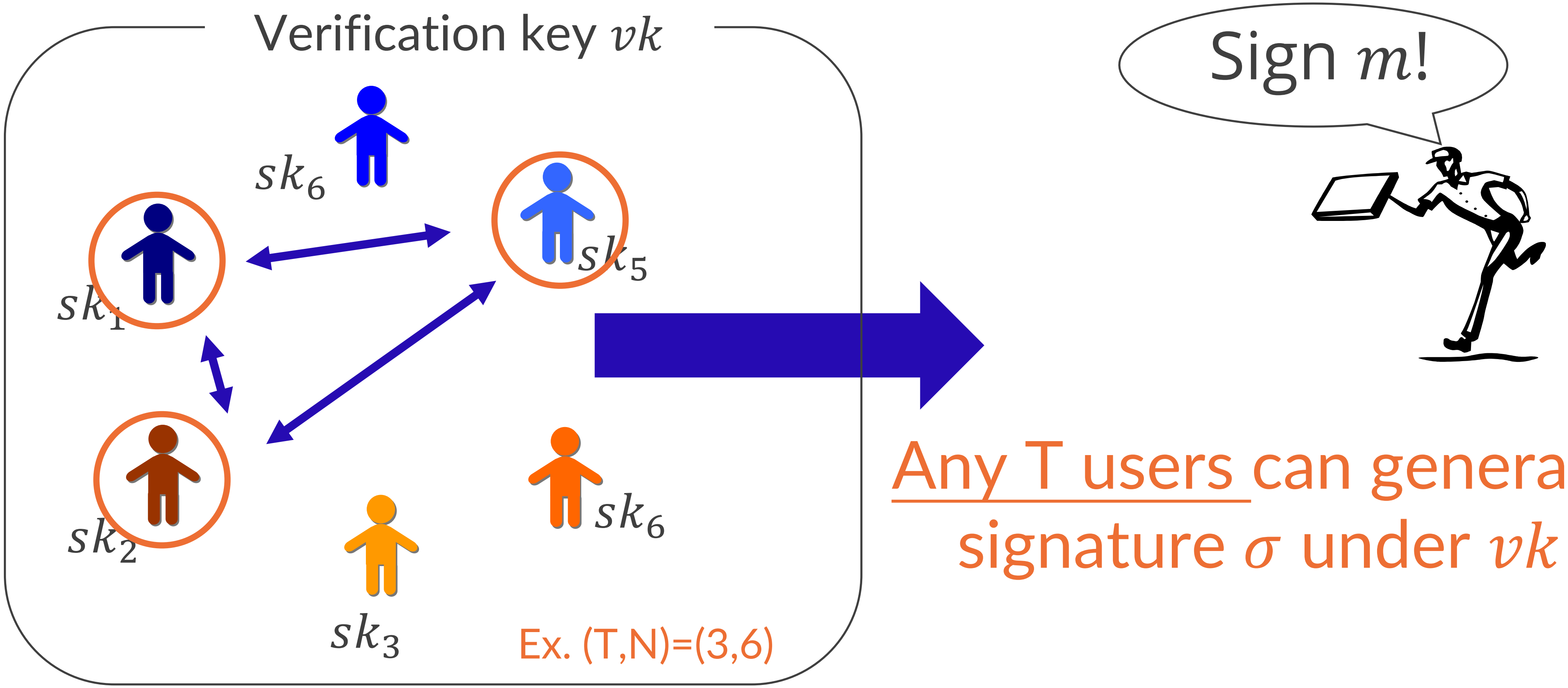


- Single vk
  (Ideally, same as existing one in practice ☺)
- Nobody knows the full signing key sk
- Given T-out-of-N partial signing keys, we can produce a signature.

*In this work, we assume the distributed key generation is performed by a trusted party. More on this at the end!

What are (T-out-of-N) Threshold Signatures?

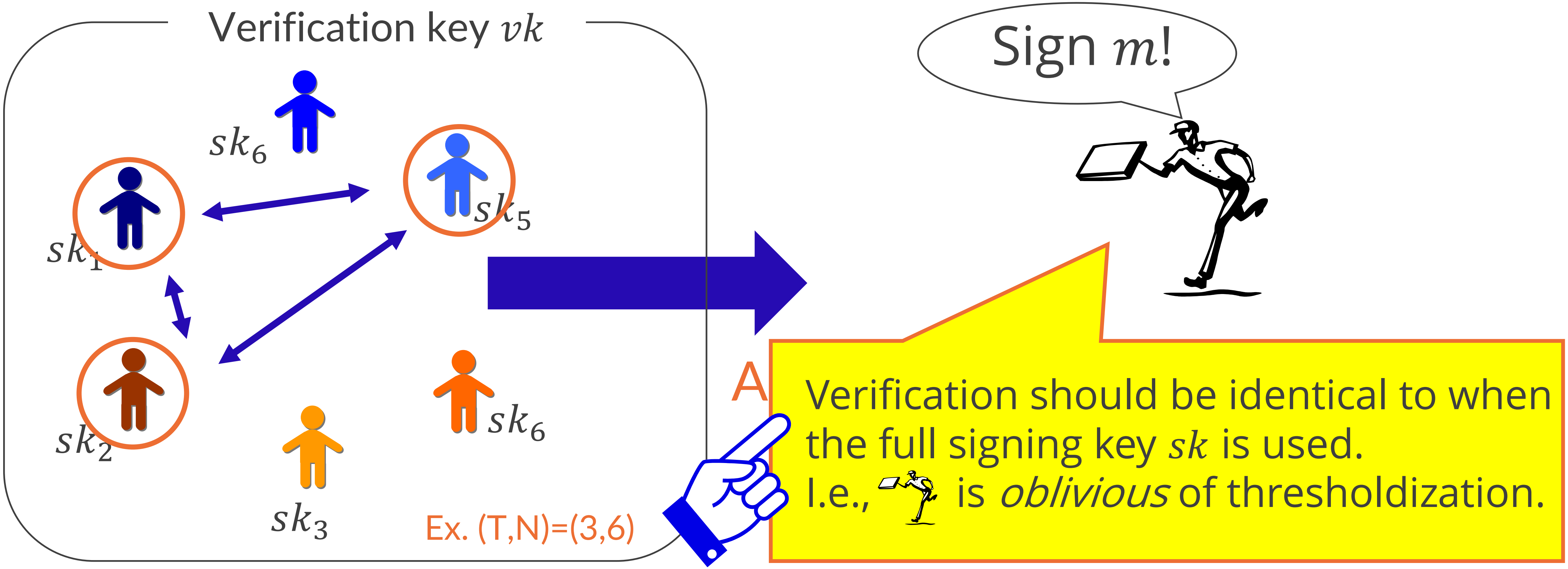
⇒ An interactive signing protocol to **"distribute trust"**.



