Better than Advertised Security for Non-Interactive Threshold Signatures

Mihir Bellare, **Elizabeth Crites**, Chelsea Komlo, Mary Maller,
Stefano Tessaro, **Chenzhi Zhu**

August 18, 2022

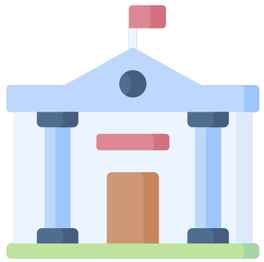
Threshold Signatures [DY90]

Single Key → Distributed Key

Single point of failure

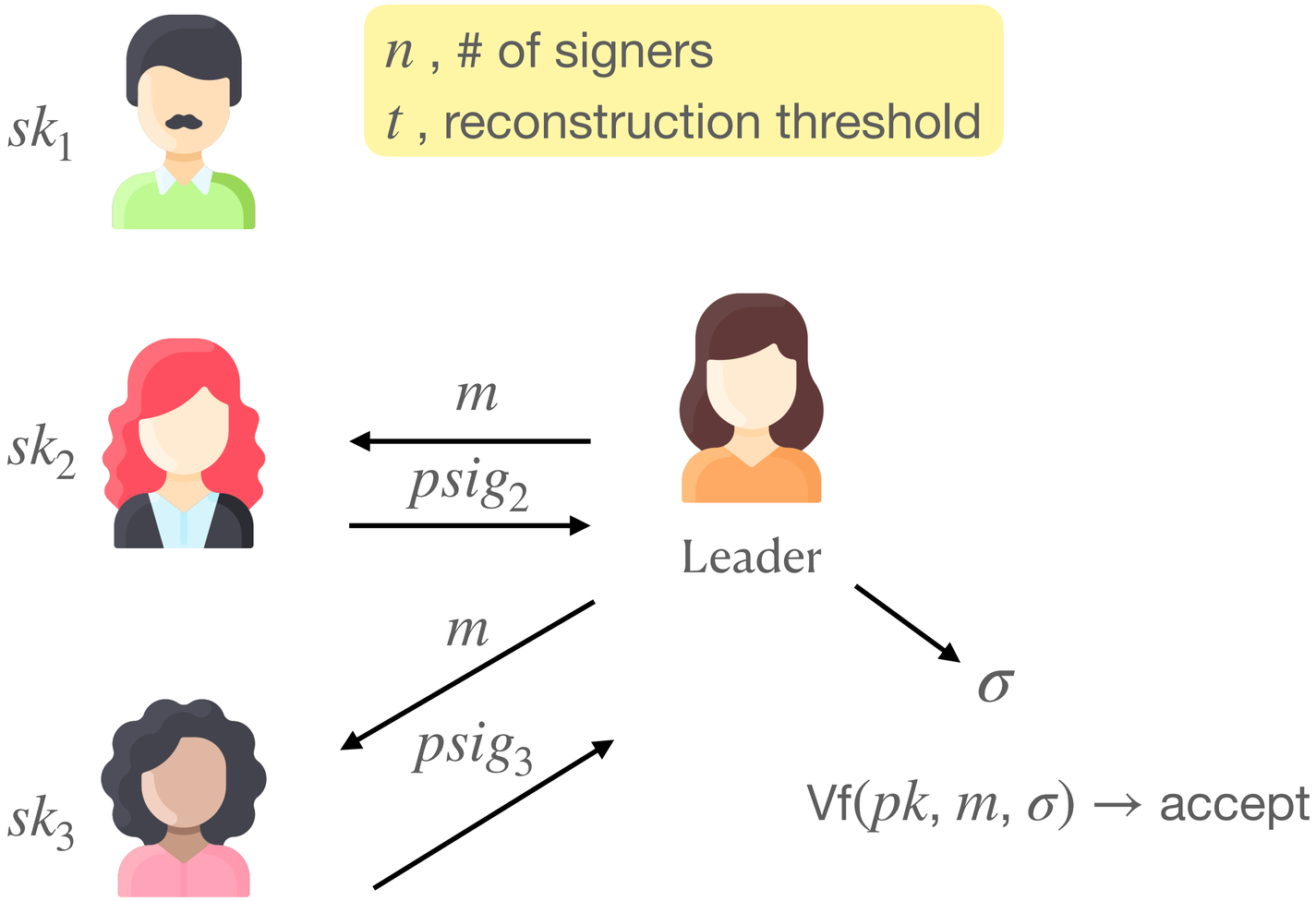
Tolerates some fraction of corrupt signers

Cryptocurrency Wallets

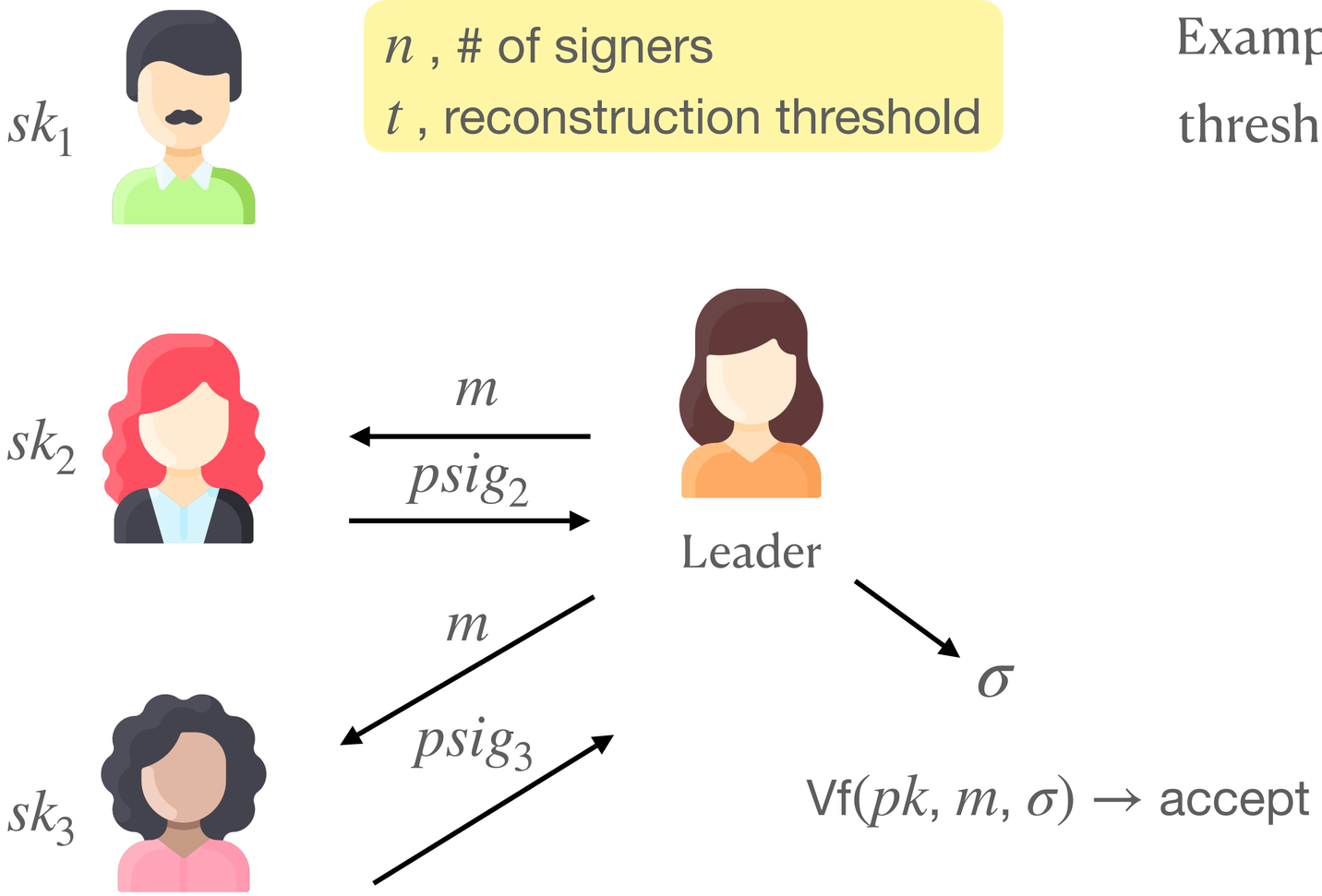


NIST Standardization

Our Focus: Non-Interactive Threshold Signatures



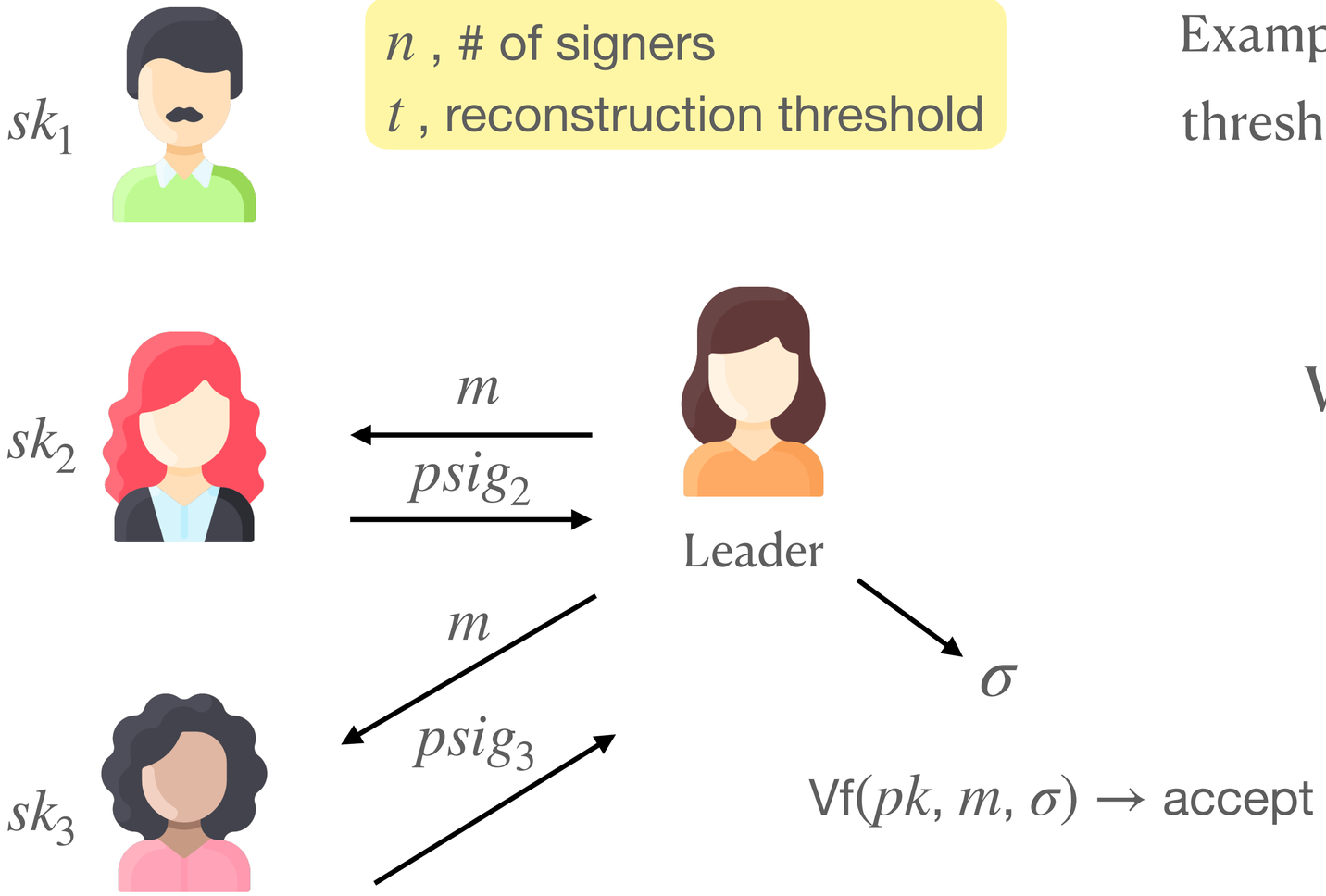
Our Focus: Non-Interactive Threshold Signatures



Examples: threshold BLS [Bol03]

threshold RSA [DSD+94, GJKR00, Sho00]

Our Focus: Non-Interactive Threshold Signatures



Examples: threshold BLS [Bol03]

threshold RSA [DSD+94, GJKR00, Sho00]

What about DL-based?

(Pairing-free)

DL-based Threshold Signatures

DL-based Threshold Signatures

No fully non-interactive scheme yet

DL-based Threshold Signatures

No fully non-interactive scheme yet

Schnorr Signatures

Single message-dependent round: FROST [KG20]

A **single-round** message-independent pre-processing

“Partially non-interactive”

DL-based Threshold Signatures

No fully non-interactive scheme yet

Schnorr Signatures

Single message-dependent round: FROST [KG20]

A **single-round** message-independent pre-processing

“Partially non-interactive”

ECDSA

Single message-dependent round: [CGG+20]

A **multi-round** message-independent pre-processing

DL-based Threshold Signatures

No fully non-interactive scheme yet

Schnorr Signatures

Single message-dependent round: FROST [KG20]

A **single-round** message-independent pre-processing

“Partially non-interactive”

ECDSA

Single message-dependent round: [CGG+20]

A **multi-round** message-independent pre-processing

Not covered

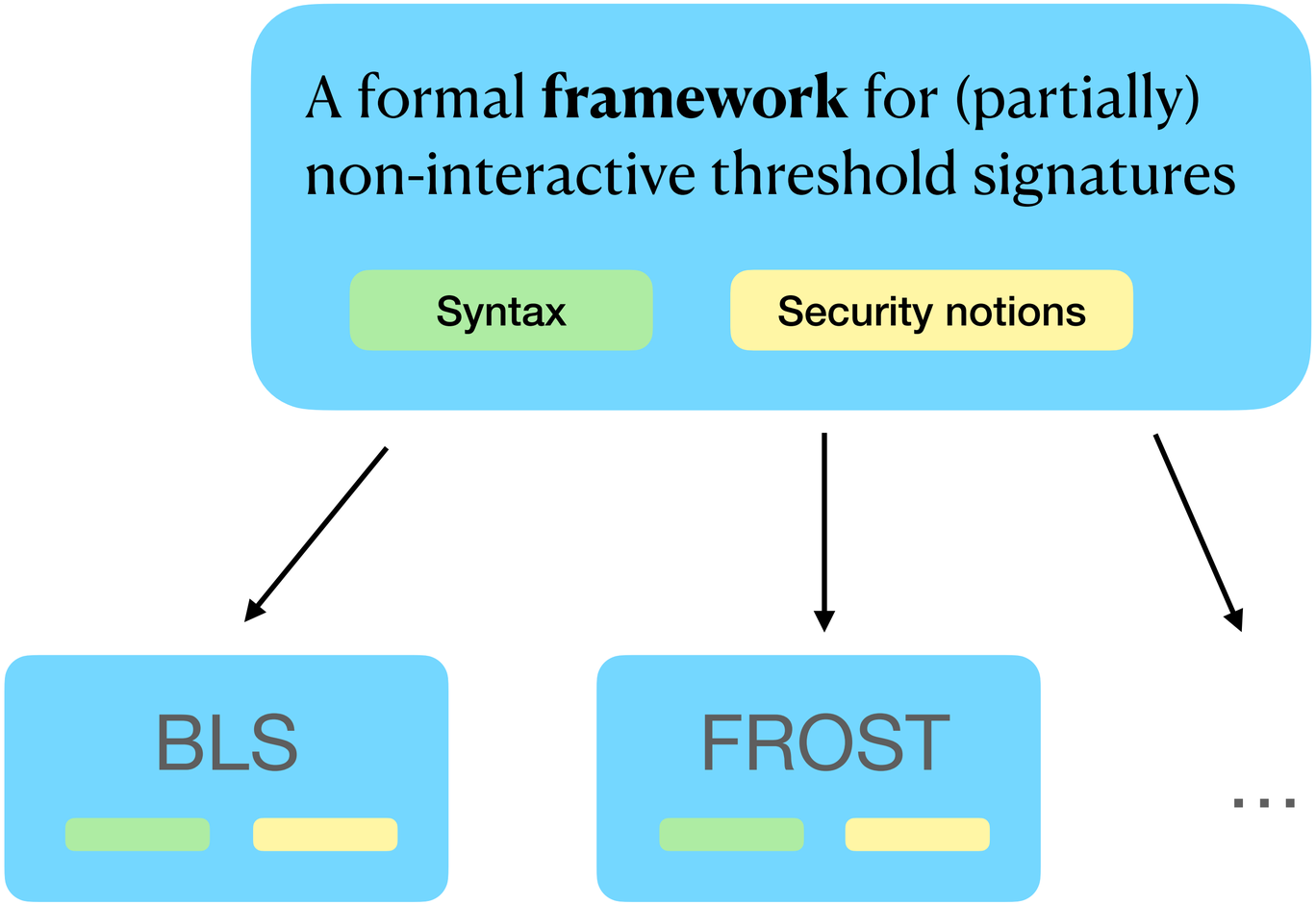
This Paper

A formal **framework** for (partially) non-interactive threshold signatures

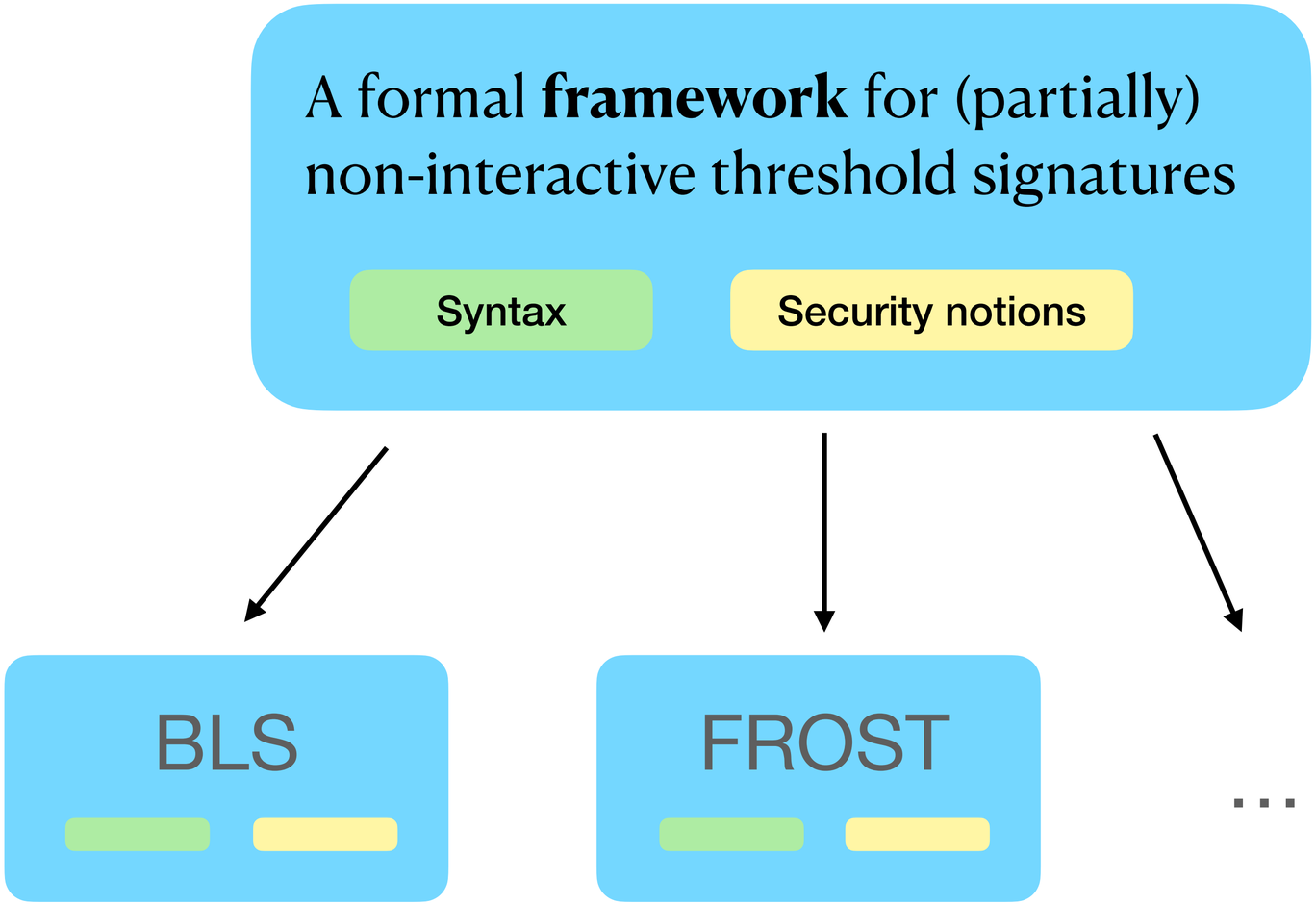
Syntax

Security notions

This Paper


A formal **framework** for (partially) non-interactive threshold signatures
Syntax
Security notions
BLS
FROST
...

This Paper



Why is this needed?

