Fully Adaptive FROST with Identifiable Aborts from AOMDL

Ruben Baecker, **Paul Gerhart**, Davide Li Calsi, Luigi Russo, Dominique Schröder, Arkady Yerukhimovich

Presented on 2026-July-07 at TCTP2: NIST Event for Threshold Call Preview Talks #2

Our Goal

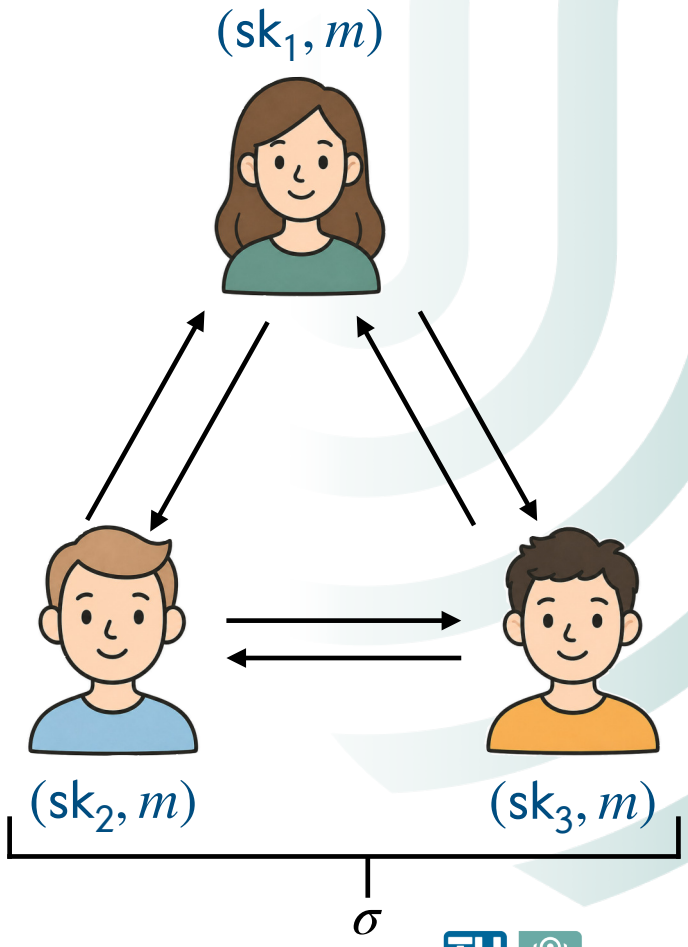
Find a Schnorr threshold signature (N1.1) scheme which

- is compatible with the **standard Schnorr** verification equation
- is unforgeable against **fully adaptive** adversaries
- has a **two-round** signing protocol
- is provable from **established assumptions**
- has **identifiable aborts**

TU WIEN PRIVACY ENHANCING TECHNOLOGIES

Threshold Signatures [Des88]

- Signing key is shared $(\mathsf{vk}, \mathsf{sk}) \rightarrow (\mathsf{vk}, (\mathsf{sk}_0, \ldots, \mathsf{sk}_n))$
- t-out-of-n signers can sign
- An adversary corrupting up to $t - 1$ signers cannot



(Single-Party) Schnorr Signatures [Sch90]

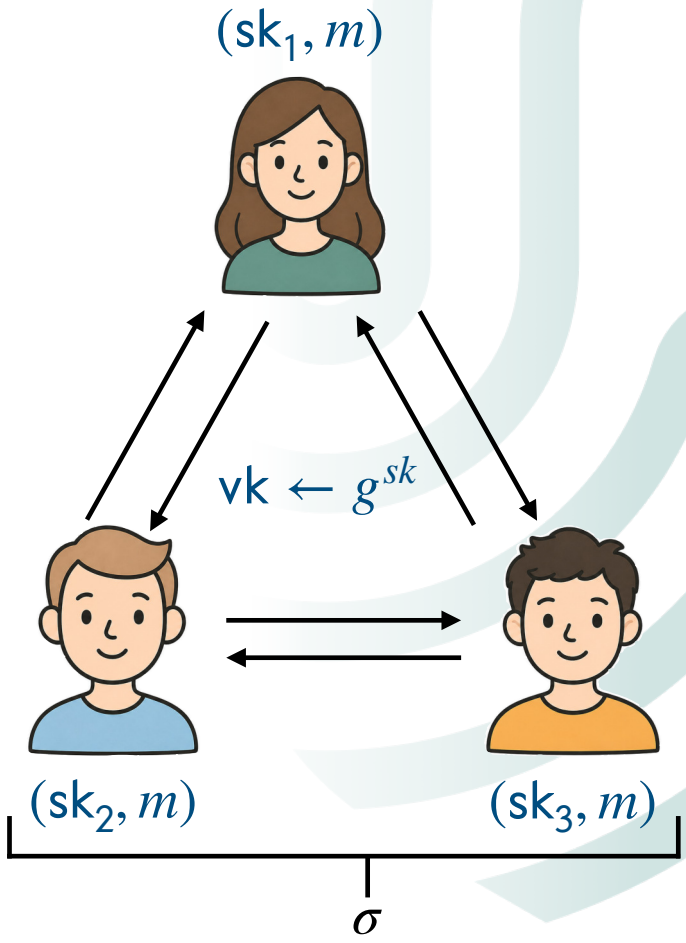
Sign(sk, m)