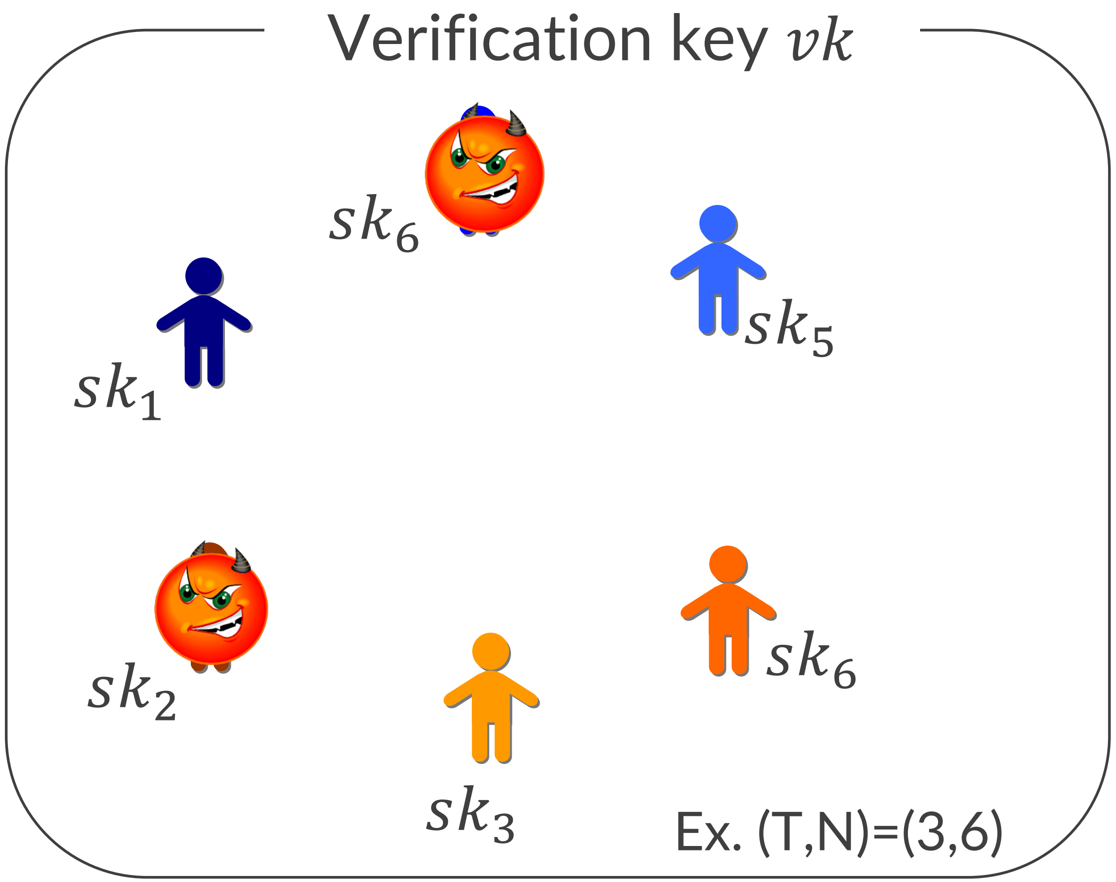
Any T users can generate signature σ under vk

What are (T-out-of-N) Threshold Signatures?

⇒ An interactive signing protocol to **<u>“distribute trust”</u>**.



Security: Unforgeability

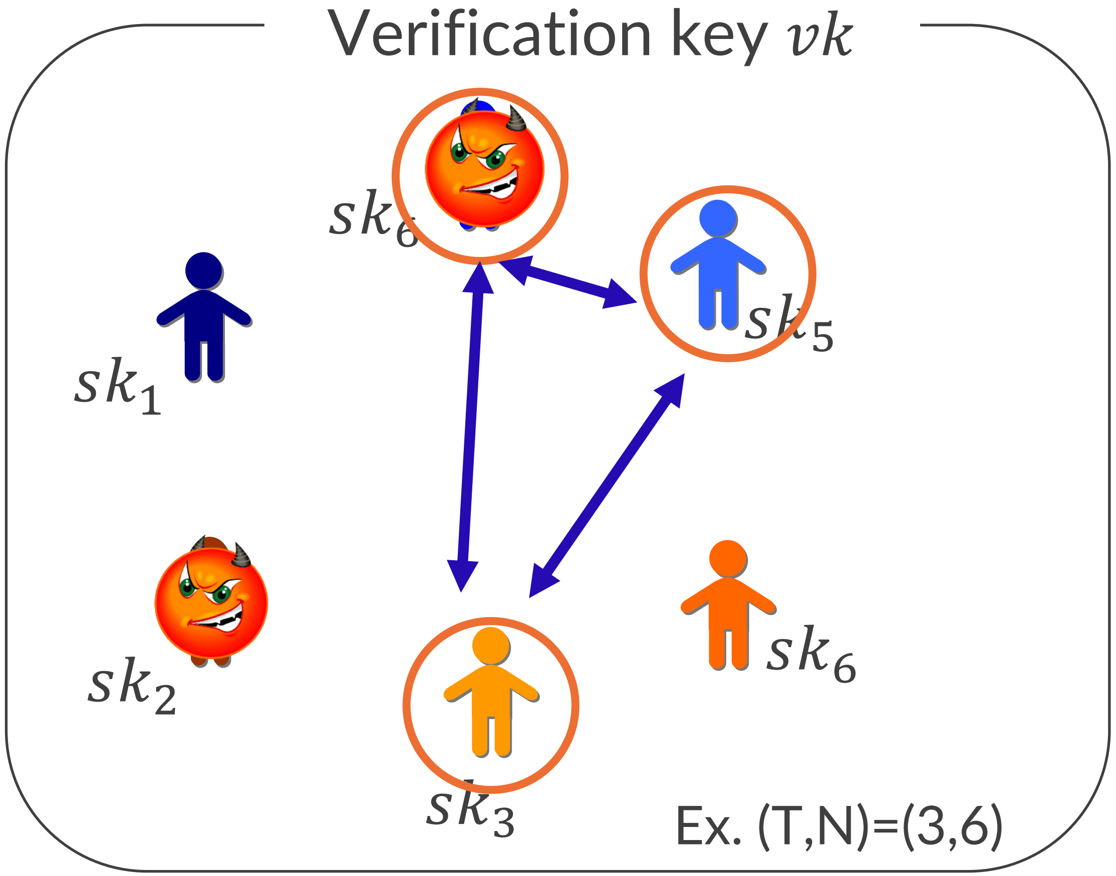


Phase 1

Adversary obtains $T - 1$ partial signing keys by corrupting users.

*We only consider “selective” corruption but more on “adaptive” corruption at the end!

