

# Defining Unforgeability for Threshold Signatures

Presented at MPTS 2026

NIST Workshop on Multi-Party Threshold Schemes

January 26, 2026

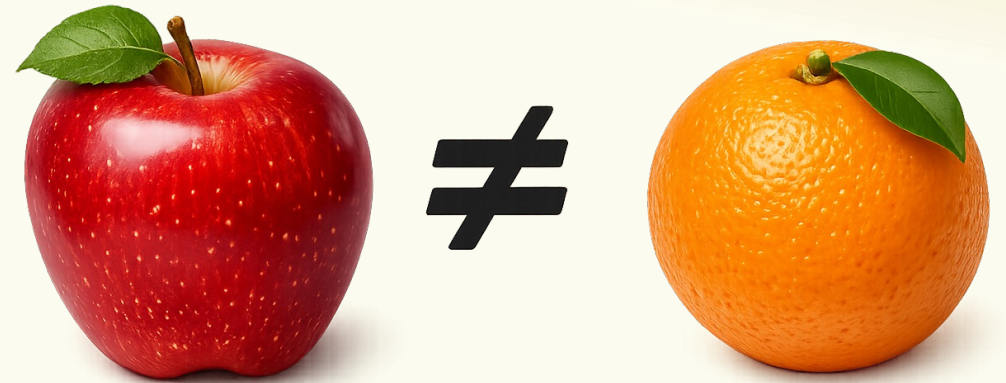
**W** PAUL G. ALLEN SCHOOL  
OF COMPUTER SCIENCE & ENGINEERING

**Stefano Tessaro**

tessaro@cs.washington.edu

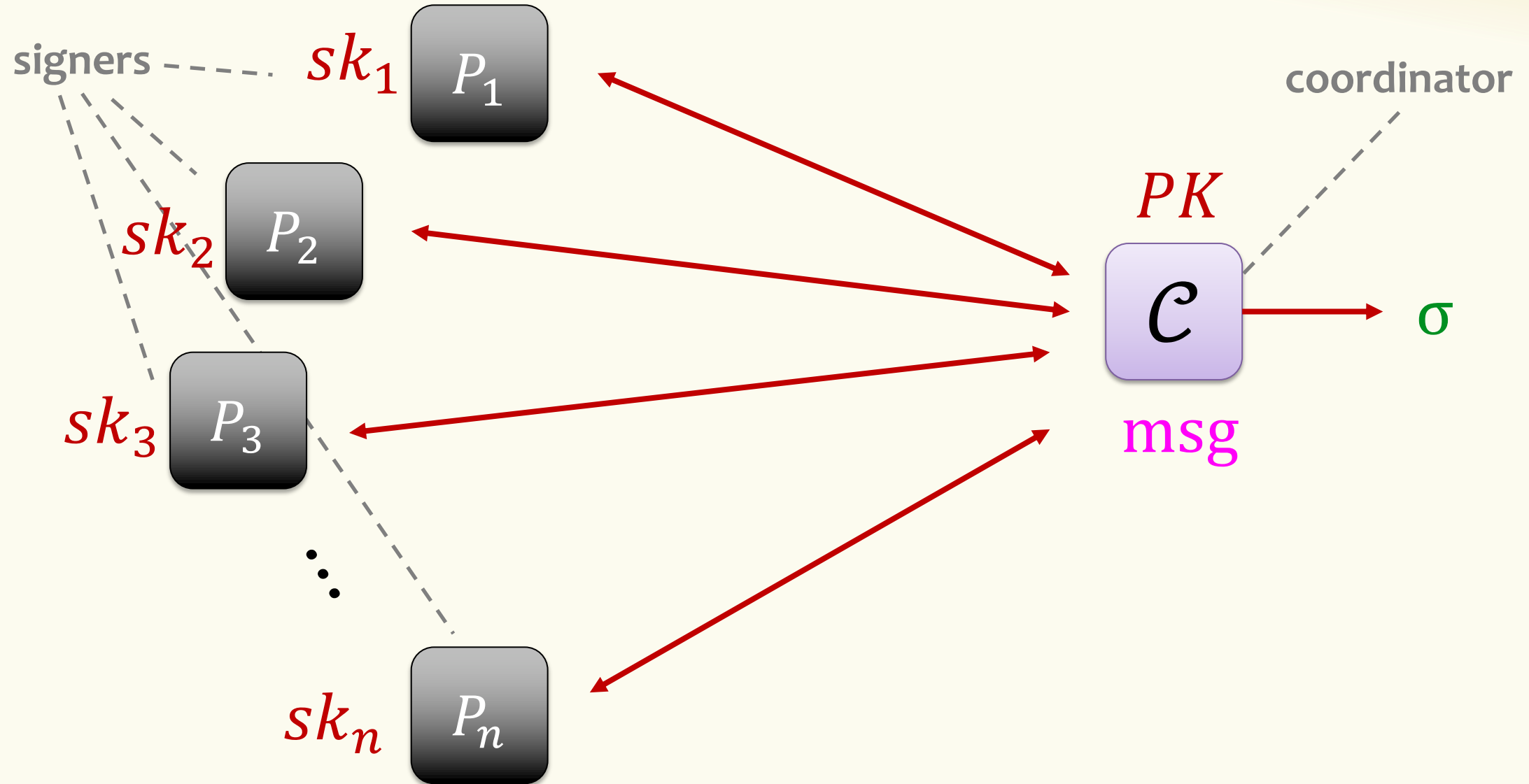
## Why this talk?