What's Missing & Our Contributions

What's Missing & Our Contributions

- No formalization for partially non-interactive schemes

What's Missing & Our Contributions

- No formalization for partially non-interactive schemes

$\Longrightarrow$

- A formal syntax for (partially) non-interactive threshold signatures

What's Missing & Our Contributions

- No formalization for partially non-interactive schemes

$\Rightarrow$

- A formal syntax for (partially) non-interactive threshold signatures

- Existing security notions are weaker than schemes can achieve

What's Missing & Our Contributions

- No formalization for partially non-interactive schemes

$\Longrightarrow$

- A formal syntax for (partially) non-interactive threshold signatures

- Existing security notions are weaker than schemes can achieve

$\Longrightarrow$

- Fine-grained security hierarchy with stronger notions

What's Missing & Our Contributions

- No formalization for partially non-interactive schemes

⟹

- A formal syntax for (partially) non-interactive threshold signatures

- Existing security notions are weaker than schemes can achieve

⟹

- Fine-grained security hierarchy with stronger notions

- Original proof for FROST [KG20] relied on heuristic assumptions

What's Missing & Our Contributions

- No formalization for partially non-interactive schemes

⟹

- A formal syntax for (partially) non-interactive threshold signatures

- Existing security notions are weaker than schemes can achieve

⟹

- Fine-grained security hierarchy with stronger notions

- Original proof for FROST [KG20] relied on heuristic assumptions

⟹

- Analysis of FROST based on our security hierarchy

What's Missing & Our Contributions

- No formalization for partially non-interactive schemes

⟹

- A formal syntax for (partially) non-interactive threshold signatures

- Existing security notions are weaker than schemes can achieve

⟹

- Fine-grained security hierarchy with stronger notions

Analysis of BLS for stronger security
Concurrent work [Gro21]

- Original proof for FROST [KG20] relied on heuristic assumptions

⟹

- Analysis of FROST based on our security hierarchy

Our Contributions

Our Contributions

- Introduce FROST2, an optimized version of FROST1

Our Contributions

- Introduce FROST2, an optimized version of FROST1

- Prove the security of FROST1/2 under OMDL + ROM in trusted DKG setting

Our Contributions

- Introduce FROST2, an optimized version of FROST1
- Prove the security of FROST1/2 under OMDL + ROM in trusted DKG setting
- Prove the security of FROST2 together with PedPoP DKG under OMDL + ROM + AGM

Our Contributions

- Introduce FROST2, an optimized version of FROST1
- Prove the security of FROST1/2 under OMDL + ROM in trusted DKG setting
- Prove the security of FROST2 together with PedPoP DKG under OMDL + ROM + AGM
- Show separation of security of FROST1 vs 2

FROST

FROST

- Flexible Round-Optimized Schnorr Threshold Signature [KG20]

FROST

- Flexible Round-Optimized Schnorr Threshold Signature [KG20]
- Consists of:
    - PedPoP distributed key generation (DKG) protocol
    - Two-round, concurrently secure signing protocol
        - First round message-independent pre-processing
        - Outputs standard, single-party Schnorr signature

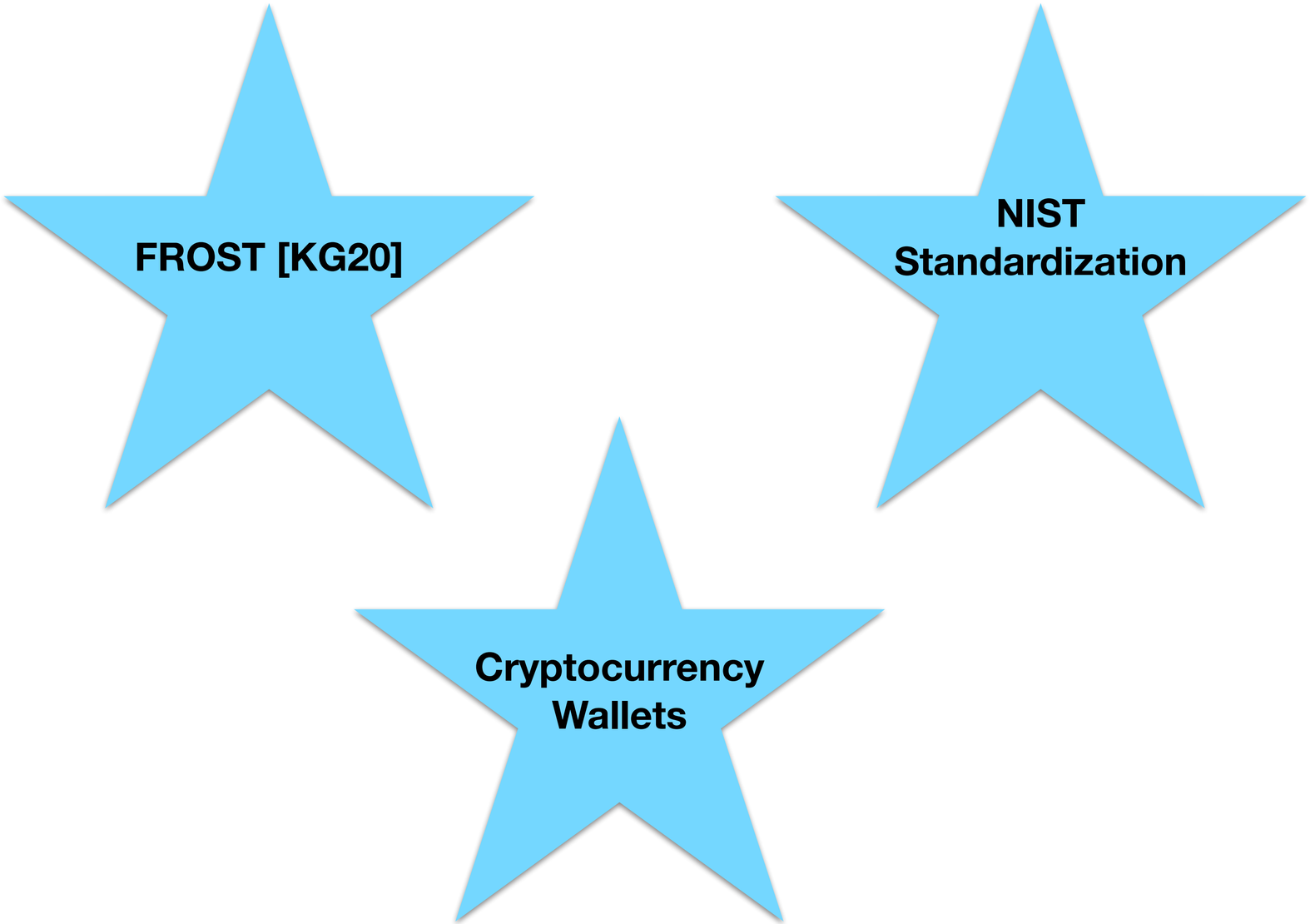
FROST

- Flexible Round-Optimized Schnorr Threshold Signature [KG20]
- Consists of:
    - PedPoP distributed key generation (DKG) protocol
    - Two-round, concurrently secure signing protocol
        - First round message-independent pre-processing
        - Outputs standard, single-party Schnorr signature
- Prior attempts at two-round Schnorr threshold signatures -> ROS attacks [BLLOR21], https://github.com/mmaller/multi_and_threshold_signature_reductions

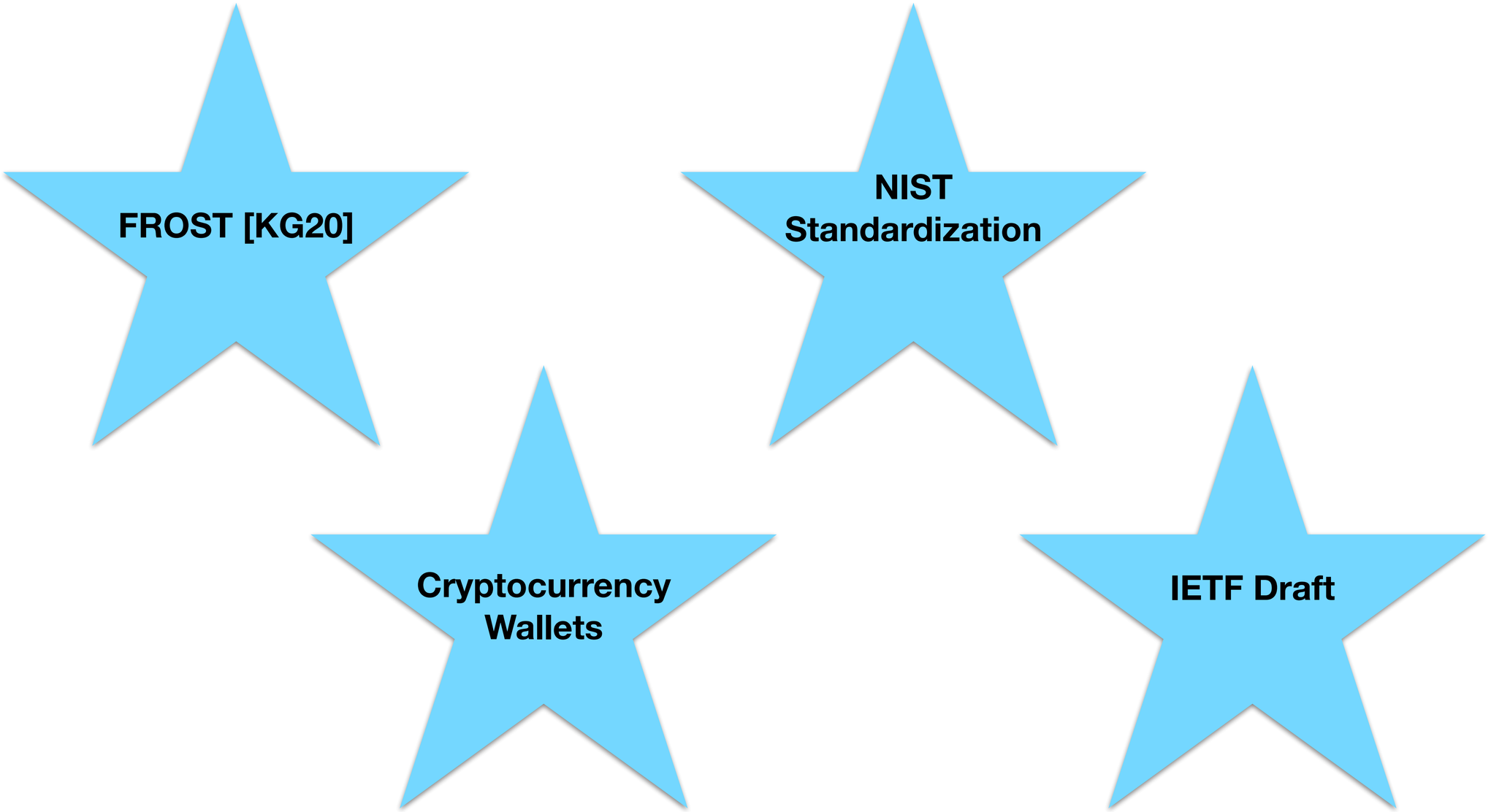
FROST

FROST

FROST



FROST


FROST [KG20]
NIST Standardization
Cryptocurrency Wallets
IETF Draft

FROST


FROST [KG20]


NIST
Standardization


5+
Implementations


Cryptocurrency
Wallets


IETF Draft

PedPoP Distributed Key Generation

- PedPoP [KG20] = Pedersen DKG + Proofs of Possession
- Proofs of possession are Schnorr signatures
    - Requires a Knowledge of Exponent assumption
    - For simplicity, instead of adding rounds to DKG
    - Allows any number of corrupt signers (not honest majority)

FROST1

sk_2

Leader

sk_3

FROST1

sk_2

$R_2 = g^{r_2}, S_2 = g^{s_2}$

Leader

sk_3

R_3, S_3

FROST1

sk_2

$R_2 = g^{r_2}, S_2 = g^{s_2}$

R_2, S_2

Leader

sk_3

R_3, S_3

R_3, S_3

FROST1

$R_2 = g^{r_2}, S_2 = g^{s_2}$

R_2, S_2 →

← $m, \{2,3\}, R_3, S_3$

Leader

R_3, S_3

R_3, S_3 →

← $m, \{2,3\}, R_2, S_2$

FROST1

sk_2

$R_2 = g^{r_2}, S_2 = g^{s_2}$

R_2, S_2 →

← $m, \{2,3\}, R_3, S_3$

pk output by DKG

$d_j = \tilde{H}(j, pk, m, \{R_j, S_j\}_2^3)$

$R = \Pi_2^3 R_j S_j^{d_j}$

$c \leftarrow H(pk, m, R)$

Leader

sk_3

R_3, S_3

R_3, S_3 →

← $m, \{2,3\}, R_2, S_2$

FROST1

sk_2

$$R_2 = g^{r_2}, S_2 = g^{s_2} \xrightarrow{R_2, S_2}$$

$$z_2 \leftarrow r_2 + d_2 s_2 + c\lambda_2 sk_2 \xleftarrow{m, \{2,3\}, R_3, S_3}$$

pk output by DKG

$$d_j = \tilde{H}(j, pk, m, \{R_j, S_j\}_2^3)$$

$$R = \Pi_2^3 R_j S_j^{d_j}$$

$$c \leftarrow H(pk, m, R)$$

Leader

sk_3

$$R_3, S_3 \xrightarrow{R_3, S_3}$$

$$z_3 \leftarrow r_3 + d_3 s_3 + c\lambda_3 sk_3 \xleftarrow{m, \{2,3\}, R_2, S_2}$$

FROST1

sk_2

$$R_2 = g^{r_2}, S_2 = g^{s_2}$$

$$z_2 \leftarrow r_2 + d_2 s_2 + c\lambda_2 sk_2$$

R_2, S_2 →

← $m, \{2,3\}, R_3, S_3$

z_2 →

pk output by DKG

$$d_j = \tilde{H}(j, pk, m, \{R_j, S_j\}_2^3)$$

$$R = \Pi_2^3 R_j S_j^{d_j}$$

$$c \leftarrow H(pk, m, R)$$

Leader

sk_3

R_3, S_3

$$z_3 \leftarrow r_3 + d_3 s_3 + c\lambda_3 sk_3$$

R_3, S_3 →

← $m, \{2,3\}, R_2, S_2$

z_3 →

FROST1

sk_2

$R_2 = g^{r_2}, S_2 = g^{s_2}$

R_2, S_2 →

← $m, \{2,3\}, R_3, S_3$

$z_2 \leftarrow r_2 + d_2 s_2 + c\lambda_2 sk_2$

z_2 →

pk output by DKG

$d_j = \tilde{H}(j, pk, m, \{R_j, S_j\}_2^3)$

$R = \Pi_2^3 R_j S_j^{d_j}$

$c \leftarrow H(pk, m, R)$

Leader

$z \leftarrow z_2 + z_3$

$\sigma = (R, z)$

sk_3

R_3, S_3 →

← $m, \{2,3\}, R_2, S_2$

$z_3 \leftarrow r_3 + d_3 s_3 + c\lambda_3 sk_3$

z_3 →

FROST1

sk_2

$R_2 = g^{r_2}, S_2 = g^{s_2}$

$z_2 \leftarrow r_2 + d_2 s_2 + c\lambda_2 sk_2$

R_2, S_2 →

← $m, \{2,3\}, R_3, S_3$

z_2 →

pk output by DKG

$d_j = \tilde{H}(j, pk, m, \{R_j, S_j\}_2^3)$

$R = \Pi_2^3 R_j S_j^{d_j}$

$c \leftarrow H(pk, m, R)$

Leader

$z \leftarrow z_2 + z_3$

$\sigma = (R, z)$

sk_3

R_3, S_3 →

← $m, \{2,3\}, R_2, S_2$

z_3 →

R_3, S_3

$z_3 \leftarrow r_3 + d_3 s_3 + c\lambda_3 sk_3$

Verify:

$c \leftarrow H(pk, m, R)$

$R \cdot pk^c \stackrel{?}{=} g^z$

FROST2

sk_2

R_2, S_2

$z_2 \leftarrow r_2 + ds_2 + c\lambda_2 sk_2$

R_2, S_2 →

← $m, \{2,3\}, R_3, S_3$

z_2 →

pk output by DKG

$d = \tilde{H}(pk, m, \{R_j, S_j\}_2^3)$

$R = \Pi_2^3 R_j S_j^{d_j}$

$c \leftarrow H(pk, m, R)$

Leader

$z \leftarrow z_2 + z_3$

$\sigma = (R, z)$

sk_3

R_3, S_3

$z_3 \leftarrow r_3 + ds_3 + c\lambda_3 sk_3$

R_3, S_3 →

← $m, \{2,3\}, R_2, S_2$

z_3 →

Verify:

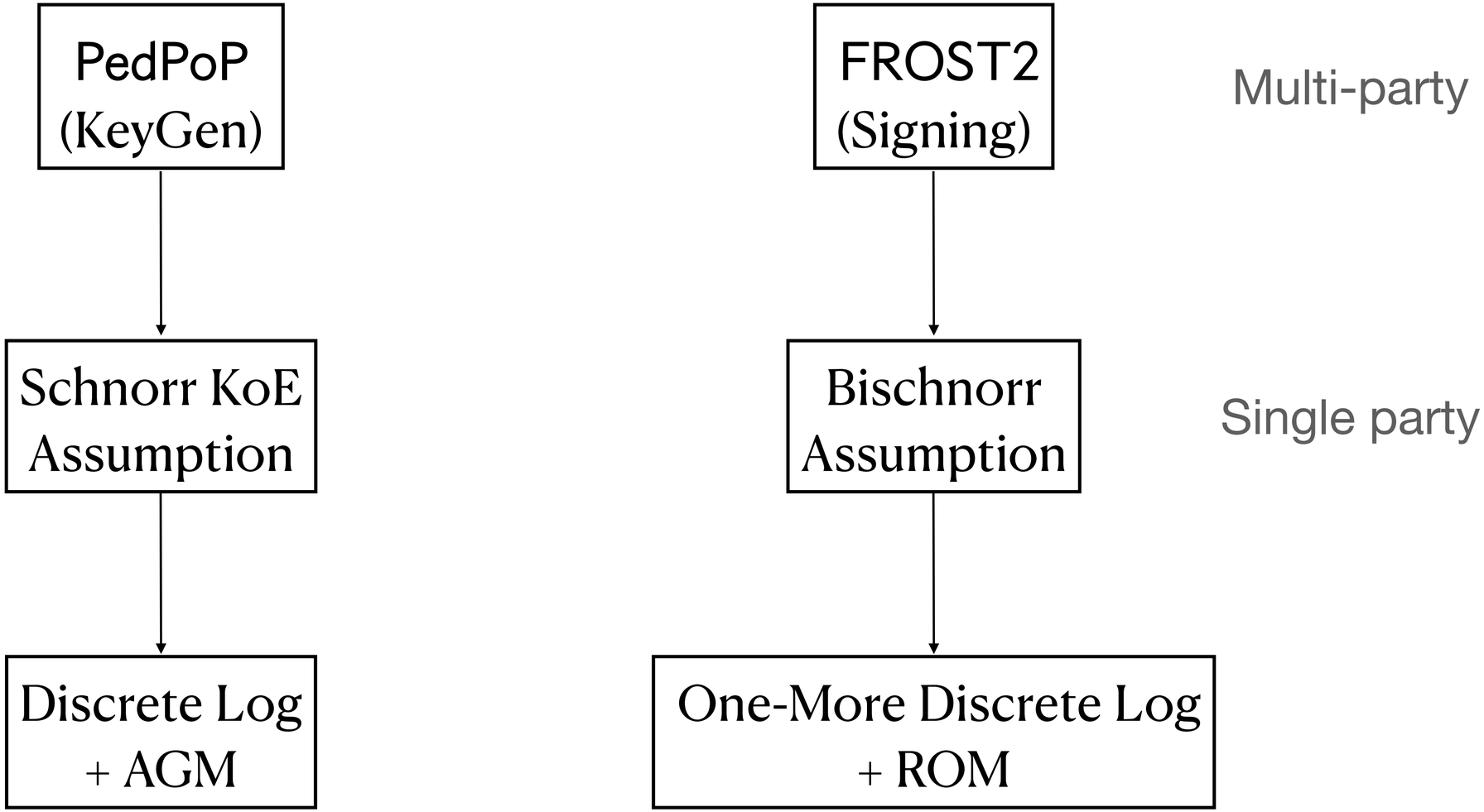
$c \leftarrow H(pk, m, R)$

$R \cdot pk^c \stackrel{?}{=} g^z$

Proving the Security of FROST2 + PedPoP

- Security reductions for multi-party signatures have two moving parts:
  1. Simulating honest users interacting with the adversary
  2. Extracting a solution to some hard problem from the adversary's responses
- Idea: Separate the two parts for a more modular reduction

Proving the Security of FROST2 + PedPoP



Framework & Security Hierarchy

Motivations

- No formalization for partially non-interactive schemes

- Existing security notions are weaker than schemes can achieve

Motivations

- No formalization for partially non-interactive schemes

- Modeled as interactive protocols [KG20]

- Existing security notions are weaker than schemes can achieve

Motivations

- No formalization for partially non-interactive schemes

- Modeled as interactive protocols [KG20]
- Simpler abstraction only for fully non-interactive schemes [Bol03, Wee11]

- Existing security notions are weaker than schemes can achieve

Motivations

- No formalization for partially non-interactive schemes

- Modeled as interactive protocols [KG20]
- Simpler abstraction only for fully non-interactive schemes [Bol03, Wee11]

- Existing security notions are weaker than schemes can achieve

- Most works: m is **considered signed** as long as **one** honest party signs it

[GJKR96, KY02, Bol03, Wee11, KG20, BGG+18]

Motivations

- No formalization for partially non-interactive schemes

- Modeled as interactive protocols [KG20]
- Simpler abstraction only for fully non-interactive schemes [Bol03, Wee11]

- Existing security notions are weaker than schemes can achieve

- Most works: m is **considered signed** as long as **one** honest party signs it [GJKR96, KY02, Bol03, Wee11, KG20, BGG+18]
- Only very few consider stronger guarantees [Sho00, LJY14, Gro21]