Security: Unforgeability



Phase 1

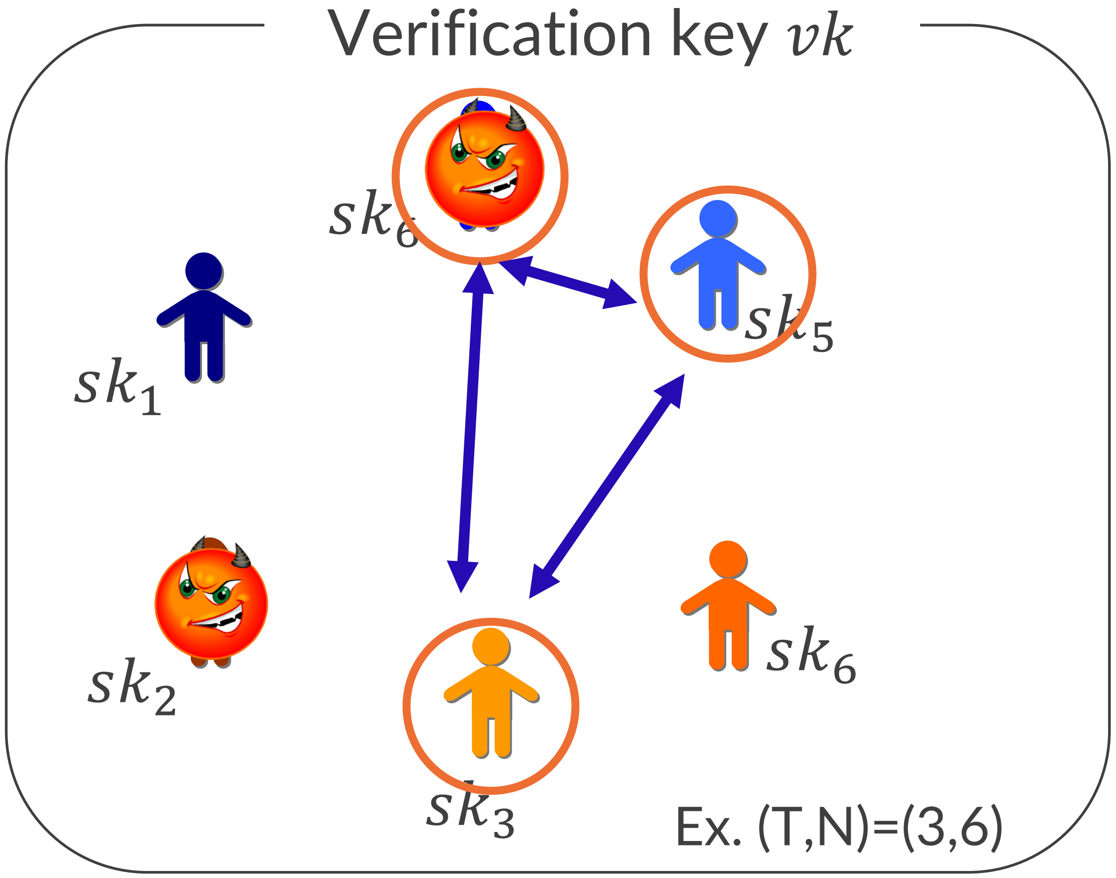
Adversary obtains $T-1$ partial signing keys by corrupting users.

Phase 2

Adversary specifies any signer set S of size T and perform a signing query.

* S can contain corrupted users, possibly deviating from the real signing protocol. ☠

Security: Unforgeability



Phase 1

Adversary obtains $T-1$ partial signing keys by corrupting users.

Phase 2

Adversary specifies any signer set S of size T and perform a signing query.

Forgery

Adversary outputs a forgery on m it didn't query in Phase 2.

Why Study Threshold Signatures?

Applications of Threshold Signature

- Distributed key management.
    - E.g., thresholdizing CA’s private keys, crypto wallets.
- Distributed consensus mechanism on blockchains
- *Let us know if there’s more application!* ☺

NIST Multi-Party Threshold Call

- Deadline expected to be late 2024

Known PQ Threshold Signatures

- TS based on FHE/Homomorphic TDF based
  - [BGG+18]: Round optimal TS via FHE.
  - [ASY22]: Optimized [BGG+18] using Renyi divergence.
  - [GKS23]: Two-round TS, further optimizing [ASY22]
- STARK-based: [KCLM22]
- "Sequential" TS based on isogenies: [CS20,DM20]
- A lot of nice **N-out-of-N** TS w/ Key Aggregation (i.e., multi-signatures)
  - [FSZ22,DOTT21,DOTT22,BTT22,Che23b]

2. An Insecure Attempt

The Basic Principle

⇒ Use (T, N)-Shamir Secret Sharing on LWE secret.

$$vk = t = \underbrace{\begin{bmatrix} A' & I \end{bmatrix}}_{A} s \in R_q^n \qquad sk = s \in R_q^m \text{ s.t. } s \text{ is "}\underline{\textbf{short}}\text{"}$$

The Basic Principle

⇒ Use (T, N)-Shamir Secret Sharing on LWE secret.

$$vk = t = \underbrace{\begin{bmatrix} A' & I \end{bmatrix}}_{A} \, s \in R_q^n \qquad sk = s \in R_q^m \text{ s.t. } s \text{ is “short”}$$

$sk_i = s_i = f(i)$

$\deg(f) = T \wedge f(0) = s$

Given T shares from users $S \subset [N]$,

$$s = \sum_{i \in S} L_{S,i} \, s_i,$$

where $L_{S,i}$ is the “Lagrange” coefficient.

The Basic Principle

⇒ Use (T, N)-Shamir Secret Sharing on LWE secret.

$vk = t = [A' \mid I]\, s \in R_q^n$ (with $A = [A' \mid I]$)

$sk = s \in R_q^m$ s.t. s is "**short**"

$sk_i = s_i = f(i)$

$\deg(f) = T \wedge f(0) = s$

Given T shares from users $S \subset [N]$,

$$s = \sum_{i \in S} L_{S,i}\, s_i,$$

where $L_{S,i}$ is the "Lagrange" coefficient.

$L_{S,i} \in R_q$ is NOT short modulo q!!

Underlying "Linear" Signature Scheme

⇒ Lyubachevsky's signature (a.k.a. Lattice-based Schnorr.

$vk = t = A\,s \in R_q^n$ $\quad sk = s \in R_q^m$

Signer

1. $r \leftarrow \chi$
2. $w = Ar \in R_q^n$

Underlying "Linear" Signature Scheme

⇒ Lyubachevsky's signature (a.k.a. Lattice-based Schnorr.

$vk = t = A s \in R_q^n$ $sk = s \in R_q^m$

Signer

1. $r \leftarrow \chi$
2. $w = Ar \in R_q^n$

3. "Small" $c = H(vk, w) \subset R_q$

Underlying “Linear” Signature Scheme

⇒ Lyubachevsky’s signature (a.k.a. Lattice-based Schnorr.

$vk = t = A\, s \in R_q^n$ $\quad sk = s \in R_q^m$

Signer

1. $r \leftarrow \chi$
2. $w = Ar \in R_q^n$

3. “Small” $c = H(vk, w) \subset R_q$

4. $z = c \cdot s + r \in R_q^m$
5. $RejSamp(z)$

Underlying "Linear" Signature Scheme

⇒ Lyubachevsky's signature (a.k.a. Lattice-based Schnorr.

$vk = t = A\,s \in R_q^n$ $sk = s \in R_q^m$

Signer

1. $r \leftarrow \chi$
2. $w = Ar \in R_q^n$

w →

3. "Small" $c = H(vk, w) \subset R_q$

4. $z = c \cdot s + r \in R_q^m$
5. $RejSamp(z)$

z →

(w, z) is a valid signature if

- z is short
- $Az = c \cdot t + w$

Underlying "Linear" Signature Scheme

⇒ Lyubachevsky's signature (a.k.a. Lattice-based Schnorr.

$vk = t = A \cdot s \in R_q^n$ $sk = s \in R_q^m$

Signer

1. $r \leftarrow \chi$
2. $w = Ar \in R_q^n$

$\xrightarrow{w}$

3. "Small" $c = H(vk, w) \subset R_q$

4. $z = c \cdot s + r \in R_q^m$

$\xrightarrow{z}$

5. R

Linear in secret s and randomness r.

(w, z) is a valid signature if

- z is short
- $Az = c \cdot t + w$

A Naïve Attempt

⇒ Adapting 3-round threshold Schnorr to lattices

Signer $i \in S$

1. $r_i \leftarrow \chi$
2. $w_i = Ar_i \in R_q^n$
3. $cmt_i = G(w_i)$

cmt_i →

← $(cmt_j)_{j \in S \setminus \{i\}}$

4. Open cmt_i and check others

w_i →

← $(w_j)_{j \in S \setminus \{i\}}$

A Naïve Attempt

⇒ Adapting 3-round threshold Schnorr to lattices

Signer $i \in S$

1. $r_i \leftarrow \chi$
2. $w_i = Ar_i \in R_q^n$
3. $cmt_i = G(w_i)$

cmt_i →

← $(cmt_j)_{j \in S \setminus \{i\}}$

4. Open cmt_i *and check others*

w_i →

← $(w_j)_{j \in S \setminus \{i\}}$

First 2 rounds are simply commit-and-open.

