



# mask-FROST: Adaptively secure 2-round threshold Schnorr signatures in the AGM

Renas Bacho (CISPA)

Yanbo Chen (University of Ottawa)

Julian Loss (Ruhr University Bochum)

Stefano Tessaro (University of Washington)

**Chenzhi Zhu (NTT Research)**

January 26, 2026

Presented at MPTS 2026 :

NIST Workshop on Multi-Party Threshold Schemes

# Threshold signatures [DF90]

$$(sk_1, \dots, sk_n, pk) \leftarrow \text{DisKeyGen}(1^\lambda)$$

Signer 1:  $sk_1$



Signer  $n$ :  $sk_n$

...



Signer 2:  $sk_2$

Threshold  $t$   
involved

Avoid single point attack

$m$



$\sigma$



Wallets



NIST

$$0/1 \leftarrow \text{Verify}(pk, m, \sigma)$$

**Unforgeability:** can't forge  $\sigma$  for  $m$  if **less than  $t$**  signers involved

# Our focus

- Producing Schnorr signatures
  - Short and efficient
  - Standardized variant: EdDSA
  - Implemented in real world (e.g. Bitcoin)
- Adaptively secure: Adversary can corrupt signers ( $< t$ ) at anytime during signing
  - Reasonable in practice

Both emphasized by the NIST call

# Adaptively secure threshold Schnorr signatures

- Early works [CGJ+99, GJKR07, SS01, AF04]:
  - Focus on robustness, require **corruption** ( $< t/2$ )
- FROST [KG20]: very efficient, 1-offline+1-online
  - Statically security under AOMDL + ROM [BCK+22]
  - Adaptively secure under **LDVR (non-standard)** + AOMDL + ROM + AGM [CKK+25]
- Standard assumptions in ROM with **more rounds**:
  - ZeroS [Mak22], [KRT24], Glacius [BDLR25a], Gargos [BDLR25b], [Chen25], [GCRS25]
- Our work: mask-FROST
  - Little tweak to FROST: 1-offline, 1-online
  - Adaptively secure under AOMDL+ROM+AGM

# Prior works

Assume ROM

	Offline+online rounds	Assumption	Need pair-wise keys	Comments
FROST [KG20]	1+1	LDVR+AOMDL+AGM	No	
ZeroS [Mak22]	1+2	DL	No	Need secure erasures
KRT [KRT24]	2+3	DL	Yes	
Glacius [BDLR25a]	2+3	DDH	No	
Gargos [BDLR25b]	1+2	DDH	No	
[GCRL25]	0+2	DDH	No	Periodic key refresh
[Chen25]	2+1	AOMDL	Yes	
<b>This work</b>	1+1	AOMDL+AGM	Yes	

# Overview

- Introduce mask-FROST
- Security of mask-FROST and proof ideas

# FROST [KG20]

$$(\mathbb{G}, p, g): \mathbb{G} = \langle g \rangle, |\mathbb{G}| = p$$

$$\text{DKG: } sk \stackrel{\$}{\leftarrow} \mathbb{Z}_p, pk \leftarrow g^{sk} \quad (sk_1, \dots, sk_n) \stackrel{\$}{\leftarrow} \text{SecSha}(t, n, sk)$$

Signer  $i: sk_i$

Coordinator

$$r_i \stackrel{\$}{\leftarrow} \mathbb{Z}_p \quad s_i \stackrel{\$}{\leftarrow} \mathbb{Z}_p$$