1) $r \leftarrow^{\$} \mathbb{Z}_p; R = g^r$
2) $c \leftarrow \mathsf{H}(\mathsf{vk}, R, m)$
3) $s \leftarrow \mathsf{sk} \cdot c + r$
4) return $\sigma := (R, s)$

Vrfy(vk, m, σ)

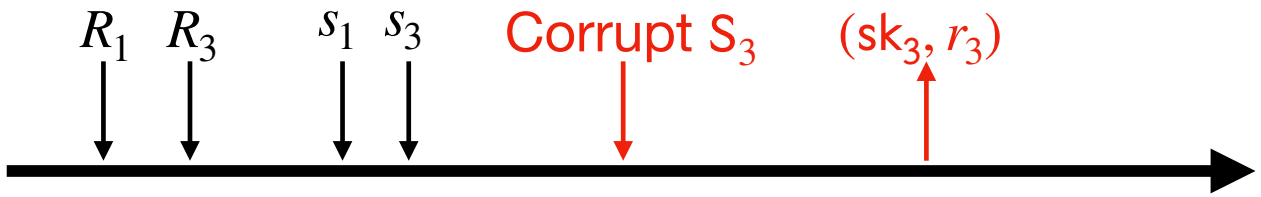
1) parse σ as (R, s)
2) $c \leftarrow \mathsf{H}(\mathsf{vk}, R, m)$
3) return $g^s = \mathsf{vk}^c \cdot R$

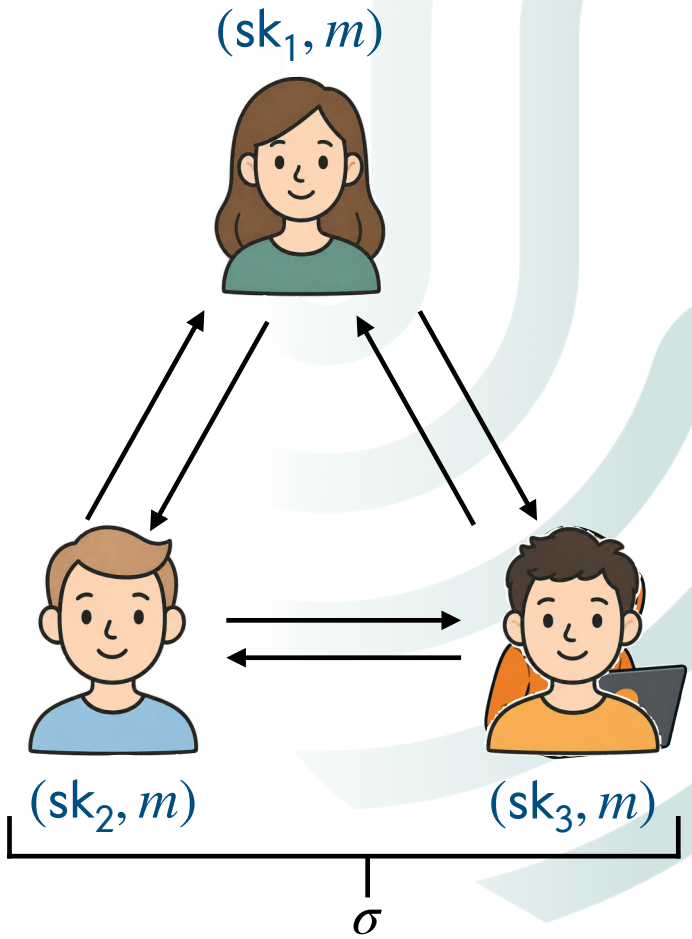
Schnorr threshold signature $\sigma = (R, s)$ must satisfy $g^s = \mathsf{vk}^c \cdot R$