Pervasive definitional inconsistencies  
make threshold signatures  
incomparable

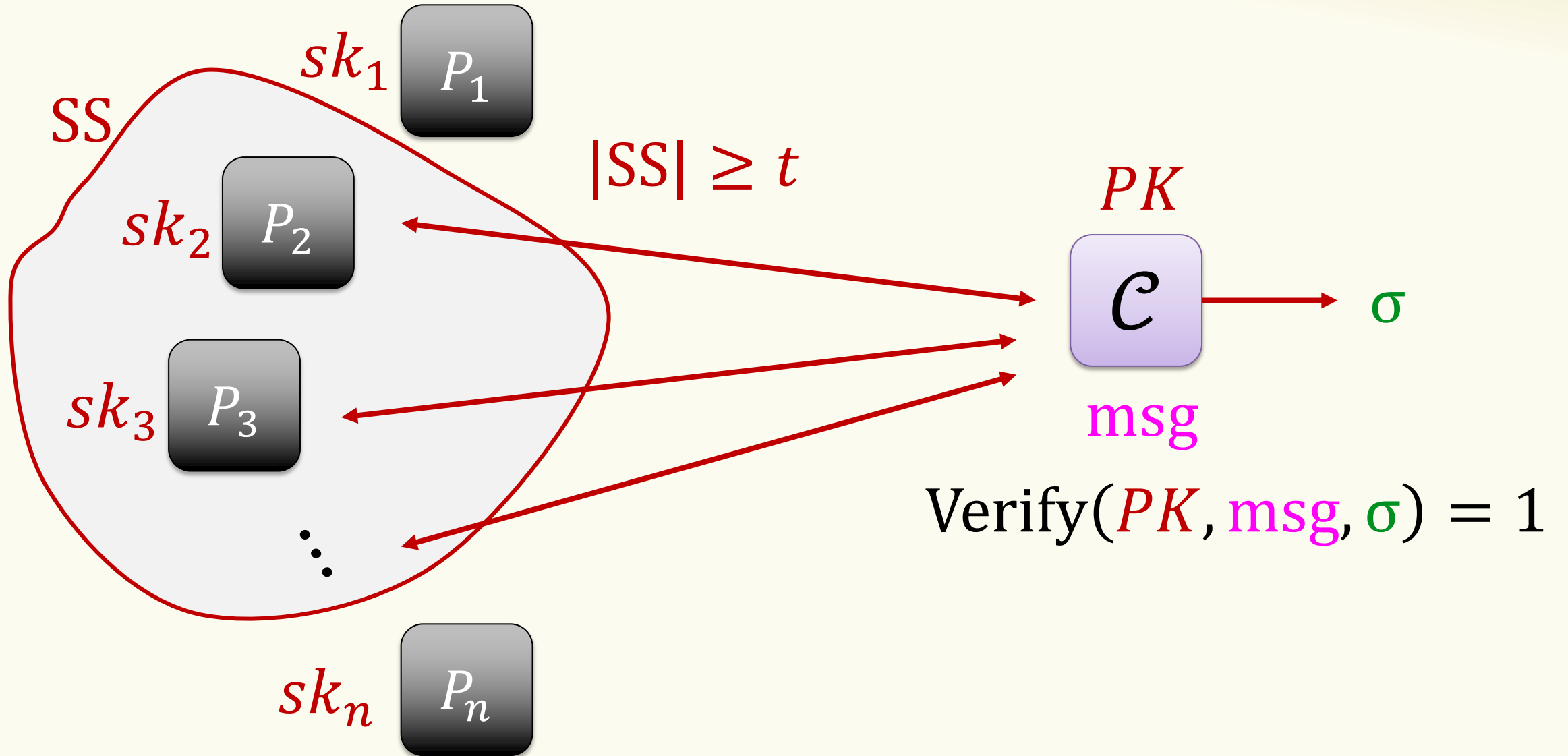


This talk: **Overview of unforgeability issues & definitions**

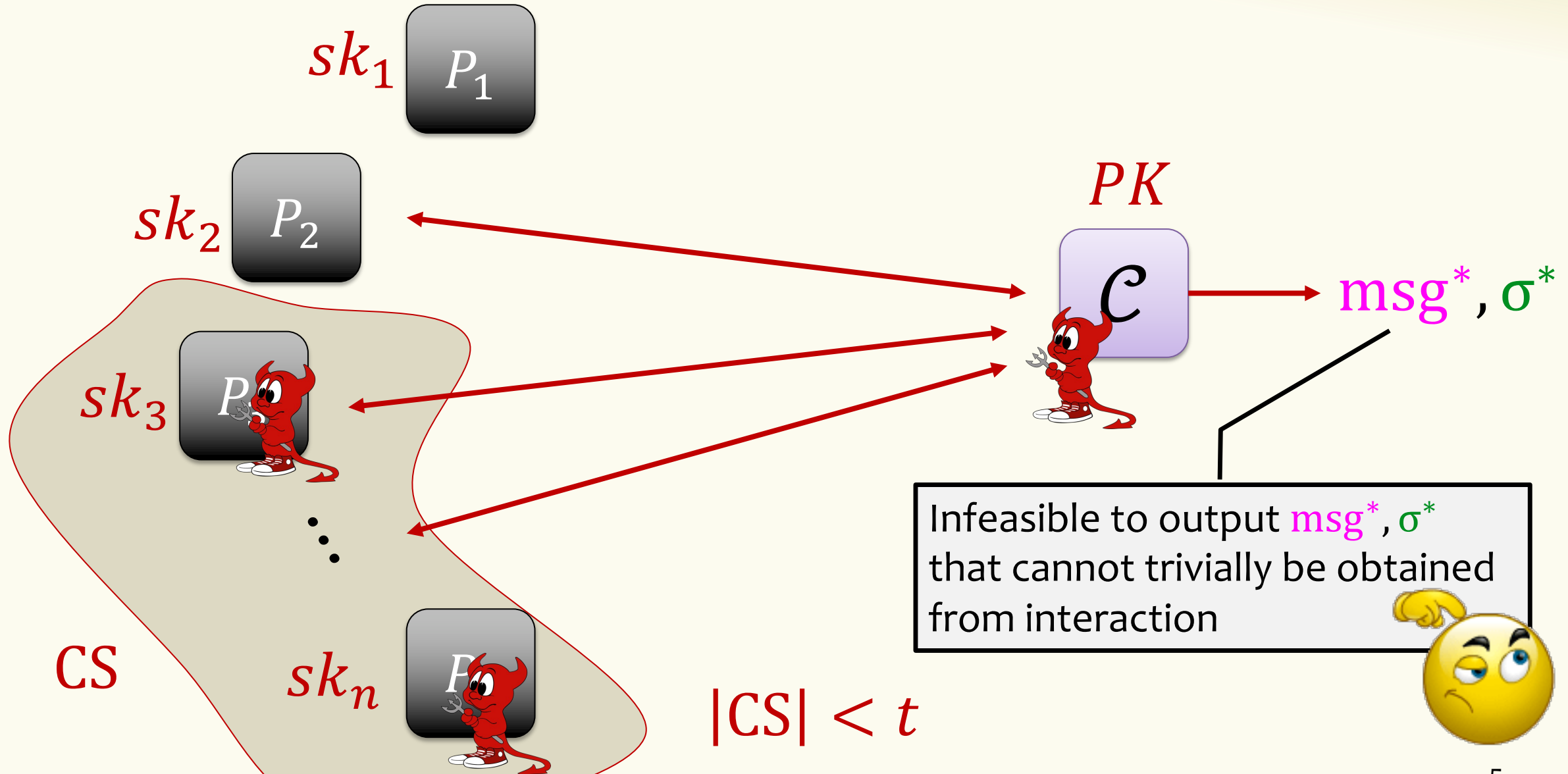
# This model: Coordinator Model



# $(t, n)$ -TS – Correctness



# $(t, n)$ -TS – Unforgeability



Challenge: it is hard to define exact set of messages for which a signature has been issued