$$R_i \leftarrow g^{r_i} \quad S_i \leftarrow g^{s_i}$$



$$d \leftarrow H'(pk, PS, m)$$

$$m, PS \leftarrow \{(j, R_j, S_j)\}_{j \in SS}$$

SS: Signer set

$$R \leftarrow \prod_{j \in SS} R_j S_j^d$$

$$z \leftarrow \sum_{i \in SS} z_i$$

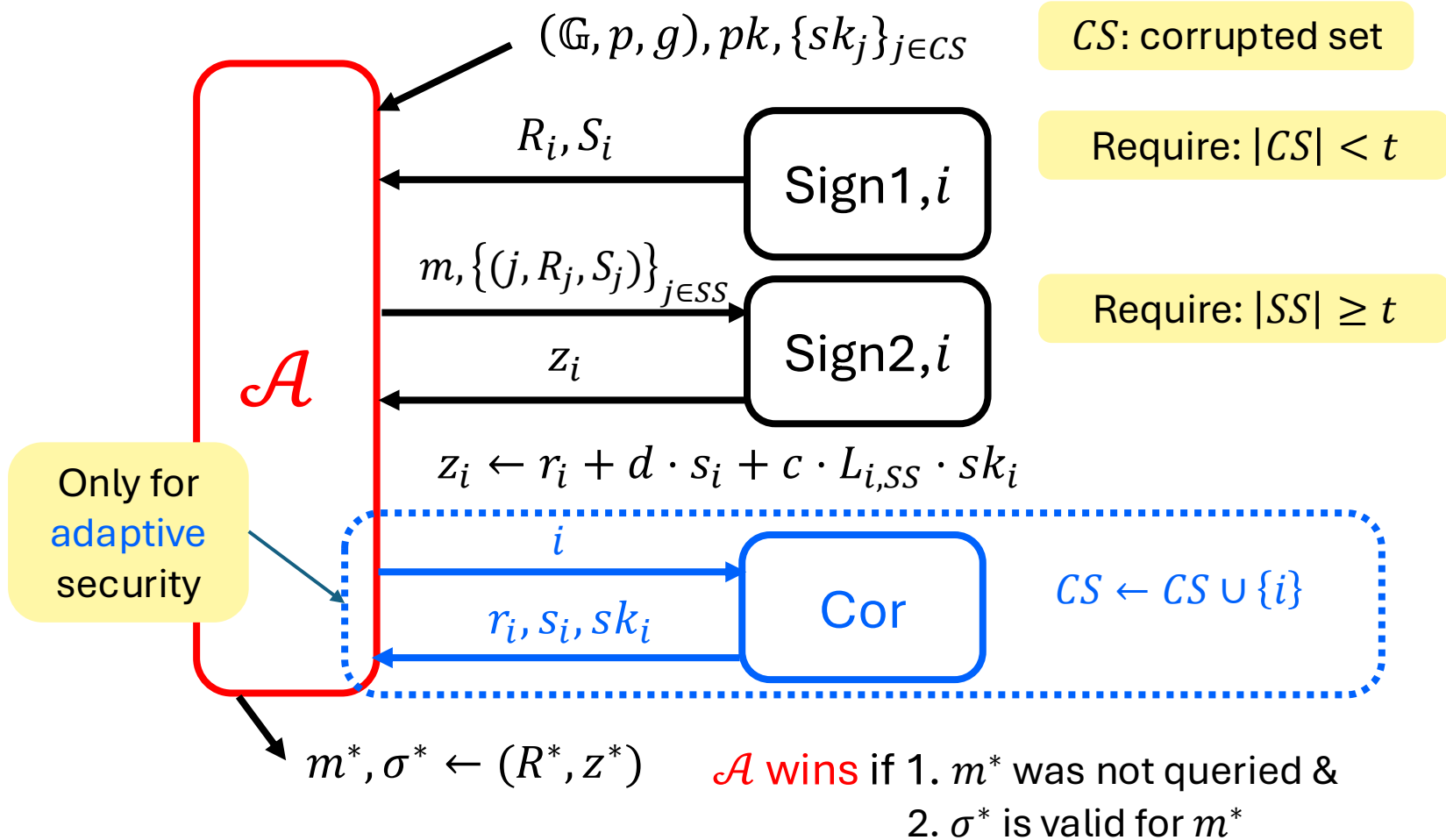
$$c \leftarrow H(pk, R, m)$$

$$\sigma \leftarrow (R, z)$$



$$z_i \leftarrow r_i + d \cdot s_i + c \cdot L_{i,SS} \cdot sk_i$$

# Security game of FROST



# Security of FROST

- Statically secure under AOMDL + ROM [BCK+22]
- **Inherent problem** in adaptive security [CS25, CKK+25]:
  - If  $(pk_1, \dots, pk_n) = (g^{sk_1}, \dots, g^{sk_n})$  are leaked and the problem P (or LDVR) is easy, then one can break adaptive security of FROST

In FROST,  $(pk_1, \dots, pk_n)$  are leaked by signing

$$z_i = r_i + d \cdot s_i + c \cdot L_{i,SS} \cdot sk_i$$



$$g^{z_i} = R_i S_i^d pk_i^{c \cdot L_{i,SS}} \longrightarrow pk_i = (g^{z_i} R_i^{-1} S_i^{-d})^{1/(c \cdot L_{i,SS})}$$

In FROST,  $(pk_1, \dots, pk_n)$  is leaked by signing

$$z_i = r_i + d \cdot s_i + c \cdot L_{i,SS} \cdot sk_i$$



$$g^{z_i} = R_i S_i^d pk_i^{c \cdot L_{i,SS}} \longrightarrow pk_i = (g^{z_i} R_i^{-1} S_i^{-d})^{1/(c \cdot L_{i,SS})}$$