Adaptive Security

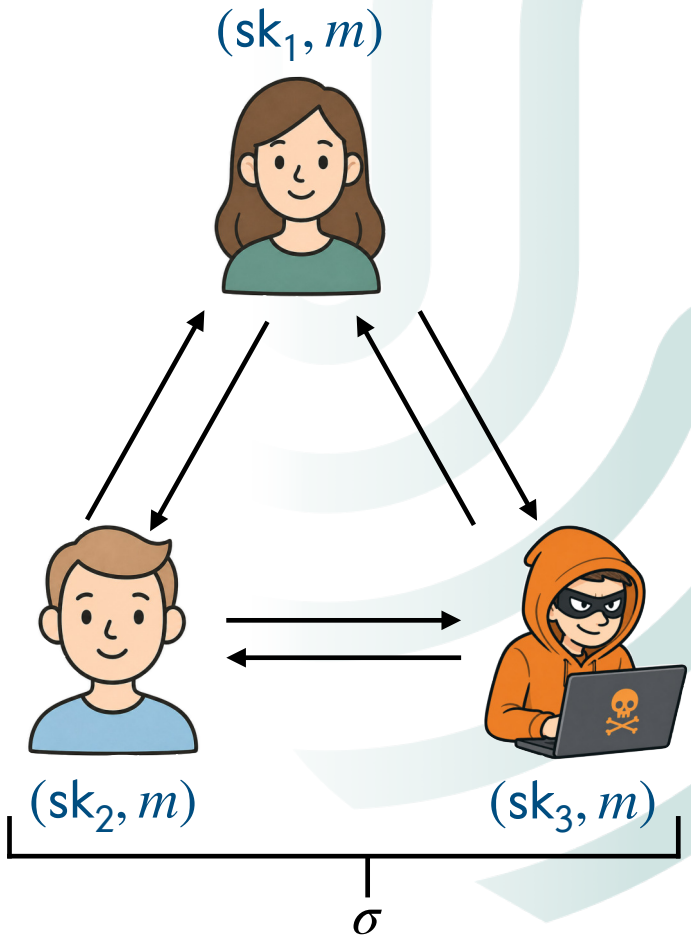
- An adversary can corrupt up to $t_c < t \leq n$ parties during experiment execution (n total signers, t out of which can sign)
- Corruption reveals signing keys and internal states
- Maximal possible corruption $t_c = t - 1$ is **fully adaptive security**





Identifiable Aborts

- If a signing session fails, the signers can decide on at least one party that misbehaved
- Enables recovery: exclude identified parties and restart the session
- Weaker than robustness, but it does not require honest majority



Existing Threshold Signatures

[DDFY94][GRJK00][Sho00][Bol03][GGN16]
[GG18][LN18][DKLs19][DOK+20][GKSS20][CGG+20]
[CCL+20][DJN+20] [YCX21][Pet21][ANO+22][GS22][BL22][DR24]
[GJKR96][GJKR07][SS01][GJKR03]
[MPS19]
[NRSW20] [KG20] [CGGMP20]
[NRS21] [CKM21]
[BTZ22] [BCKMTZ22] [RRJSS22] [Lin22]
[CGRS23] [CKM23]
[KRT24]
[BCLTZ25][CKKTZ25][BDLR25][BDLR25][Chen25]

Our goal:

- **standard Schnorr** verification
- **fully adaptive** adversaries
- **two-round** signing protocol
- **established assumptions**
- **identifiable aborts**

Our Starting Point: FROST [KomGol20]

$\mathsf{S}_1(\mathsf{sk}_1, m)$

// Round 1

1) $(d_1, e_1) \leftarrow^{\$} \mathbb{Z}_p^2$
2) $D_1 \leftarrow g^{d_1};\ E_1 \leftarrow g^{e_1}$

$\xrightarrow{(D_1, E_1)}$
$\xleftarrow{(D_2, E_2)}$

// Round 2

3) $D \leftarrow \prod D_j;\ E \leftarrow \prod E_j$
4) $b \leftarrow \mathsf{H}_{\mathsf{non}}(\mathsf{vk}, S, m, \{\mathsf{pm}_1\})$
5) $R \leftarrow D \cdot E^b$
6) $c \leftarrow \mathsf{H}_{\mathsf{sig}}(\mathsf{vk}, R, m)$
7) $s_1 \leftarrow c \cdot \mathsf{sk}_1 + b \cdot e_1 + d_1$

$\xrightarrow{s_1}$
$\xleftarrow{s_2}$

- Round 1: Exchange **two** nonce commitments
  - Build R as a random linear combination of both commitment products (in the exponent)
  - Randomization helps avoid ROS attacks
- Round 2: Exchange partial signature