(Partially) Non-interactive Threshold Sigs

$n = 3,\ t = 2$

Signer 1

Signer 2

Signer 3

Leader

(Partially) Non-interactive Threshold Sigs

$n = 3,\ t = 2$

sk_1

Signer 1

sk_2

Signer 2

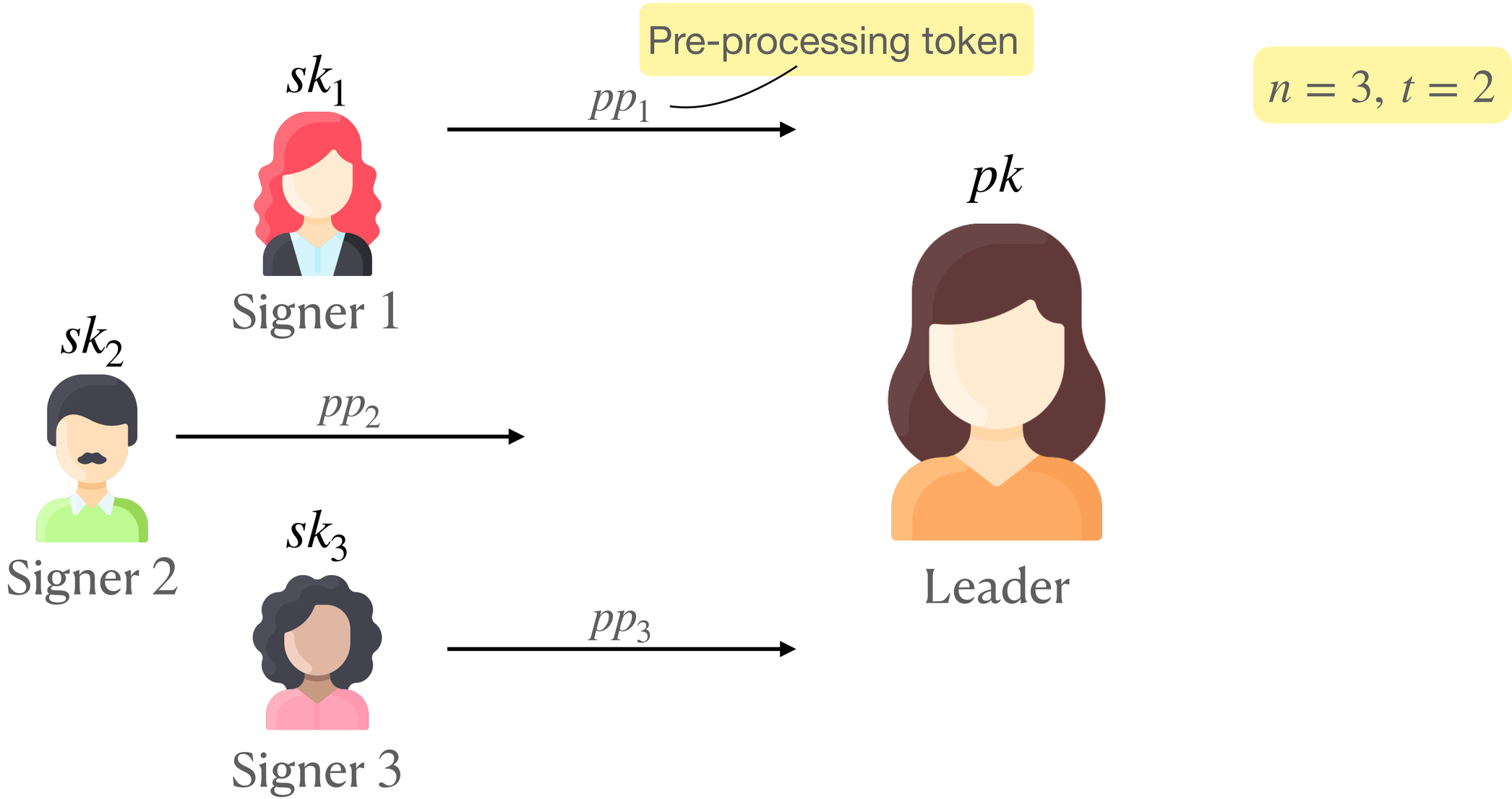
sk_3

Signer 3

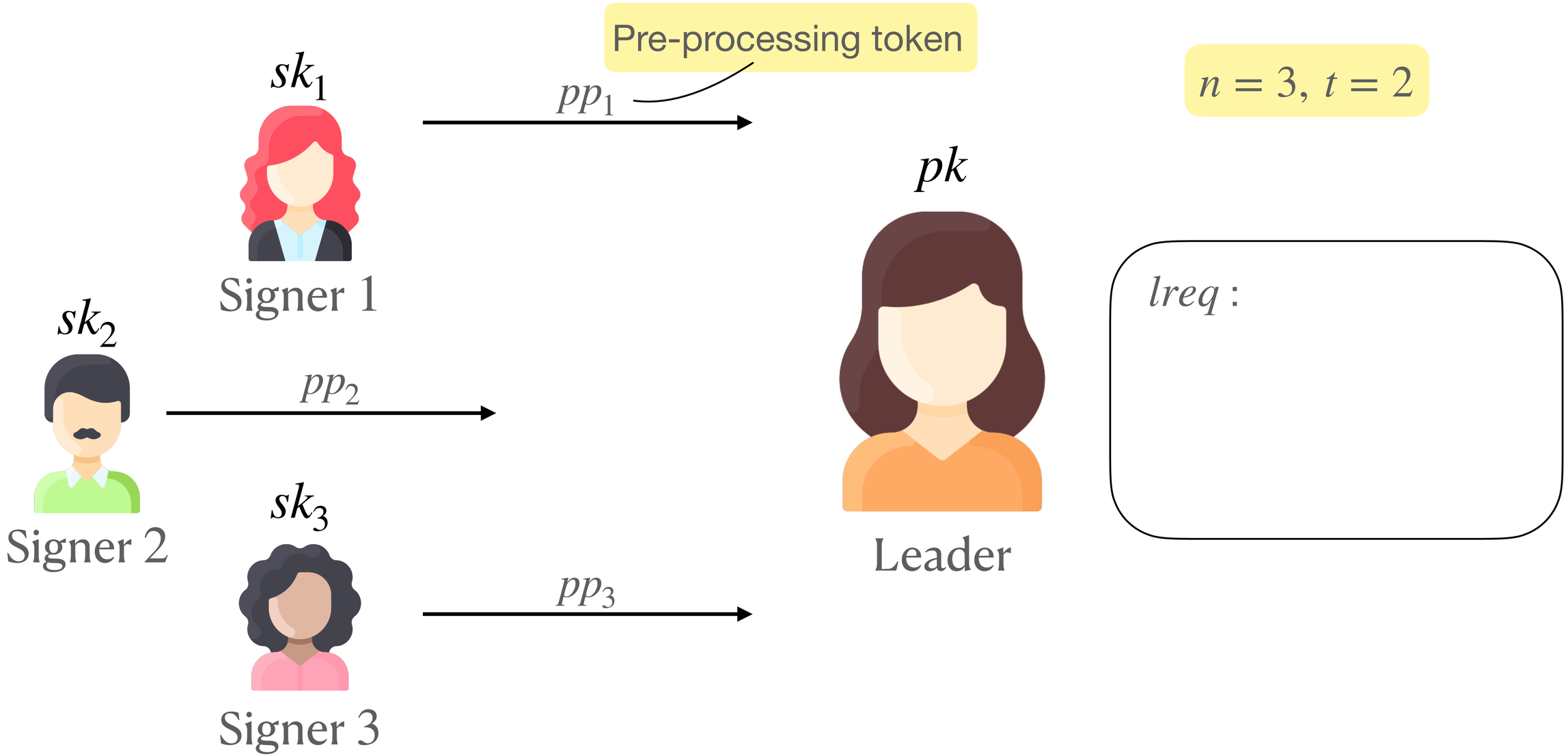
pk

Leader

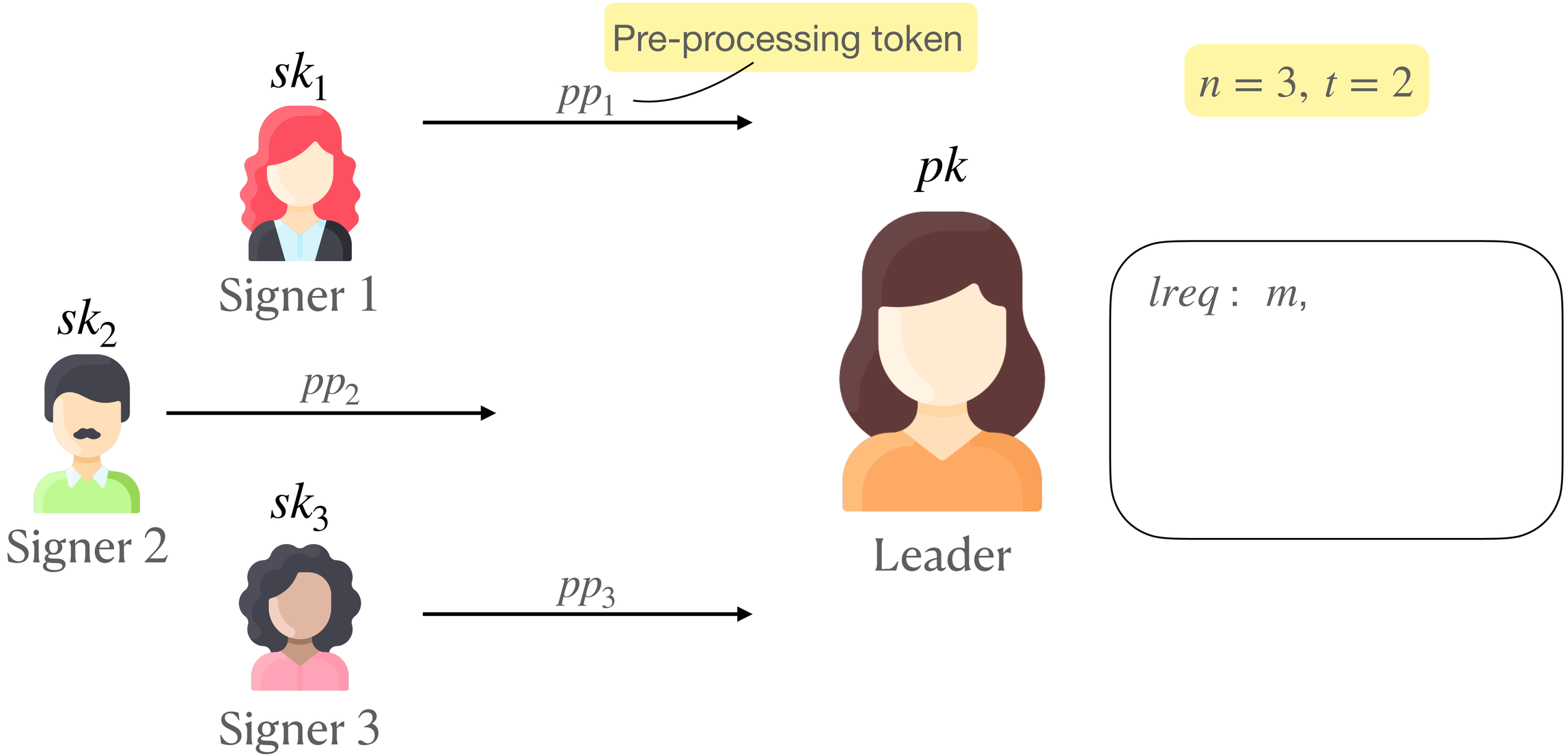
(Partially) Non-interactive Threshold Sigs



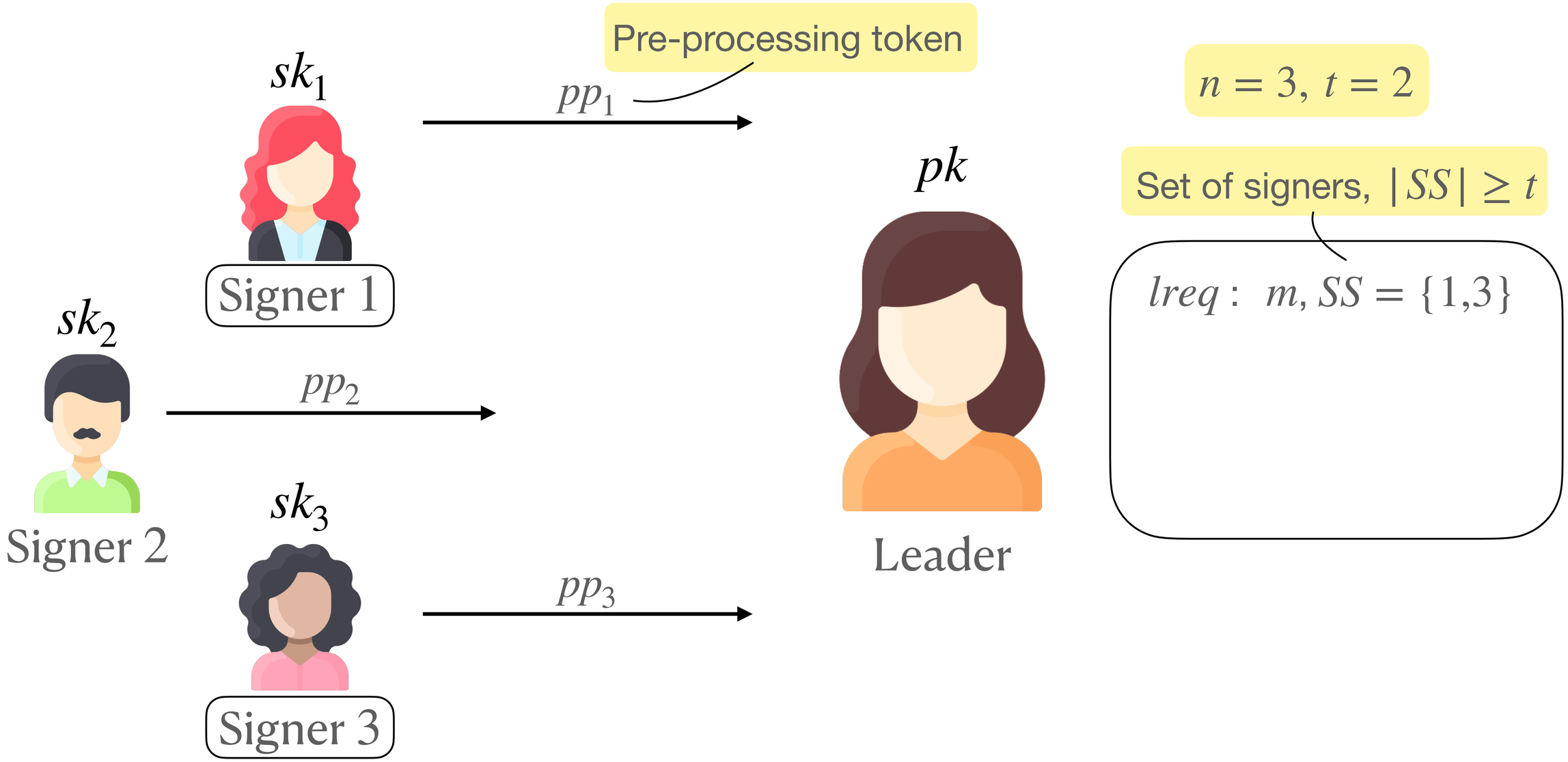
(Partially) Non-interactive Threshold Sigs



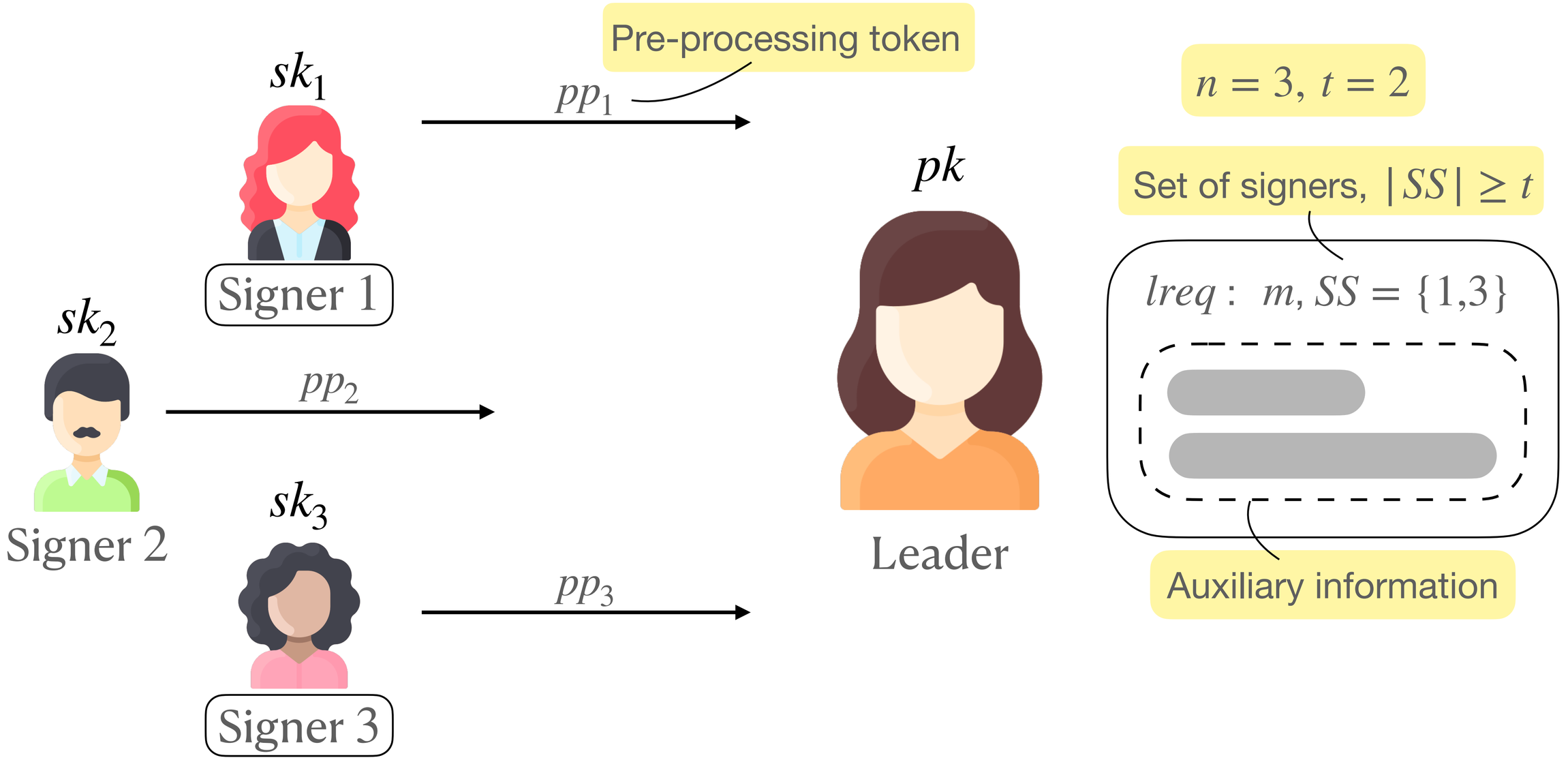
(Partially) Non-interactive Threshold Sigs



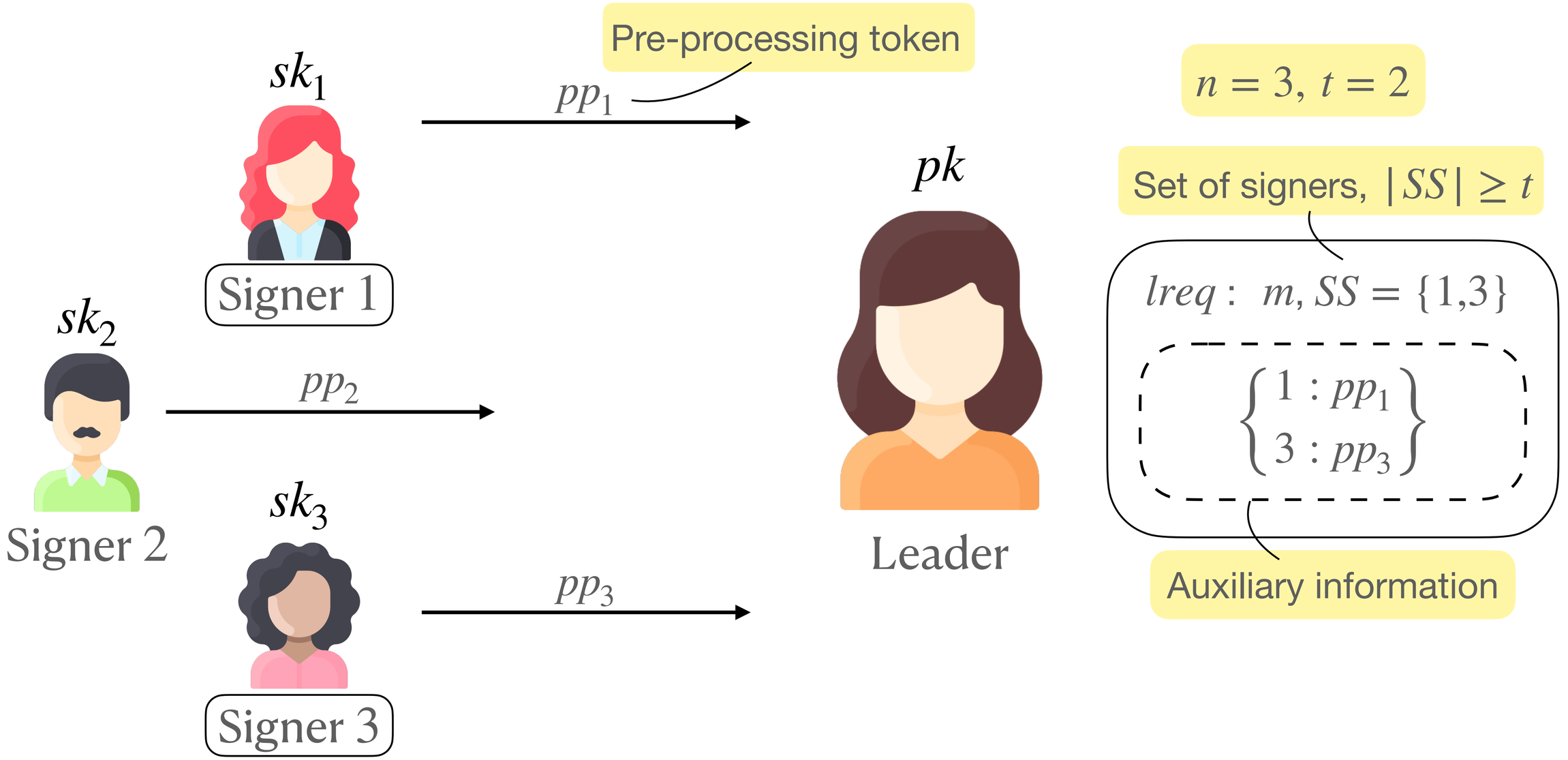
(Partially) Non-interactive Threshold Sigs



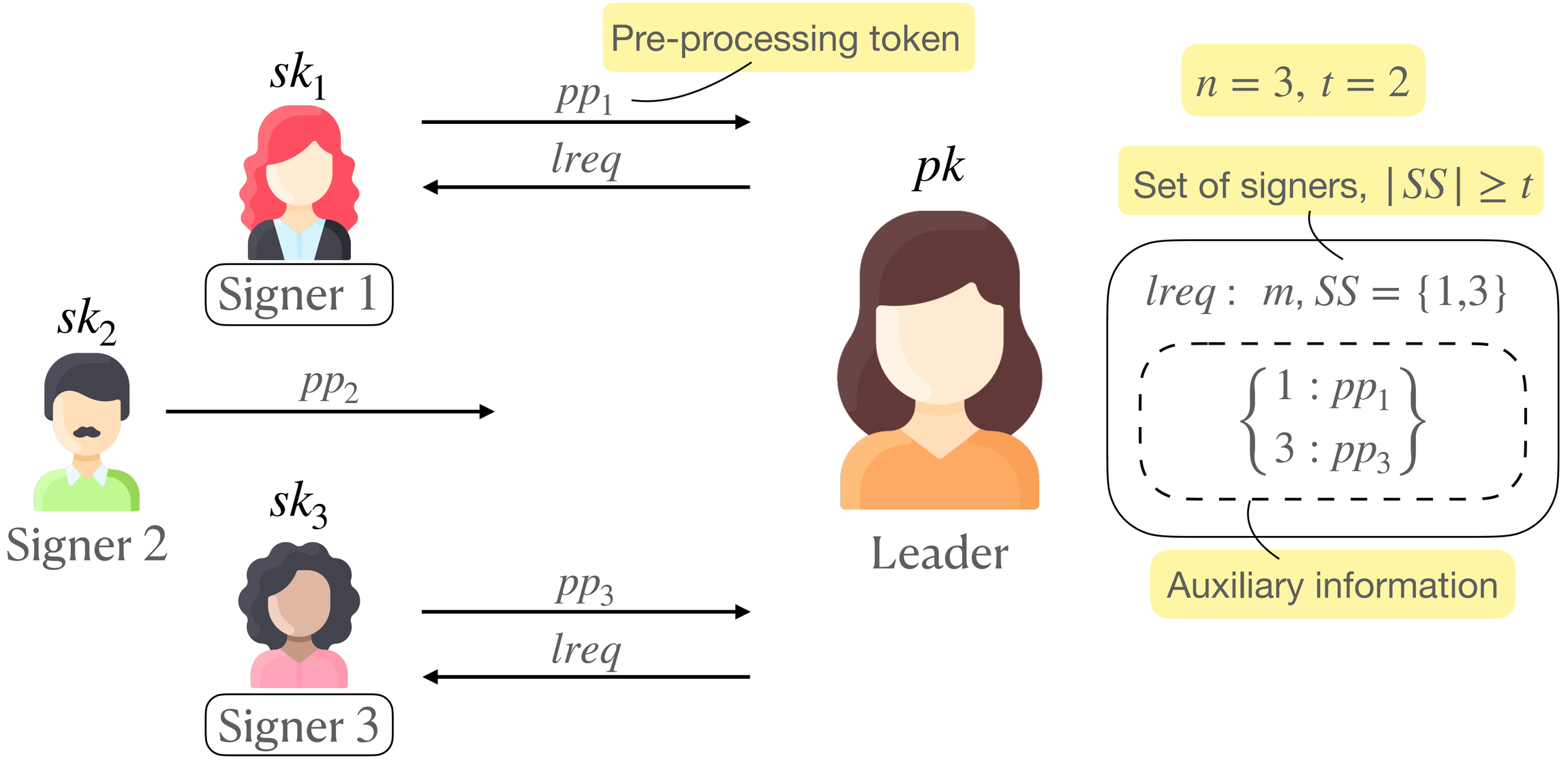
(Partially) Non-interactive Threshold Sigs



