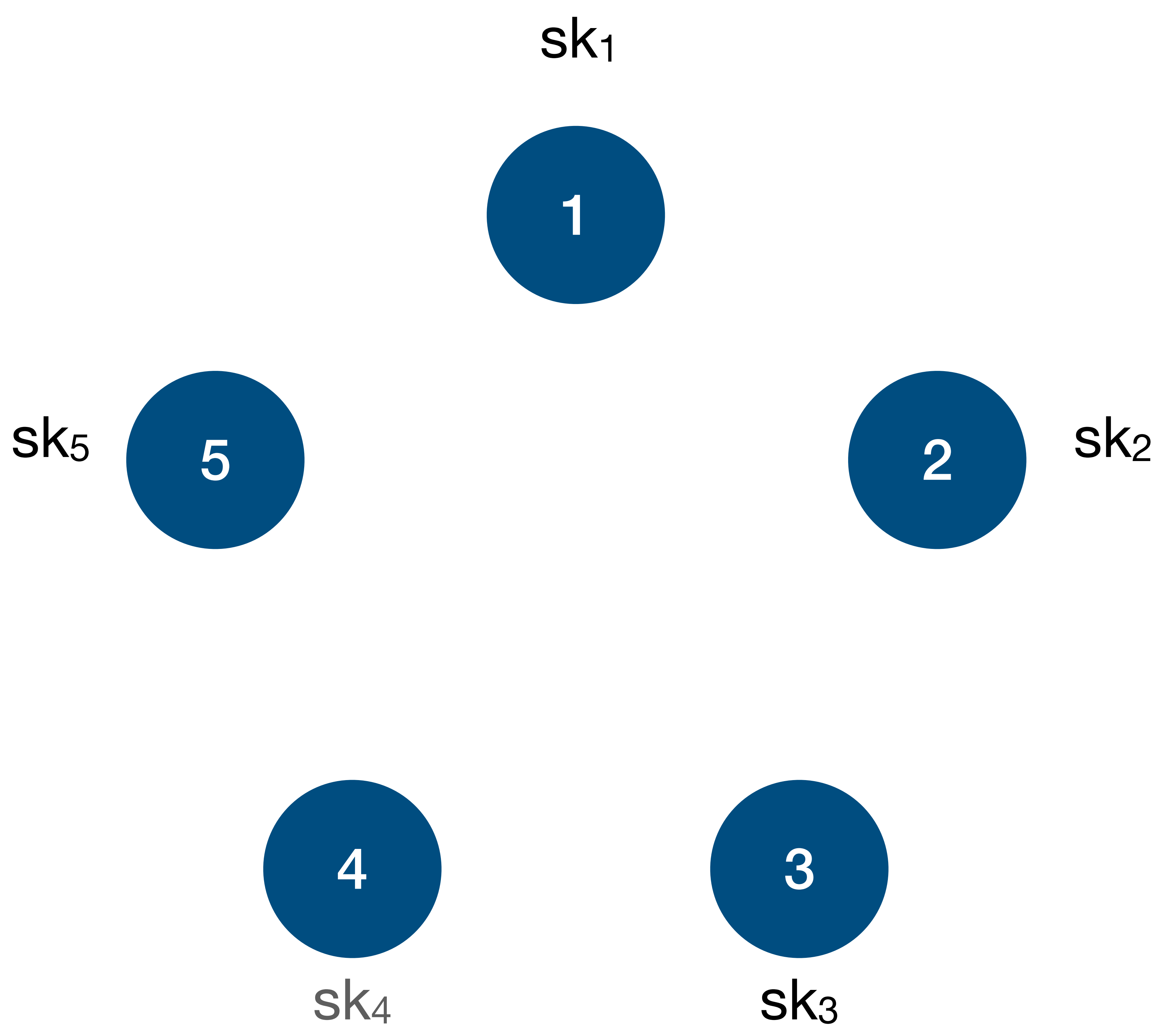
Adaptively Secure 5 Round Threshold Signatures from MLWE / MSIS and DL with Rewinding

- Shuichi Katsumata — PQShield — AIST
- Michael Reichle — ETH Zurich
- Kaoru Takemure — PQShield — AIST