Idea: using masks to hide  $z_i$

Each signer sends  $\tilde{z}_i \leftarrow z_i + mask_i$

$\{mask_i\}_{i \in SS}$  is additive shares of 0

[KRT24, Chen25] also use masks but more complicated

# mask-FROST

Signer  $i$ :  $sk_i$

Coordinator

$$r_i \stackrel{\$}{\leftarrow} \mathbb{Z}_p \quad s_i \stackrel{\$}{\leftarrow} \mathbb{Z}_p$$

$$R_i \leftarrow g^{r_i} \quad S_i \leftarrow g^{s_i}$$

$$d \leftarrow H'(pk, PS, m)$$

$$m, PS \leftarrow \{(j, R_j, S_j)\}_{j \in SS}$$

$$R \leftarrow \prod_{j \in SS} R_j S_j^d$$

$$z \leftarrow \sum_{i \in SS} \tilde{z}_i$$

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

$$\sigma \leftarrow (R, z)$$

$$mask_i \leftarrow \sum_{j \in SS} H_M(Seed_{i,j}, cnt) - H_M(Seed_{j,i}, cnt)$$

$$\tilde{z}_i \leftarrow r_i + d \cdot s_i + c \cdot L_{i,SS} \cdot sk_i + mask_i$$

$(Seed_{i,j}, Seed_{j,i})$ :  
 $\lambda$ -bits secret keys  
shared only between  $i$   
and  $j$  generated during  
DKG

$$cnt = (pk, PS, m)$$

$H_M$  modeled as ROM

# Our results

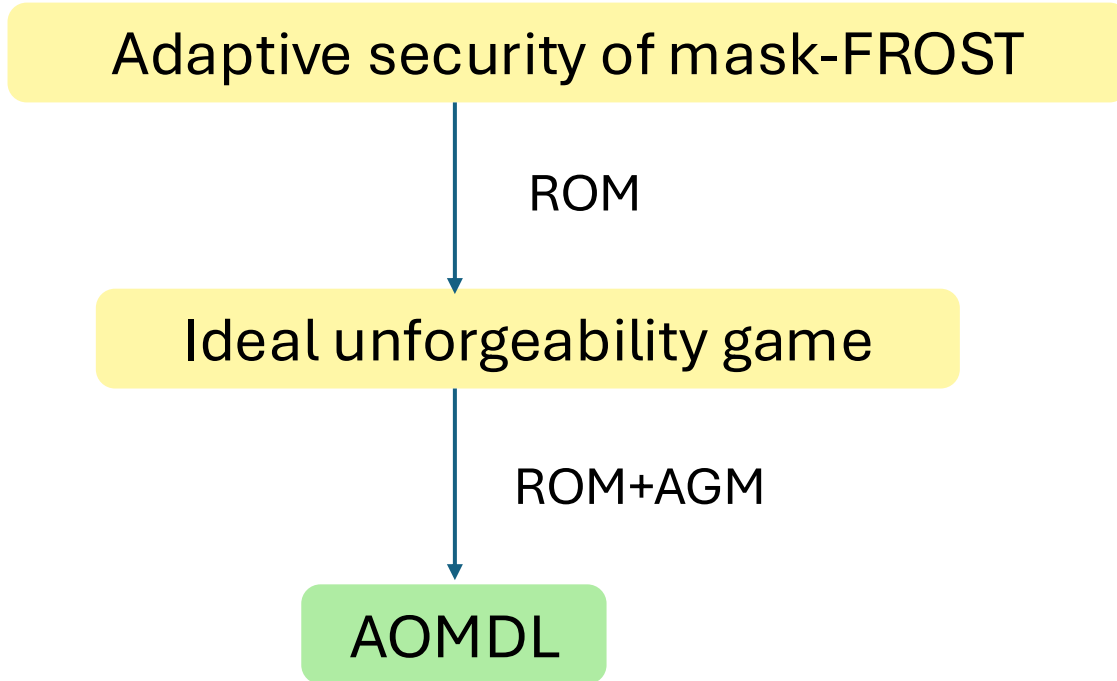
**Theorem 1.** (informal)  $\forall$  *algebraic*  $\mathcal{A}$  for adp-TSUF game of mask-FROST,  $\exists \mathcal{B}$  for AOMDL that

$$\text{Adv}_{\text{mask-FROST}}^{\text{adp-tsuf}}(\mathcal{A}) \leq \text{Adv}_{\text{AOMDL}}(\mathcal{B}) + \frac{q_h(2n^2 + t + q_h)}{2^\lambda}$$