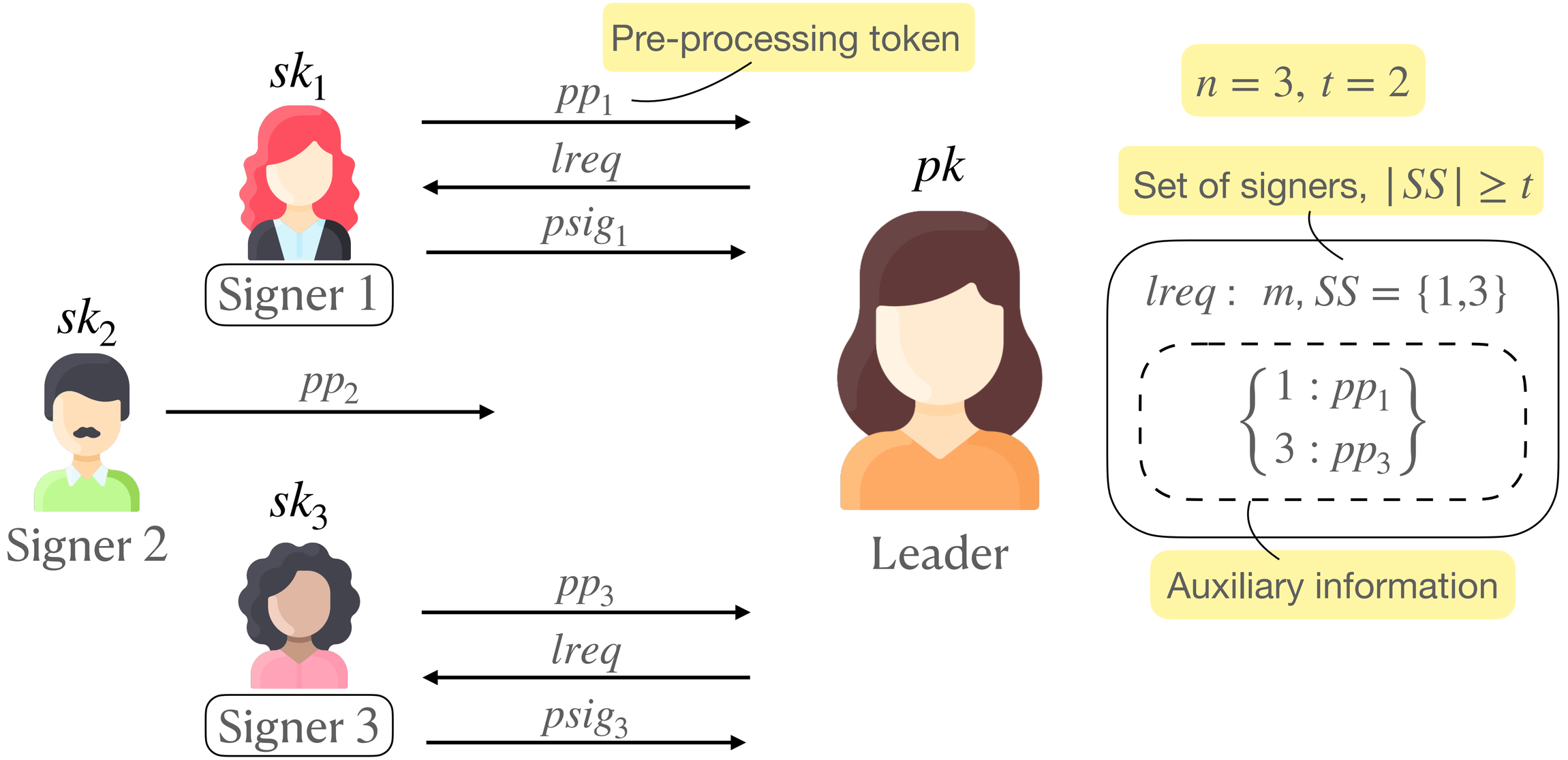
(Partially) Non-interactive Threshold Sigs



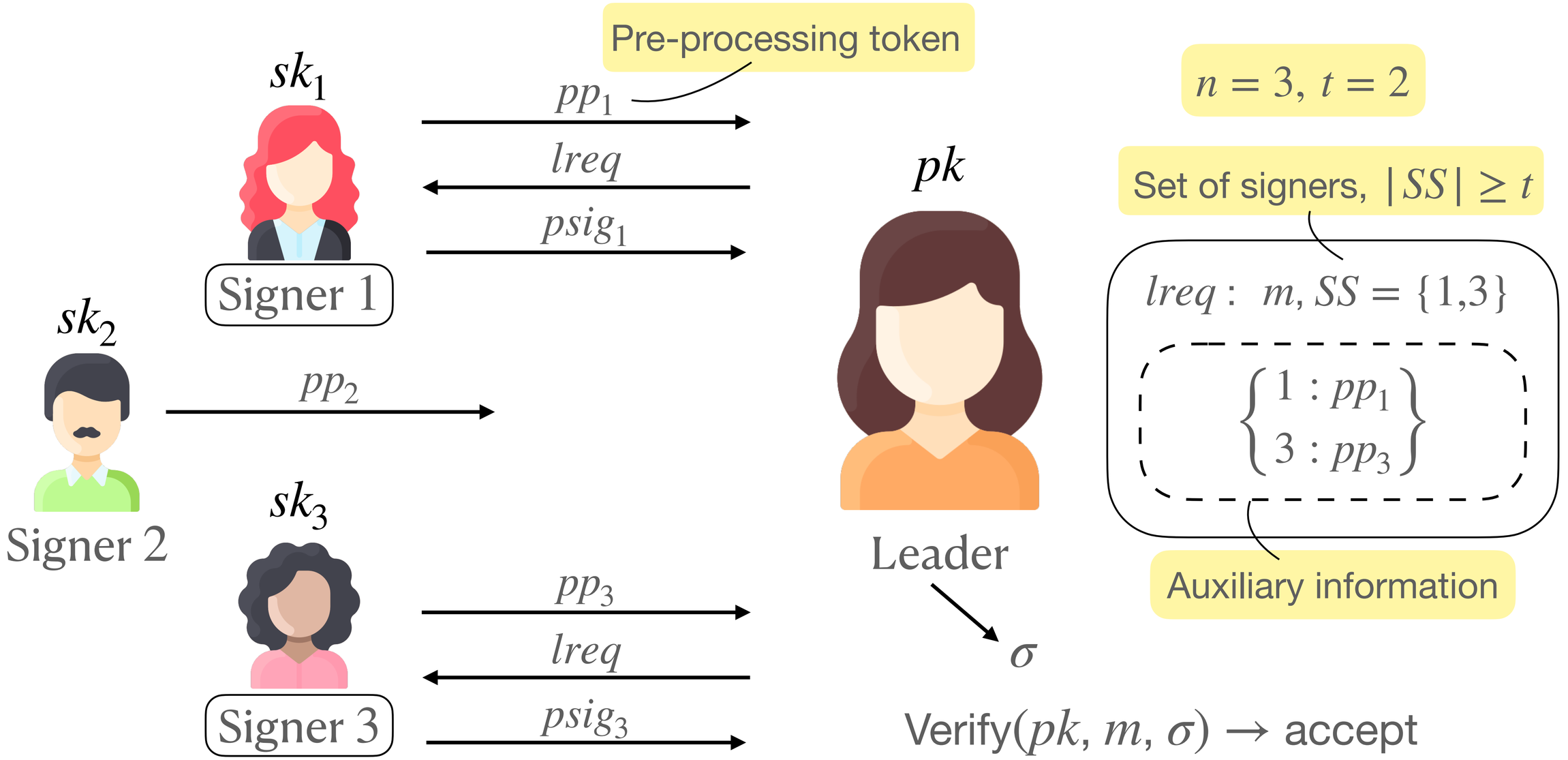
(Partially) Non-interactive Threshold Sigs



(Partially) Non-interactive Threshold Sigs



(Partially) Non-interactive Threshold Sigs



Unforgeability

sk_1

Signer 1

sk_2

Signer 2

sk_3

Signer 3

pk

Leader

Unforgeability

sk_1

Signer 1

sk_2

Signer 2

sk_3

Signer 3

pk

Adversary

Unforgeability

sk_1

Signer 1

sk_2

Signer 2

sk_3

Signer 3

pk

Adversary

Unforgeability

sk_1

Signer 1

sk_2

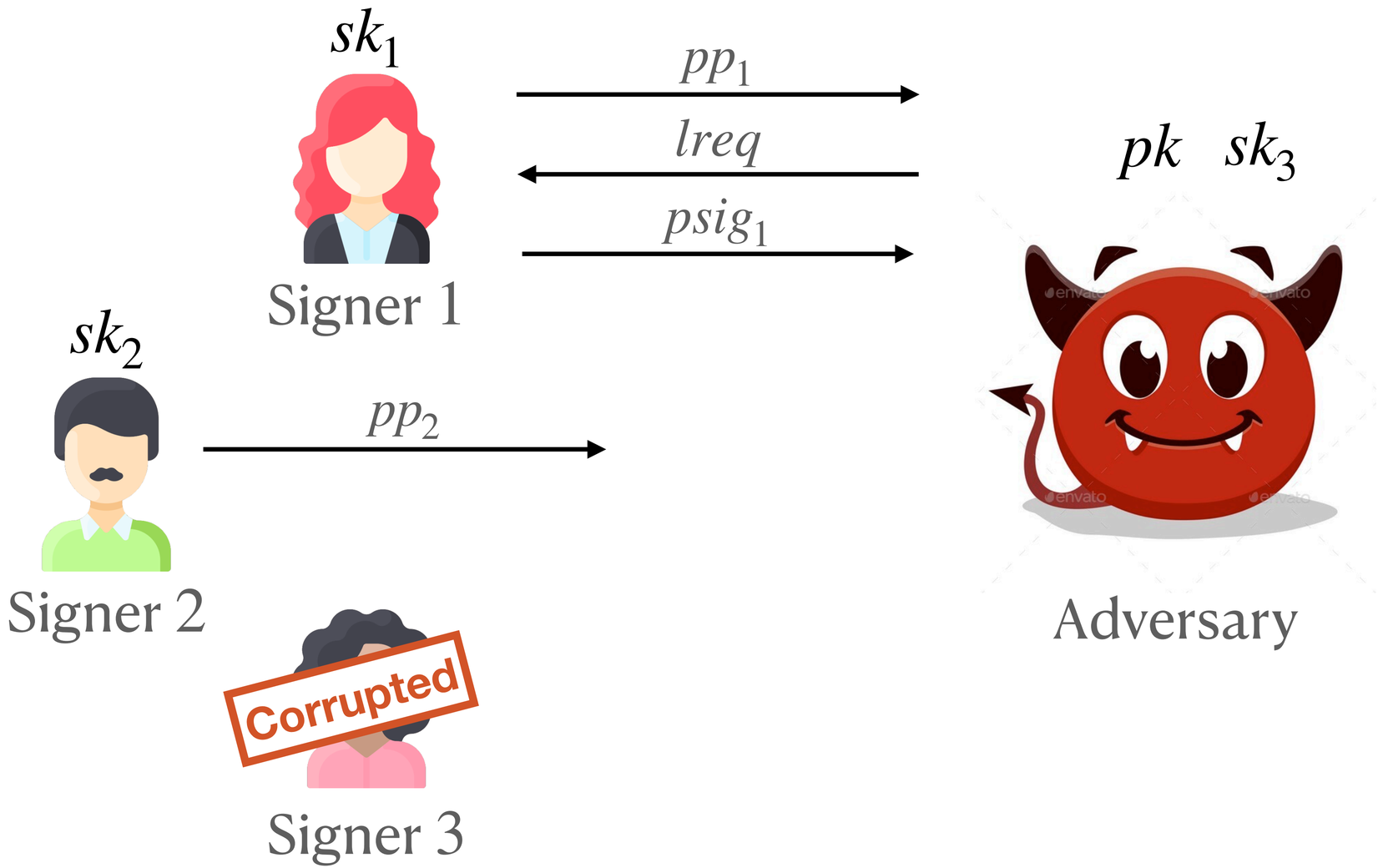
Signer 2

Signer 3

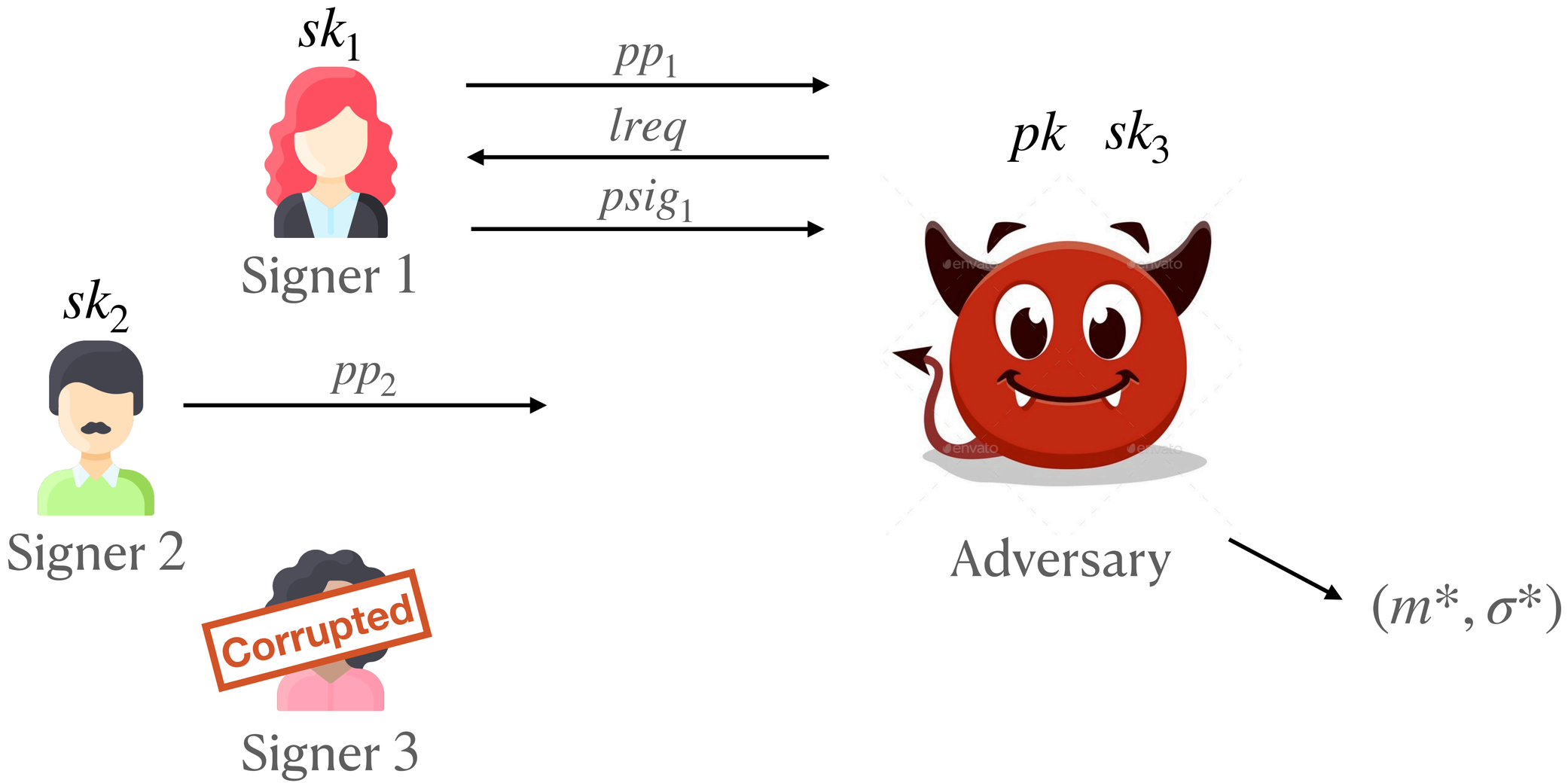
pk sk_3

Adversary

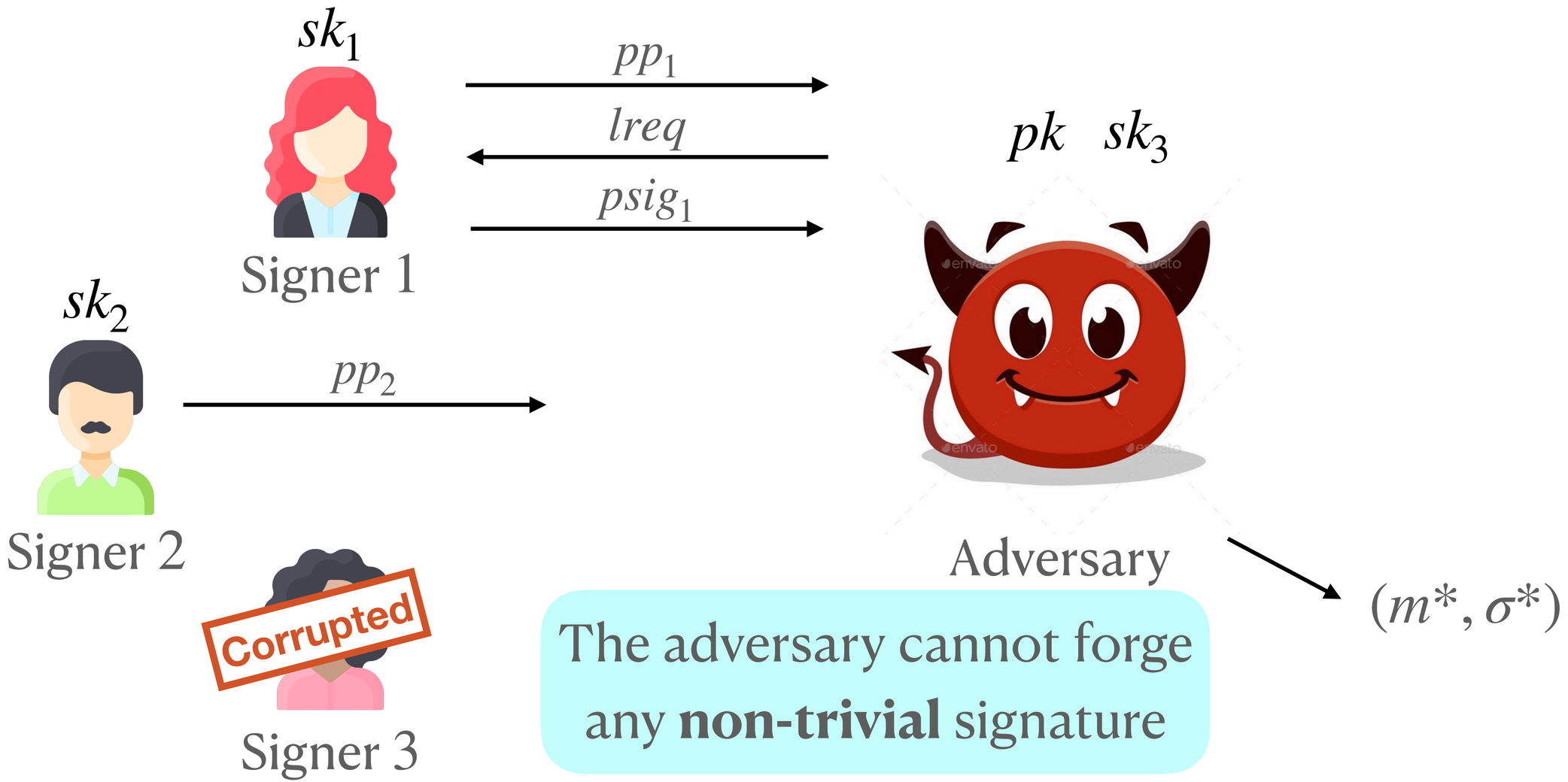
Unforgeability



Unforgeability



Unforgeability



Unforgeability

This work:

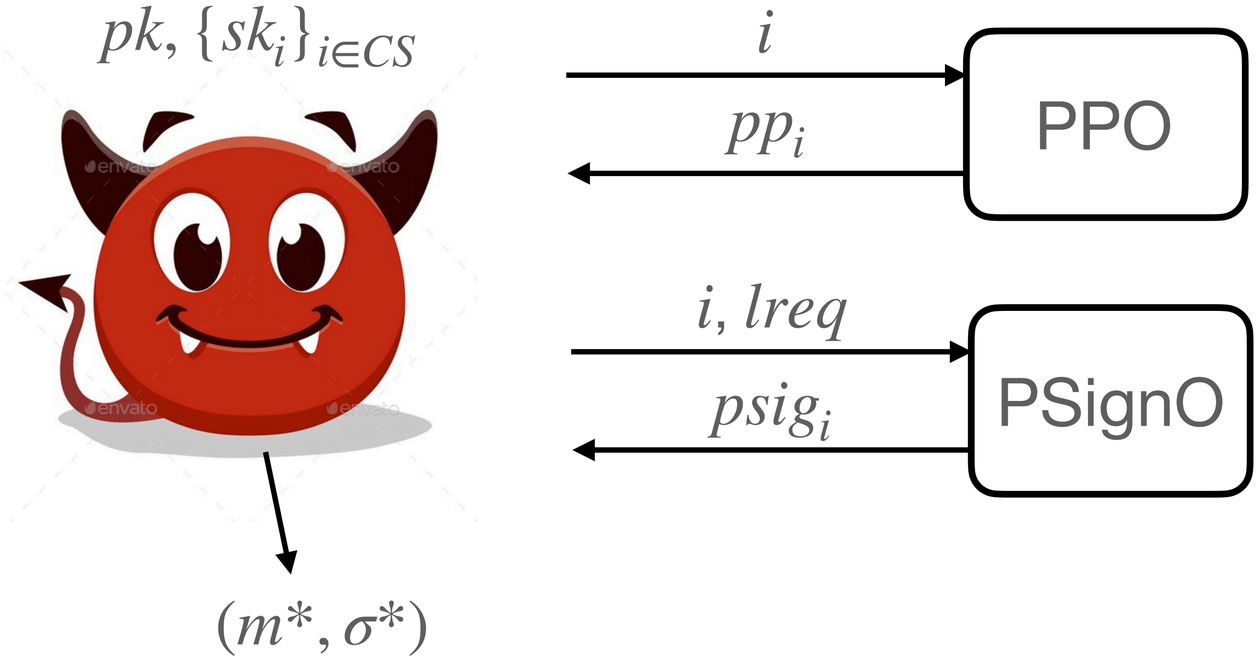
What is a trivial forgery?

