"Preview Writeup": In anticipation of a package submission to the NIST Threshold Call

Title: FaFROST

Subtitle: A FROST-based Threshold Schnorr Signature Scheme with Identifiable Aborts and Fully Adaptive Security from AOMDL

Version: Preview Writeup 1.1 (2026-07-24)[1]

Team name: FaFROST

Team members: Ruben Baecker, Paul Gerhart, Davide Li Calsi, Luigi Russo, Dominique Schröder, Arkady Yerukhimovich

Abstract: FaFROST is a two-round threshold Schnorr signature scheme achieving fully adaptive unforgeability under the Algebraic One-More Discrete Logarithm (AOMDL) assumption in the Algebraic Group and Random Oracle Models. Full adaptive security means the adversary may corrupt up to $t-1$ signers at any point during the protocol execution, including after observing signing transcripts. FaFROST differs from FROST, which achieves this security level only under the additional Low-Dimensional Vector Representation (LDVR) assumption, in a single design choice: the signers blind their second-round messages with pairwise masks derived from keys shared during key generation, and the masks cancel during aggregation. FaFROST is thus compatible with standard Schnorr verification and preserves the communication structure of FROST, with a message-independent first round that supports preprocessing. An identifiable abort extension enables the protocol to identify misbehaving signers upon failure.

Proposed crypto-systems:

- (I) FaFROST (Categories N1.1)

Keywords: Threshold Cryptography; Threshold Signatures; Schnorr Signatures; Fully Adaptive Unforgeability; Identifiable Abort

Document License: CC BY 4.0 International

[1] Adapted version submitted after the "Preview Talk" (at MPTS 2026).

Preview writeup. This document is provided to NIST for online publication, to foster public awareness and support public discussion within the scope of the NIST First Call for Multi-Party Threshold Schemes [NIST-IR8214C]. This "preview writeup" represents a good-faith plan for a subsequent "package submission". However, until the deadline for package submission, the team may still modify its own composition and the submission plan, including possible changes to the technical scope, and/or the used techniques or achieved results.

Team members: Ruben Baecker [i1,a2*], Paul Gerhart [i2,a1*], Davide Li Calsi [i3,a1*], Luigi Russo [i4,a1], Dominique Schröder [i5,a1, a2], Arkady Yerukhimovich [i6,a3‡]

Open Researcher and Contributor Identifiers (ORCID):

i1 (0009-0008-9310-8964); i2 (0000-0002-0164-0187); i3 (0000-0002-4745-5374); i4 (0000-0001-9869-786X); i5 (0000-0001-6943-8914); i6 (0000-0002-9045-748X)

Affiliations:

[a1] TU Wien @ Vienna, Austria

[a2] Friedrich-Alexander-Universität Erlangen–Nürnberg @ Erlangen, Germany

[a3] George Washington University @ Washington D.C., USA

Main contacts:

- **Team mailing list:** fafrost@pet.tuwien.ac.at
- **Primary technical contact person:** Paul Gerhart, paul.gerhart@tuwien.ac.at
- **Secondary contact person 1:** Dominique Schröder, dominique.schroeder@tuwien.ac.at
- **Secondary contact person 2:** Ruben Baecker, ruben.baecker@fau.de

Produced by humans. The team hereby confirms that the content in this preview writeup: (i) was produced by the team members, and (ii) was not produced by generative artificial intelligence (AI), with the possible exception of AI-proposed grammar improvements, minor integrated suggestions, or some well-identified and short localized portions of auxiliary content (e.g., some illustration); and (iii) was proofread by the team members.

1. Introduction

FaFROST [BGCRSY25] is a two-round threshold Schnorr signature scheme intended for submission to Category N1.1 of the NIST First Call for Multi-Party Threshold Schemes [NIST-IR8214C]. It produces signatures that verify under the standard Schnorr verification algorithm, making it a drop-in threshold layer for any Schnorr-based verification infrastructure. In particular, when instantiated over the edwards25519 curve with SHA-512, the produced signatures are interchangeable with single-signer signatures under the EdDSA verification algorithm standardized by NIST in FIPS 186-5 [Nat23]. Signing requires two communication rounds; the first is message-independent and can be preprocessed, reducing the online signing phase to a single round in practice.