The AOMDL Assumption and AGM

AOMDL = Algebraic One-More Dlog Assumption [NicRufSeu21] AGM = Algebraic Group Model [FucKilLos18]

- AOMDL: algebraic version of the one-more Dlog assumption
  - Adversary learns Q challenge Dlog elements
  - Has Q-1 times access to a Dlog oracle
  - Finds an additional Dlog
- Algebraic Group Model (AGM):
  - When the adversary sends a group element, it also sends an explanation

$$Z = g^{\alpha} \cdot \prod Y_i^{\beta_i}$$

TU WIEN PRIVACY ENHANCING TECHNOLOGIES

Impossibilities for Adaptive Security

On the Adaptive Security of FROST

Elizabeth Crites[1], Jonathan Katz[2], Chelsea Komlo[3], Stefano Tessaro[4], and Chenzhi Zhu[4]

[1] Web3 Foundation elizabeth@web3.foundation
[2] Google, USA jkatz2@gmail.com
[3] University of Waterloo, NEAR One, Waterloo, Canada ckomlo@uwaterloo.ca
[4] Paul G. Allen School of Computer Science & Engineering
University of Washington, Seattle, US
{zhucz20,tessaro}@cs.washington.edu

On the Adaptive Security of Key-Unique Threshold Signatures

Elizabeth Crites[1], Chelsea Komlo[2], and Mary Maller[3]

[1] Web3 Foundation
[2] University of Waterloo & NEAR One
[3] Ethereum Foundation & PQShield
elizabeth@web3.foundation, ckomlo@uwaterloo.ca, mary.maller@ethereum.org

A Plausible Attack on the Adaptive Security of Threshold Schnorr Signatures

Elizabeth Crites and Alistair Stewart

Web3 Foundation
firstname@web3.foundation

Proving FROST Adaptively Secure[CKKTZ25]

- Rewinding-based security proofs to the AOMDL assumption are ruled out by [CKM'25]
- What if we use the algebraic group model (AGM)?

Simulate $S_1(sk, m)$

Key Generation

1) $\mathsf{vk}_i \leftarrow \mathsf{Chal}()$

First round simulation

2) $D, E \leftarrow \mathsf{Chal}()$

Second round simulation

3) $s_i \leftarrow \mathsf{DLog}(\mathsf{vk}_i^c \cdot R)$

$$\begin{pmatrix} c_1 & 1 & 1 & 1 & 0 & 0 & 0 & b_1 & b_1 & b_1 & 0 & 0 & 0 \\ c_2 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & b_2 & b_2 & b_2 \\ c^* + \alpha_{\mathbf{vk}} & \beta_1^{(1)} & \beta_1^{(2)} & \beta_1^{(3)} & \beta_2^{(1)} & \beta_2^{(2)} & \beta_2^{(3)} & \gamma_1^{(1)} & \gamma_1^{(2)} & \gamma_1^{(3)} & \gamma_2^{(1)} & \gamma_2^{(2)} & \gamma_2^{(3)} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{sk} \\ d_1^{(1)} \\ d_1^{(2)} \\ \vdots \\ d_2^{(3)} \\ e_1^{(1)} \\ \vdots \\ e_2^{(3)} \end{pmatrix}$$

Goal:

- The forgery is linearly independent of the previously solved equations
- Then, the forgery can be used to solve for one more DLog

TU WIEN PRIVACY ENHANCING TECHNOLOGIES

The LDVR Assumption[CKKTZ25]

$$\begin{pmatrix} c_1 & 1 & 1 & 1 & 0 & 0 & 0 & b_1 & b_1 & b_1 & 0 & 0 & 0 \\ c_2 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & b_2 & b_2 & b_2 \\ c^* + \alpha_{\mathbf{vk}} & \beta_1^{(1)} & \beta_1^{(2)} & \beta_1^{(3)} & \beta_2^{(1)} & \beta_2^{(2)} & \beta_2^{(3)} & \gamma_1^{(1)} & \gamma_1^{(2)} & \gamma_1^{(3)} & \gamma_2^{(1)} & \gamma_2^{(2)} & \gamma_2^{(3)} \end{pmatrix}$$

Problem:

- The adversary can adaptively corrupt signers after fixing the forgery (R^*, s^*)
  - This eliminates columns from the matrix associated to each corrupted signers
- What if the adversary finds a linear combination $(\alpha_1, \ldots, \alpha_k)$, such that

$$g^{s^*} = \prod_{i \in [C]} \mathsf{vk}_i^{\alpha_i}$$

For a set C with $|C| < t$

The forgery is **not linearly independent** of the already signed messages and does **not reveal a solution to the AOMDL assumption.**