The adversary cannot forge any **non-trivial** signature

(General) Unforgeability Game (TS-UF)

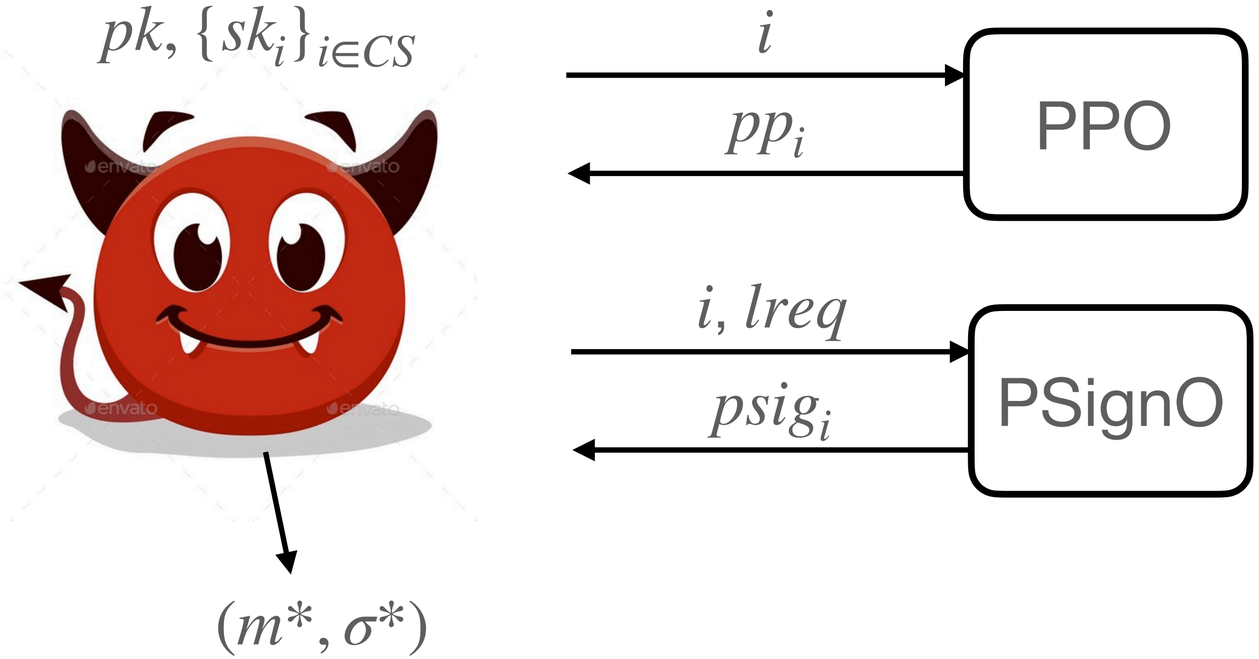
(General) Unforgeability Game (TS-UF)

CS, Set of corrupted signers



(General) Unforgeability Game (TS-UF)

CS, Set of corrupted signers



i is an honest signer

(General) Unforgeability Game (TS-UF)

CS, Set of corrupted signers

i

pp_i

PPO

i is an honest signer

Arbitrary number of times

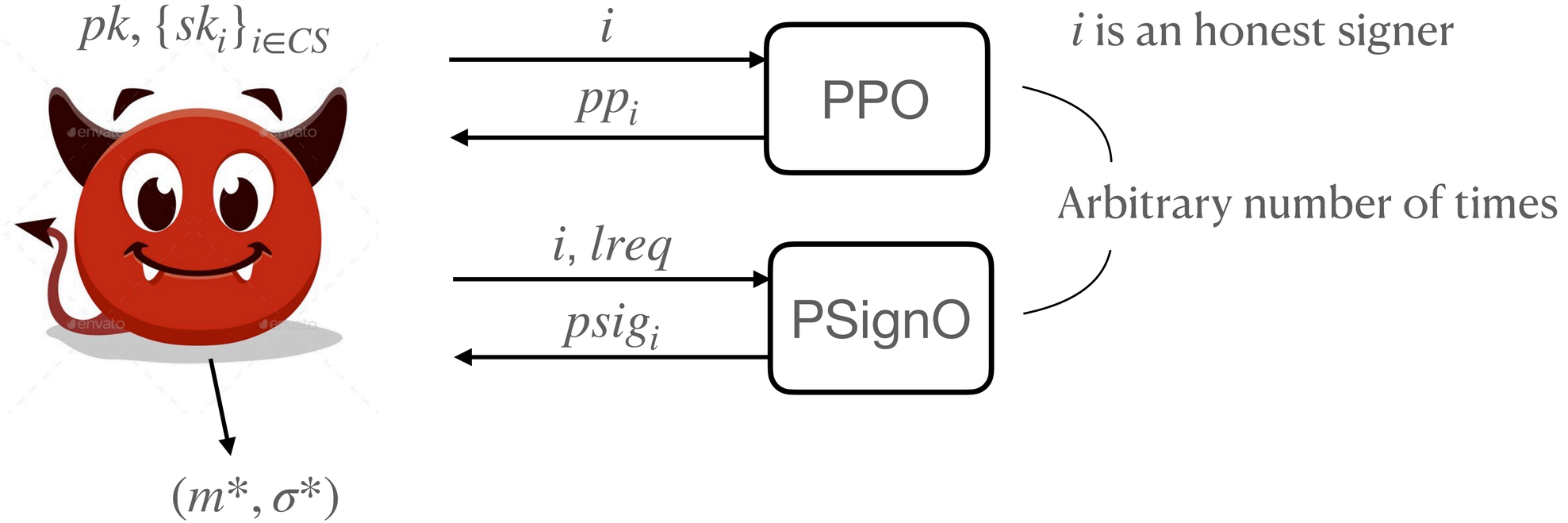
$i, lreq$

$psig_i$

PSignO

(General) Unforgeability Game (TS-UF)

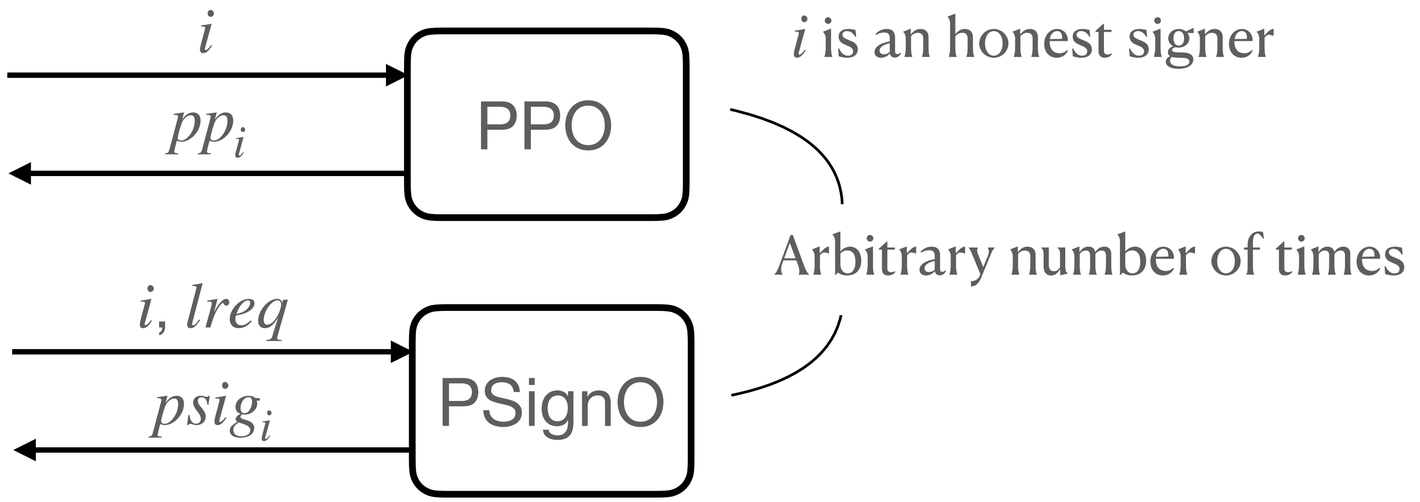
CS, Set of corrupted signers



Adversary wins iff (m^*, σ^*) is **not** a **trivial forgery**

(General) Unforgeability Game (TS-UF)

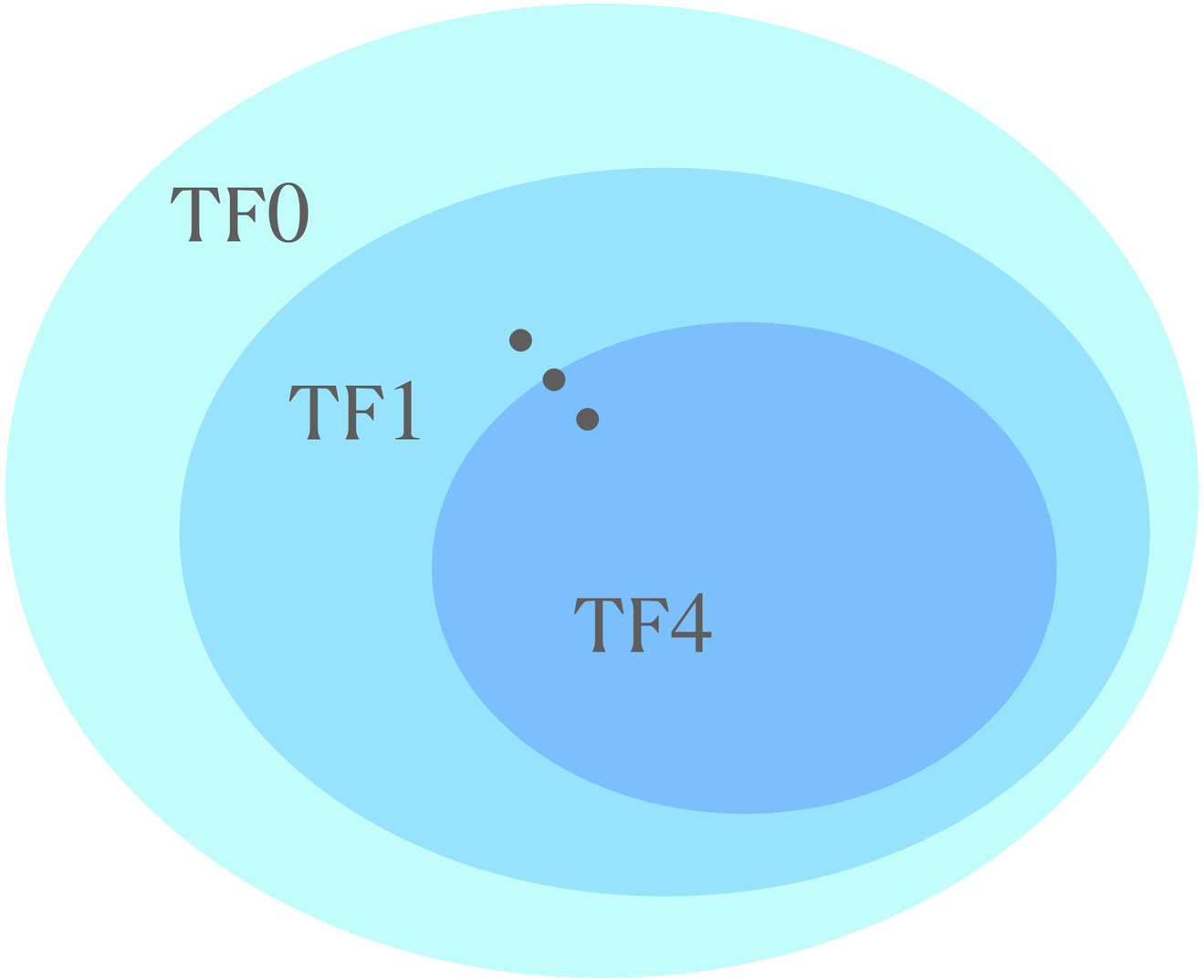
CS, Set of corrupted signers

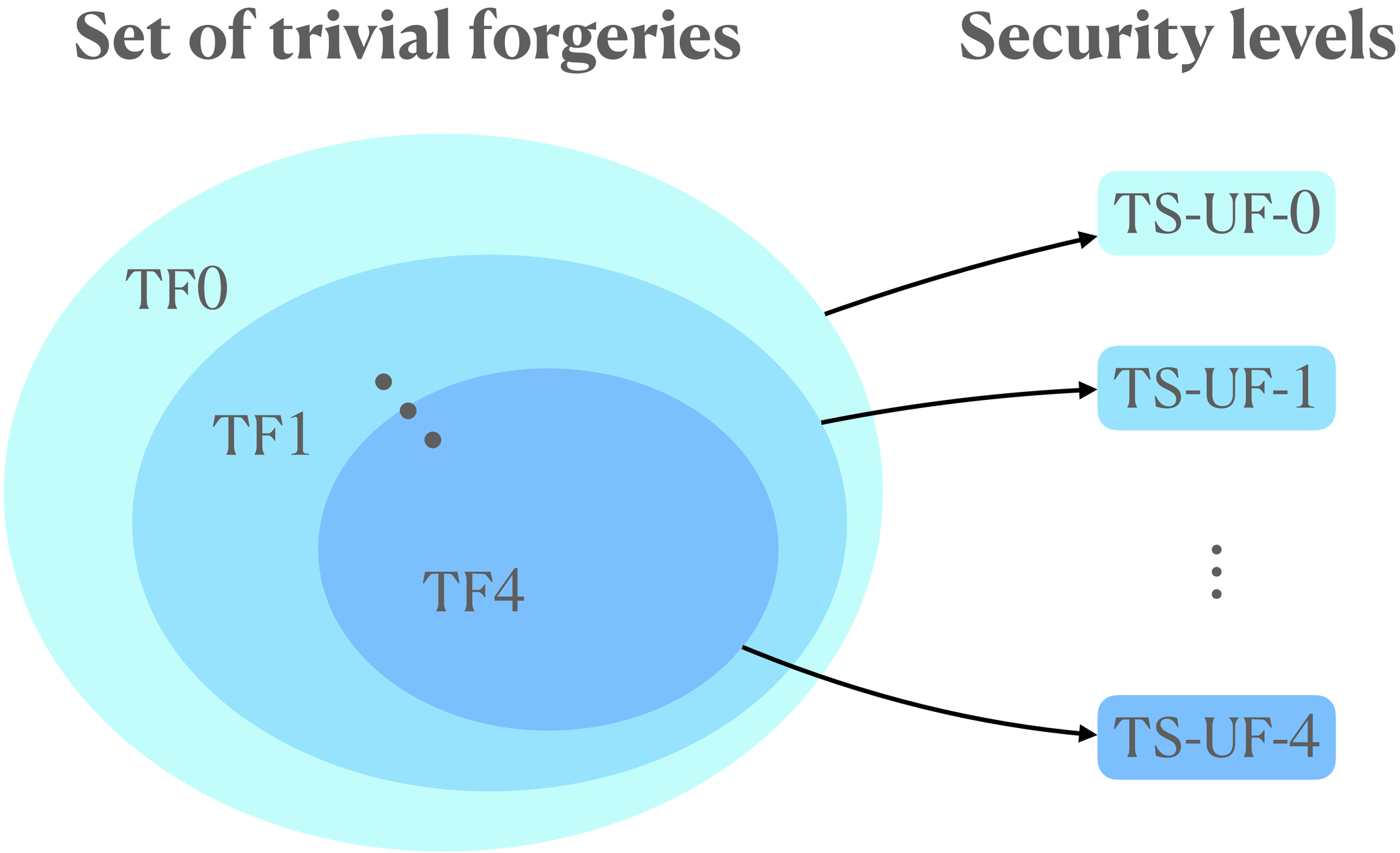


What is a trivial forgery?

Adversary wins iff (m^*, σ^*) is **not** a **trivial forgery**

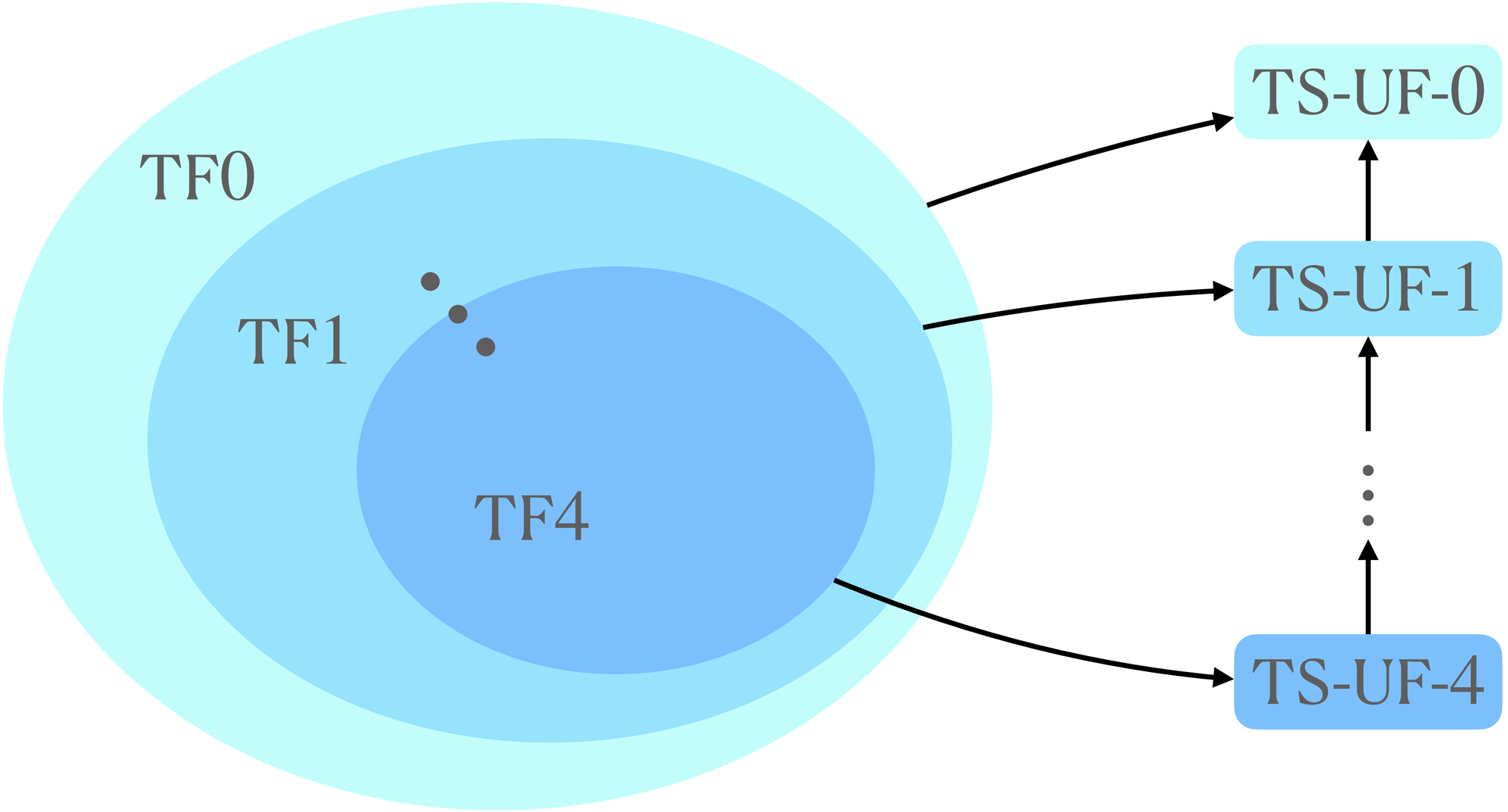
Set of trivial forgeries




Set of trivial forgeries
Security levels
TF0
TF1
TF4
TS-UF-0
TS-UF-1
TS-UF-4

Set of trivial forgeries

Security hierarchy



The Simplest

L : set of $(i, lreq)$ queries to PSignO

TS-UF-0 (m^*, σ^*) is a trivial forgery :

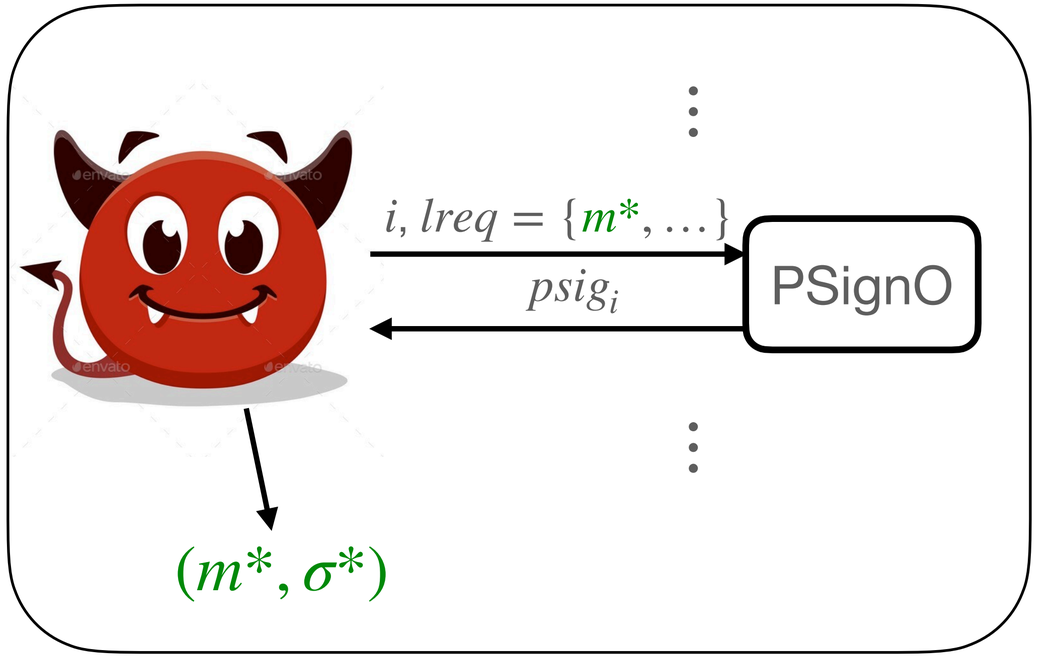
$$\exists (i, lreq) \in L : lreq\,.msg = m^*$$

The Simplest

L : set of $(i, lreq)$ queries to PSignO

TS-UF-0 (m^*, σ^*) is a trivial forgery :

$$\exists (i, lreq) \in L : lreq\,.msg = m^*$$



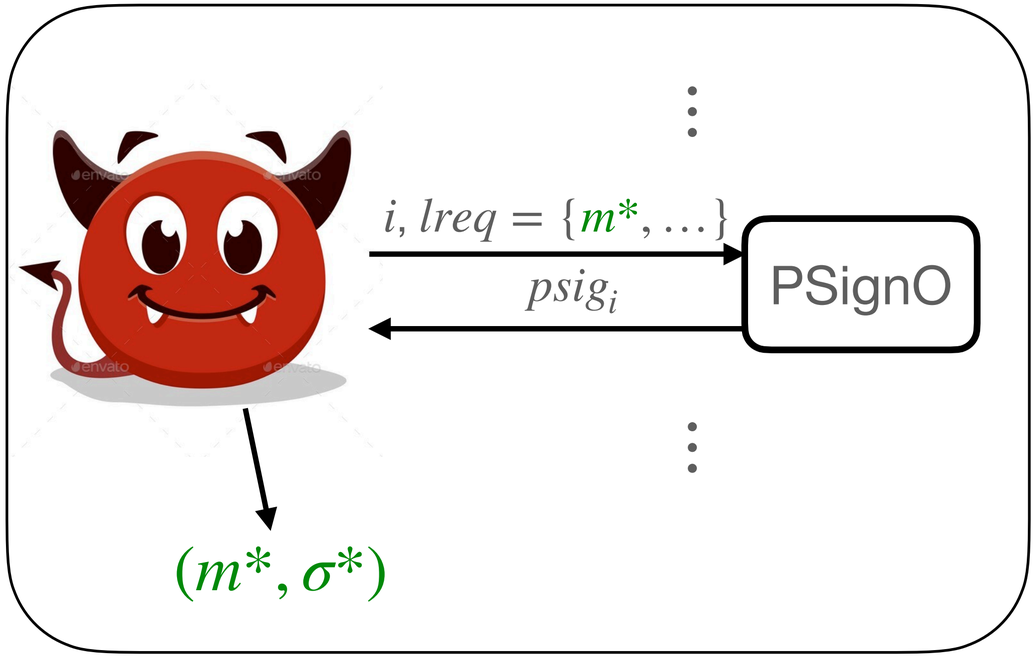
The Simplest

L : set of $(i, lreq)$ queries to PSignO

TS-UF-0 (m^*, σ^*) is a trivial forgery :

$$\exists (i, lreq) \in L : lreq\,.msg = m^*$$

At least one honest signer signed m^*



The Simplest

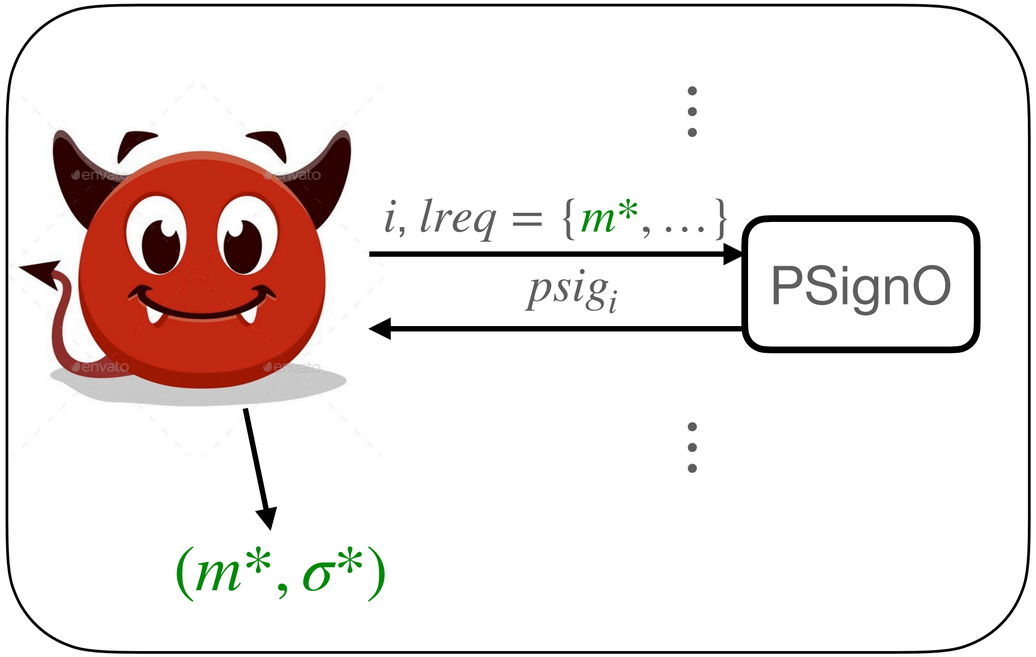
L : set of $(i, lreq)$ queries to PSignO