W/o it, it is (potentially) insecure against efficient ROS attacks.

A Naïve Attempt

⇒ Adapting 3-round threshold Schnorr to lattices

Signer $i \in S$

1. $r_i \leftarrow \chi$
2. $w_i = Ar_i \in R_q^n$
3. $cmt_i = G(w_i)$

cmt_i →

← $(cmt_j)_{j \in S \setminus \{i\}}$

4. Open cmt_i and check others

w_i →

← $(w_j)_{j \in S \setminus \{i\}}$

5. $c = H(vk, \sum_{j \in S} w_j)$
6. $z_i = c \cdot L_{S,i}\, s_i + r_i \in R_q^m$
7. $RejSamp(z)$

z_i →

← $(z)_{j \in S \setminus \{i\}}$

A Naïve Attempt

⇒ Adapting 3-round threshold Schnorr to lattices

Signer $i \in S$

1. $r_i \leftarrow \chi$
2. $w_i = Ar_i \in R_q^n$
3. $cmt_i = G(w_i)$

cmt_i →

← $(cmt_j)_{j \in S \setminus \{i\}}$

4. Open cmt_i and check others

w_i →

← $(w_j)_{j \in S \setminus \{i\}}$

5. $c = H(vk, \sum_{j \in S} w_j)$
6. $z_i = c \cdot L_{S,i}\, s_i + r_i \in R_q^m$
7. $RejSamp(z)$

z_i →

← $(z)_{j \in S \setminus \{i\}}$

- **Correctness**

Using linearity,

$$z = \sum_{i \in S} z_i = \sum_{i \in S} (c \cdot L_{S,i}\, s_i + r_i)$$

A Naïve Attempt

⇒ Adapting 3-round threshold Schnorr to lattices

Signer $i \in S$

1. $r_i \leftarrow \chi$
2. $w_i = Ar_i \in R_q^n$
3. $cmt_i = G(w_i)$

cmt_i →

← $(cmt_j)_{j \in S \setminus \{i\}}$

4. *Open* cmt_i *and check others*

w_i →

← $(w_j)_{j \in S \setminus \{i\}}$

5. $c = H(vk, \sum_{j \in S} w_j)$
6. $z_i = c \cdot L_{S,i}\, s_i + r_i \in R_q^m$
7. $RejSamp(z)$

z_i →

← $(z)_{j \in S \setminus \{i\}}$

Correctness

Using linearity,

$$z = \sum_{i \in S} z_i = \sum_{i \in S} (c \cdot L_{S,i}\, s_i + r_i) = cs + r$$

Via Shamir SS

A Naïve Attempt

⇒ Adapting 3-round threshold Schnorr to lattices

Signer $i \in S$

1. $r_i \leftarrow \chi$
2. $w_i = Ar_i \in R_q^n$
3. $cmt_i = G(w_i)$

cmt_i →

← $(cmt_j)_{j \in S \setminus \{i\}}$

4. Open cmt_i and check others

w_i →

← $(w_j)_{j \in S \setminus \{i\}}$

5. $c = H(vk, \sum_{j \in S} w_j)$
6. $z_i = c \cdot L_{S,i}\, s_i + r_i \in R_q^m$
7. $RejSamp(z)$

z_i →

← $(z)_{j \in S \setminus \{i\}}$

- **Correctness**

  Using linearity,

$$z = \sum_{i \in S} z_i = \sum_{i \in S} (c \cdot L_{S,i}\, s_i + r_i) = \underbrace{cs + r}_{\text{Via Shamir SS}}$$

$$w = \sum_{i \in S} w_i = \sum_{i \in S} Ar_i = Ar$$

A Naïve Attempt

⇒ Adapting 3-round threshold Schnorr to lattices

Signer $i \in S$

1. $r_i \leftarrow \chi$
2. $w_i = Ar_i \in R_q^n$
3. $cmt_i = G(w_i)$

cmt_i →

← $(cmt_j)_{j \in S \setminus \{i\}}$

4. *Open cmt_i and check others*

w_i →

← $(w_j)_{j \in S \setminus \{i\}}$

5. $c = H(vk, \sum_{j \in S} w_j)$
6. $z_i = c \cdot L_{S,i}\, s_i + r_i \in R_q^m$
7. $RejSamp(z)$

z_i →

← $(z)_{j \in S \setminus \{i\}}$

Correctness

Using linearity,

$$z = \sum_{i \in S} z_i = \sum_{i \in S} (c \cdot L_{S,i}\, s_i + r_i) = cs + r$$

Via Shamir SS

$$w = \sum_{i \in S} w_i = \sum_{i \in S} Ar_i = Ar$$

Exactly a standard signature!

(The Elephant in the Room)

⇒ Adapting 3-round threshold Schnor

Signer $i \in S$

☐ Correctness

1.
2.
3.

4.
ch

Remove rejection sampling by increasing parameters. (Use hint MLWE rather than RD.)

Same ideology as **Raccoon**@NIST Additional PQ Sig: larger signature but ***no restarts*** ☺.