**Signing Oracle**

Plain signatures

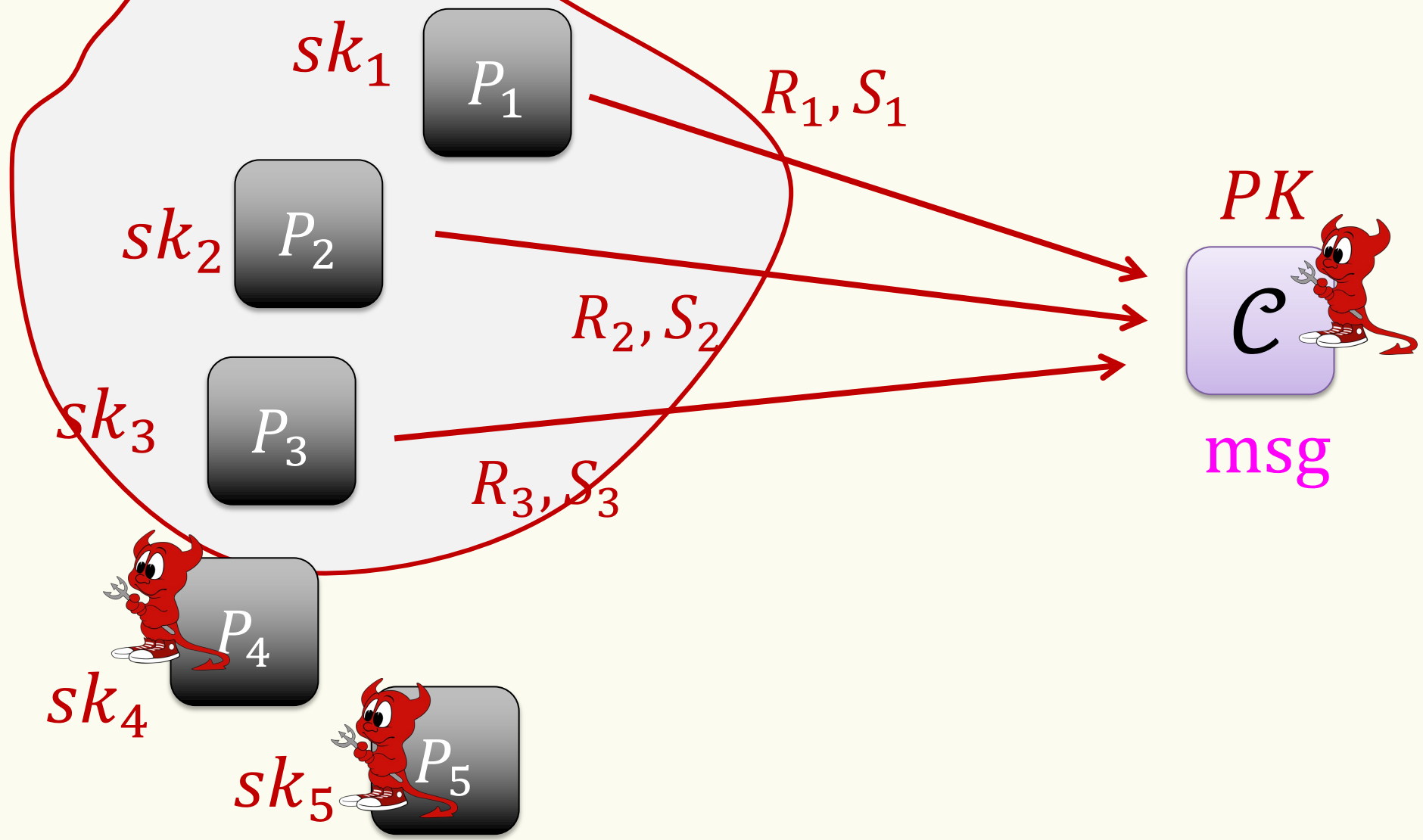


**(Partial) Interactive Execution**

Threshold signatures

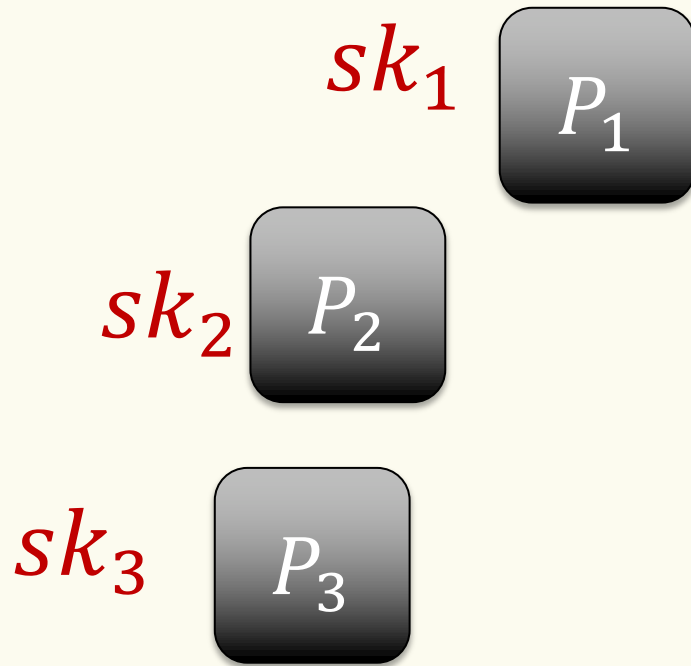
# Example – FROST1

$t = 3, SS = \{1,2,3\}, CS = \{4,5\}$



# Example – FROST1

$t = 3, SS = \{1,2,3\}, CS = \{4,5\}$



$R_1, S_1$       msg

$\tilde{R}_2 = \tilde{r}_2 G, \tilde{S}_2 = \tilde{s}_2 G$

$\tilde{R}_3 = \tilde{r}_3 G, \tilde{S}_3 = \tilde{s}_3 G$



msg  $\tilde{r}_2, \tilde{r}_3, \tilde{s}_2, \tilde{s}_3$   
 $sk_4, sk_5$

# Example – FROST1

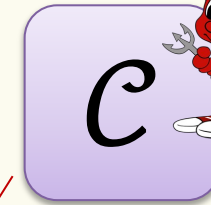