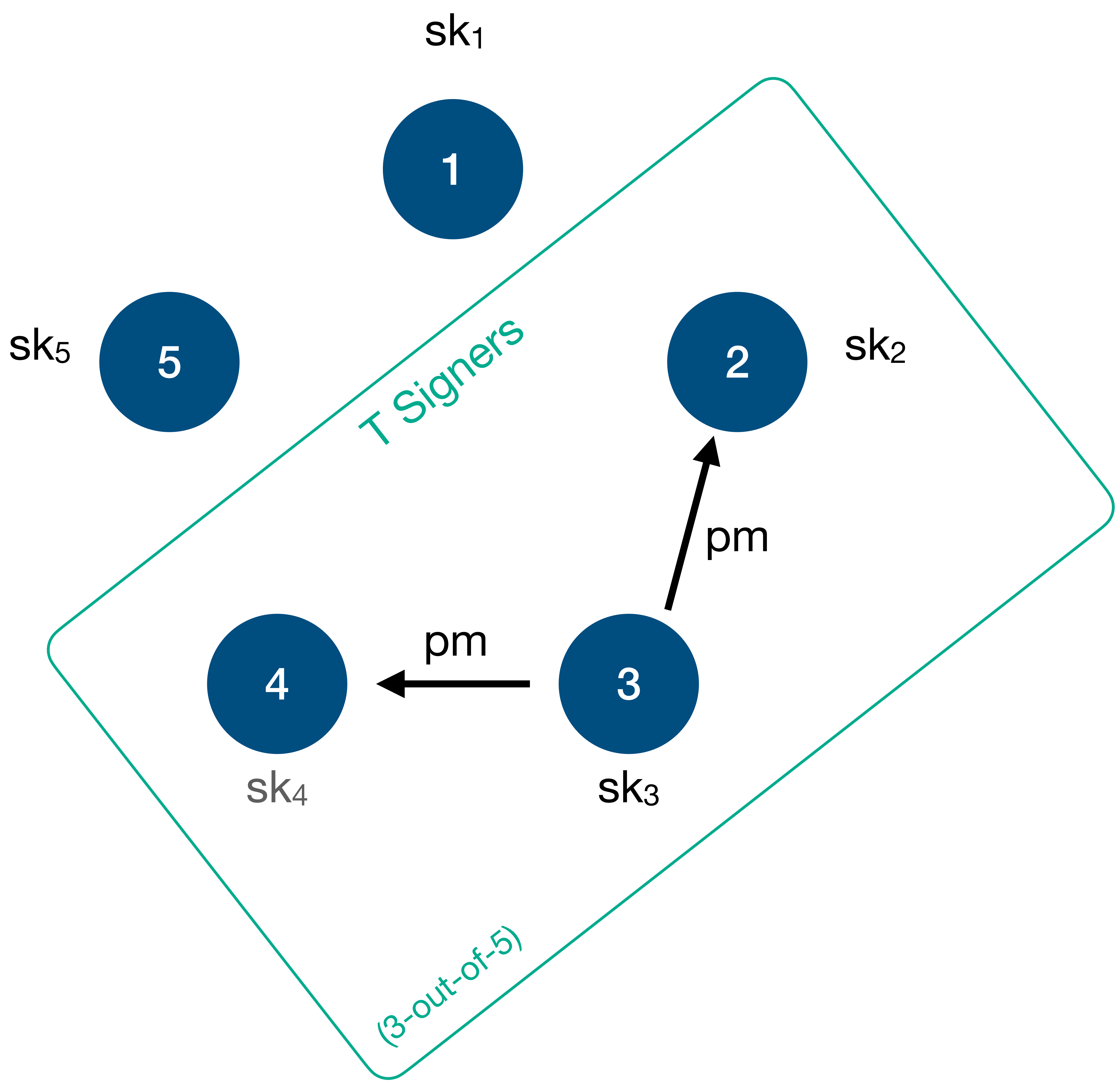
(T-out-of-N) Threshold Signatures

Protocol



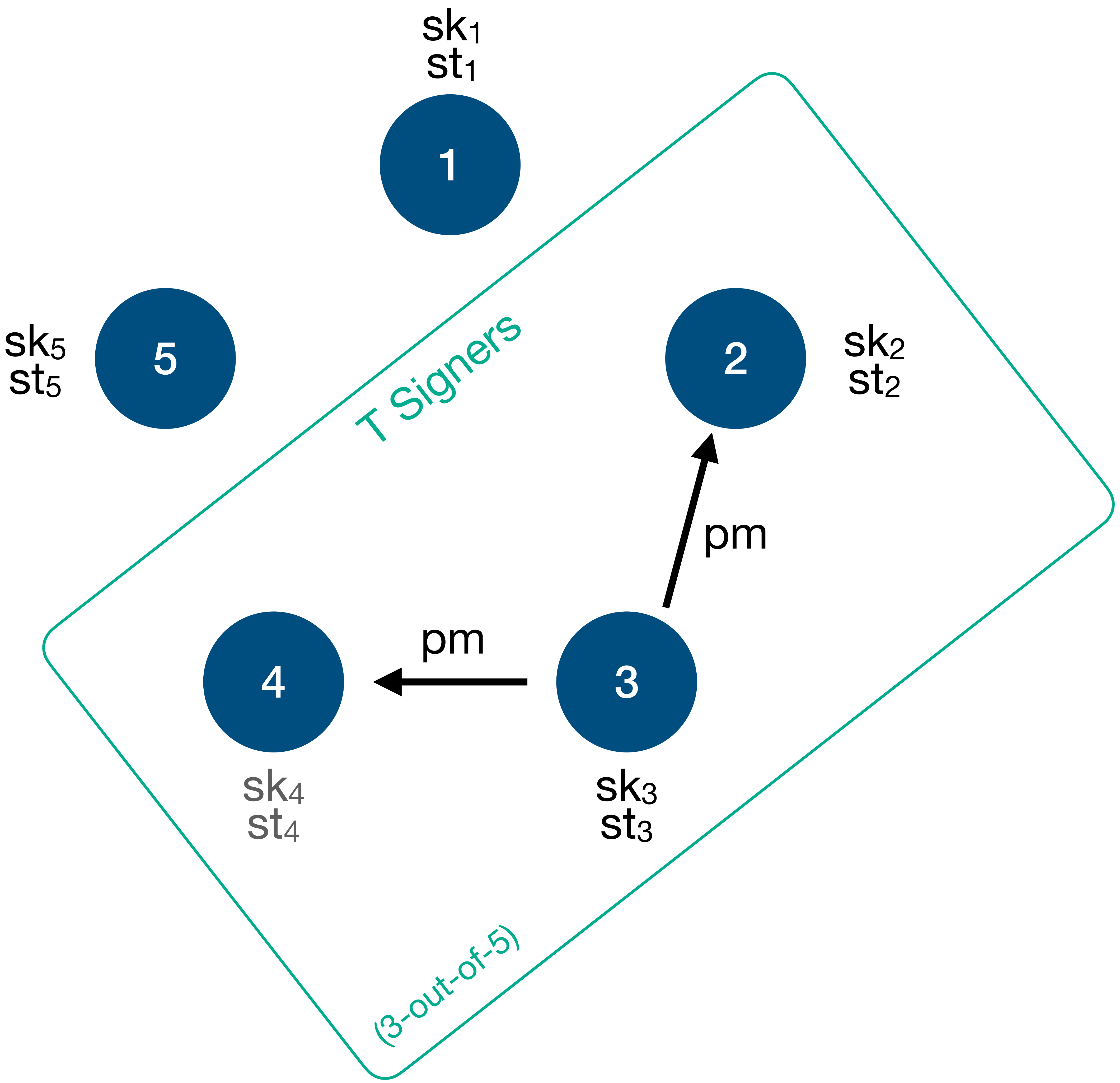
(T-out-of-N) Threshold Signatures

Protocol



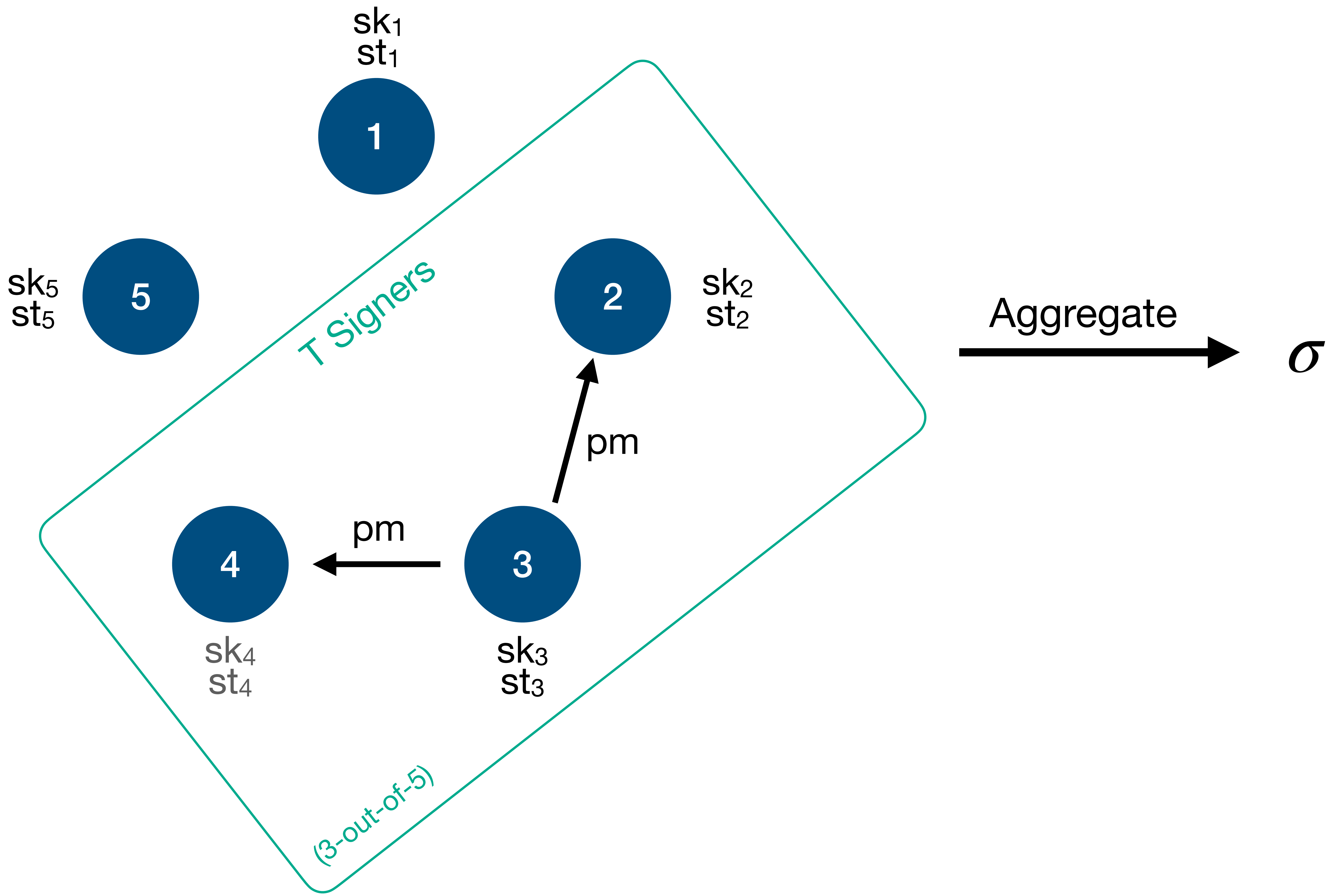
(T-out-of-N) Threshold Signatures

Protocol



(T-out-of-N) Threshold Signatures

Protocol



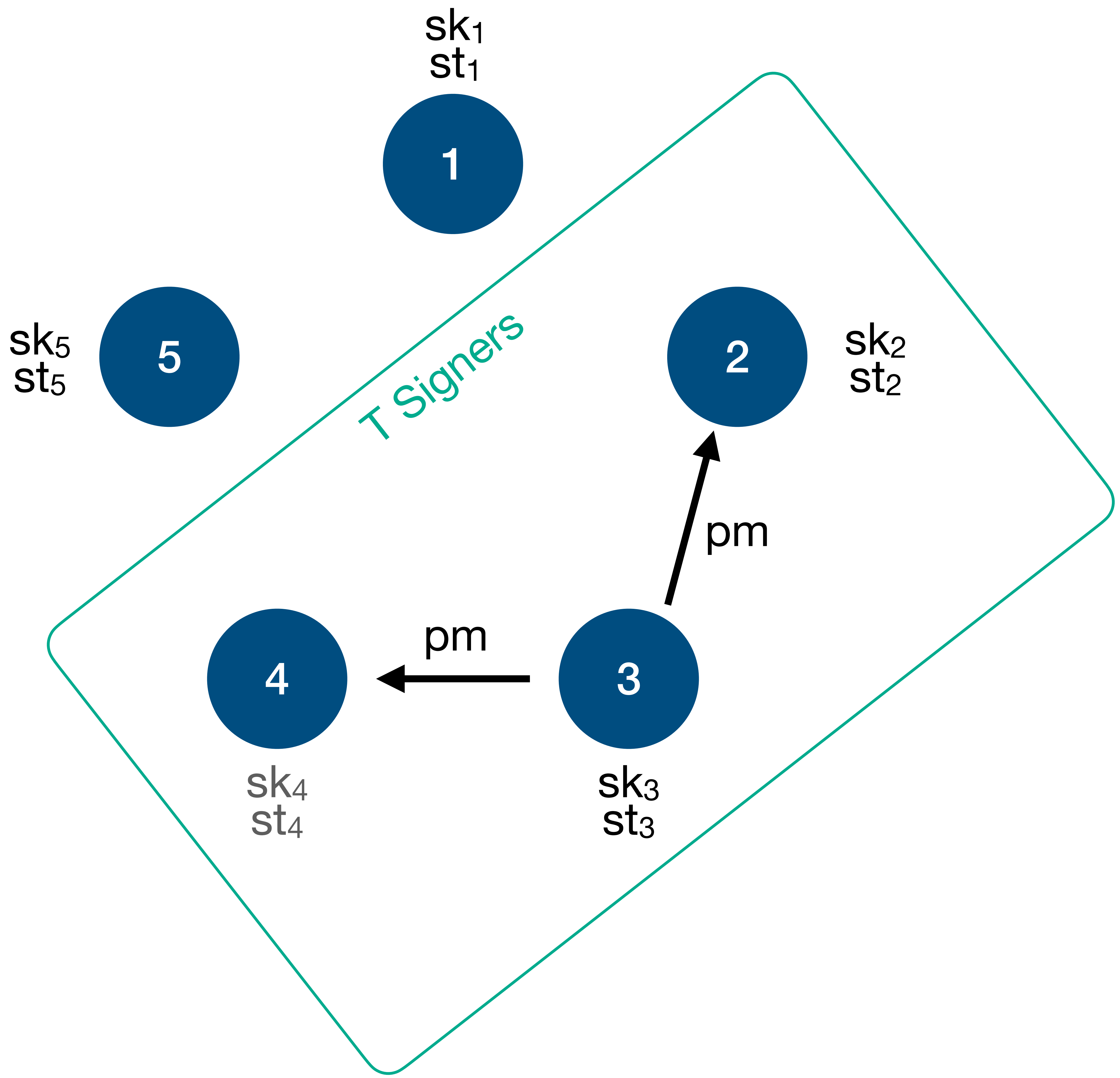
Security

Unforgeability:

- It is hard to find a non-trivial forgery, even in presence of at most T-1 corrupted signers
- Selective: corrupted signers are initially fixed
- Adaptive: signers are corrupted adaptively

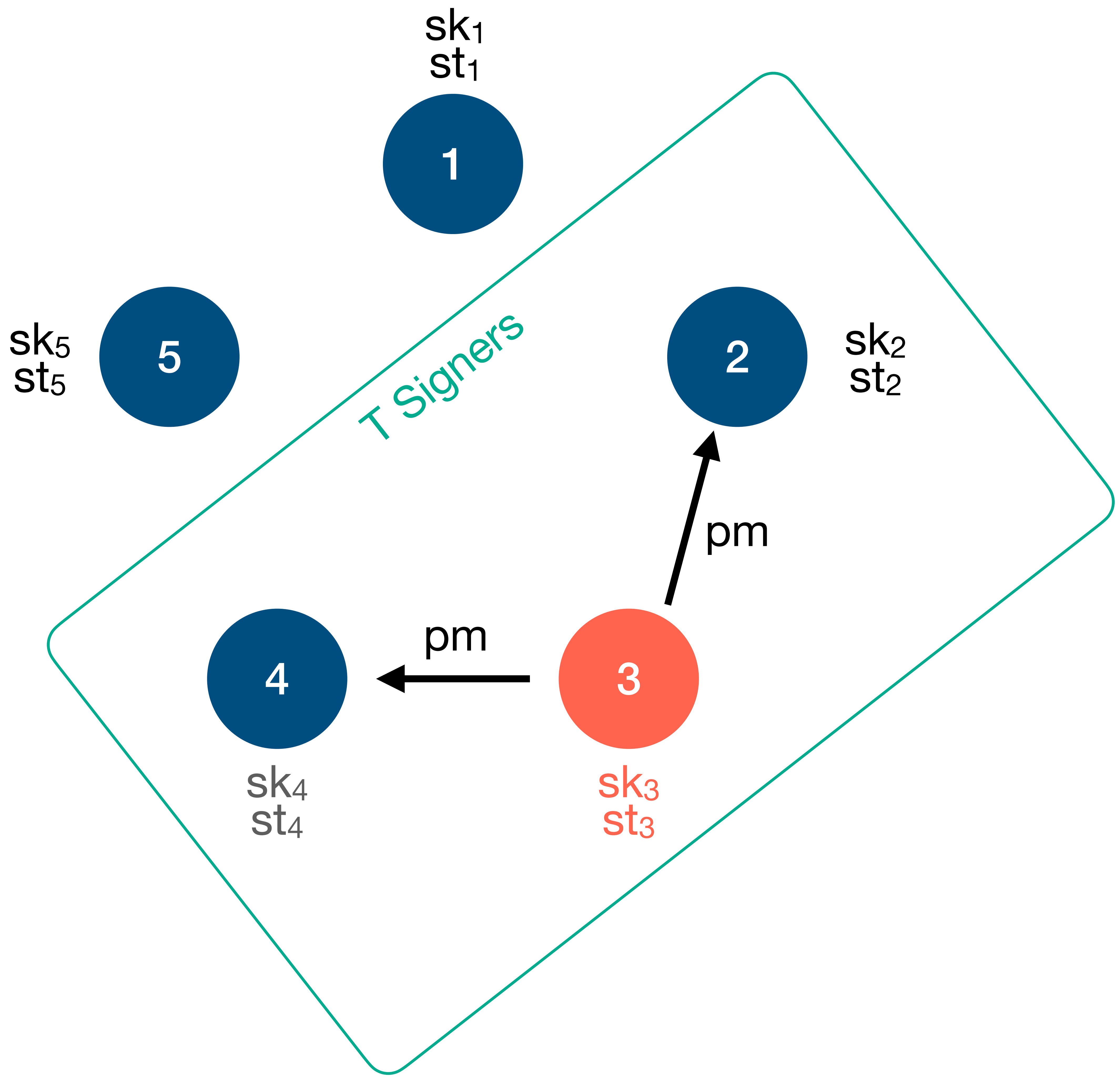
(T, N) Threshold Signatures

Unforgeability



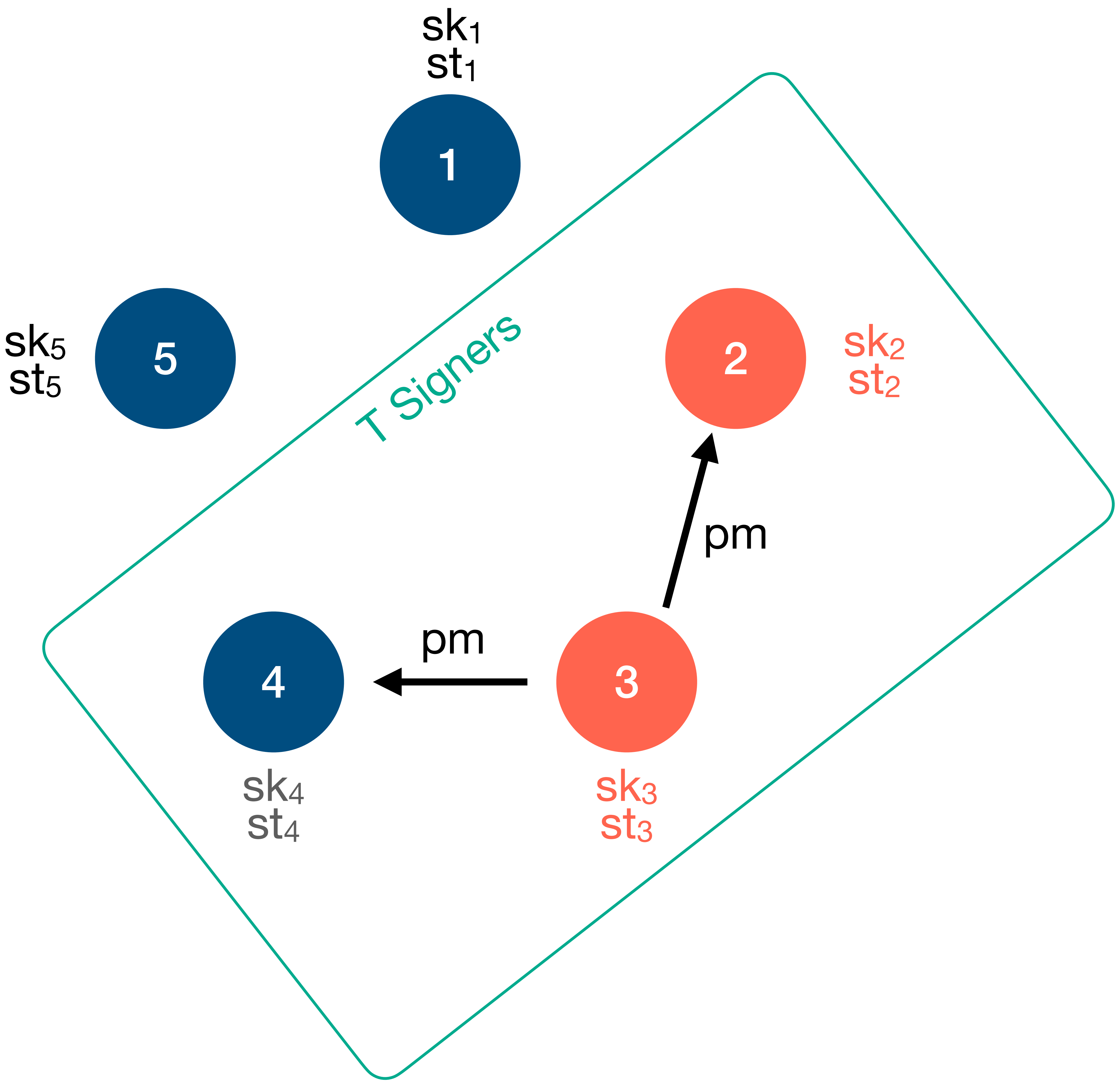
(T, N) Threshold Signatures

Unforgeability



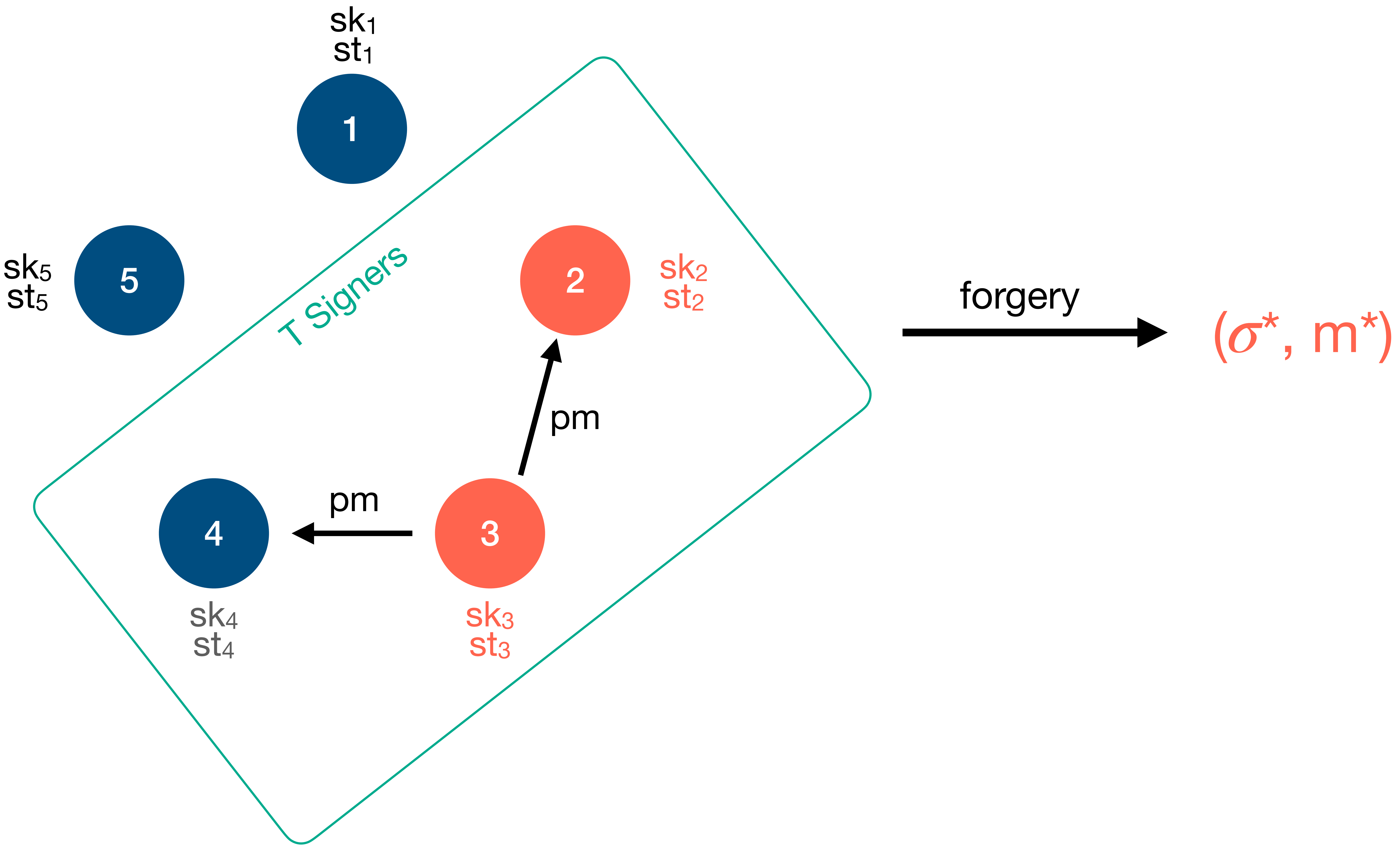
(T, N) Threshold Signatures

Unforgeability



(T, N) Threshold Signatures

Unforgeability



State-of-the-Art

Fiat-Shamir based Threshold Signatures

Selective Security:

- Many efficient protocols (Threshold Raccoon, Threshold Schnorr, …)
- Often relies on ROM and standard assumptions (MLWE / MSIS, DLOG, …)

State-of-the-Art

Fiat-Shamir based Threshold Signatures

Selective Security:

- Many efficient protocols (Threshold Raccoon, Threshold Schnorr, …)
- Often relies on ROM and standard assumptions (MLWE / MSIS, DLOG, …)

Adaptive Security:

- [CKM23]: Adaptive security under AGM, ROM and AOMDL for Schnorr
- [BLTWZ24]: Adaptive security under ROM and DDH for Schnorr-variant

This Work

Results:

This Work

Results:

- **Main Result:** Techniques for adaptive security under minimal assumptions in the ROM

This Work

Results:

- **Main Result:** Techniques for adaptive security under minimal assumptions in the ROM
  - **Schnorr:** 5 round protocol under DL

This Work

Results:

- **Main Result:** Techniques for adaptive security under minimal assumptions in the ROM
  - **Schnorr:** 5 round protocol under DL
  - **Raccoon:** 5 round protocol under MLWE / MSIS

This Work

Results:

- **Main Result:** Techniques for adaptive security under minimal assumptions in the ROM
  - **Schnorr:** 5 round protocol under DL
  - **Raccoon:** 5 round protocol under MLWE / MSIS
- **Others:**

This Work

Results:

- **Main Result:** Techniques for adaptive security under minimal assumptions in the ROM
  - **Schnorr:** 5 round protocol under DL
  - **Raccoon:** 5 round protocol under MLWE / MSIS
- **Others:**
  - State-free security proof for Threshold Raccoon

This Work

Results:

- **Main Result:** Techniques for adaptive security under minimal assumptions in the ROM
  - **Schnorr:** 5 round protocol under DL
  - **Raccoon:** 5 round protocol under MLWE / MSIS
- **Others:**
  - State-free security proof for Threshold Raccoon