$+ r$

mir SS

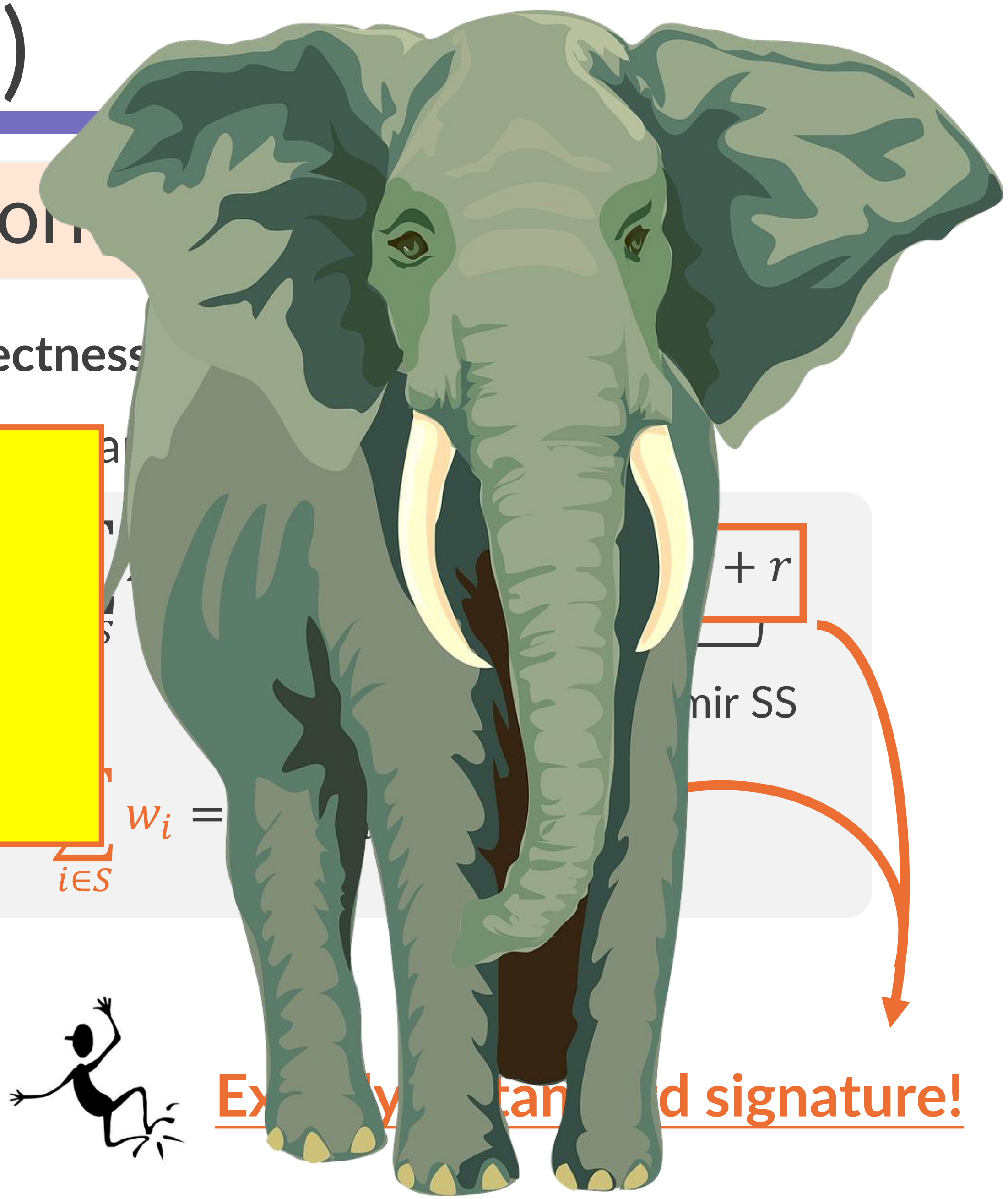
$w_i =$

$i \in S$

5. $c = H(vk, \sum_j$
6. $z_i = c \cdot L_{S,i}\, s_i$ ... $r_i \in R_q^m$
7. $RejSamp(z)$

z_i

$(z)_{j \in S \setminus \{i\}}$



Ex... d signature!

Question: Is This Naïve Attempt Secure??

The **classical 3-round threshold Schnorr would be secure** ☺ [CKM23]

Question: Is This Naïve Attempt Secure??

The **classical 3-round threshold Schnorr would be secure** ☺ [CKM23]

Sadly, **it's insecure in the lattice setting** ☹

The Attack in a Nutshell

Signer $i \in S$

1. $r_i \leftarrow \chi$
2. $w_i = A r_i \in R_q^n$
3. $cmt_i = G(w_i)$

cmt_i →

← $(cmt_j)_{j \in S \setminus \{i\}}$

4. Open cmt_i and check others

w_i →

← $(w_j)_{j \in S \setminus \{i\}}$

5. $c = H(vk, \sum_{j \in S} w_j)$
6. $z_i = c \cdot L_{S,i}\, s_i + r_i \in R_q^m$

z_i →

← $(z)_{j \in S \setminus \{i\}}$

The Vulnerability

Focus on user i's partial signature z_i,

- c, r_i are **small**
- $L_{S,i}$ is **large**
- $L_{S,i}$ is controllable by the adversary

The Attack in a Nutshell

Signer $i \in S$

1. $r_i \leftarrow \chi$
2. $w_i = Ar_i \in R_q^n$
3. $cmt_i = G(w_i)$

$\xrightarrow{cmt_i}$

$\xleftarrow{(cmt_j)_{j\in S\setminus\{i\}}}$

4. Open cmt_i and check others

$\xrightarrow{w_i}$

$\xleftarrow{(w_j)_{j\in S\setminus\{i\}}}$

5. $c = H(vk, \sum_{j\in S} w_j)$
6. $z_i = c \cdot L_{S,i}\, s_i + r_i \in R_q^m$

$\xrightarrow{z_i}$

$\xleftarrow{(z)_{j\in S\setminus\{i\}}}$

The Vulnerability

Focus on user i's partial signature z_i,

- c, r_i are **small**
- $L_{S,i}$ is **large**
- $L_{S,i}$ is controllable by the adversary

(Informal) Attack:

① Find signer sets S, S' with i such that $L_{S,i} = p \cdot L_{S',i}$, where $p < q$ but sufficiently large.

The Attack in a Nutshell

Signer $i \in S$

1. $r_i \leftarrow \chi$
2. $w_i = Ar_i \in R_q^n$
3. $cmt_i = G(w_i)$

cmt_i →

← $(cmt_j)_{j \in S \setminus \{i\}}$

4. Open cmt_i and check others

w_i →

← $(w_j)_{j \in S \setminus \{i\}}$

5. $c = H(vk, \sum_{j \in S} w_j)$
6. $\mathbf{z_i = c \cdot L_{S,i}\, s_i + r_i \in R_q^m}$

z_i →

← $(z)_{j \in S \setminus \{i\}}$

The Vulnerability

Focus on user i's partial signature z_i,

- c, r_i are **small**
- $L_{S,i}$ is **large**
- $L_{S,i}$ is controllable by the adversary

(Informal) Attack:

① Find signer sets S, S' with i such that $\mathbf{L_{S,i} = p \cdot L_{S',i}}$, where $p < q$ but sufficiently large.

② Obtain two partial signatures

$$z_i = c \cdot L_{S,i}\, s_i + r_i \quad \text{and} \quad z'_i = c' \cdot L_{S',i}\, s_i + r'_i$$

The Attack in a Nutshell

Signer $i \in S$

1. $r_i \leftarrow \chi$
2. $w_i = Ar_i \in R_q^n$
3. $cmt_i = G(w_i)$

cmt_i →

← $(cmt_j)_{j \in S \setminus \{i\}}$

4. Open cmt_i and check others

w_i →

← $(w_j)_{j \in S \setminus \{i\}}$

5. $c = H(vk, \sum_{j \in S} w_j)$
6. $\boldsymbol{z_i = c \cdot L_{S,i}\, s_i + r_i \in R_q^m}$

z_i →

← $(z)_{j \in S \setminus \{i\}}$

The Vulnerability

Focus on user i's partial signature z_i,

- c, r_i are **small**
- $L_{S,i}$ is **large**
- $L_{S,i}$ is controllable by the adversary

(Informal) Attack:

① Find signer sets S, S' with i such that $\boldsymbol{L_{S,i} = p \cdot L_{S',i}}$, where $p < q$ but sufficiently large.

② Obtain two partial signatures

$$z_i = c \cdot L_{S,i}\, s_i + r_i \quad \text{and} \quad z'_i = c' \cdot L_{S',i}\, s_i + r'_i$$

③ Recover $\boldsymbol{c \cdot p \cdot r'_i - c' \cdot r_i}$ **over** $\mathbb{Z}$.

The Attack in a Nutshell

Signer $i \in S$

1. $r_i \leftarrow \chi$
2. $w_i = Ar_i \in R_q^n$
3. $cmt_i = G(w_i)$

→ cmt_i

← $(cmt_j)_{j \in S \setminus \{i\}}$

4. Open cmt_i and check others

→ w_i

← $(w_j)_{j \in S \setminus \{i\}}$

5. $c = H(vk, \sum_{j \in S} w_j)$
6. $z_i = c \cdot L_{S,i}\, s_i + r_i \in R_q^m$

→ z_i

← $(z)_{j \in S \setminus \{i\}}$

The Vulnerability

Focus on user i's partial signature z_i,

- c, r_i are **small**
- $L_{S,i}$ is **large**
- $L_{S,i}$ is controllable by the adversary

(Informal) Attack:

① Find signer sets S, S' with i such that $L_{S,i} = p \cdot L_{S',i}$, where $p < q$ but sufficiently large.