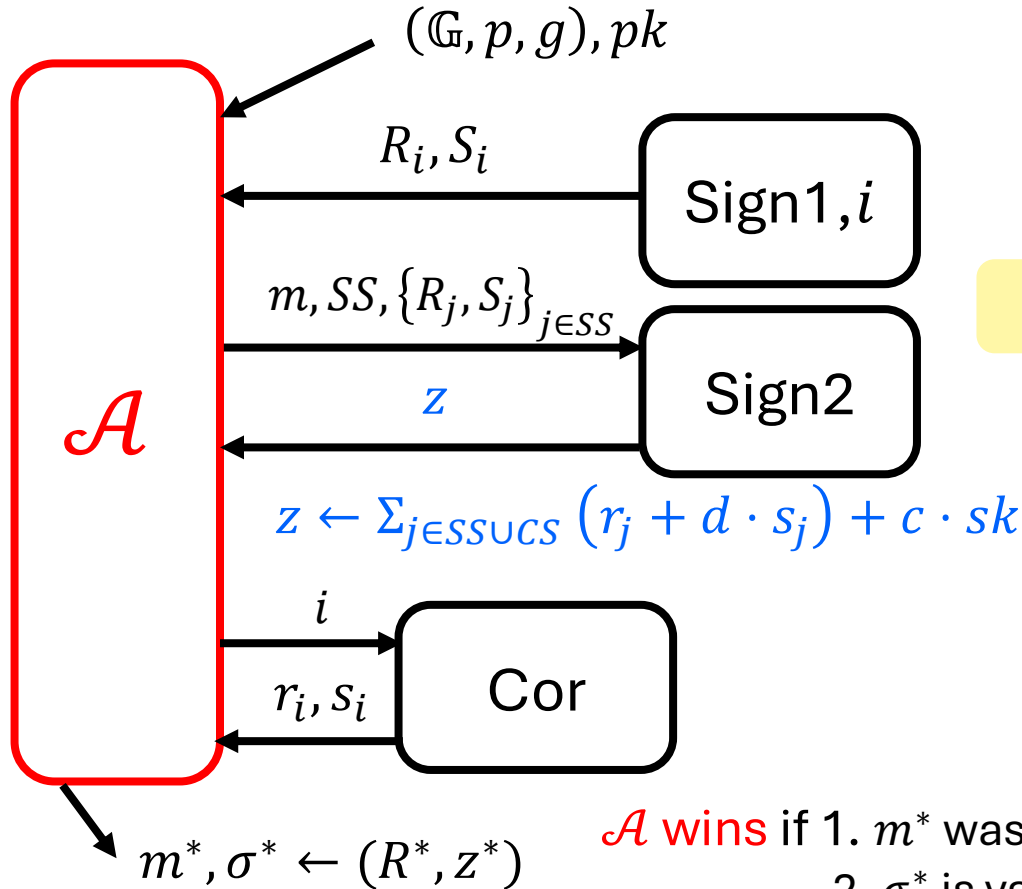
$q_h$ : # of RO queries

**Theorem 2.** (informal) There is no efficient algebraic reduction can reduce adp-TSUF of mask-FROST to AOMDL in ROM given AOMDL is hard

# Security proof overview



# Ideal unforgeability game (IUF)



Require:  $|SS| \geq t$

- Benefits of IUF:
- Simple: no key shares or masks involved
  - Show effect of masks

**A wins** if 1.  $m^*$  was not queried & 2.  $\sigma^*$  is valid

# Consider following $\mathcal{A}$ for IUF



2)  $\mathcal{A}$  tries  $d_1 \leftarrow H'(pk, \{(1, R_1, S_1), (2, \tilde{R}_2, \tilde{S}_2)\}, m)$

$$c_1 \leftarrow H(pk, R_1 S_1^{d_1} \tilde{R}_2 \tilde{S}_2^{d_1}, m)$$

$\tilde{R}_1, \tilde{S}_1, \tilde{R}_2, \tilde{S}_2, m$  picked arbitrarily

$\mathcal{A}$  tries  $d_2 \leftarrow H'(pk, \{(1, \tilde{R}_1, \tilde{S}_1), (2, R_2, S_2)\}, m)$