  - Techniques to proof stronger unforgeability notions for simulation-based signatures

Threshold Raccoon

Masking-based Threshold Signature

Threshold Raccoon

[dKMMPS24]

<u>Key Material:</u>

- $vk = As$ with $A = [\bar{A} \,|\, I]$
- $sk_i = s_i$ such that $s = \sum_{j \in S} L_{S,i} \cdot s_i$

<u>Signature:</u>

- $\sigma = (w, z)$ such that (i) $Az = c \cdot vk + w$ (iii) z is short

  (ii) $c = H(vk, w, m)$

Threshold Raccoon

[dKMMPS24]

Key Material:

- $vk = As$ with $A = [\bar{A} \,|\, I]$
- $sk_i = s_i$ such that $s = \sum_{j \in S} L_{S,i} \cdot s_i$

Signature:

- $\sigma = (w, z)$ such that (i) $Az = c \cdot vk + w$ (iii) z is short

  (ii) $c = H(vk, w, m)$

Security:

Threshold Raccoon

[dKMMPS24]

<u>Key Material:</u>

- $vk = As$ with $A = [\bar{A} \mid I]$
- $sk_i = s_i$ such that $s = \sum_{j \in S} L_{S,i} \cdot s_i$

<u>Signature:</u>

- $\sigma = (w, z)$ such that (i) $Az = c \cdot vk + w$ (iii) z is short
  (ii) $c = H(vk, w, m)$

<u>Security:</u>

- EUF-CMA under MLWE / MSIS in the ROM

Threshold Raccoon

[dKMMPS24]

Round 1:

- $r_i \leftarrow \chi$
- $w_i \leftarrow A \cdot r_i$
- $\mathsf{cmt}_i = G(w_i)$
- *send* cmt_i

Threshold Raccoon

[dKMMPS24]

Round 1:

- $r_i \leftarrow \chi$
- $w_i \leftarrow A \cdot r_i$
- $\mathsf{cmt}_i = G(w_i)$
- *send* cmt_i

Round 2:

- *send* w_i

Threshold Raccoon

[dKMMPS24]

Round 1:

- $r_i \leftarrow \chi$
- $w_i \leftarrow A \cdot r_i$
- $\mathsf{cmt}_i = G(w_i)$
- *send* cmt_i

Round 2:

- *send* w_i

Round 3:

- *check* $\mathsf{cmt}_i = G(w_i)$
- $w = \sum_{j \in S} w_i$
- $c = H(vk, w, m)$
- *sample 0-share* Δ_i
- *send* $z_i = c \cdot L_{S,i} \cdot s_i + r_i + \Delta_i$

Threshold Raccoon

[dKMMPS24]

<u>Round 1:</u>

- $r_i \leftarrow \chi$
- $w_i \leftarrow A \cdot r_i$
- $\mathsf{cmt}_i = G(w_i)$
- *send* cmt_i

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

<u>Round 2:</u>

- *send* w_i

<u>Round 3:</u>

- *check* $\mathsf{cmt}_i = G(w_i)$
- $w = \sum_{j \in S} w_i$
- $c = H(vk, w, m)$
- *sample 0-share* Δ_i
- *send* $z_i = c \cdot L_{S,i} \cdot s_i + r_i + \Delta_i$

Threshold Raccoon

[dKMMPS24]

<u>Selective Security:</u>

Threshold Raccoon

[dKMMPS24]

Selective Security:

- Simulation of signing oracles without shares s_i:

Threshold Raccoon

[dKMMPS24]

Selective Security:

- Simulation of signing oracles without shares s_i:
  - Simulate a commitment-response pair (w_i, z) for challenge c via HVZK

Threshold Raccoon
[dKMMPS24]

<u>Selective Security:</u>

- Simulation of signing oracles without shares s_i:
  - Simulate a commitment-response pair (w_i, z) for challenge c via HVZK

➡ $Az = c \cdot vk + w_i$ but r_i is unknown

Threshold Raccoon
[dKMMPS24]

<u>Selective Security:</u>

- Simulation of signing oracles without shares s_i:
  - Simulate a commitment-response pair (w_i, z) for challenge c via HVZK

    ➡ $Az = c \cdot vk + w_i$ but r_i is unknown
  - Use properties of zero-share Δ_i to embed z into the signing session

Threshold Raccoon

[dKMMPS24]

Selective Security:

- Simulation of signing oracles without shares s_i:
  - Simulate a commitment-response pair (w_i, z) for challenge c via HVZK

    ➡ $Az = c \cdot vk + w_i$ but r_i is unknown

  - Use properties of zero-share Δ_i to embed z into the signing session
- Rewind to extract MSIS solution s

Threshold Raccoon

Adaptive Security

Adaptive Security:

Threshold Raccoon

Adaptive Security

Adaptive Security:

- Simulate a commitment-response pair (w_i, z) for challenge c via HVZK

Threshold Raccoon

Adaptive Security

Adaptive Security:

- Simulate a commitment-response pair (w_i, z) for challenge c via HVZK

➡ $Az = c \cdot vk + w_i$ but r_i is unknown

Threshold Raccoon

Adaptive Security

Adaptive Security:

- Simulate a commitment-response pair (w_i, z) for challenge c via HVZK

➡ $Az = c \cdot vk + w_i$ but r_i is unknown

- Use properties of zero-share Δ_i to embed z into the signing session

Threshold Raccoon

Adaptive Security

Adaptive Security:

- Simulate a commitment-response pair (w_i, z) for challenge c via HVZK

➡ $Az = c \cdot vk + w_i$ but r_i is unknown

- Use properties of zero-share Δ_i to embed z into the signing session
- If signer i is corrupted: PANIC

Threshold Raccoon

Adaptive Security

Adaptive Security:

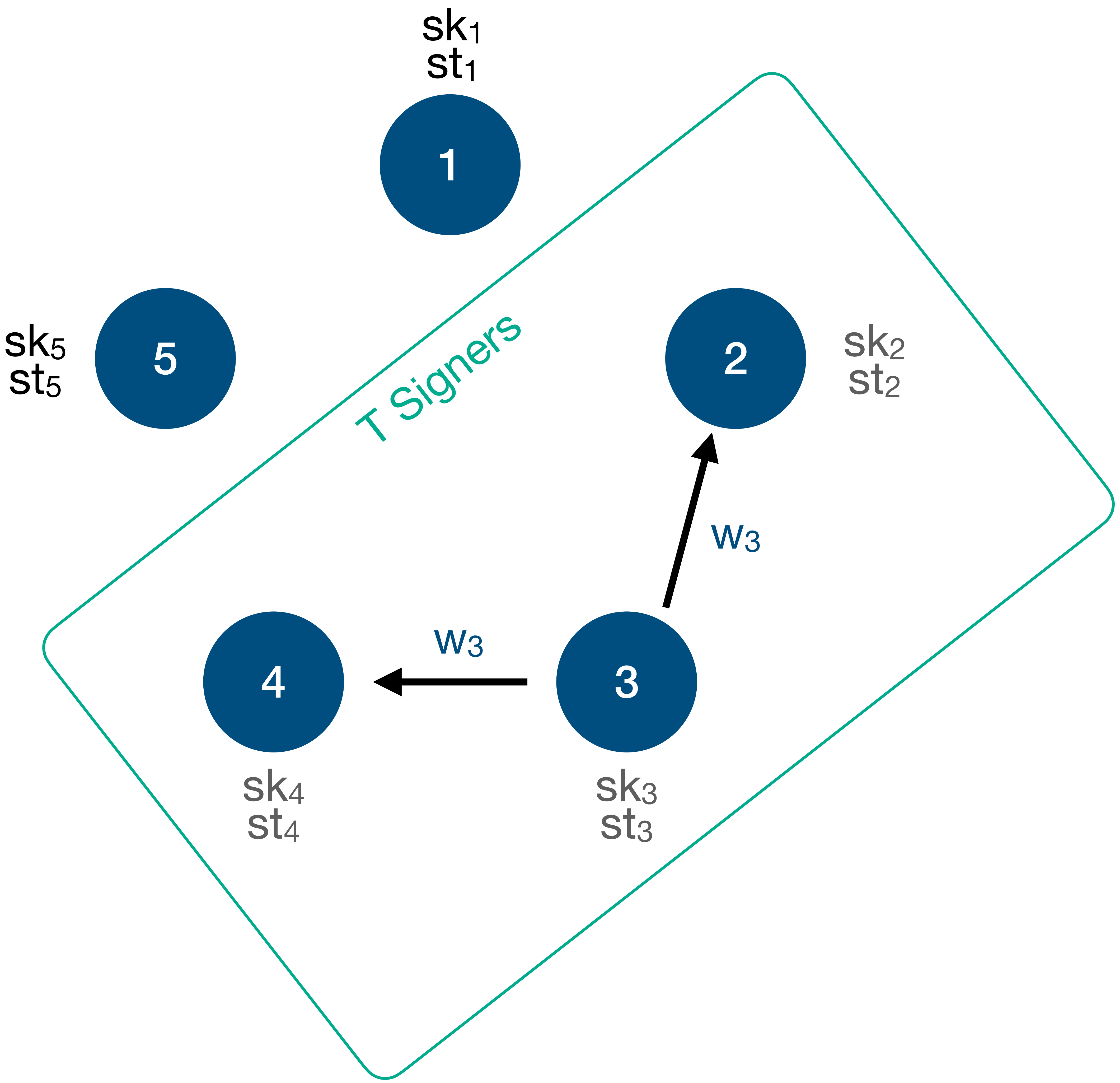
- Simulate a commitment-response pair (w_i, z) for challenge c via HVZK

➡ $Az = c \cdot vk + w_i$ but r_i is unknown

- Use properties of zero-share Δ_i to embed z into the signing session
- If signer i is corrupted: PANIC

Threshold Raccoon

Adaptive Security

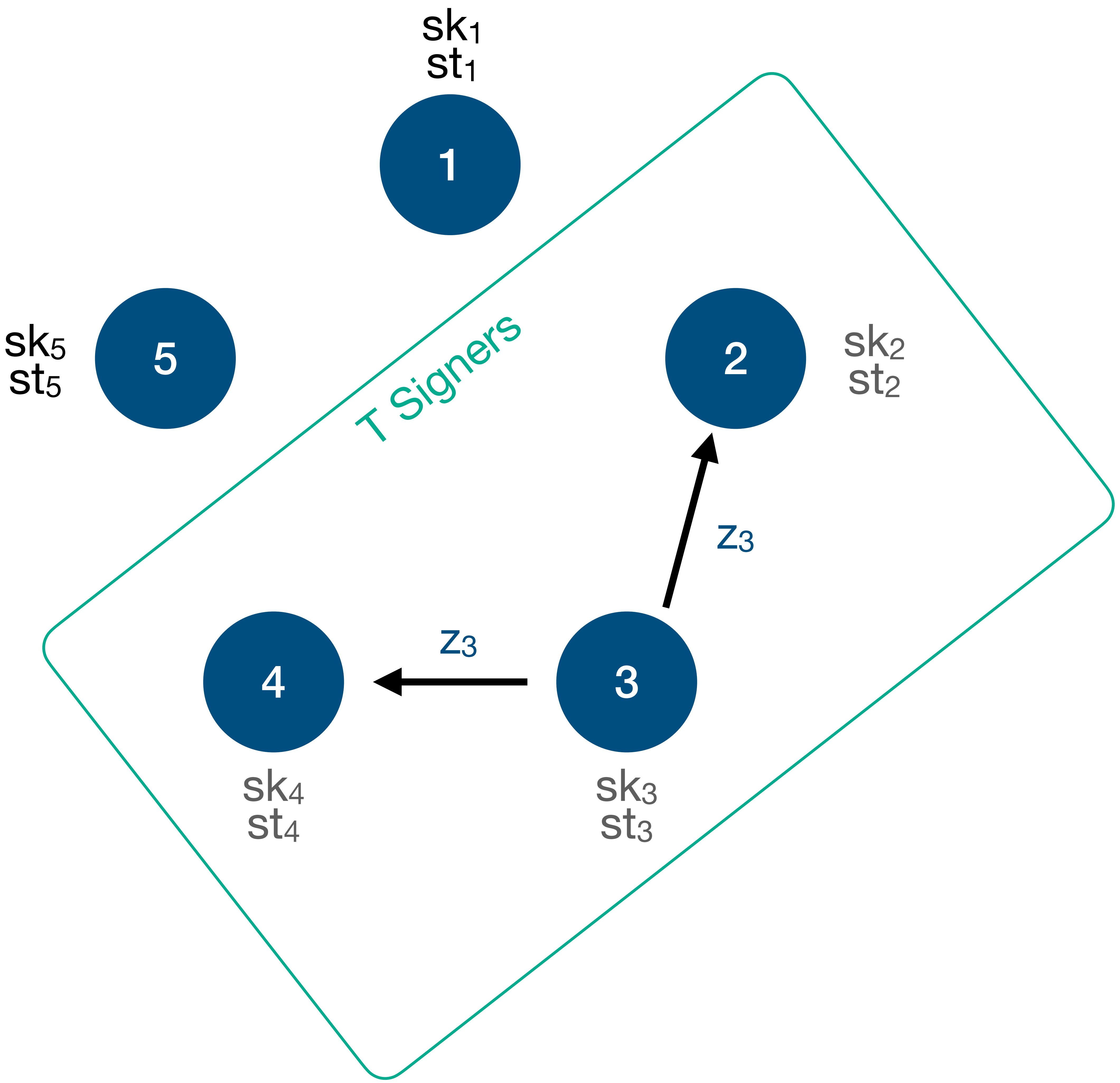


Scenario:

- $\mathscr{A}$ observes (w_2, z_2), (w_3, z_3), (w_4, z_4)

Threshold Raccoon

Adaptive Security

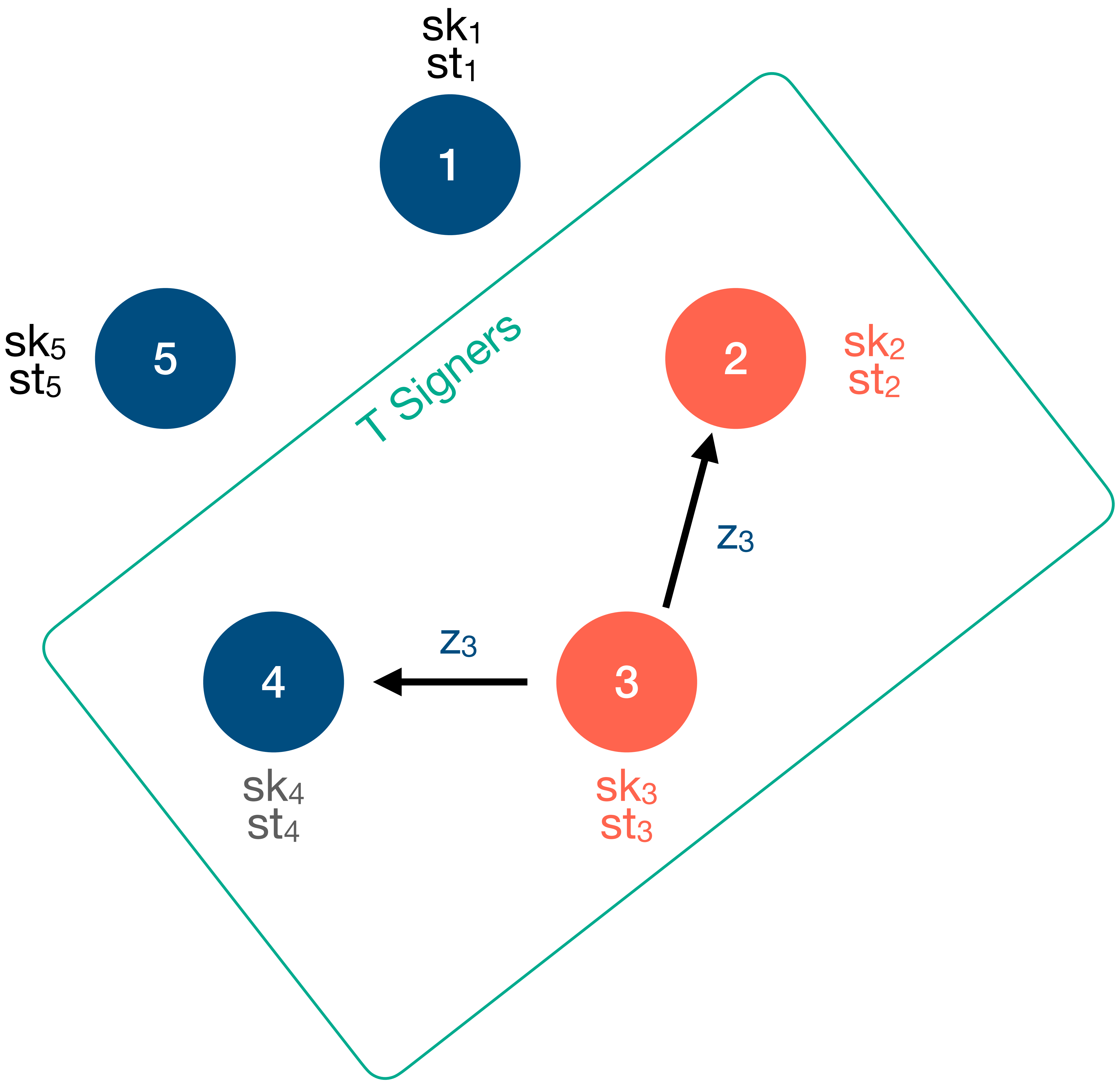


Scenario:

- $\mathscr{A}$ observes (w_2, z_2), (w_3, z_3), (w_4, z_4)

Threshold Raccoon

Adaptive Security

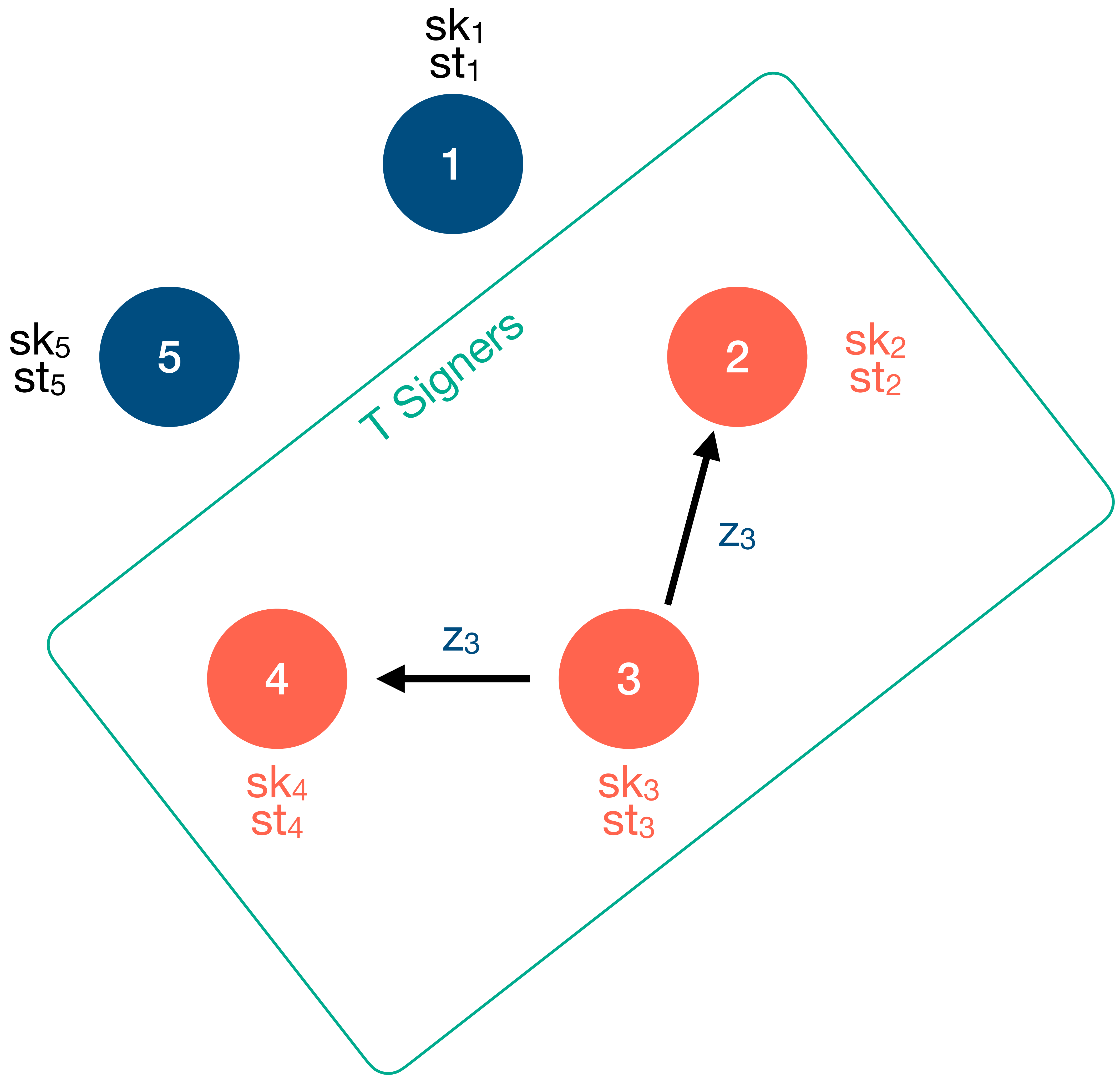


Scenario:

- $\mathscr{A}$ observes (w_2, z_2), (w_3, z_3), (w_4, z_4)
- $\mathscr{A}$ corrupts signer 2 and 3