TS-UF-0 (m^*, σ^*) is a trivial forgery :

$$\exists (i, lreq) \in L : lreq\,.msg = m^*$$

At least one honest signer signed m^*

Most of previous works consider this
[GJKR96, KY02, Bol03, Wee11, KG20, BGG+18]



The Simplest

L : set of $(i, lreq)$ queries to PSignO

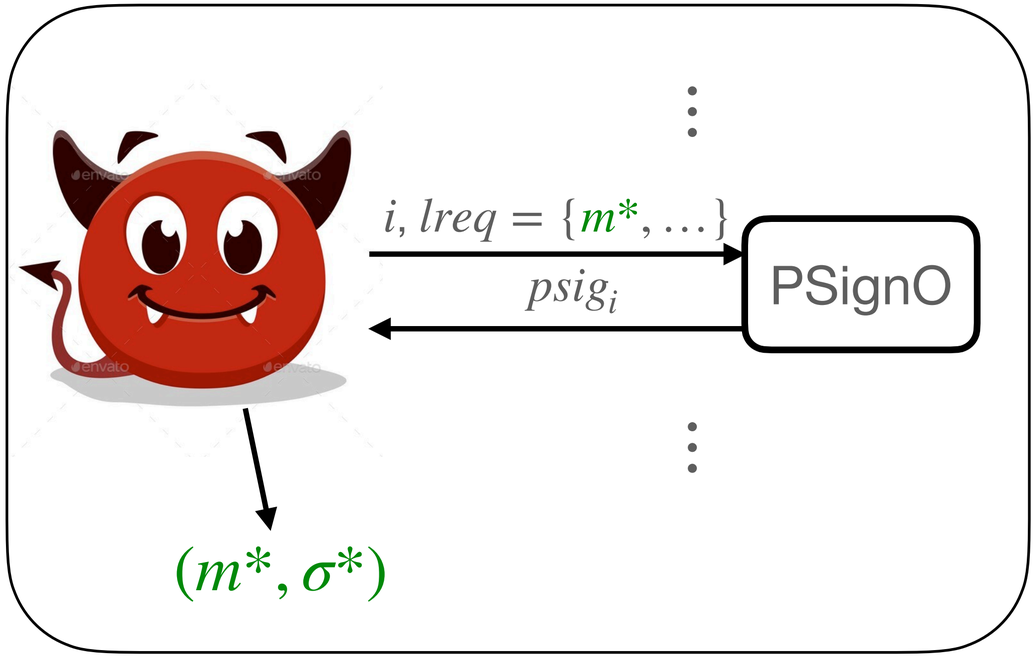
TS-UF-0 (m^*, σ^*) is a trivial forgery :

$$\exists (i, lreq) \in L : lreq\,.msg = m^*$$

At least one honest signer signed m^*

Most of previous works consider this
[GJKR96, KY02, Bol03, Wee11, KG20, BGG+18]

When $|CS| < t - 1$, there are **stronger** security considerations



The Simplest

L : set of $(i, lreq)$ queries to PSignO

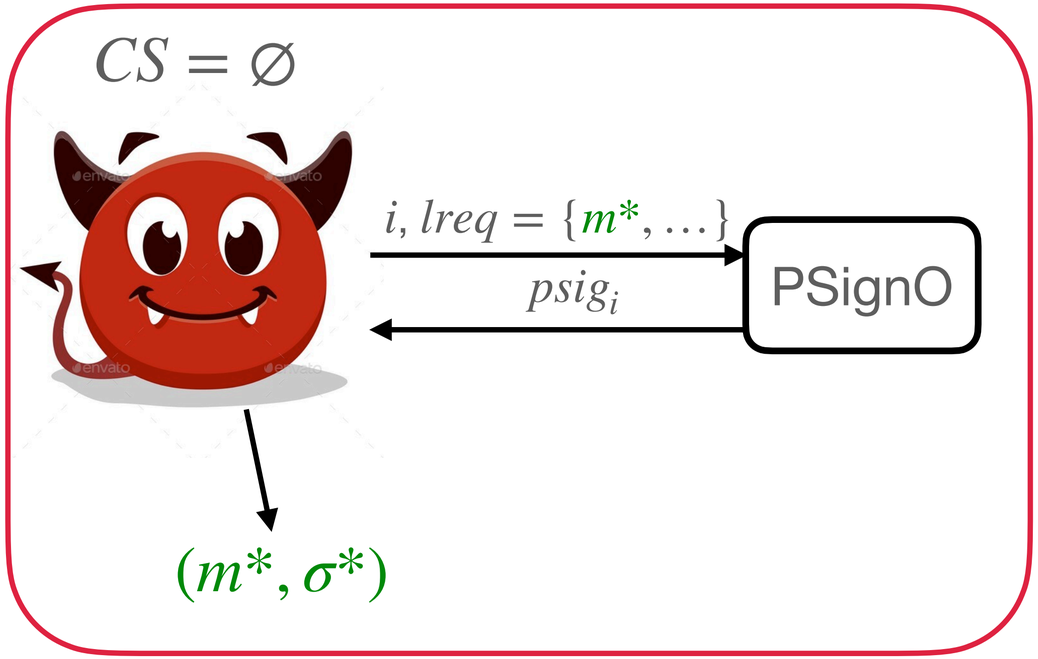
TS-UF-0 (m^*, σ^*) is a trivial forgery :

$$\exists (i, lreq) \in L : lreq\,.msg = m^*$$

At least one honest signer signed m^*

Most of previous works consider this
[GJKR96, KY02, Bol03, Wee11, KG20, BGG+18]

When $|CS| < t - 1$, there are **stronger** security considerations



The Simplest

L : set of $(i, lreq)$ queries to PSignO

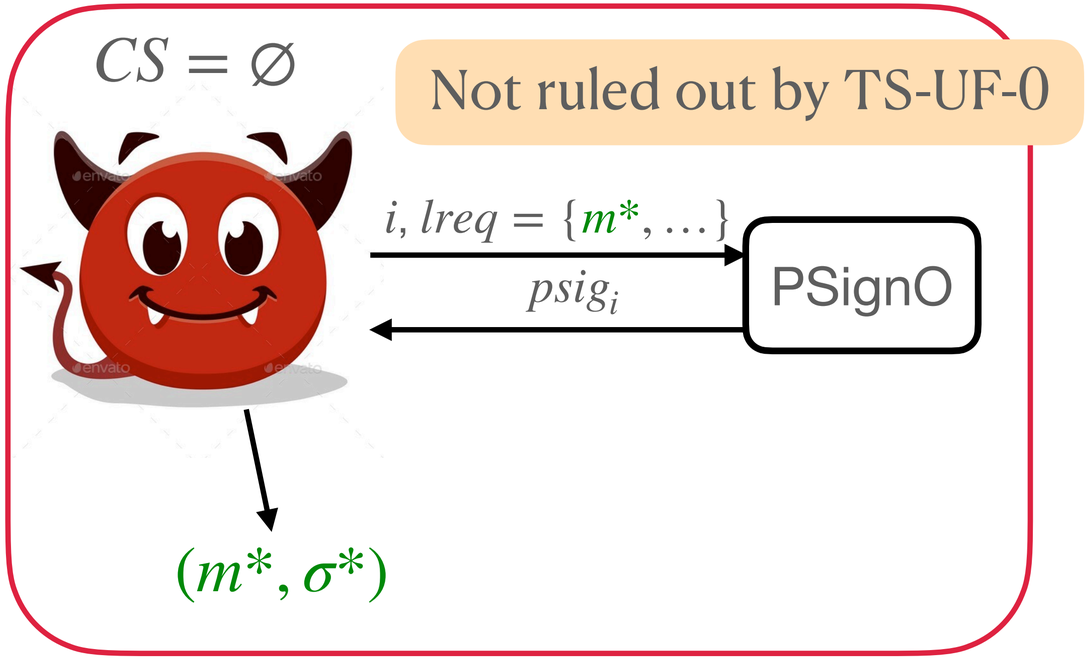
TS-UF-0 (m^*, σ^*) is a trivial forgery :

$$\exists (i, lreq) \in L : lreq\,.msg = m^*$$

At least one honest signer signed m^*

Most of previous works consider this
[GJKR96, KY02, Bol03, Wee11, KG20, BGG+18]

When $|CS| < t - 1$, there are **stronger** security considerations



The Simplest

L : set of $(i, lreq)$ queries to PSignO

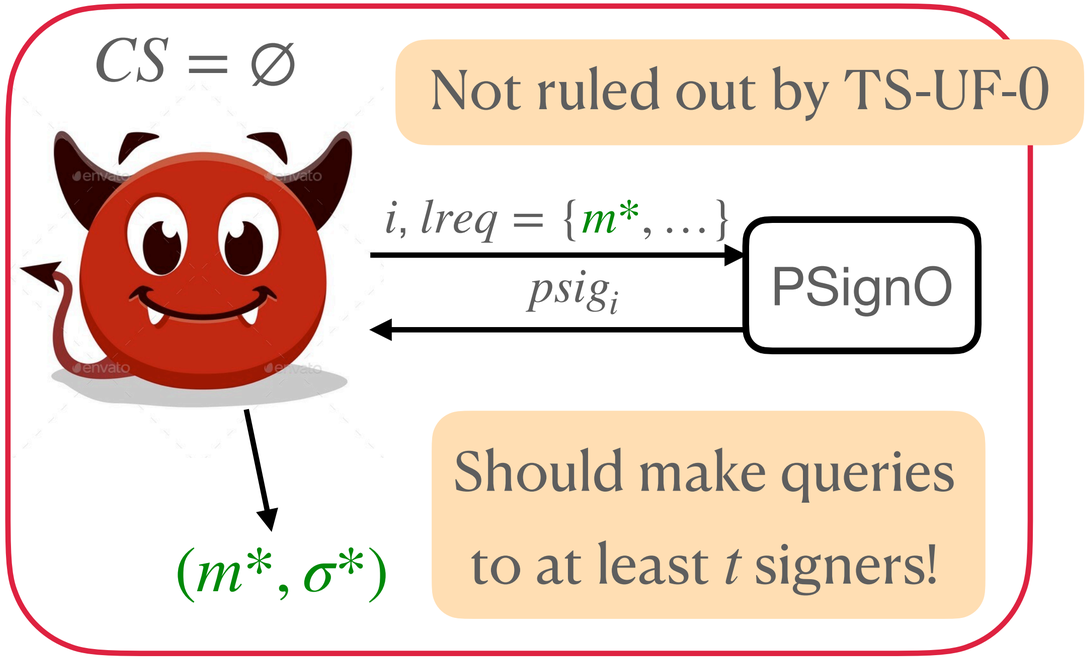
TS-UF-0 (m^*, σ^*) is a trivial forgery :

$$\exists (i, lreq) \in L : lreq\,.msg = m^*$$

At least one honest signer signed m^*

Most of previous works consider this
[GJKR96, KY02, Bol03, Wee11, KG20, BGG+18]

When $|CS| < t - 1$, there are **stronger** security considerations



Next One

Next One

$$mSS(m) := \{i : \exists lreq \text{ s.t. } (i, lreq) \in L,\ lreq\,.msg = m\}$$

Next One

$$mSS(m) := \{i : \exists lreq \text{ s.t. } (i, lreq) \in L,\ lreq\,.\,msg = m\}$$

The set of honest signers that signed m

Next One

$mSS(m) := \{i : \exists lreq \text{ s.t. } (i, lreq) \in L,\ lreq\,.\,msg = m\}$

The set of honest signers that signed m

TS-UF-0 (m^*, σ^*) is a trivial forgery : $|mSS(m^*)| > 0$

Next One

$mSS(m) := \{i : \exists lreq \text{ s.t. } (i, lreq) \in L,\ lreq\,.\,msg = m\}$

The set of honest signers that signed m

TS-UF-0 (m^*, σ^*) is a trivial forgery : $|mSS(m^*)| > 0$

TS-UF-1 (m^*, σ^*) is a trivial forgery : $|mSS(m^*)| \geq t - |CS|$

Next One

$mSS(m) := \{i : \exists lreq \text{ s.t. } (i, lreq) \in L,\ lreq\,.\,msg = m\}$

The set of honest signers that signed m

TS-UF-0 (m^*, σ^*) is a trivial forgery : $|mSS(m^*)| > 0$

TS-UF-1 (m^*, σ^*) is a trivial forgery : $|mSS(m^*)| \geq t - |CS|$

Very few previous works consider this
[Sho00, LJY14, Gro21]

Next One

$mSS(m) := \{i : \exists lreq \text{ s.t. } (i, lreq) \in L,\ lreq\,.\,msg = m\}$

The set of honest signers that signed m

TS-UF-0 (m^*, σ^*) is a trivial forgery : $|mSS(m^*)| > 0$

TS-UF-1 (m^*, σ^*) is a trivial forgery : $|mSS(m^*)| \geq t - |CS|$

Very few previous works consider this
[Sho00, LJY14, Gro21]

Even stronger?

Next One

$mSS(m) := \{i : \exists lreq \text{ s.t. } (i, lreq) \in L,\ lreq\,.\,msg = m\}$

The set of honest signers that signed m

TS-UF-0 (m^*, σ^*) is a trivial forgery : $|mSS(m^*)| > 0$

TS-UF-1 (m^*, σ^*) is a trivial forgery : $|mSS(m^*)| \geq t - |CS|$

Very few previous works consider this
[Sho00, LJY14, Gro21]

Even stronger?

Partial sigs for m^* but from **different** $lreq$ can be combined

Go Beyond

TS-UF-1 (m^*, σ^*) is a trivial forgery : $|mSS(m^*)| \geq t - |CS|$

Go Beyond

TS-UF-1 (m^*, σ^*) is a trivial forgery : $|mSS(m^*)| \geq t - |CS|$

$rSS(lreq) := \{i : (i, lreq) \in L\}$

Go Beyond

TS-UF-1 (m^*, σ^*) is a trivial forgery : $|mSS(m^*)| \geq t - |CS|$

$rSS(lreq) := \{i : (i, lreq) \in L\}$

The set of honest signers that answered $lreq$

Go Beyond

TS-UF-1 (m^*, σ^*) is a trivial forgery : $|mSS(m^*)| \geq t - |CS|$

$rSS(lreq) := \{i : (i, lreq) \in L\}$

The set of honest signers that answered $lreq$

TS-UF-2 (m^*, σ^*) is a trivial forgery : $\exists\ lreq : lreq.msg = m^*$

$|rSS(lreq)| \geq t - |CS|$

Go Beyond

TS-UF-1 (m^*, σ^*) is a trivial forgery : $|mSS(m^*)| \geq t - |CS|$

$rSS(lreq) := \{i : (i, lreq) \in L\}$

The set of honest signers that answered $lreq$

TS-UF-2 (m^*, σ^*) is a trivial forgery : $\exists\ lreq : lreq.msg = m^*$

$|rSS(lreq)| \geq t - |CS|$

TS-UF-4 (m^*, σ^*) is a trivial forgery : $\exists\ lreq : lreq.msg = m^*$

$rSS(lreq) = lreq.SS \setminus CS$

Go Beyond

TS-UF-1 (m^*, σ^*) is a trivial forgery : $|mSS(m^*)| \geq t - |CS|$

$rSS(lreq) := \{i : (i, lreq) \in L\}$

The set of honest signers that answered $lreq$

TS-UF-2 (m^*, σ^*) is a trivial forgery : $\exists\ lreq : lreq.msg = m^*$

$|rSS(lreq)| \geq t - |CS|$

TS-UF-4 (m^*, σ^*) is a trivial forgery : $\exists\ lreq : lreq.msg = m^*$

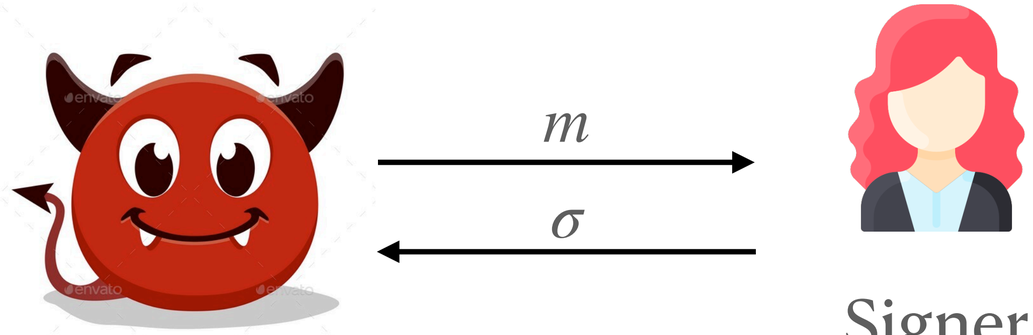
Our Highest

$rSS(lreq) = lreq.SS \setminus CS$

Strong Unforgeability

Strong Unforgeability

For signature schemes



The adversary cannot forge $\sigma^*(\neq \sigma)$ for m

Strong Unforgeability

For threshold signatures

$lreq$

$psig_i$

Signer i

Strong Unforgeability

For threshold signatures

$lreq$

$psig_i$

Signer i

What is the issued signature?

Strong Unforgeability

For threshold signatures

$lreq$

$psig_i$

Signer i

What is the issued signature?

For schemes with **deterministic signing**

$$(pk, lreq) \longrightarrow (m, \sigma)$$

Strong Unforgeability

For threshold signatures

$lreq$

$psig_i$

Signer i

What is the issued signature?

For schemes with **deterministic signing**

$$(pk, lreq) \xrightarrow{\Phi} (m, \sigma)$$

Strong Unforgeability

For threshold signatures

$lreq$

$psig_i$

Signer i

What is the issued signature?

For schemes with **deterministic signing**

$$(pk, lreq) \xrightarrow{\Phi} (m, \sigma)$$

$m = lreq\,.msg$

Strong Unforgeability

For threshold signatures

$lreq$

$psig_i$

Signer i

What is the issued signature?

For schemes with **deterministic signing**

$$(pk, lreq) \xrightarrow{\Phi} (m, \sigma)$$

Depends on $lreq$ so not unique for m

$m = lreq\,.msg$

Strong Unforgeability

For threshold signatures

$lreq$

$psig_i$

Signer i

What is the issued signature?

For schemes with **deterministic signing**

$(pk, lreq) \xrightarrow{\Phi} (m, \sigma)$

Depends on $lreq$ so not unique for m

FROST1/2 have this property

$m = lreq\,.msg$

Strong Unforgeability

Strong Unforgeability

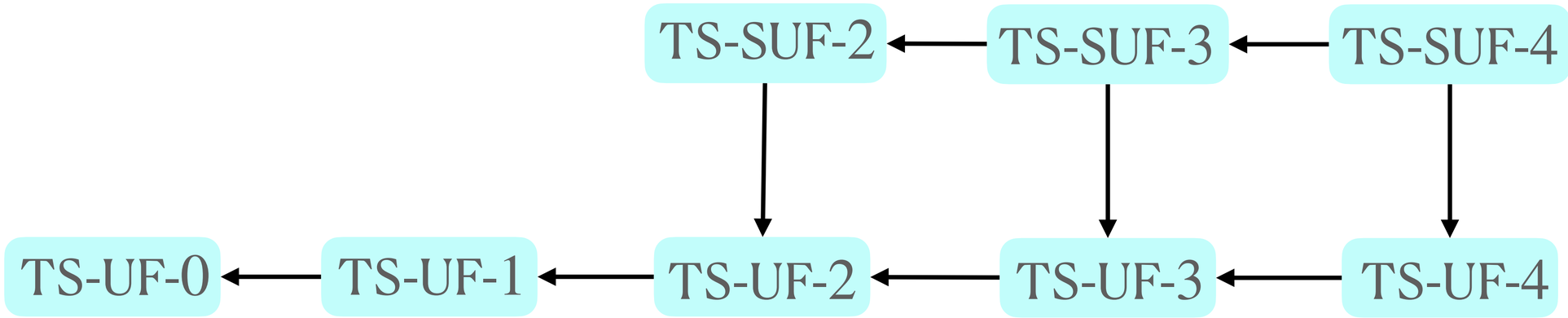
TS-SUF-2 (m^*, σ^*) is a trivial forgery : $\exists\ lreq : \Phi(pk, lreq) = (m^*, \sigma^*)$

$$|rSS(lreq)| \geq t - |CS|$$

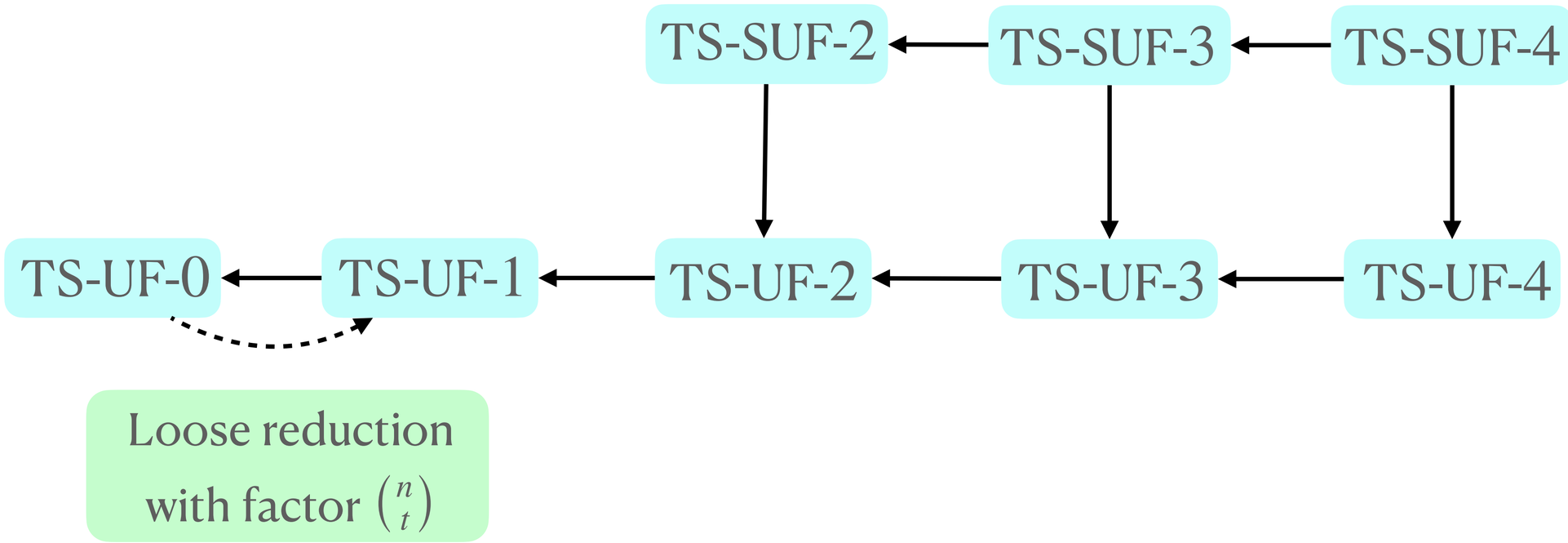
TS-SUF-4 (m^*, σ^*) is a trivial forgery : $\exists\ lreq : \Phi(pk, lreq) = (m^*, \sigma^*)$