$$t = 3, \text{ SS} = \{1,2,3\}, \text{ CS} = \{4,5\}$$

$sk_1$



$z_1$

$PK$



$\sigma = (R, z)$

$msg, \sigma$

$msg \quad \tilde{r}_2, \tilde{r}_3, \tilde{s}_2, \tilde{s}_3$   
 $sk_4, sk_5$

$$R = R_1 + d_1 S_1 + \tilde{R}_2 + d_2 \tilde{S}_2 + \tilde{R}_3 + d_3 \tilde{S}_3$$

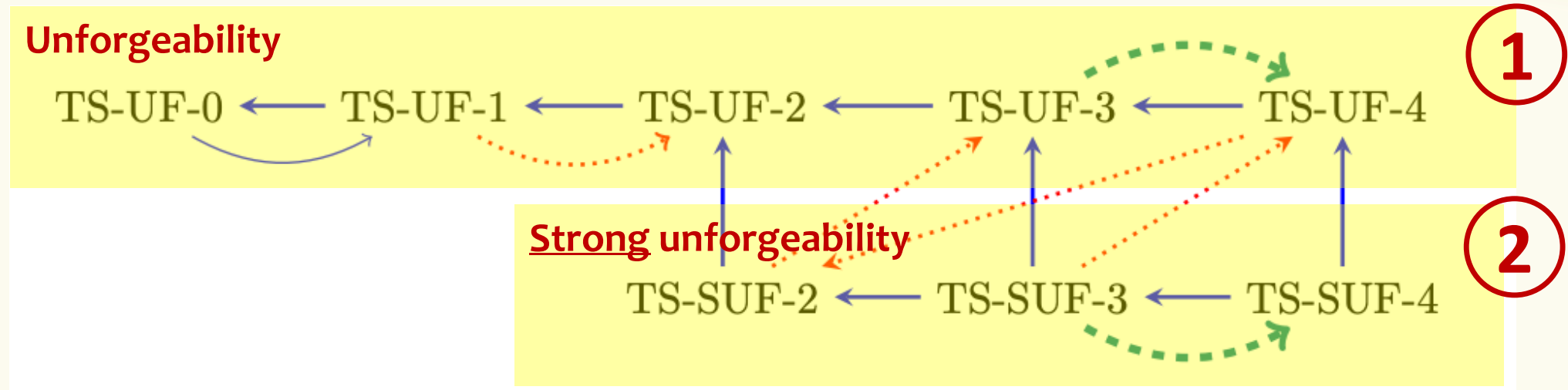
$$d_i = H_0(PK, \{R_1, S_1, \tilde{R}_2, \tilde{S}_2, \tilde{R}_3, \tilde{S}_3\}, i)$$

$$z_1 = r_1 + d_1 s_1 + c \lambda_1 sk_1$$

$$c = H(PK, R, msg)$$

$$z = z_1 + \tilde{r}_2 + d_2 \tilde{s}_2 + \tilde{r}_3 + d_3 \tilde{s}_3 + c(\lambda_4 sk_4 + \lambda_5 sk_5)$$