For signer set sizes of $n \geq 128$ and more than $t/2$ corrupted parties the hardness of this problem is **only conjectured.**

Avoiding the LDVR Assumption

- If we remove the partial verification keys vk_i
  - The adversary cannot find this linear combination effectively

Removing vk_i is non-trivial, as $g^{s_i} \cdot R_i^{-c} = \mathsf{vk}_i$

TU WIEN PRIVACY ENHANCING TECHNOLOGIES

Avoid Committing using Blinding

Katsumata, Reichle, Takemure — CRYPTO'24

- Each pair of signers (i, j) receives a secret key $k_{i,j} \leftarrow^{\$} \{0,1\}^{\lambda}$
- For each other signer in the signing set, the signer i computes the blinding value $B_{i,j} = \mathsf{H}(k_{i,j}, sid)$

  In addition, use the Kronecker Delta $\Delta_{i,j} = \begin{cases} -1, & \text{if } i < j, \\ +1, & \text{if } i > j\,. \end{cases}$

- Combining both, each pair of blindings cancels out

$$B_{i,j} \cdot \Delta_{i,j} + B_{j,i} \cdot \Delta_{j,i} = 0$$

TU WIEN PRIVACY ENHANCING TECHNOLOGIES

Our Scheme: FaFROST

$\mathsf{S}_1(\mathsf{sk}, m)$

1) $(d_1, e_1) \leftarrow^{\$} \mathbb{Z}_p^2$
2) $D_1 \leftarrow g^{d_1};\ E_1 \leftarrow g^{e_1}$

(D_1, E_1) ⇄ (D_2, E_2)

3) $D \leftarrow \prod D_j;\ E \leftarrow \prod E_j$
4) $b \leftarrow \mathsf{H}_{\mathsf{non}}(\mathsf{vk}, S, m, \{\mathsf{pm}_1\})$
5) $R \leftarrow D \cdot E^b$
6) $c \leftarrow \mathsf{H}_{\mathsf{sig}}(\mathsf{vk}, R, m)$
7) $B_1^s \leftarrow \sum \mathsf{H}_{\mathsf{s}}(k_{i,j}, \{\mathsf{pm}_1\}) \cdot \Delta_{i,j}$
8) $s_1 \leftarrow c \cdot \mathsf{sk}_1 + b \cdot e_1 + d_1$
9) $s_1' \leftarrow s_1 + B_1^s$

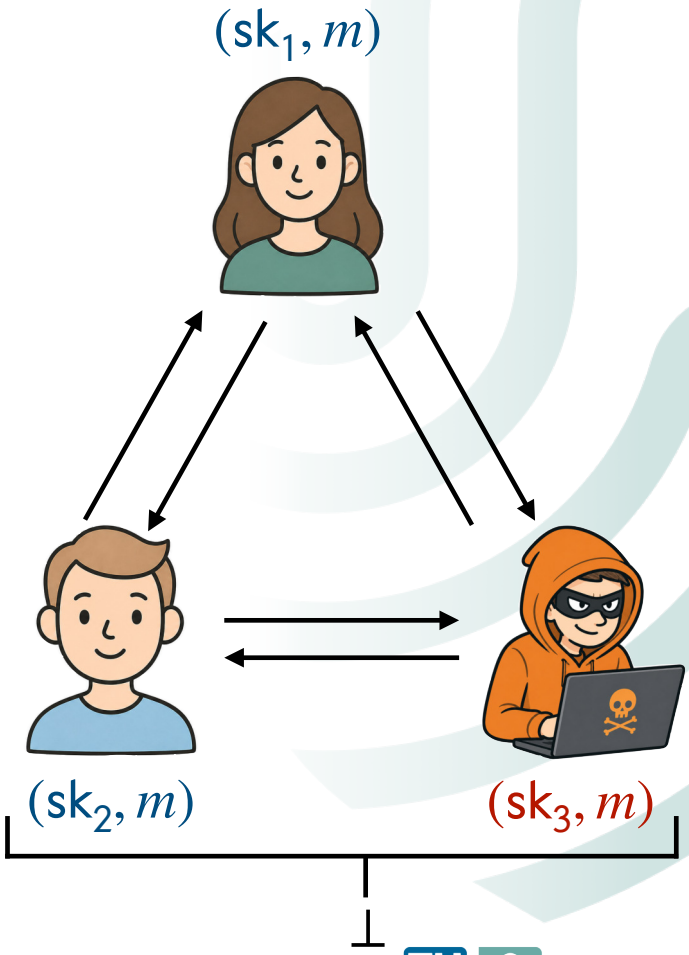
s_1' ⇄ s_2'