② Obtain two partial signatures

$$z_i = c \cdot L_{S,i}\, s_i + r_i \quad \text{and} \quad z'_i = c' \cdot L_{S',i}\, s_i + r'_i$$

③ Recover $c \cdot p \cdot r'_i - c' \cdot r_i$ **over** $\mathbb{Z}$.

④ If $p > |c' \cdot r_i|$, we can recover r_i, r_i, and then **recover s_i!!**

The Attack in a Nutshell

Signer $i \in S$

1. $r_i \leftarrow \chi$

cmt_i

This attack won't work in the classical setting as $c \cdot p \cdot r_i' - c' \cdot r_i$ is distributed uniformly **over $\mathbb{Z}_p$**.

5. $c = H(vk, \sum_{j \in S} w_j)$
6. $\boldsymbol{z_i = c \cdot L_{S,i}\, s_i + r_i \in R_q^m}$

z_i

$(z)_{j \in S \setminus \{i\}}$

The Vulnerability

Focus on user i's partial signature z_i,

- c, r_i are **small**
- $L_{S,i}$ is **large**
- $L_{S,i}$ is controllable by the adversary

(Informal) Attack:

1. Find signer sets S, S' with i such that $\boldsymbol{L_{S,i} = p \cdot L_{S',i}}$, where $p < q$ but sufficiently large.
2. Obtain two partial signatures
$$z_i = c \cdot L_{S,i}\, s_i + r_i \quad \text{and} \quad z'_i = c' \cdot L_{S',i}\, s_i + r'_i$$
3. Recover $\boldsymbol{c \cdot p \cdot r_i' - c' \cdot r_i}$ **over $\mathbb{Z}$**.
4. If $p > |c' \cdot r_i|$, we can recover r_i, r_i', and then **recover s_i!!**

The Main Issue

Lagrange coefficients $L_{S,i}$ can be large over $mod\ q$.

BUT, it seems we need $\left|c \cdot L_{S,i}\ s_i\right| < |r_i|$ for a "short" r_i.

Notorious in lattice-based cryptography

- Use {0,1}/{-1,0,1} linear secret sharing with O(N^4) share size ☹
- Argue $L_{S,i}$ is "small" over $mod\ q$ using exp. large modulus q ☹

3. Threshold Raccoon

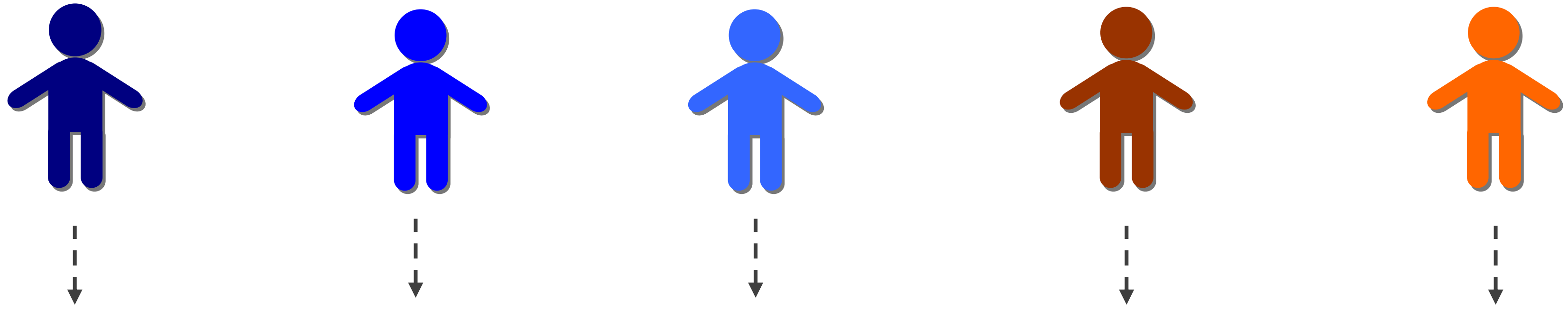
Our Simple Key Idea

"Mask" the partial signature z_i by additive shares of zero!

Intuition

✓ Individual partial signature z_i won't reveal anything.
✓ Collectively, they add up to the real signature z.

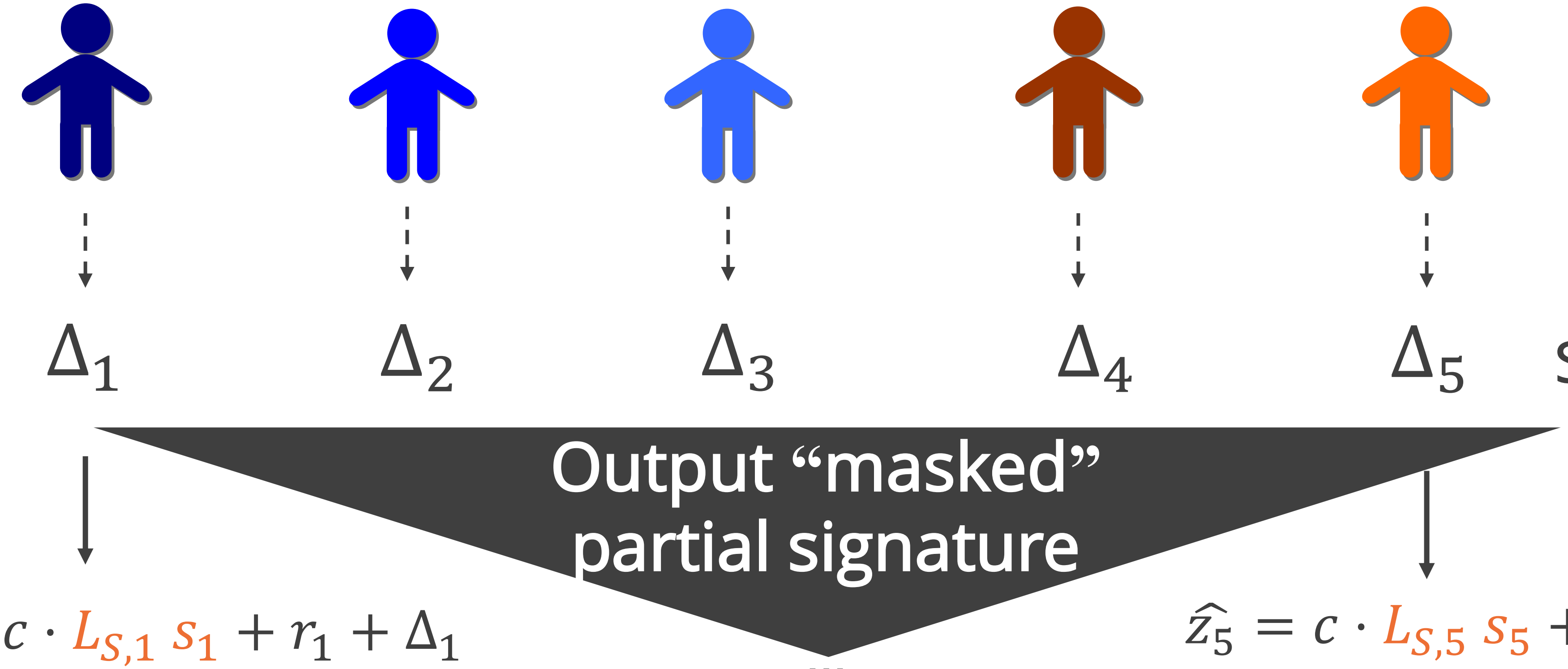
Additive Zero Share



Δ_1 Δ_2 Δ_3 Δ_4 Δ_5 s.t. $\sum_{i \in S} \Delta_i = 0$

Locally generate zero share

Additive Zero Share



$\Delta_1 \quad \Delta_2 \quad \Delta_3 \quad \Delta_4 \quad \Delta_5$ s.t. $\sum_{i \in S} \Delta_i = 0$

$$\hat{z}_1 = \underbrace{c \cdot L_{S,1}\, s_1 + r_1}_{z_1} + \Delta_1$$

...