# The Definitional Hierarchy [Bellare, Crites, Komlo, Maller, T, Zhu, CRYPTO '22]



$\longrightarrow$  implication

$\dashrightarrow$  transformation

$\xrightarrow{\text{lossy}}$  (lossy) implication

$\cdots\rightarrow$  separation

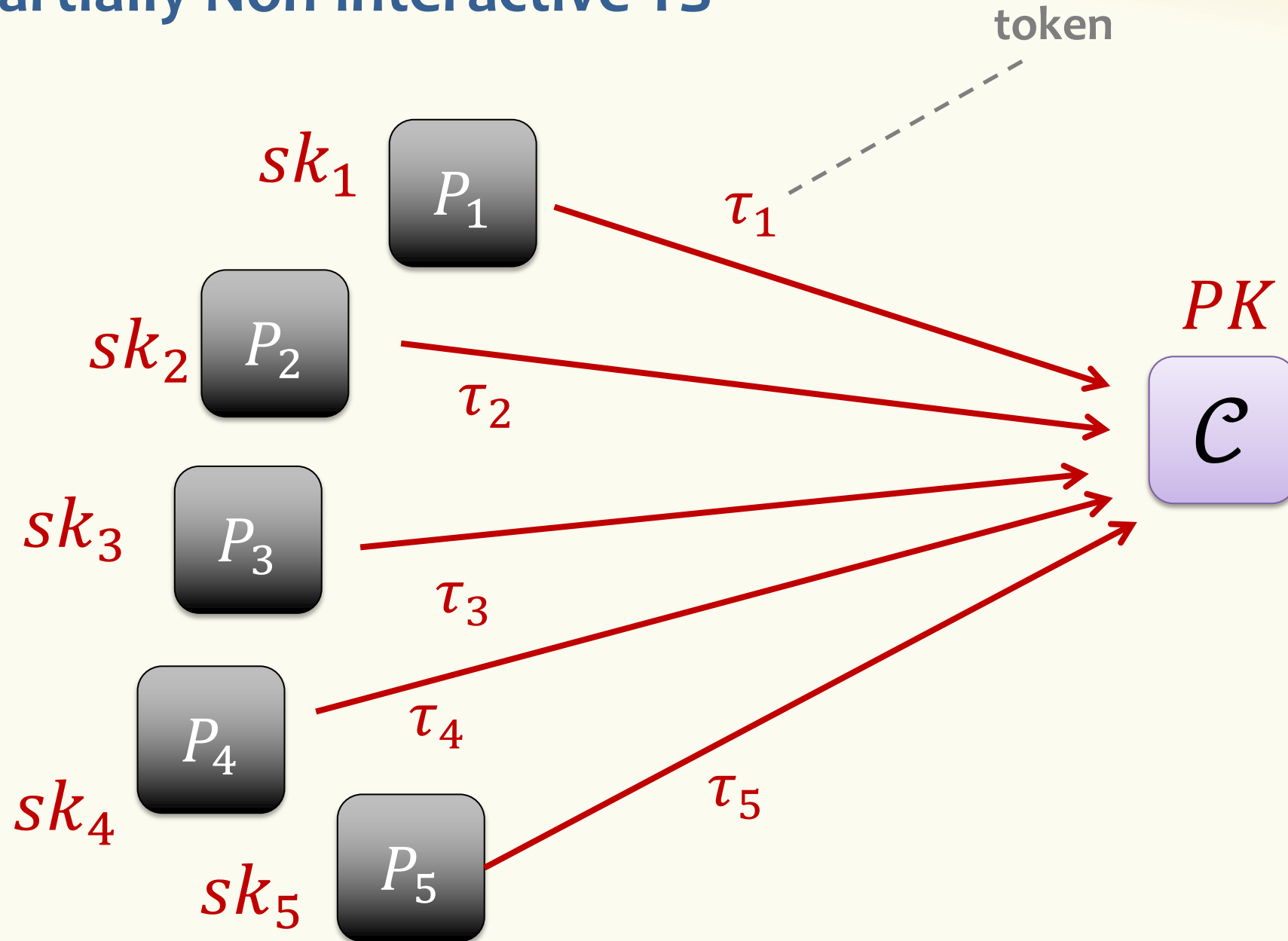
Tailored at “**partially non-interactive schemes**” like FROST

Extensions (multi-round, DKG, ...) considered by follow-ups

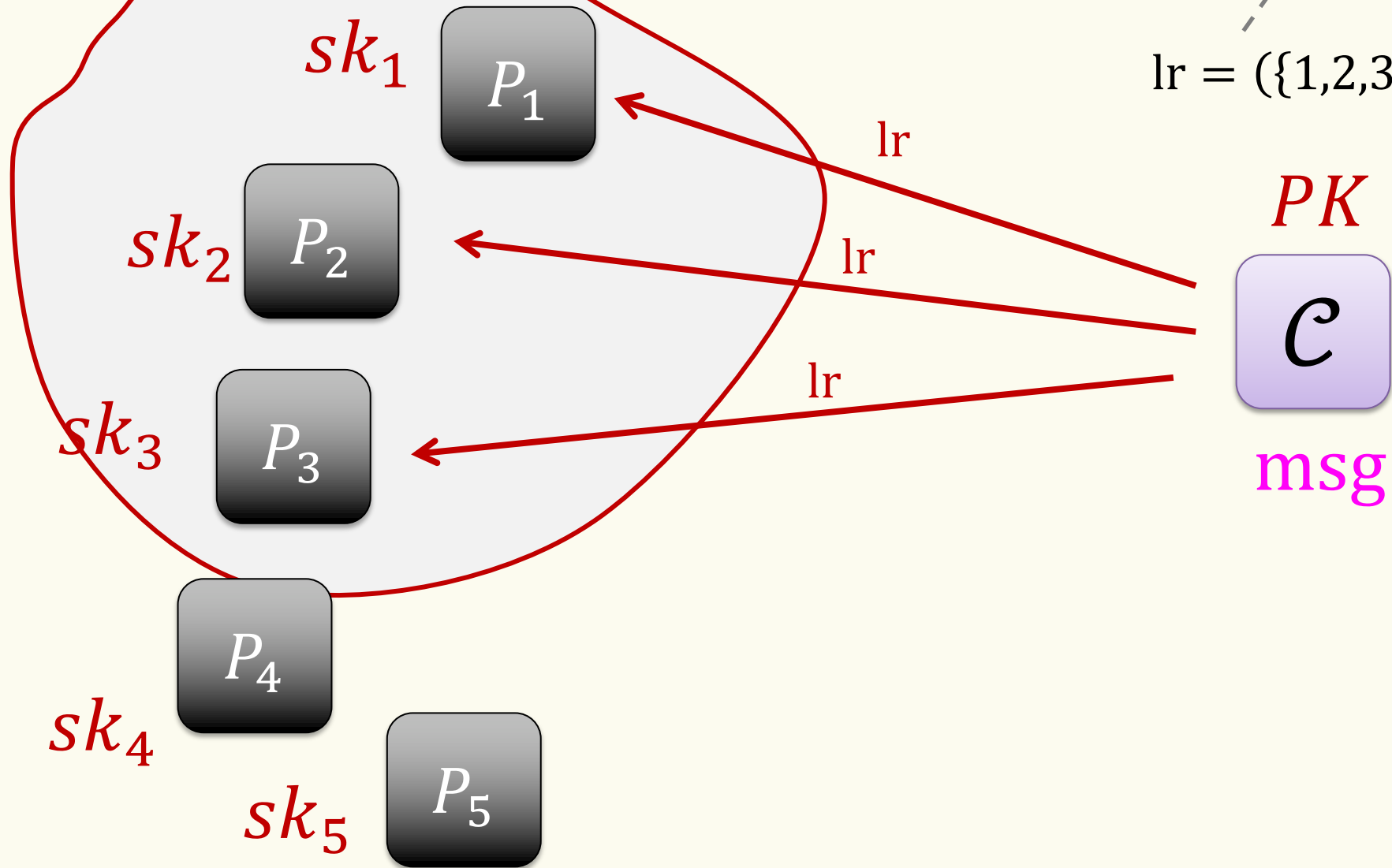
[Cremers, Peltonen, Zhao, ePrint '24] [Lehmann, Nazarian, Özbay, EUROCRYPT '25]