- Round 1:
  - Exchange **two** nonce commitments (unchanged)
- Round 2:
  - Compute partial signatures
  - Blind partial signatures using pairwise keys
  - Exchange blinded partial signatures (leaves communication unchanged)

TU WIEN PRIVACY ENHANCING TECHNOLOGIES

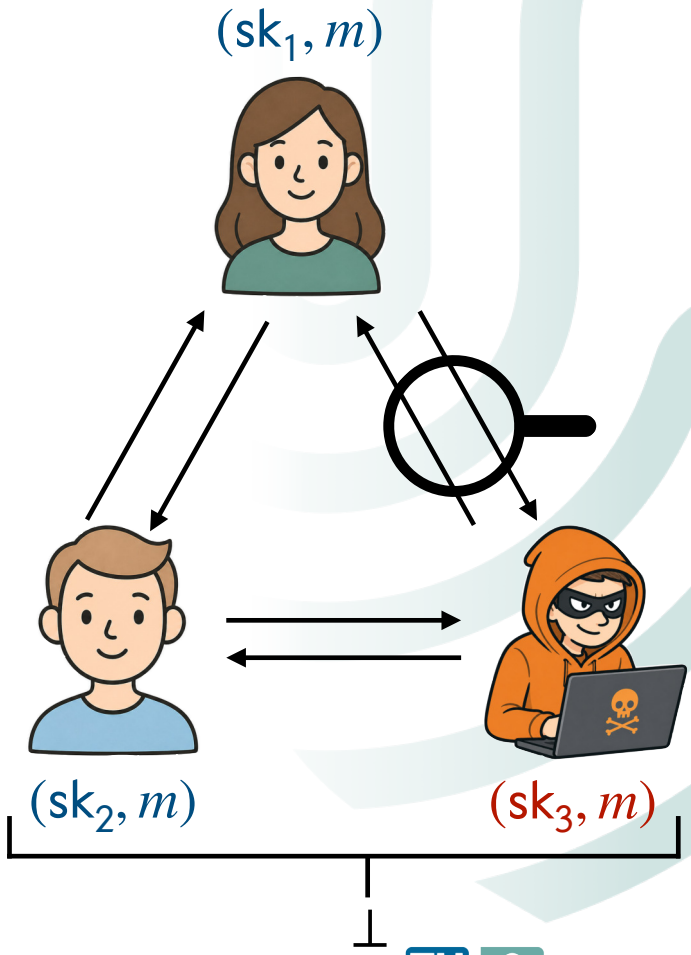
Identifiable Aborts

- Before combining: All protocol messages appear **uniformly random**
  - **No party** can identify if they are well-formed
- **A single malicious** party can stall the protocol by sending malformed messages
  - $s_1' + z' \neq s_1' + s_3'$, so the combined signature does not verify
- Not an issue in FROST: $g^{s_3} = R_3 \cdot \mathsf{vk}^c$ can be checked efficiently



Unforgeability with Identifiable Aborts

- Run the protocol optimistically
- If the combined signature does not verify:
  - run a dedicated **identifiable aborts protocol**
    - At least one misbehaving party will be identified
    - No honest party will be identified



How to Identify Misbehaving Parties?

$\mathsf{S}_1(\mathsf{sk}_1, m)$

1) $(d_1, e_1) \leftarrow^{\$} \mathbb{Z}_p^2$
2) $D_1 \leftarrow g^{d_1};\ E_1 \leftarrow g^{e_1}$

$\xrightarrow{(D_1, E_1)}$
$\xleftarrow{(D_2, E_2)}$

3) $c \leftarrow \mathsf{H}_{\mathsf{sig}}(\mathsf{vk}, R, m)$
4) $s_1 \leftarrow c \cdot \mathsf{sk}_1 + b \cdot e_1 + d_1$
5) $B_1^s \leftarrow \sum \mathsf{H}_{\mathsf{s}}(k_{i,j}, \{\mathsf{pm}_1\}) \cdot \Delta_{i,j}$
6) $s_1' \leftarrow s_1 + B_1^s$

$\xrightarrow{s_1'}$
$\xleftarrow{s_2'}$

$\mathsf{vk}^c \cdot R \neq g^{s_1' + s_2'}$

Idea:

- If the protocol fails, parties prove in zero-knowledge that s_i' it is well-formed with respect to sk_i and $\{k_{i,j}\}$

Problem:

- We do not want to instantiate the RO in a zero-knowledge proof

TU WIEN PRIVACY ENHANCING TECHNOLOGIES

How to Identify Misbehaving Parties?