The primary design goal of FaFROST is *fully adaptive unforgeability*: an adversary may corrupt up to $t-1$ signers at any point during the protocol execution, including after observing signing transcripts, and still cannot produce a forgery. FROST [CKKTZ25] achieves this only under the additional Low-Dimensional Vector Representation (LDVR) assumption, a newly introduced hardness assumption that has not yet been subject to broad cryptanalytic scrutiny. FaFROST eliminates this reliance by introducing *blinded* second-round messages, derived from pairwise shared keys established during key generation. This modification leaves the communication structure and computational cost essentially unchanged compared to FROST: the blinding overhead accounts for less than 5% of signing time across all evaluated configurations.

The blinding, however, conceals which signer deviated from the protocol, making it harder to attribute failures. FaFROST therefore incorporates an *identifiable abort* (IA) extension: when a signing session fails, a post-failure diagnostic procedure identifies misbehaving signers, without falsely accusing honest parties, enabling accountability in deployments where repeated, unattributed failures would otherwise stall the system.

2. Specification

We assume the following when describing FaFROST.

- **Setup Parameters.** Fix integers $0 \leq t_c < t \leq n$, where n is the number of signers holding a key share, t is the signing threshold (the number of signers that must collaborate to produce a signature), and t_c is the *corruption threshold*, the maximum number of signers the adversary may corrupt. Not all n signers need to participate in a signing session. FaFROST achieves *full adaptive security* when $t_c = t-1$, tolerating the maximum possible number of corruptions.
- **Idealized Key Generation.** We model key generation as an idealized functionality that produces a public key pk and n secret key shares $\{\mathsf{sk}_i\}_{i\in[n]}$ via t-out-of-n Shamir secret sharing. Each signer i additionally holds pairwise shared keys $\{k_{i,j}\}_{j\neq i}$, which are used to compute blinding factors during signing. A compatible distributed key generation (DKG) protocol that handles all required key material under adaptive corruptions is given in [BGJNRS26].

- **Communication and Coordinator Role.** Following [CKKTZ25], we assume a PKI and authenticated channels between all participants. Message passing is mediated by a coordinator, a role one of the signers may take. The coordinator is trusted only for liveness: since messages are authenticated, a corrupted coordinator cannot alter them undetected, so the worst it can do is drop messages and abort a session. This is not an extra trust assumption: our analysis gives the adversary full control over message scheduling and delivery, so a corrupted coordinator is subsumed by the adversary and can affect only liveness, never unforgeability.
- **Signing.** Only a subset $S \subseteq [n]$ with $|S| \geq t$ participates in each signing session. The signing set S is part of every signing request, and the combined nonce binds S together with the first-round messages of the session. Several subsets may therefore run signing sessions concurrently, even for the same message: the sessions are separated cryptographically, unforgeability is proven in a model in which the adversary schedules arbitrarily many concurrent sessions, and no coordination between subsets needs to be enforced. FaFROST requires $k = 2$ rounds, which is round-optimal amongst existing Schnorr threshold signatures. The first round is message-independent and may be preprocessed. If no party misbehaves, the output σ verifies against pk with probability 1.

Security. We target adversaries that can adaptively corrupt up to t_c signers over time. Upon corrupting a signer, the adversary learns its full secret state and controls its future behavior. The adversary may also trigger new signing sessions, choosing the message m and signing set S freely. A scheme is *fully adaptive* when it remains unforgeable under $t_c = t - 1$ such corruptions.

FaFROST achieves fully adaptive unforgeability in the Algebraic Group Model (AGM) [FKL18] and the Random Oracle Model (ROM) [BR93] under the AOMDL assumption [NRS21]. The AGM captures adversaries that can compute group elements only via explicit algebraic operations and is widely used throughout the threshold signature literature [CKKTZ25; CKM21; BTZ22]. Concretely, we bound the advantage of an adversary against the *adaptive threshold signature unforgeability* of FaFROST, denoted adpTSUF: any efficient adversary $\mathcal{A}$ breaking the adpTSUF of FaFROST with advantage $\mathsf{Adv}^{\mathsf{adpTSUF}}_{\mathcal{A},\text{FA-FROST}}(t_c, t, p)$, where p is the prime order of the underlying group, implies an efficient AOMDL adversary with advantage at least

$$\mathsf{Adv}^{\mathsf{adpTSUF}}_{\mathcal{A},\text{FA-FROST}}(t_c, t, p) - \frac{4 \cdot q_{\text{sig}}^2}{p} - \frac{q_{\text{non}} \cdot q_{\text{sig}}}{p} - \frac{q_{\mathsf{H}_s} \cdot p^2}{2^\lambda} - \frac{q_{\text{sig}} \cdot t}{p(t - t_c)},$$

where $q_{\text{sig}}, q_{\text{non}}, q_{\mathsf{H}_s}$ are the number of queries $\mathcal{A}$ makes to $\mathsf{H}_{\text{sig}}, \mathsf{H}_{\text{non}}$ and H_s, respectively. All error terms are negligible for groups of standard cryptographic size ($p \approx 2^{256}$).