$$\hat{z}_5 = \underbrace{c \cdot L_{S,5}\, s_5 + r_5}_{z_5} + \Delta_5$$

Additive Zero Share

Δ_1 Δ_2 Δ_3 Δ_4 Δ_5 s.t. $\sum_{i \in S} \Delta_i = 0$

Output "masked" partial signature

$\hat{z}_1 = \underbrace{c \cdot L_{S,1}\, s_1 + r_1}_{z_1} + \Delta_1$

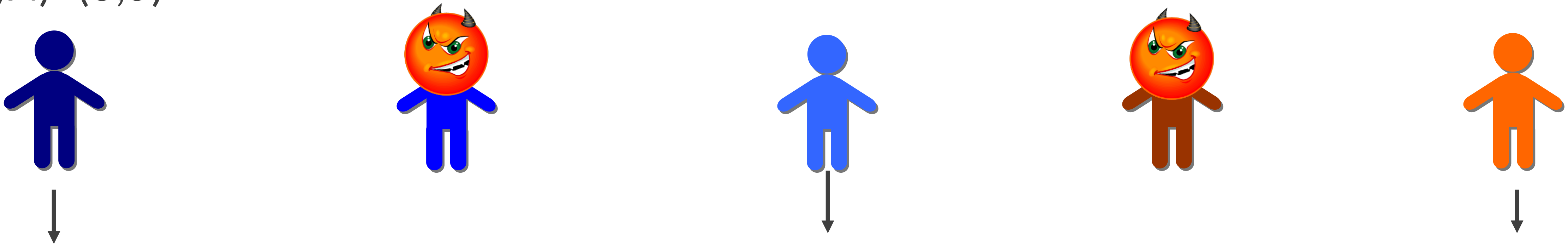
...

$\hat{z}_5 = c \cdot L_{S,5}\, s_5 + r_5 + \Delta_5$

$$z = \sum_{i \in S} \hat{z}_i = \sum_{i \in S} (c \cdot L_{S,i} \cdot s_i + r_i + \Delta_i) = c \cdot s + r$$

Intuition of Why It's Secure

Ex. (T,N)=(3,5)

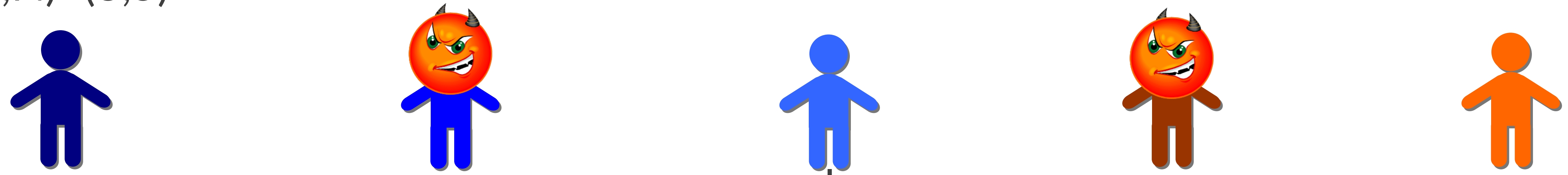


$\widehat{z_1} = c \cdot L_{S,1}\, s_1 + r_1 + \Delta_1$ $\widehat{z_3} = c \cdot L_{S,3}\, s_3 + r_3 + \Delta_3$ $\widehat{z_5} = c \cdot L_{S,5}\, s_5 + r_5 + \Delta_5$

View of 

Intuition of Why It's Secure

Ex. (T,N)=(3,5)



$\widehat{z_1} = c \cdot L_{S,1}\, s_1 + r_1 + \Delta_1$

$\widehat{z_3} = c \cdot L_{S,3}\, s_3 + r_3 + \Delta_3$

$\widehat{z_5} = c \cdot L_{S,5}\, s_5 + r_5 + \Delta_5$

$\approx$

$\widehat{z_1} = c \cdot L_{S,1}\, s_1 + r_1 + \Delta_1$

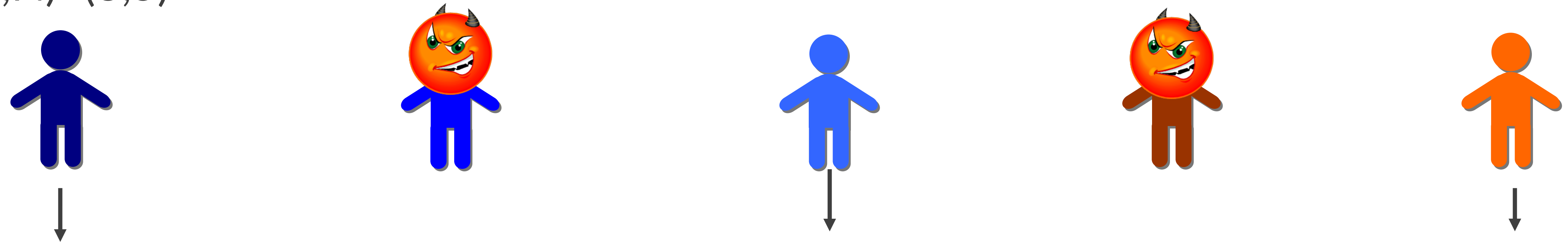
$\widehat{z_3} = c \cdot L_{S,3}\, s_3 + r_3 - \Delta_2 - \Delta_3 - \Delta_1 - \Delta_5$

$\widehat{z_5} = c \cdot L_{S,5}\, s_5 + r_5 + \Delta_5$

Via correctness of zero share

Intuition of Why It's Secure

Ex. (T,N)=(3,5)



$\widehat{z_1} = c \cdot L_{S,1}\, s_1 + r_1 + \Delta_1$ $\widehat{z_3} = c \cdot L_{S,3}\, s_3 + r_3 + \Delta_3$ $\widehat{z_5} = c \cdot L_{S,5}\, s_5 + r_5 + \Delta_5$

$\approx$

$\widehat{z_1} = c \cdot L_{S,1}\, s_1 + r_1 + \Delta_1$ $\widehat{z_3} = c \cdot L_{S,3}\, s_3 + r_3 - \Delta_2 - \Delta_3 - \Delta_1 - \Delta_5$ $\widehat{z_5} = c \cdot L_{S,5}\, s_5 + r_5 + \Delta_5$