[Azari, Boschini, Hostáková, Reichle, TCC '25] [Niot, Reichle, Takemure, ePrint '25]

# Partially Non-interactive TS



# Partially Non-interactive TS



leader request

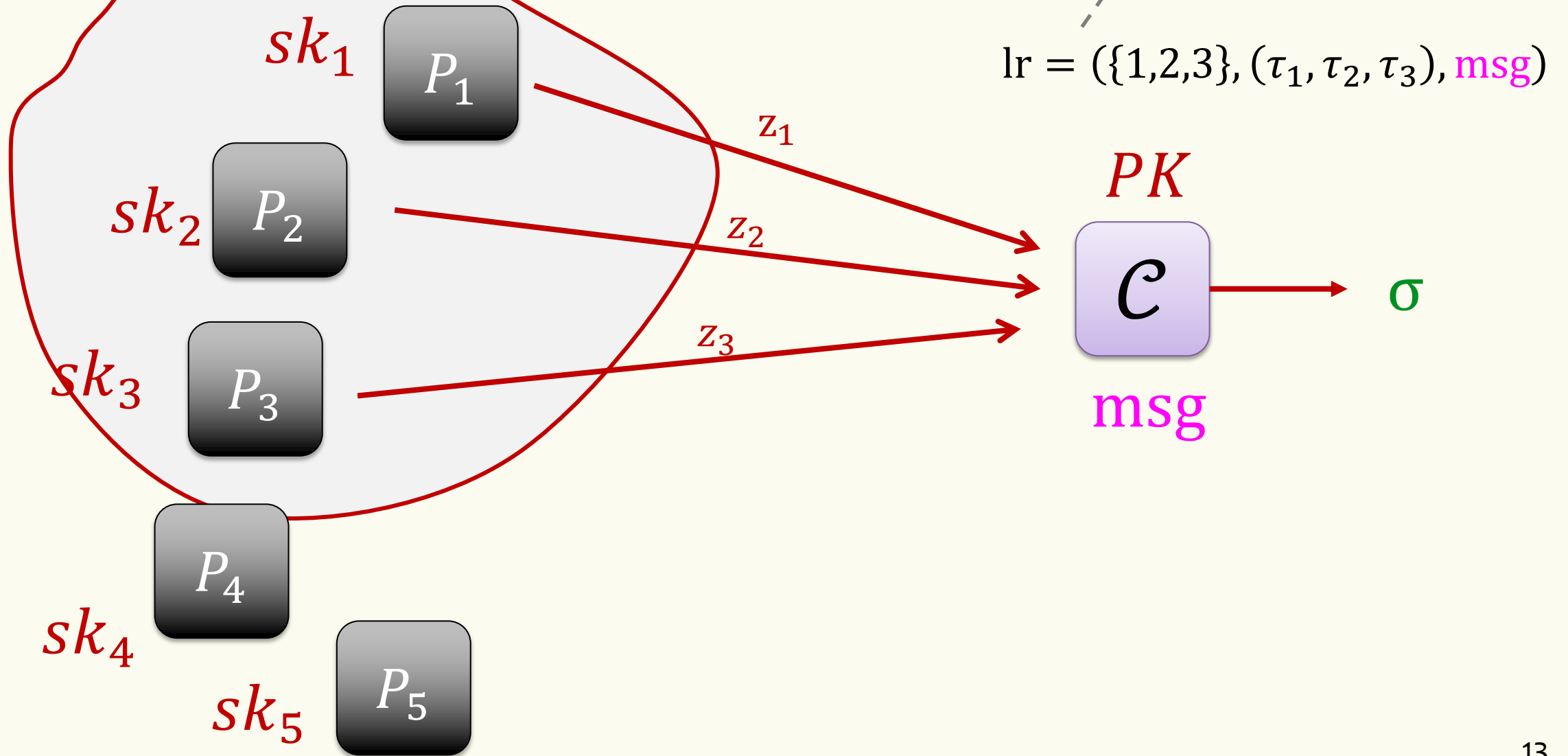
$$lr = (\{1,2,3\}, (\tau_1, \tau_2, \tau_3), msg)$$