$$rSS(lreq) = lreq\,.SS \setminus CS$$

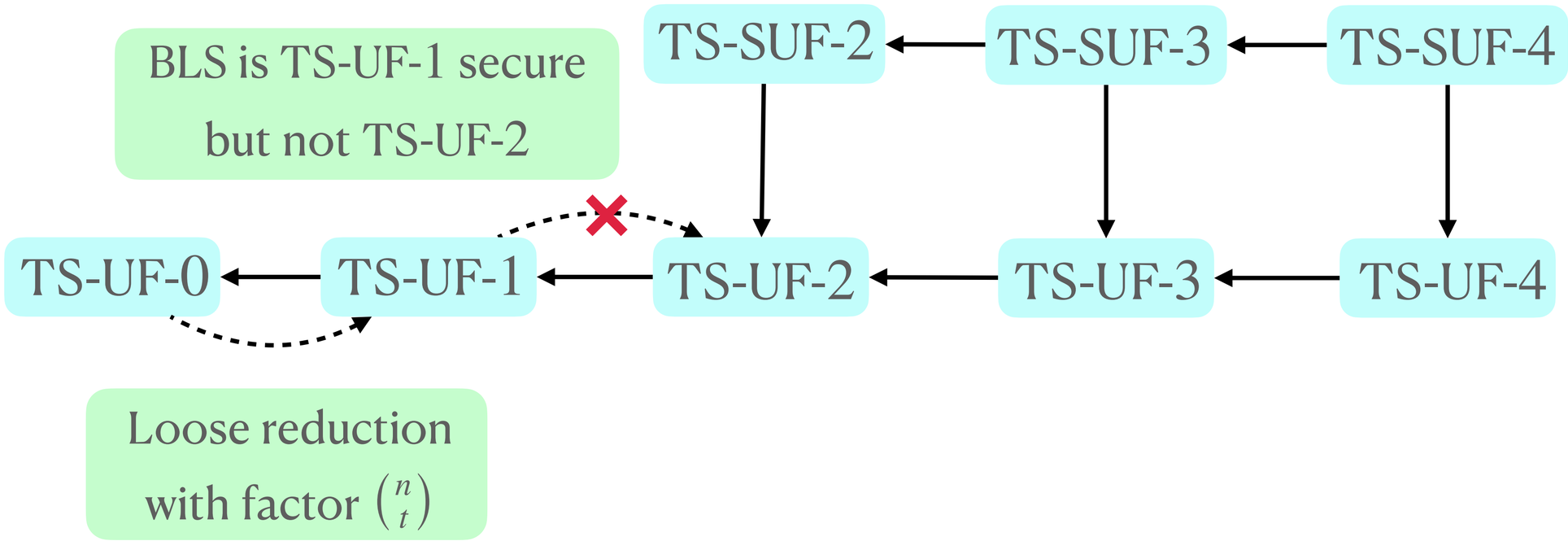
Full Picture



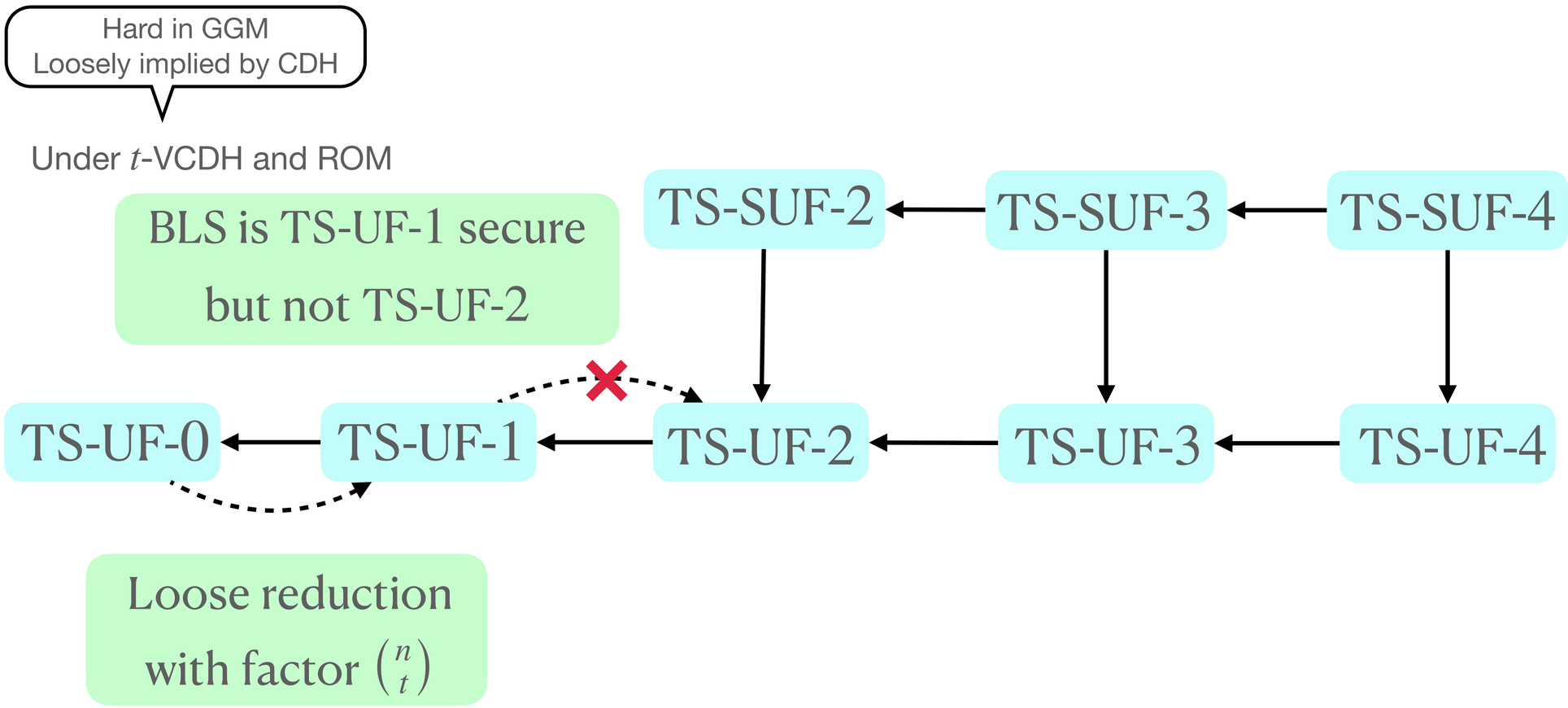
Full Picture



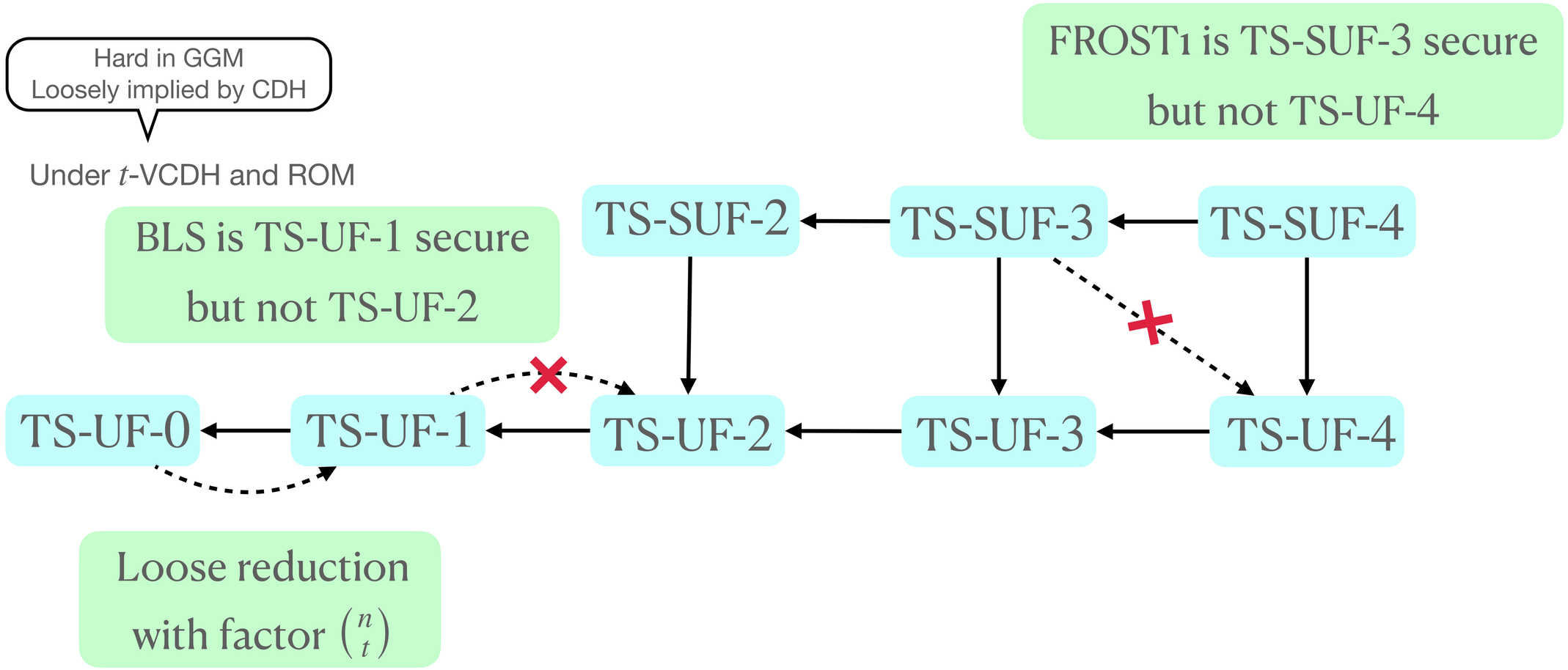
Full Picture



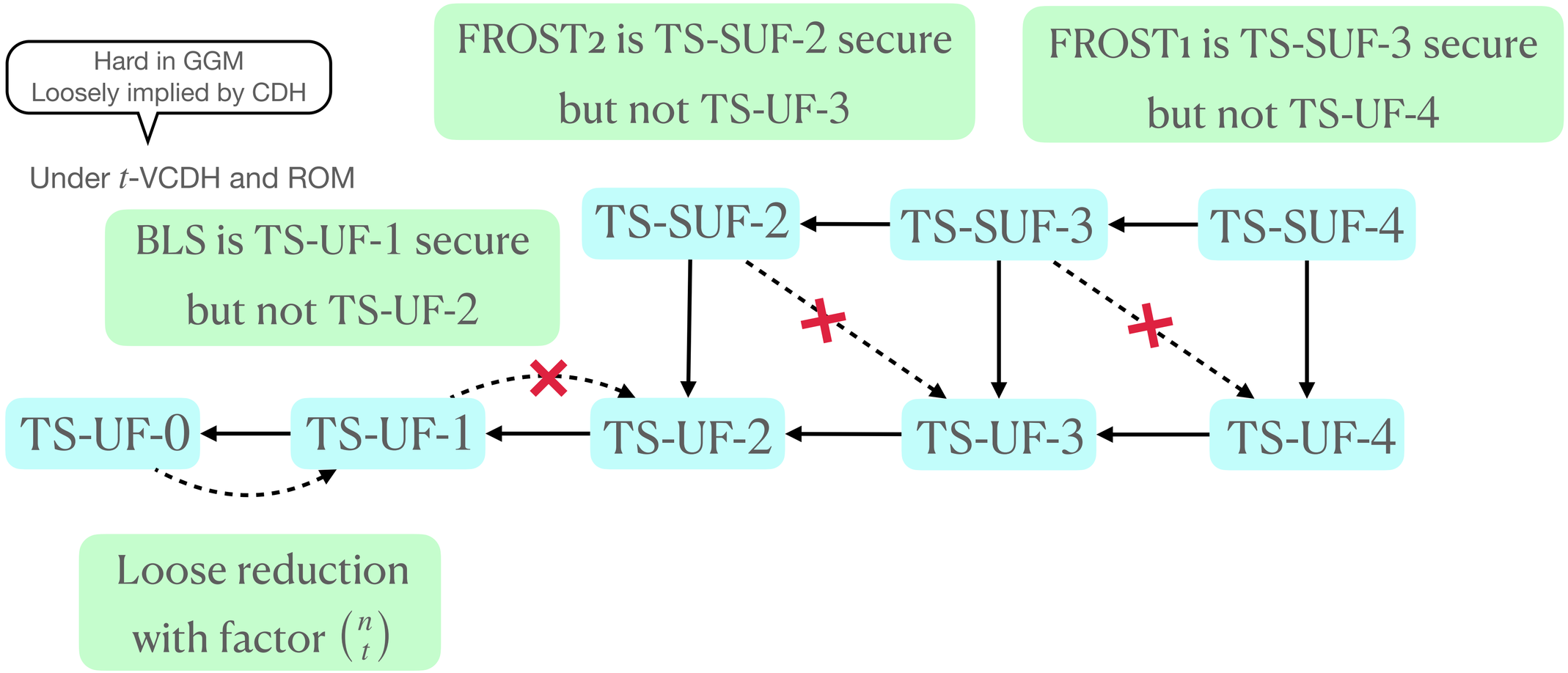
Full Picture



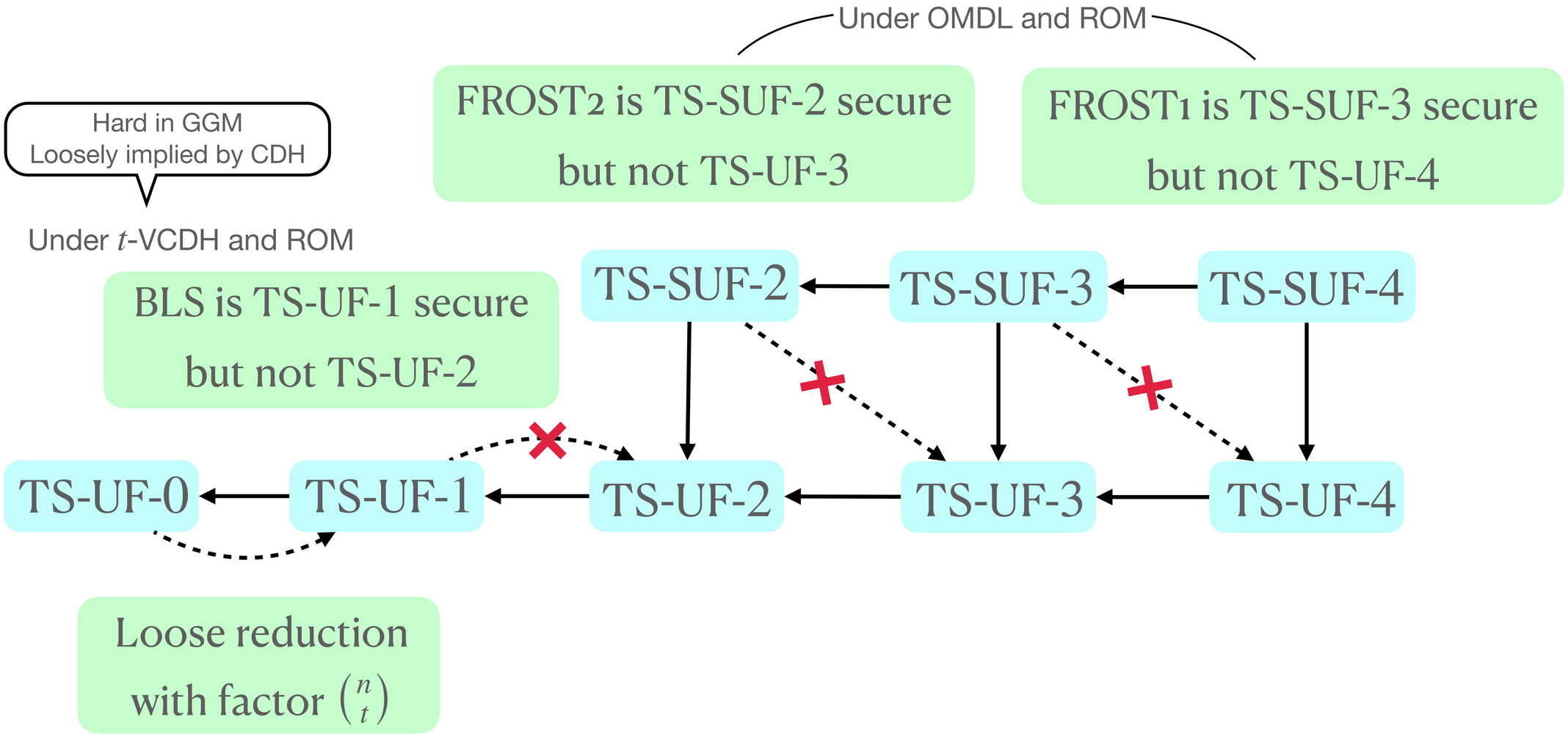
Full Picture



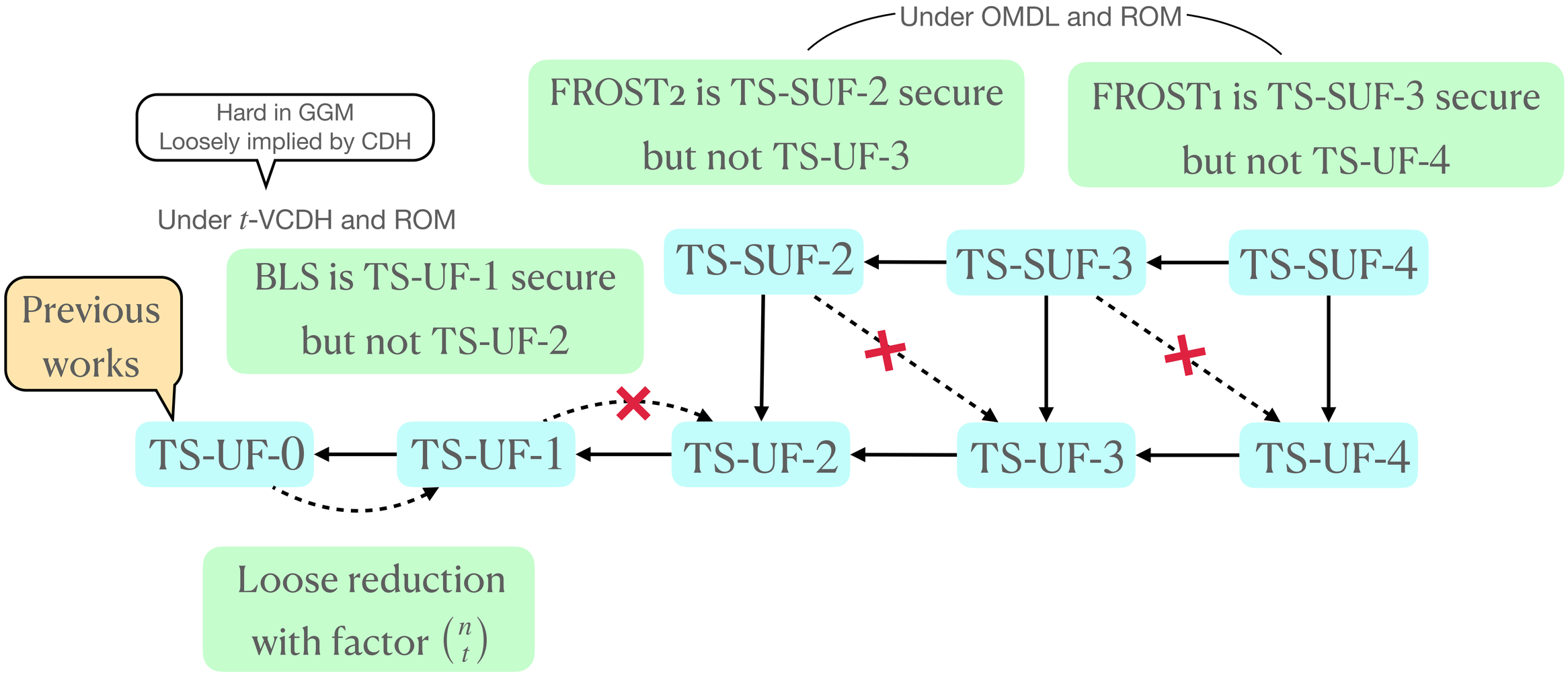
Full Picture



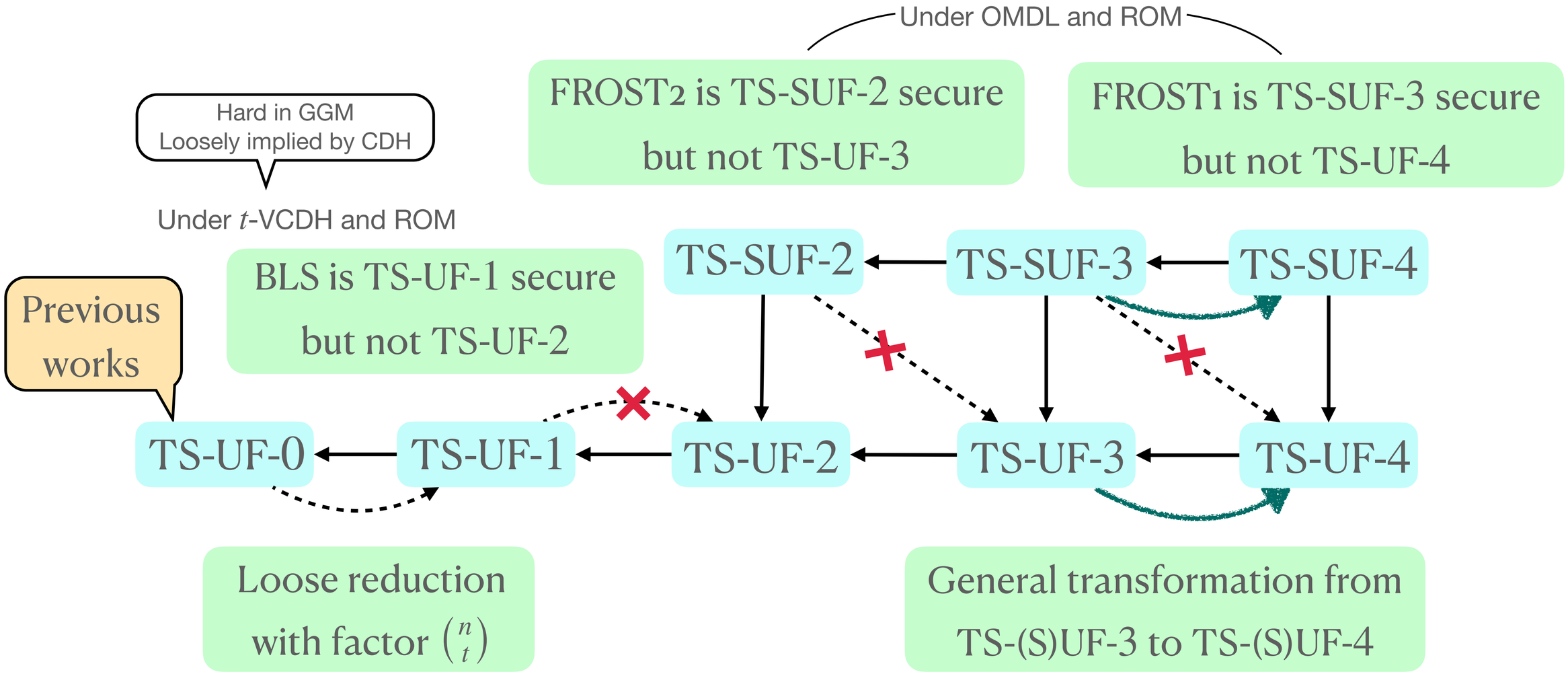
Full Picture



Full Picture



Full Picture



Separation of FROST1/2

Separation of FROST1/2

An adversary for FROST2

Separation of FROST1/2

An adversary for FROST2

$n = 4,\ t = 3$

Separation of FROST1/2

An adversary for FROST2

$n = 4,\ t = 3$

1 2 3 4

Separation of FROST1/2

An adversary for FROST2

$n = 4,\ t = 3$

$CS = \{3,4\}$

pk, sk_3, sk_4

1 2 3 4

Separation of FROST1/2

An adversary for FROST2

$n = 4,\ t = 3$

$CS = \{3,4\}$

pk, sk_3, sk_4

1,2

PPO

1 2 3 4

Separation of FROST1/2

An adversary for FROST2

$n = 4,\ t = 3$

$CS = \{3,4\}$

pk, sk_3, sk_4

1,2

$(R_1, S_1), (R_2, S_2)$

PPO

1 2 3 4

Separation of FROST1/2

An adversary for FROST2

$n = 4,\ t = 3$

$CS = \{3,4\}$

pk, sk_3, sk_4

1,2

$(R_1, S_1), (R_2, S_2)$

PPO

1 2 3 4

$lreq:\ m^*, SS = \{1,2,3\}$

Separation of FROST1/2

An adversary for FROST2

$n = 4,\ t = 3$

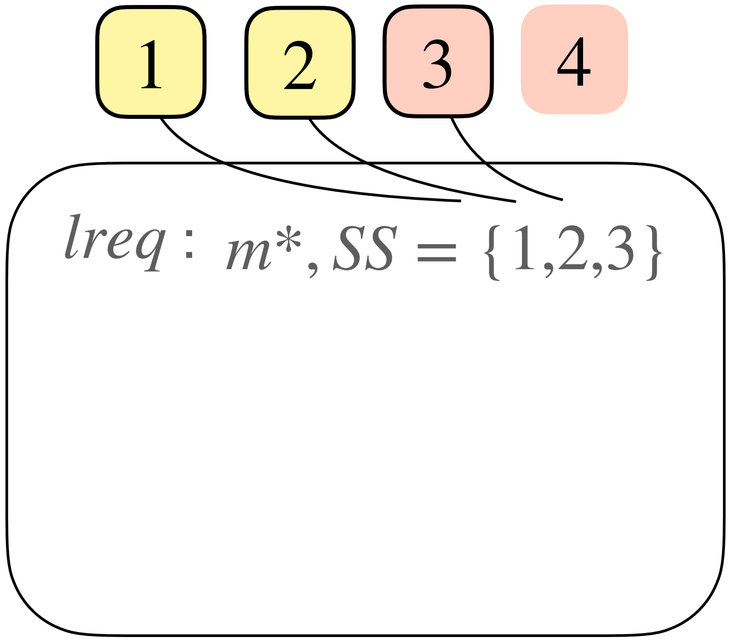
$CS = \{3,4\}$

pk, sk_3, sk_4

1,2

$(R_1, S_1), (R_2, S_2)$

PPO



Separation of FROST1/2

An adversary for FROST2

$n = 4,\ t = 3$

$CS = \{3,4\}$

pk, sk_3, sk_4

1,2

$(R_1, S_1), (R_2, S_2)$

PPO

1 2 3 4

$lreq:\ m^*, SS = \{1,2,3\}$