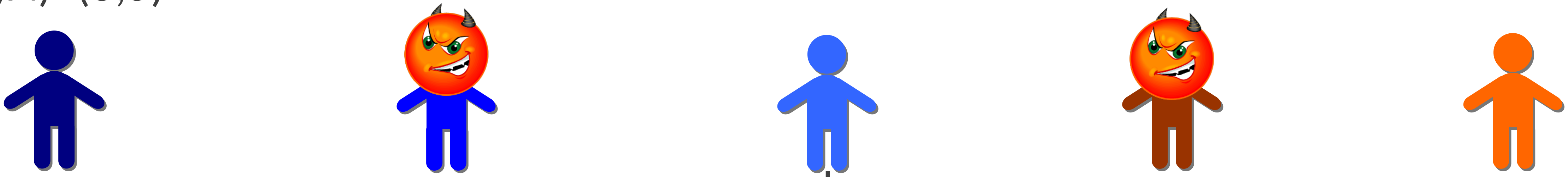
$\approx$

$\widehat{z_1} = \Delta_1^*$ $\widehat{z_3} = c \cdot \sum_{i \in \{1,3,5\}} L_{S,i} s_i + r_3 - \Delta_2 - \Delta_3 - \Delta_1^* - \Delta_5^*$ $\widehat{z_5} = \Delta_5^*$

Via security of zero share

Intuition of Why It's Secure

Ex. (T,N)=(3,5)



$\widehat{z_1} = c \cdot L_{S,1}\, s_1 + r_1 + \Delta_1$

$\widehat{z_3} = c \cdot L_{S,3}\, s_3 + r_3 + \Delta_3$

$\widehat{z_5} = c \cdot L_{S,5}\, s_5 + r_5 + \Delta_5$

$\approx$

$\widehat{z_1} = c \cdot L_{S,1}\, s_1 + r_1 + \Delta_1$

$\widehat{z_3} = c \cdot L_{S,3}\, s_3 + r_3 - \Delta_2 - \Delta_3 - \Delta_1 - \Delta_5$

$\widehat{z_5} = c \cdot L_{S,5}\, s_5 + r_5 + \Delta_5$

$\approx$

$\widehat{z_1} = \Delta_1^*$

$\widehat{z_3} = c \cdot \sum_{i \in \{1,3,5\}} L_{S,i} s_i + r_3 - \Delta_2 - \Delta_3 - \Delta_1^* - \Delta_5^*$

$\widehat{z_5} = \Delta_5^*$

$\approx$

Via correctness of Shamir

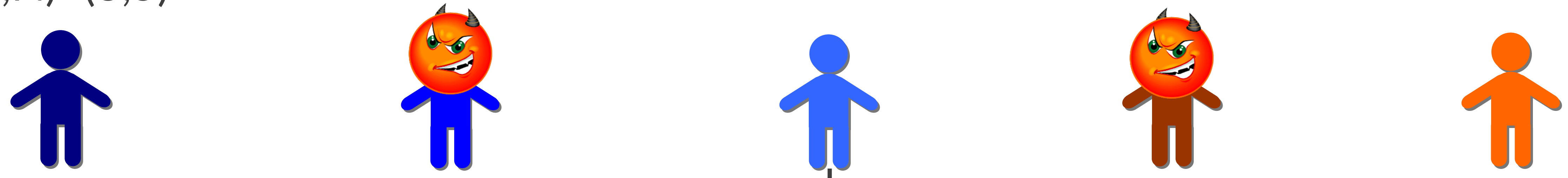
$\widehat{z_1} = \Delta_1^*$

$\widehat{z_3} = c \cdot s + r_3 - c \cdot \sum_{i \in \{2,4\}} (L_{S,i} s_i + \Delta_i) - \Delta_1^* - \Delta_5^*$

$\widehat{z_5} = \Delta_5^*$

Intuition of Why It's Secure

Ex. (T,N)=(3,5)



$\widehat{z}_1 = c \cdot L_{S,1}\, s_1 + r_1 + \Delta_1$ $\widehat{z}_3 = c \cdot L_{S,3}\, s_3 + r_3 + \Delta_3$ $\widehat{z}_5 = c \cdot L_{S,5}\, s_5 + r_5 + \Delta_5$

$\widehat{z}_1$ … $_2 - \Delta_3 - \Delta_1 - \Delta_5$ $\widehat{z}_5 = c \cdot L_{S,5}\, s_5 + r_5 + \Delta_5$

No more Lagrange coefficient!

Proof boils down to non-threshold Raccoon signature ☺

$\widehat{z}_1 = \Delta_1^*$ $\widehat{z}_3 = c$ … $L_{S,i} s_i + r_3 - \Delta_2 - \Delta_3 - \Delta_1^* - \Delta_5^*$ $\widehat{z}_5 = \Delta_5^*$

≈ Via correctness of Shamir

$\widehat{z}_1 = \Delta_1^*$ $\widehat{z}_3 = c \cdot s + r_3 - c \cdot \sum_{i \in \{2,4\}} (L_{S,i} s_i + \Delta_i) - \Delta_1^* - \Delta_5^*$ $\widehat{z}_5 = \Delta_5^*$

Simple Way to Implement Zero Share

During KeyGen, give user i, PRF keys $\left(k_{i,j}, k_{j,i}\right)_{j\in[N]}$.

ZeroShare

$$\Delta_i = \sum_{j\in S} PRF\left(k_{i,j}, sid\right) - PRF(k_{j,i}, sid)$$

Simple Way to Implement Zero Share

During KeyGen, give user i, PRF keys $\left(k_{i,j}, k_{j,i}\right)_{j\in[N]}$.

ZeroShare

$$\Delta_i = \sum_{j \in S} \underbrace{PRF\left(k_{i,j}, sid\right)}_{m_{i,j}} - \underbrace{PRF\left(k_{j,i}, sid\right)}_{m_{j,i}}$$

$$= m_i - m_i^*$$

	1		2		3		4		5		
1	$m_{1,1}$	+	$m_{1,2}$	+	$m_{1,3}$	+	$m_{1,4}$	+	$m_{1,5}$	=	m_1
	+		+		+		+		+		+
2	$m_{2,1}$	+	$m_{2,2}$	+	$m_{2,3}$	+	$m_{2,4}$	+	$m_{2,5}$	=	m_2
	+		+		+		+		+		+
3	$m_{3,1}$	+	$m_{3,2}$	+	$m_{3,3}$	+	$m_{3,4}$	+	$m_{3,5}$	=	m_3
	+		+		+		+		+		+
4	$m_{4,1}$	+	$m_{4,2}$	+	$m_{4,3}$	+	$m_{4,4}$	+	$m_{4,5}$	=	m_4
	+		+		+		+		+		+
5	$m_{5,1}$	+	$m_{5,2}$	+	$m_{5,3}$	+	$m_{5,4}$	+	$m_{5,5}$	=	m_5
	‖		‖		‖		‖		‖		‖
	m_1^*	+	m_2^*	+	m_3^*	+	m_4^*	+	m_5^*	=	m

Performances

Bit security	T	|vk|	|sig|	Comm. / Signer	Runtime / Signer
128	4 16 64 256 1024	**3.9 KB**	**12.7 KB**	**40.8 KB**	11 ms 13 ms 24 ms 72 ms 256 ms

- Asymptotically
  - $|sig| = \tilde{O}(1)$
  - Communication cost/signer = $\tilde{O}(1)$
  - Runtime/signer is $= \tilde{O}(T)$

Thank You For Listening

Some Follow Up Works

- "Two-Round Threshold Signature from Algebraic One-More LWE" [C:EKT24]
- "Adaptively Secure 5 Round Threshold Signatures from MLWE/MSIS and DL with Rewinding" [C:KTR24]
- "Flood and submerse: Verifiable short secret sharing and application to robust threshold signatures on lattices" [C:ENP24]