$PK$

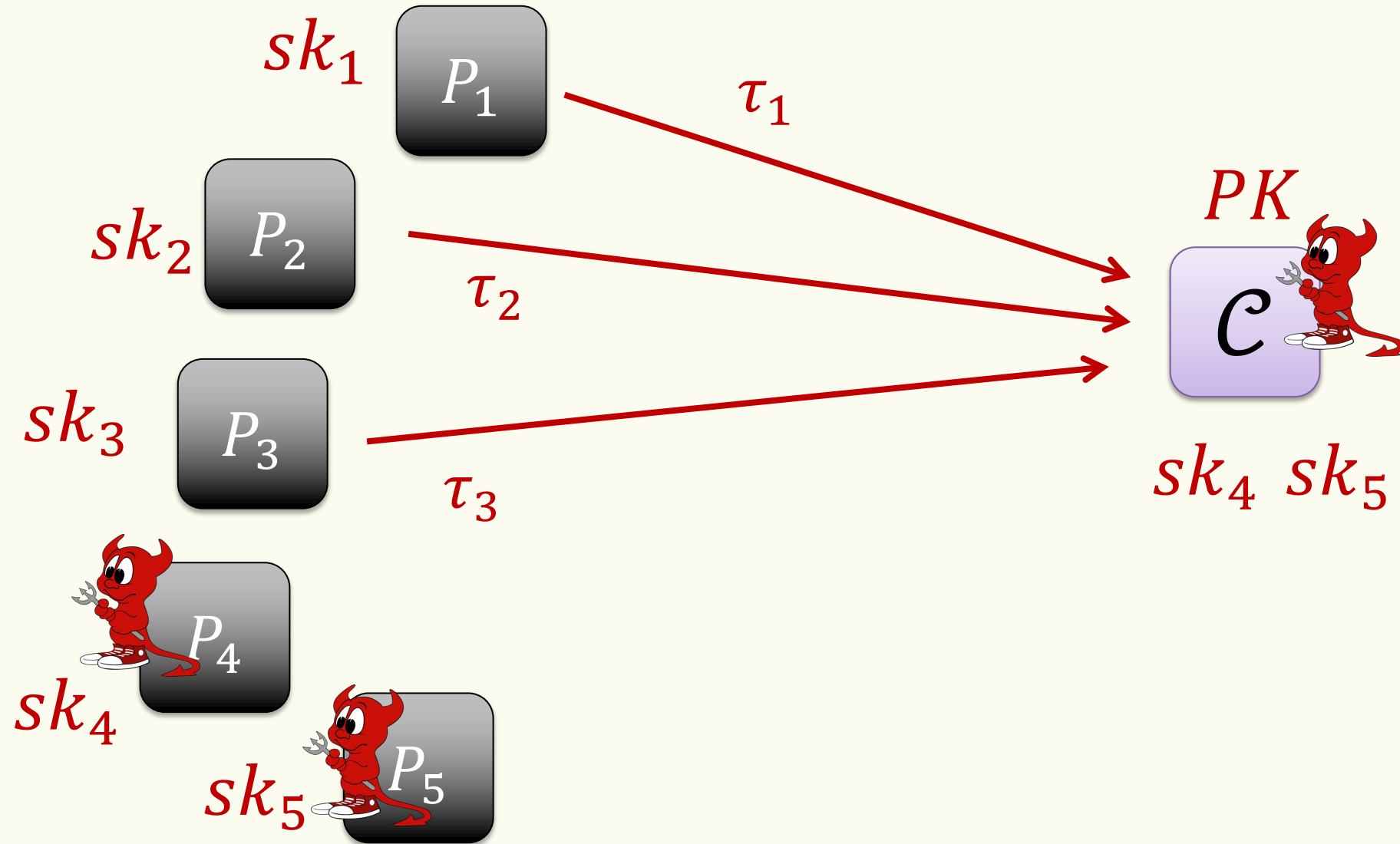
$C$

$msg$

# Partially Non-interactive TS

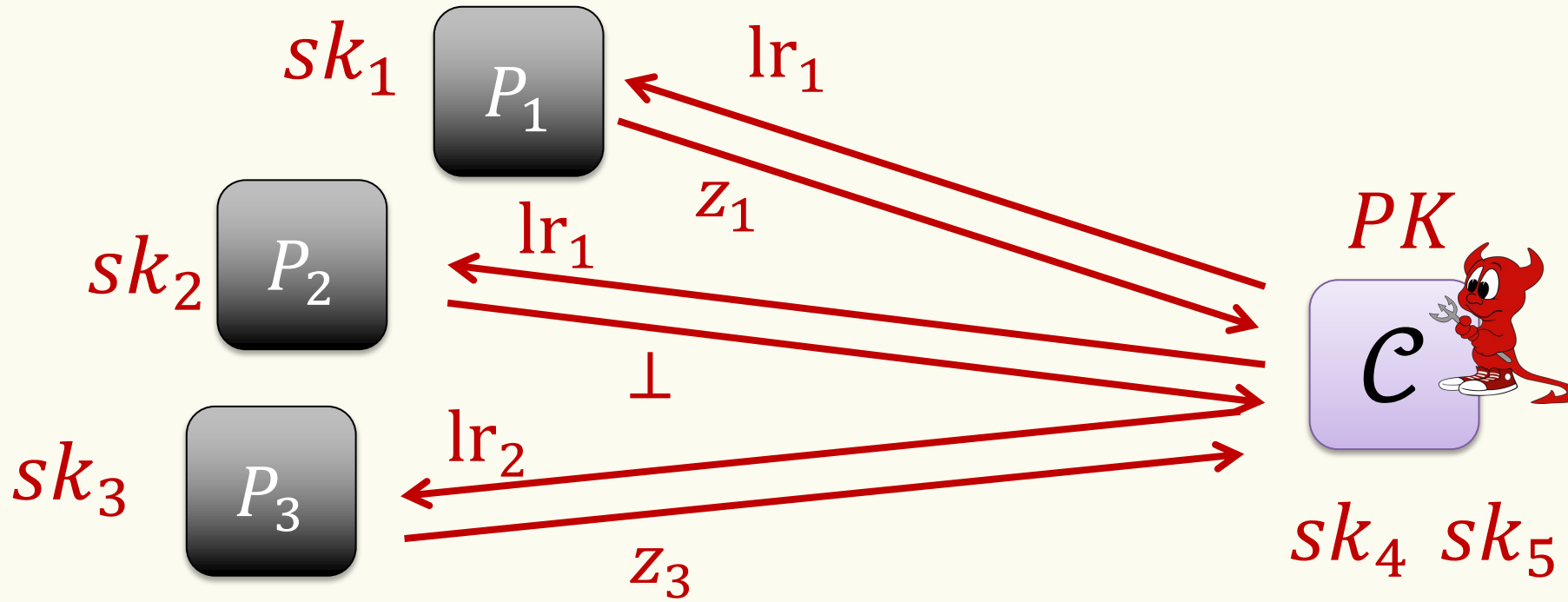


# Attack Model



# Attack Model

$lr_1 \neq lr_2$ , both for  $msg$



$$S_1(msg) = \{1,3\}$$

signers that answered some LR for  $msg$

$$S_2(lr_1) = \{1\}$$

$$S_2(lr_2) = \{3\}$$

signers that answered specific LR

# The Hierarchy

A signature for  $msg$  has been issued if:

- **TS-UF-0**  $S_1(msg) \neq \emptyset$  **Widely used!**
- **TS-UF-1**  $|S_1(msg)| \geq t - |CS|$   
**BLS** [Groth '21, BCKMTZ '22] **RSA** [Shoup, '00]
- **TS-UF-2**  $\exists lr: lr.msg = msg \wedge |S_2(lr)| \geq t - |CS|$   
**FROST2** [BCKMTZ '22]
- **TS-UF-3**  $\exists lr: lr.msg = msg \wedge |S_2(lr)| \geq t - |CS| \wedge S_2(lr) = S_3(lr)$   
**FROST1** [BCKMTZ '22]
- **TS-UF-4**  $\exists lr: lr.msg = msg \wedge |S_2(lr)| \geq t - |CS| \wedge S_2(lr) = S_4(lr)$
- **???**

signers in  $lr$  associated with honest token

Honest signers in  $lr$

# Achieving TS-UF-4

Two modifications, both require PKI:

1. **Generic:** Each signer signs own tokens; LR discarded if any token not validly signed

[Bellare, Crites, Komlo, Maller, T, Zhu, CRYPTO '22]

2. **Less generic:** Replace  $z_i$  with  $z'_i = z_i + \delta_i$  s.t.  $\sum_{i \in SS} \delta_i = 0$

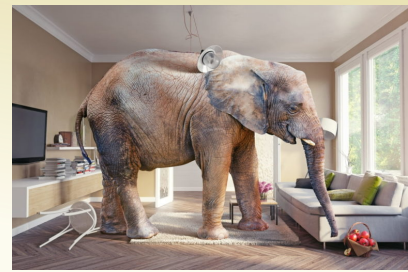
[Crites, Katz, Komlo, T, Zhu, CRYPTO '25] for FROST, in the AGM

[Zhu, T, CRYPTO '25] Transformation

[Niot, Reichle, Takemure, ePrint '25]

...

# What about UC security? (Addressing the elephant in the room)



Actually: Unforgeability issues addressed by hierarchy **entirely orthogonal** to UC security definitions

**FUNCTIONALITY 5** (Schnorr Signing Functionality  $\mathcal{F}$ ).

Upon receiving  $(\text{sign}, \text{sid}, m, Q, Q_1, \dots, Q_n, d_i)$  from  $t+1$  different parties  $P_i$ , functionality  $\mathcal{F}$  verifies that all parties sent the same  $(\text{sid}, m, Q, Q_1, \dots, Q_n)$ , that  $Q_i = d_i \cdot G$  for all inputs received, and that there exists a degree- $t$  polynomial  $p(x)$  such that  $p(i) \cdot G = Q_i$  for  $i = 1, \dots, n$ , and  $p(0) \cdot G = Q$ . If no, then it does nothing. Else, it chooses a random  $k \leftarrow \mathbb{Z}_q$ , and computes  $R = k \cdot G$ ,  $e = H(m \| Q \| R)$  and  $s = k - e \cdot d \pmod{q}$ . Then,  $\mathcal{F}$  sends  $(\text{sid}, e, s)$  to  $\mathcal{C}$ .

[Lindell, CiC '24]

Functionality reveals signature to adversary as soon as sufficiently many parties have started the protocol

# The Definitional Hierarchy [Bellare, Crites, Komlo, Maller, T, Zhu, CRYPTO '22]

## Unforgeability

TS-UF-0 ← TS-UF-1 ← TS-UF-2 ← TS-UF-3 ← TS-UF-4

1

## Strong unforgeability

TS-SUF-2 ← TS-SUF-3 ← TS-SUF-4

2

→ implication

- - - → transformation

→ (lossy) implication

⋯ → separation

# Strong Unforgeability

Adversary wins even if it outputs a new signature for already signed message

- Requires defining which signatures have been learned
- Even more annoying at the general level\* 😓
  - [CBKMTZ22] gives definition for schemes where LR request uniquely defines signature (TS-SUF- $\{2,3,4\}$ )

Better: **one-more unforgeability (OMUF)** [Navot, T, '24]

- equivalent to strong unforgeability for (plain) signatures

\* Easier to define within UC security

## One-more unforgeability (OMUF) [Navot, T, ASIACRYPT '24]

Informally: Adversary wins if it outputs more message-signature pairs than number of signing sessions it "concludes"

Caveat: How to define "conclude"??

- [Navot, T, ASIACRYPT '24] "at least an honest party sends final message"
- [Azari, Boschini, Hostáková, Reichle, TCC '25] "all honest parties send final message" (TS-UF-4 analogue)

## One-more unforgeability – Notes

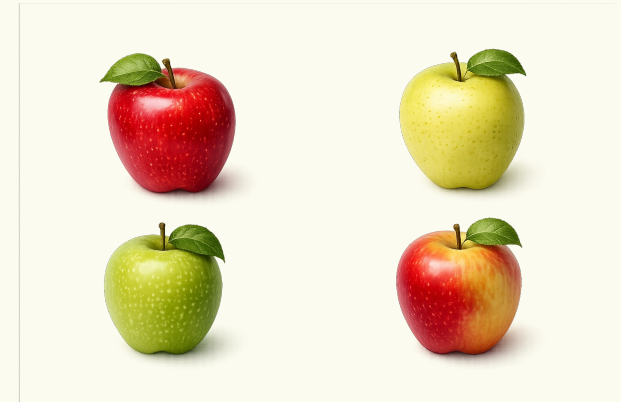
- Many schemes can be proved to satisfy OMUF with little modification to existing proofs
- If underlying signature scheme is strongly unforgeable, threshold signing protocol does not need to be OMUF
  - Counterexample in [Navot, T, ASIACRYPT '24]
- Generic transformation from unforgeability to OMUF [Azari, Boschini, Hostáková, Reichle, TCC '25]
  - Add two offline + one online round

## In conclusion

Unforgeability definitions are subtle!

Best practices to support standardization:

- Be very explicit about what security your scheme achieves
- State and prove the strongest possible notion
- Adopt unified models as far as possible (e.g., coordinator model)



Wanted research:

- Upstream implications of using weaker or strong unforgeability notions?

**Thank you!**