$$\left\{\begin{array}{ll} 1: & (R_1, S_1) \\ 2: & (R_2, S_2) \\ 3: & (R_3, S_3) \end{array}\right\}$$

Separation of FROST1/2

An adversary for FROST2

$n = 4,\ t = 3$

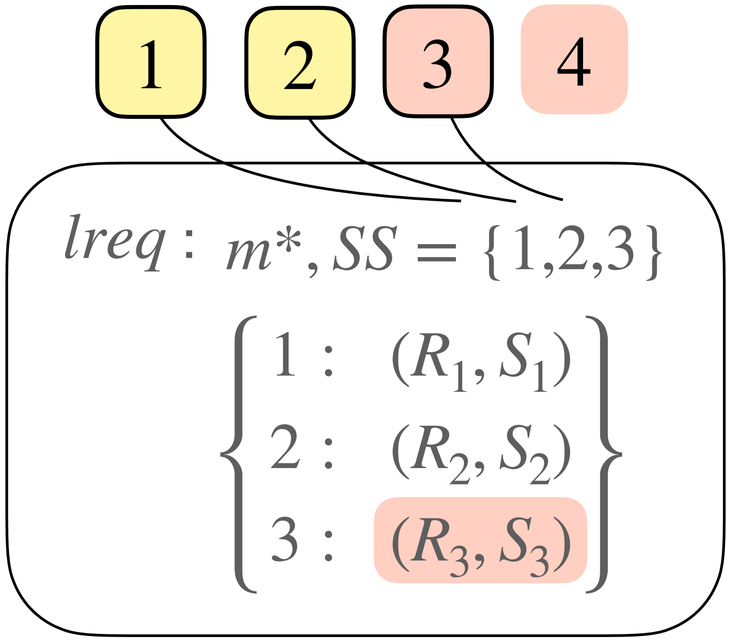
$CS = \{3,4\}$

pk, sk_3, sk_4

$1,2$

$(R_1, S_1), (R_2, S_2)$

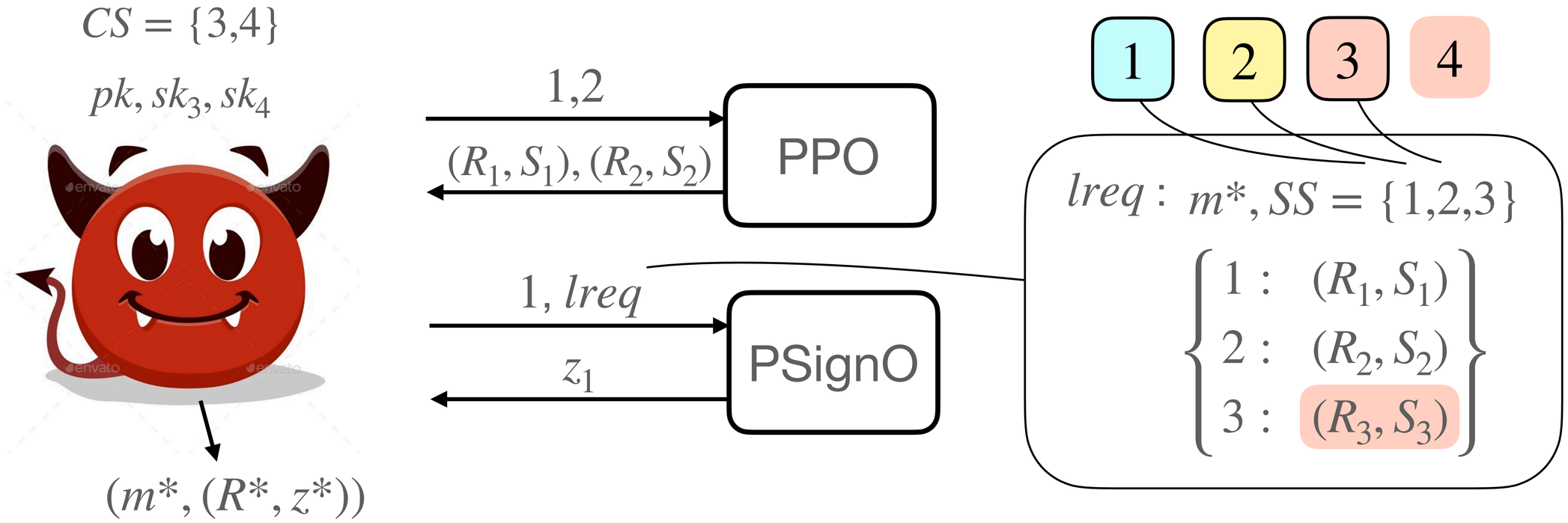
PPO



Separation of FROST1/2

An adversary for FROST2

$n = 4,\ t = 3$

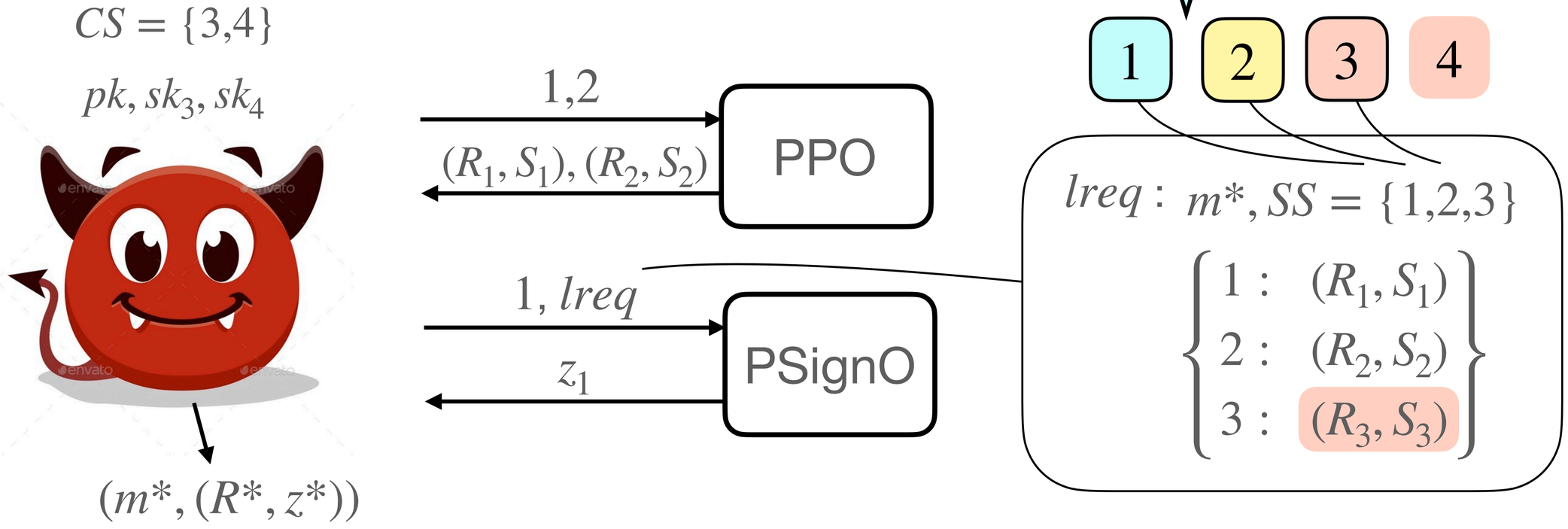


Separation of FROST1/2

An adversary for FROST2

$n = 4,\ t = 3$

For FROST1, the adversary has to query **both signer 1 and 2**

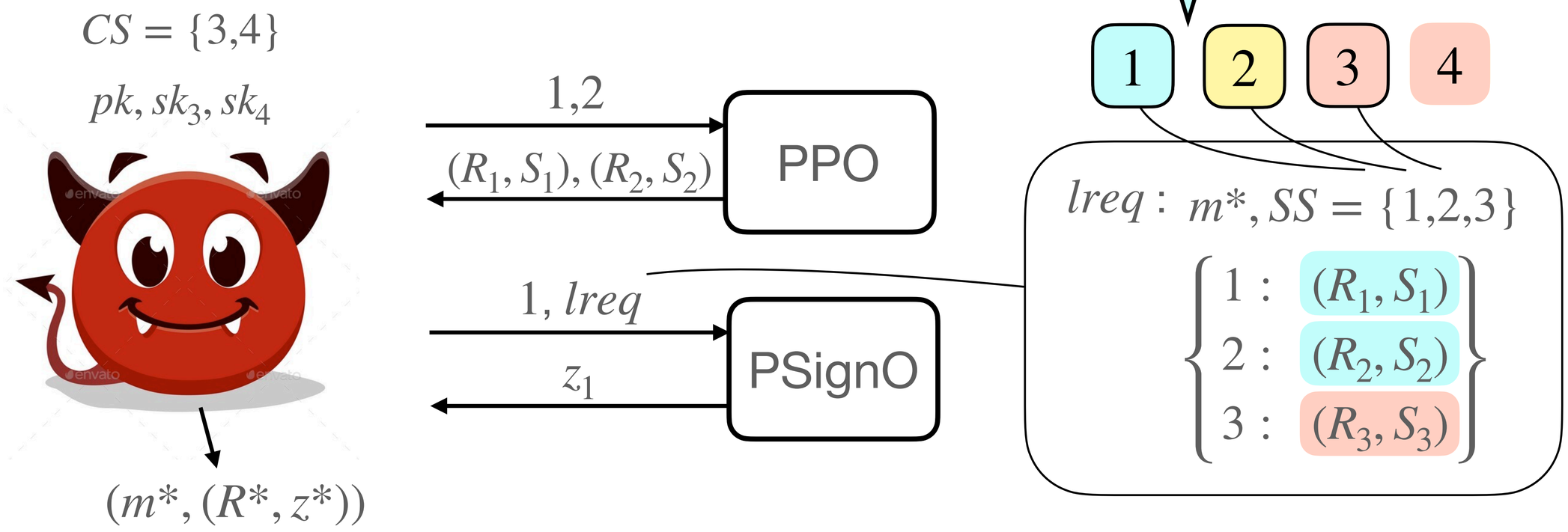


Separation of FROST1/2

An adversary for FROST2

$n = 4,\ t = 3$

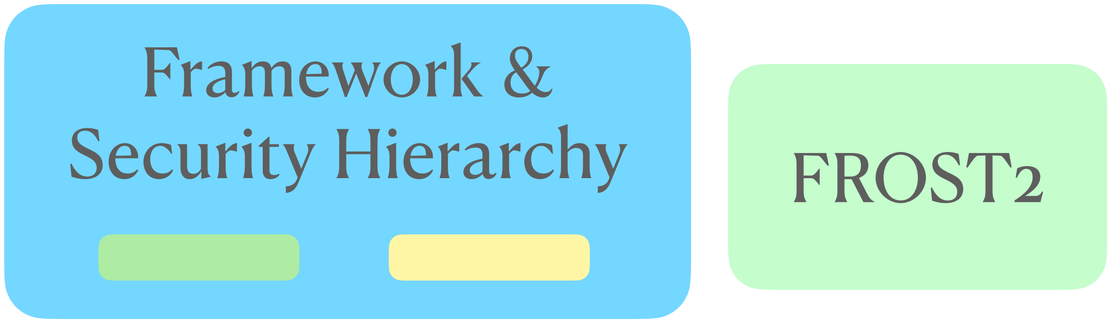
For FROST1, the adversary has to query **both signer 1 and 2**



Conclusion & Future work

Conclusion & Future work

Conclusion & Future work



Conclusion & Future work

Framework &
Security Hierarchy

FROST2

Better-than-advertised
security of BLS, FROST

Conclusion & Future work

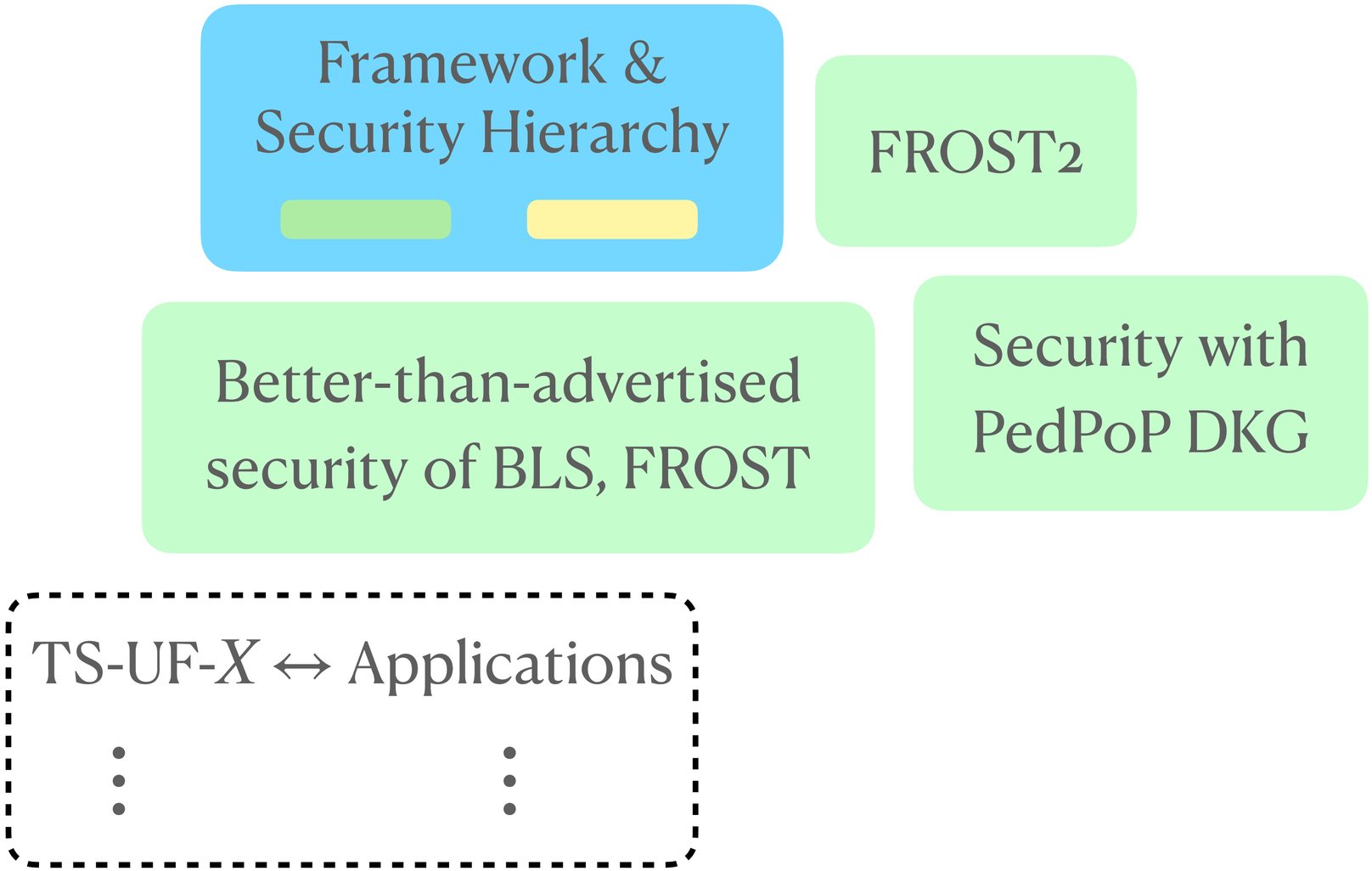
Framework & Security Hierarchy

FROST2

Better-than-advertised security of BLS, FROST

Security with PedPoP DKG

Conclusion & Future work



Conclusion & Future work

Framework & Security Hierarchy

FROST2

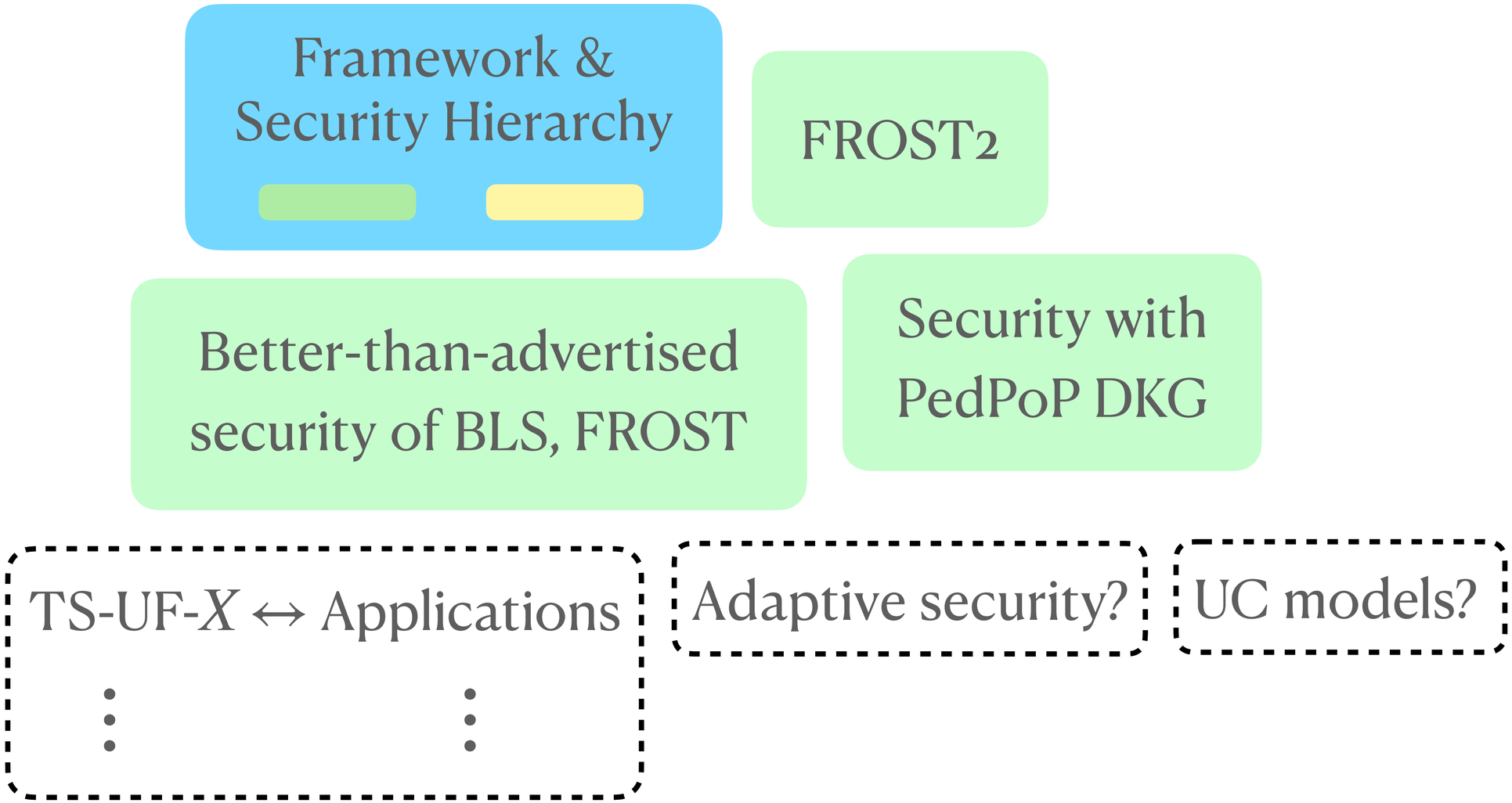
Better-than-advertised security of BLS, FROST

Security with PedPoP DKG

TS-UF-X ↔ Applications

Adaptive security?

Conclusion & Future work



Conclusion & Future work

Framework & Security Hierarchy

FROST2

Better-than-advertised security of BLS, FROST

Security with PedPoP DKG

TS-UF-X ↔ Applications

Adaptive security?

UC models?

Framework for general DKG protocols?

Thank you!