$\mathsf{S}_1(\mathsf{sk}_1, m)$

1) $(d_1, e_1) \leftarrow^{\$} \mathbb{Z}_p^2$
2) $D_1 \leftarrow g^{d_1};\ E_1 \leftarrow g^{e_1}$

$\xrightarrow{(D_1, E_1)}$
$\xleftarrow{(D_2, E_2)}$

3) $c \leftarrow \mathsf{H}_{\mathsf{sig}}(\mathsf{vk}, R, m)$
4) $s_1 \leftarrow c \cdot \mathsf{sk}_1 + b \cdot e_1 + d_1$
5) $B_1^s \leftarrow \sum \mathsf{H}_{\mathsf{s}}(k_{i,j}, \{\mathsf{pm}_1\}) \cdot \Delta_{i,j}$
6) $s_1' \leftarrow s_1 + B_1^s$

$\xrightarrow{s_1'}$
$\xleftarrow{s_2'}$

$\mathsf{vk}^c \cdot R \neq g^{s_1' + s_2'}$

Idea:

- Commit to sk_i and $\mathsf{H}_\mathsf{s}(k_{i,j}, \{\mathsf{pm}_1\})$ in Pedersen commitments
  - $\mathsf{C}_\mathsf{i} = g^{\mathsf{sk}_i} \cdot h^{\rho_i}$
  - $\mathsf{C}_{\mathsf{i,j}} = g^{\mathsf{H}_\mathsf{s}(k_{i,j}, \{\mathsf{pm}_1\})} \cdot h^{\rho_{i,j}}$
- Prove well-formedness of s_i' w.r.t. these commitments

Problem:

- This still does not guarantee if $\mathsf{H}_\mathsf{s}(k_{i,j}, \{\mathsf{pm}_1\})$ is well-formed

How to Identify Misbehaving Parties?

$\mathsf{S}_1(\mathsf{sk}_1, m)$

1) $(d_1, e_1) \leftarrow^{\$} \mathbb{Z}_p^2$
2) $D_1 \leftarrow g^{d_1};\ E_1 \leftarrow g^{e_1}$

$\xrightarrow{(D_1, E_1)}$
$\xleftarrow{(D_2, E_2)}$

3) $c \leftarrow \mathsf{H}_{\mathsf{sig}}(\mathsf{vk}, R, m)$
4) $s_1 \leftarrow c \cdot \mathsf{sk}_1 + b \cdot e_1 + d_1$
5) $B_1^s \leftarrow \sum \mathsf{H}_{\mathsf{s}}(k_{i,j}, \{\mathsf{pm}_1\}) \cdot \Delta_{i,j}$
6) $s_1' \leftarrow s_1 + B_1^s$

$\xrightarrow{s_1'}$
$\xleftarrow{s_2'}$

$\mathsf{vk}^c \cdot R \neq g^{s_1' + s_2'}$

Idea:

- **Deterministically** commit to $\mathsf{H}_{\mathsf{s}}(k_{i,j}, \{\mathsf{pm}_1\})$ in Pedersen commitments
  - $\mathsf{C}_{\mathsf{i,j}} = g^{\mathsf{H}_{\mathsf{s}}(k_{i,j},\{\mathsf{pm}_1\})} \cdot h^{\mathsf{H}_{\mathsf{s}}(k_{i,j},\{\mathsf{pm}_1,\mathsf{pm}_2\})}$
- Signer i and j have the same commitments $\mathsf{C}_{\mathsf{i,j}} = \mathsf{C}_{\mathsf{j,i}}$

TU WIEN PRIVACY ENHANCING TECHNOLOGIES

Our Identification Protocol

1. If two signers received **different first-round messages** from the same party j
   $\rightarrow$ Mark j as malicious (requires authenticated channels)
2. If a proof π_j does **not verify** $\rightarrow$ Mark j as malicious
3. If no misbehavior is detected above, there is one pair (i, j) with $\mathsf{C}_{i,j} \neq \mathsf{C}_{j,i}$
   - Both parties i and j reveal $k_{i,j}$ (fine, since key is already known to the adversary)
   - Recompute $\mathsf{C}_{i,j} = g^{\mathsf{H_s}(k_{i,j},\{\mathsf{pm}_1\})} \cdot h^{\mathsf{H_s}(k_{i,j},\{\mathsf{pm}_1,\mathsf{pm}_2\})}$

TU WIEN PRIVACY ENHANCING TECHNOLOGIES

Our Submission

https://github.com/pGerhart/FaFrost

- We target Category N1.1
- Our packet includes the specification, reference implementation, and security analysis
- We include two ciphersuites
  - sepc256k1 (BIP340)
  - Ed25519 (FIPS 186-5 EdDSA)
- Round one is message-independent and can be preprocessed (one online round)
- Our implementation is in Rust and will be available on Github

Prototype Implementation

On both Edwards25519 (EdDSA) and Secp256k1 (BIP340) curves

Table 3. Median runtimes per signer on an Apple M3 Pro. The commit phase is independent of the signer set size. The computational overhead compared to FROST is just the blinding time whose overhead remains below 2.2% of signing time.