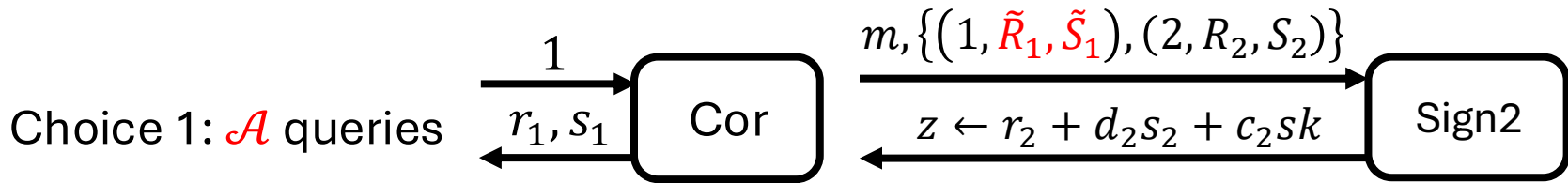
$$c_2 \leftarrow H(pk, \tilde{R}_1 \tilde{S}_1^{d_2} R_2 S_2^{d_2}, m)$$

3)  $\mathcal{A}$  computes  $R^* \leftarrow R_1 S_1^{d_1} R_2 S_2^{d_2}$

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

$$m^* \neq m$$

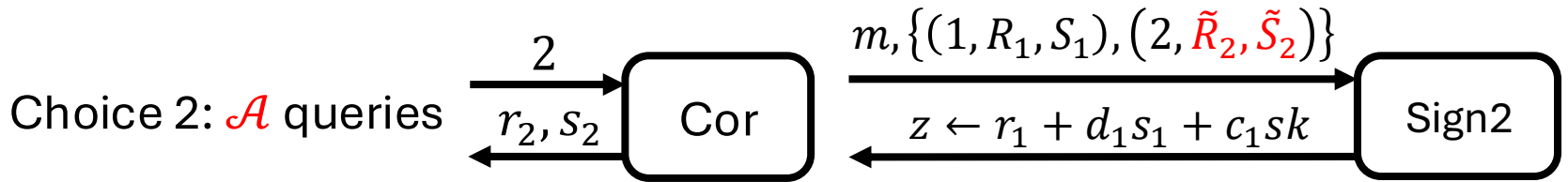
Two choices:  $\mathcal{A}$  either corrupts 1 and queries  $m, \{(1, \tilde{R}_1, \tilde{S}_1), (2, R_2, S_2)\}$   
or corrupts 2 and queries  $m, \{(1, R_1, S_1), (2, \tilde{R}_2, \tilde{S}_2)\}$



$$R^* \leftarrow R_1 S_1^{d_1} R_2 S_2^{d_2}, c^* \leftarrow H(pk, R^*, m^*)$$

If  $c_2 = c^*$ ,  $\mathcal{A}$  computes  $z^* \leftarrow r_1 + s_1 d_1 + z$

$(R^*, z^*)$  is a valid sig for  $m^*$ :  $g^{z^*} = R_1 S_1^{d_1} g^z = \boxed{R_1 S_1^{d_1} R_2 S_2^{d_2}} pk^{c_2}$



If  $c_1 = c^*$ ,  $\mathcal{A}$  computes  $z^* \leftarrow r_2 + s_2 d_2 + z \longrightarrow (R^*, z^*)$  is valid for  $m^*$

Two possible  $c^*$  s.t.  $\mathcal{A}$  wins the IUF game

In general, it can be more  $c^*$ , but we can show the total is bounded by  $t$

Otherwise,  $\mathcal{A}$  may act arbitrarily

# To bound advantage of $\mathcal{A}$

**Fact 1.**  $\Pr[c^* \in \{c_1, c_2\}] \leq 2/p$

**Fact 2.**  $\exists$  AOMDL  $\mathcal{B}$  for AOMDL s.t.  $\mathcal{B}$  wins given  $\mathcal{A}$  wins and  $c^* \notin \{c_1, c_2\}$

**Fact 2.**  $\exists$  AOMDL  $\mathcal{B}$  for AOMDL s.t.  $\mathcal{B}$  wins given  $\mathcal{A}$  wins and  $c^* \notin \{c_1, c_2\}$

Suppose  $\mathcal{A}$  makes Choice 1 and  $c^* \neq c_2$

$(R^*, z^*)$  is a valid sig for  $m^*$ :  $g^{z^*} = R^*pk^{c^*}$

$$z \leftarrow r_2 + d_2s_2 + c_2sk$$

$$R^* = R_1S_1^{d_1}R_2S_2^{d_2} = g^{r_1+d_1s_1}g^zpk^{-c_2}$$

$$g^{z^*} = g^{r_1+d_1s_1+z}pk^{c^*-c_2}$$

$$sk = (z^* - r_1 - d_1s_1 - z)/(c^* - c_2)$$

# Open problems

- Without AGM?
- Good DKG protocol?
- Achieving identifiable abort

Thank you!

Our paper: [ia.cr/2025/1953](https://ia.cr/2025/1953)