Threshold Raccoon

Adaptive Security

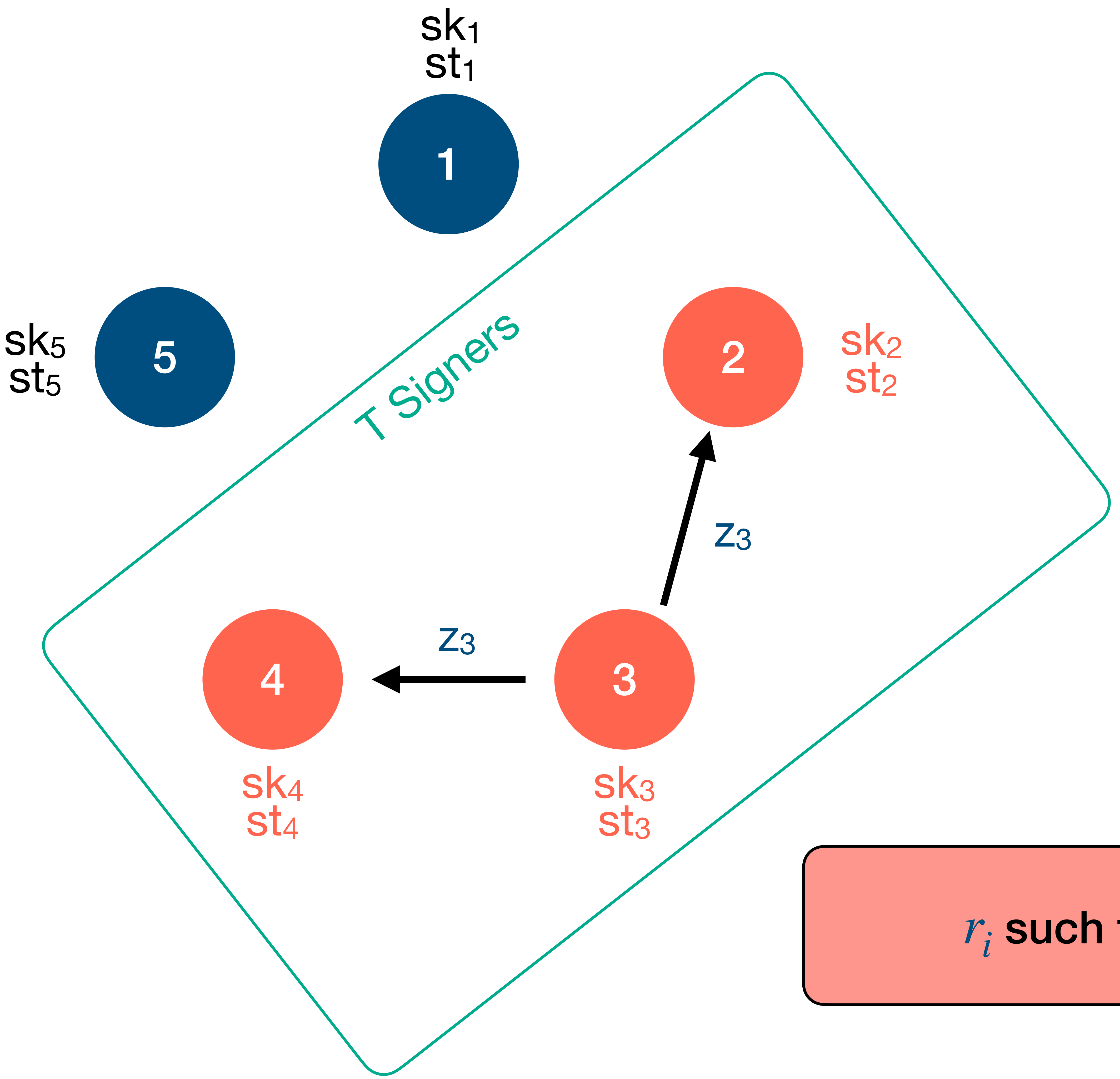


Scenario:

- $\mathscr{A}$ observes $(w_2, z_2), (w_3, z_3), (w_4, z_4)$
- $\mathscr{A}$ corrupts signer 2 and 3
- During rewinding, $\mathscr{A}$ corrupts user 4

Threshold Raccoon

Adaptive Security



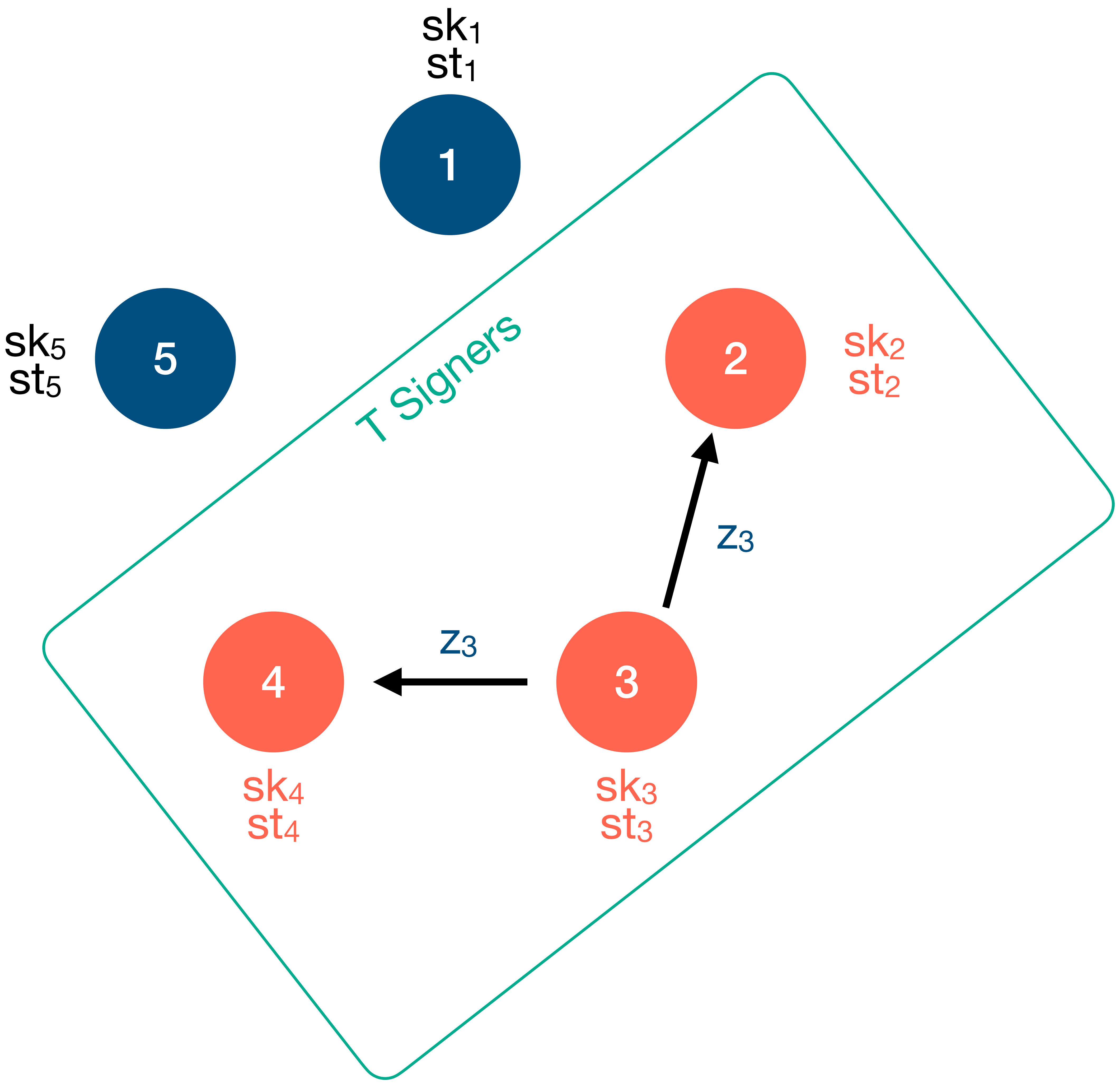
Scenario:

- $\mathscr{A}$ observes (w_2, z_2), (w_3, z_3), (w_4, z_4)
- $\mathscr{A}$ corrupts signer 2 and 3
- During rewinding, $\mathscr{A}$ corrupts user 4

r_i such that $w_i = Ar_i$ is expected in st_i

Threshold Raccoon

Adaptive Security



<u>Scenario:</u>

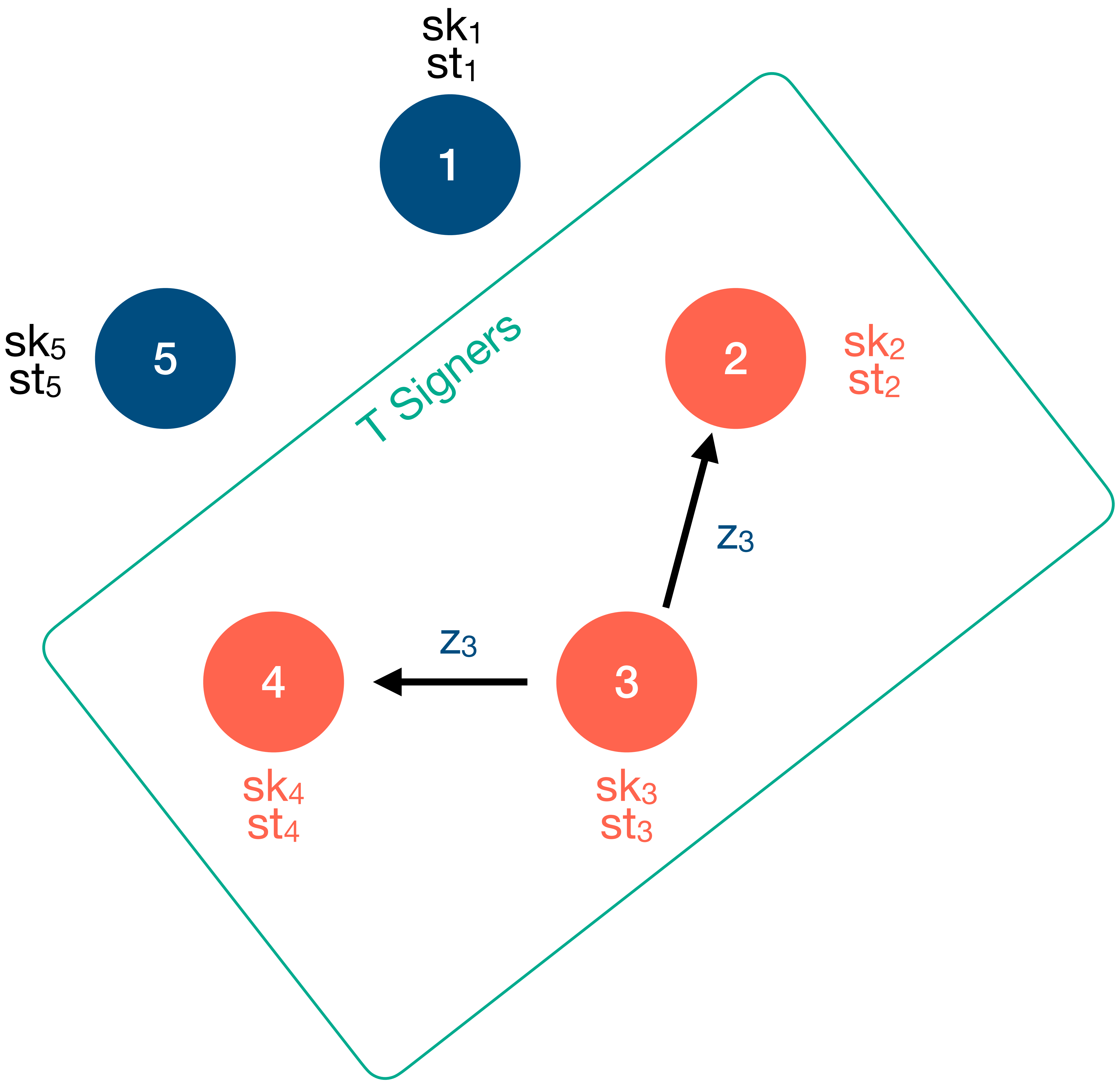
- $\mathscr{A}$ observes (w_2, z_2), (w_3, z_3), (w_4, z_4)
- $\mathscr{A}$ corrupts signer 2 and 3
- During rewinding, $\mathscr{A}$ corrupts user 4

<u>Conclusion:</u>

- The reduction has no space to embed a simulated w_i

Threshold Raccoon

Adaptive Security



Scenario:

- $\mathscr{A}$ observes (w_2, z_2), (w_3, z_3), (w_4, z_4)
- $\mathscr{A}$ corrupts signer 2 and 3
- During rewinding, $\mathscr{A}$ corrupts user 4

Conclusion:

- The reduction has no space to embed a simulated w_i
- The secret keys sk_i cannot be fixed in advance

Our Solution

More masking solves the problem

Our Protocol

4-round Threshold Raccoon

Round 1:

- $r_i \leftarrow \chi$
- $w_i \leftarrow A \cdot r_i$
- *sample 0-share* $\tilde{\Delta}_i$
- $\tilde{w}_i \leftarrow w_i + \tilde{\Delta}_i$
- $\mathsf{cmt}_i = G(\tilde{w}_i)$
- *send* cmt_i

Our Protocol

4-round Threshold Raccoon

<u>Round 1:</u>

- $r_i \leftarrow \chi$
- $w_i \leftarrow A \cdot r_i$
- *sample 0-share* $\tilde{\Delta}_i$
- $\tilde{w}_i \leftarrow w_i + \tilde{\Delta}_i$
- $\mathsf{cmt}_i = G(\tilde{w}_i)$
- *send* cmt_i

0-shares are sampled via RO

$\Delta_i = \sum_{j \in S} F(k_{i,j}, \mathsf{sid}) - F(k_{j,i}, \mathsf{sid})$

Note:
Requires non-repeating *sid*
which requires state-keeping

This *sid* can be established in
additional round

Our Protocol

4-round Threshold Raccoon

Round 1:

- $r_i \leftarrow \chi$
- $w_i \leftarrow A \cdot r_i$
- *sample 0-share* $\tilde{\Delta}_i$
- $\tilde{w}_i \leftarrow w_i + \tilde{\Delta}_i$
- $\mathsf{cmt}_i = G(\tilde{w}_i)$
- *send* cmt_i

Round 2:

- *sign view*

0-shares are sampled via RO

$$\Delta_i = \sum_{j \in S} F(k_{i,j}, \mathsf{sid}) - F(k_{j,i}, \mathsf{sid})$$

Our Protocol

4-round Threshold Raccoon

<u>Round 1:</u>

- $r_i \leftarrow \chi$
- $w_i \leftarrow A \cdot r_i$
- *sample 0-share* $\tilde{\Delta}_i$
- $\tilde{w}_i \leftarrow w_i + \tilde{\Delta}_i$
- $\mathsf{cmt}_i = G(\tilde{w}_i)$
- *send* cmt_i

<u>Round 2:</u>

- *sign view*

<u>Round 3:</u>

- *check signature*
- *send* $\tilde{w}_i$

0-shares are sampled via RO

$\Delta_i = \sum_{j \in S} F(k_{i,j}, \mathsf{sid}) - F(k_{j,i}, \mathsf{sid})$

Our Protocol

4-round Threshold Raccoon

Round 1:

- $r_i \leftarrow \chi$
- $w_i \leftarrow A \cdot r_i$
- *sample 0-share* $\tilde{\Delta}_i$
- $\tilde{w}_i \leftarrow w_i + \tilde{\Delta}_i$
- $\mathsf{cmt}_i = G(\tilde{w}_i)$
- *send* cmt_i

Round 2:

- *sign view*

Round 3:

- *check signature*
- *send* $\tilde{w}_i$

Round 4:

- *check* $\mathsf{cmt}_i = G(\tilde{w}_i)$
- $w = \sum_{j \in S} \tilde{w}_i$
- $c = H(vk, w, m)$
- *sample 0-share* Δ_i
- $z_i = c \cdot L_{S,i} \cdot s_i + r_i + \Delta_i$
- send z_i

0-shares are sampled via RO

$$\Delta_i = \sum_{j \in S} F(k_{i,j}, \mathsf{sid}) - F(k_{j,i}, \mathsf{sid})$$

Simplified

Intuition:

- The masking via 0-shares $\tilde{\Delta}_i$ and Δ_i minimize information learned from signing sessions:

  - $s = \sum_{j \in S} L_{S,j}\, s_i$
  - $\tilde{w}_i = w_i + \tilde{\Delta}_i$
  - $\tilde{z}_i = c \cdot L_{S,i} \cdot s_i + r_i + \Delta_i$
  - $w_i = [A \mid I]\, r_i$
  - $0 = \sum_{j \in S} \tilde{\Delta}_j = \sum_{j \in S} \Delta_j$

statistically hidden

determined

Simplified

Intuition:

The protocol message are uniform conditioned on the final signature verifying

- The masking via 0-shares $\tilde{\Delta}_i$ and Δ_i minimize information learned from signing sessions:

- $s = \sum_{j \in S} L_{S,j}\, s_i$
- $\tilde{w}_i = w_i + \tilde{\Delta}_i$
- $\tilde{z}_i = c \cdot L_{S,i} \cdot s_i + r_i + \Delta_i$

- $w_i = [A \mid I]\, r_i$
- $0 = \sum_{j \in S} \tilde{\Delta}_j = \sum_{j \in S} \Delta_j$

statistically hidden

determined

Security Proof

Simplified

Intuition:

The protocol message are uniform conditioned on the final signature verifying

Security Proof

Simplified

Intuition:

The protocol message are uniform conditioned on the final signature verifying

Security Proof

Simplified

Intuition:

The protocol message are uniform conditioned on the final signature verifying

- Simulate one w_i via HVZK and the others honestly (allows to simulate signing)

Security Proof

Simplified

Intuition:

The protocol message are uniform conditioned on the final signature verifying

- Simulate one w_i via HVZK and the others honestly (allows to simulate signing)
- On corruption, sample s_i at random and choose one honest w_i per session

Security Proof

Simplified

Intuition:

The protocol message are uniform conditioned on the final signature verifying

- Simulate one w_i via HVZK and the others honestly (allows to simulate signing)
- On corruption, sample s_i at random and choose one honest w_i per session
- Program RO for 0-shares for consistency

Summary

Results:

- Proof technique for adaptive security in the ROM
- State-free security proof for Threshold Raccoon
- Techniques to prove stronger unforgeability notions