(t, n)	Commit	Sign	Blind	Agg	IA1	IA2	Decide
$(4, 32)$	55 µs	158 µs	1.14 µs	162 ns	776 µs	179 ns	3.1 ms
$(8, 32)$	55 µs	231 µs	3.22 µs	369 ns	1.57 ms	393 ns	10.0 ms
$(16, 32)$	55 µs	382 µs	6.70 µs	783 ns	3.18 ms	902 ns	38.0 ms
$(32, 64)$	55 µs	668 µs	13.40 µs	1.62 µs	6.53 ms	1.93 µs	145.8 ms
$(48, 64)$	55 µs	962 µs	20.45 µs	2.45 µs	9.99 ms	3.27 µs	326.6 ms
$(64, 128)$	55 µs	1.25 ms	27.08 µs	3.28 µs	13.61 ms	4.56 µs	588.1 ms

TU WIEN PRIVACY ENHANCING TECHNOLOGIES

Communication Cost

- Round 1: 64 B per signer
- Round 2: 32 B per signer

Same as FROST

- IA 1: $t - 1$ Pedersen commitments + one proof linear in $t \approx t \cdot 96$ B per signer
- IA 2: pair $(j, k_{i,j})$ for each identified misbehavior $\leq t - 1 \cdot 32$ B

Key Storage per Signer

- Signing key: 32 B
- Verification keys: $n \cdot 32$ B

Same as FROST

- Blinding keys: $n^2 \cdot 32$ B

TU WIEN PRIVACY ENHANCING TECHNOLOGIES

Fully-Adaptive Schnorr Threshold Signatures

Scheme	Rounds	Model	Assumption	IA
KRT [KRT24]	5	ROM	DL	✗
Glacius [BDLR25]	5	ROM	DDH	✓
Gargos [BDLR25]	3	ROM	DDH	✓
Frost3R [Chen25]	3	ROM	AOMDL	✗
Rackle [NRT26]	3	ROM	DL	✗
Classic Schnorr*** [AABCDGM26]	3	UC + ROM	Schnorr	✓
Cryos [GLRS26]	2*	ROM	DDH	✓
FROST [CKKTZ25]	2	ROM + AGM	AOMDL + LDVR	✓
msFROST** [BCLTZ25]	2	ROM + AGM	AOMDL	✗
FaFROST [BGLRSY25]	2	ROM + AGM	AOMDL	✓

* requires key updates after d signing rounds

** msFROST and FaFROST are the same signature scheme, which were found independently
msFROST has no protocol for IA, but our protocol can be used for msFROST as well

*** not adaptively secure, but submitted to the call

TU WIEN PRIVACY ENHANCING TECHNOLOGIES

Key Generation

- We built a distributed key generation scheme
  - Fully adaptively secure from CDH
  - Two versions
    - one round (biasable keys)
    - two rounds (unbiasable keys)

ia.cr/2026/892

The AOMDL Assumption and AGM

AOMDL = Algebraic One-More Dlog Assumption [NicRufSeu21]

AGM = Algebraic Group Model [FucKilLos18]

Chal()

1) $x \leftarrow^{\$} \mathbb{Z}_p$
2) $X \leftarrow^{\$} g^x$
3) $C \leftarrow C + 1$
4) return X

DLog(X)

1) find x s.t. $g^x = X$
2) $Q \leftarrow Q + 1$
3) return x

$\text{AOMDL}^{\mathscr{A}}_{\mathbb{G},pg}$

1) $Q = C = 0$
2) $(x_1, \ldots, x_C) \leftarrow \mathscr{A}^{\text{Chal,DLog}}()$
3) return $Q < C \wedge g^{x_i} = X_i$ for $i \in [C]$

In the *AGM*, when $\mathscr{A}$ outputs a value $Z \in \mathbb{G}$, it also provides an algebraic representation with respect to previously learned group elements $\{Y_i\}$

$$Z = g^{\alpha} \cdot \prod_i Y_i^{\beta_i}$$

TU WIEN PRIVACY ENHANCING TECHNOLOGIES