2.1. Protocol Description

Setup and Key Generation. Given a target security parameter $\lambda \in \mathbb{N}$, select a cyclic elliptic curve group $\mathbb{G}$ of prime order p with generator g. Select independent hash functions $\mathsf{H}_{\text{sig}}, \mathsf{H}_{\text{non}}, \mathsf{H}_s$ mapping arbitrary-length bitstrings to scalars in $\mathbb{Z}_p$.

For any pair $(i, j) \in [n]^2$ of distinct users $i < j$ sample a uniformly random $k_{i,j} \leftarrow\!\$\ \{0,1\}^\lambda$ and set $k_{j,i} = k_{i,j}$. Sample $\mathsf{sk} \leftarrow\!\$\ \mathbb{Z}_p$ and generate n shares $\mathsf{sk}_1, \ldots, \mathsf{sk}_n$ via t-out-of-n Shamir

secret sharing. For each $i \in [n]$ the i-th secret key share, received by signer i, consists of $(\{k_{i,j}\}_{j\in[n]\setminus\{i\}}, \mathsf{sk}_i)$. The global public key is set to $\mathsf{pk} \leftarrow g^{\mathsf{sk}}$.

First Signing Round. Each signer $i \in S$ samples two random and independent nonces $e_i, d_i \leftarrow\!\!\$\ \mathbb{Z}_p$, to be temporarily stored as part of its state, bound to the published values below and used for at most one signing session. It then publishes $E_i \leftarrow g^{e_i}$ and $D_i \leftarrow g^{d_i}$. Since this round is message-independent, it can also be preprocessed, that is, executed by any potential signer $i \in [n]$ before the signing set S and the message are determined.

Second Signing Round. Each signer $i \in S$ aggregates $D \leftarrow \prod_{j\in S} D_j, \quad E \leftarrow \prod_{j\in S} E_j$ and computes $b \leftarrow \mathsf{H}_{\mathrm{non}}(\mathsf{pk}, S, m, \{D_j, E_j\}_{j\in S})$. It computes the nonce $R \leftarrow D \cdot E^b$ and from the latter the challenge $c \leftarrow \mathsf{H}_{\mathrm{sig}}(\mathsf{pk}, R, m)$. The signer computes a blinding factor of the form

$$B_i^s \leftarrow \sum_{j\in S\setminus\{i\}} \mathsf{H}_s(k_{i,j}, \{D_\ell, E_\ell\}_{\ell\in S}, m, S) \cdot \Delta_{i,j}$$

where $\Delta_{i,j} = -1$ if $i < j$ and 1 otherwise. Finally, it outputs a partial signature share

$$s_i \leftarrow c \cdot \Lambda_i \cdot \mathsf{sk}_i + d_i + b \cdot e_i + B_i^s,$$

where Λ_i denotes the Lagrange coefficient of i within set S.

Aggregation. The individual signatures are aggregated through simple addition, i.e. $s \leftarrow \sum_{i\in S} s_i$. The final signature is then $\sigma = (R, s)$.

Verification. FaFROST is compatible with the standard Schnorr verification procedure. That is, a party accepts if

$$\mathsf{pk}^{\mathsf{H}_{\mathrm{sig}}(\mathsf{pk},R,m)} \cdot R = g^s.$$

2.1.1. Identifiable Abort Extension

Since our protocol blinds the second-round messages, a misbehaving signer might cause the protocol to abort without producing a signature, leaving the source of the failure unknown. The *identifiable abort* (IA) extension addresses this: upon failure, a diagnostic procedure consisting of two additional communication rounds identifies misbehaving signers, without falsely accusing honest parties, which is essential in large signing groups.

We endow FaFROST with two IA rounds (IA1 and IA2) and a decision algorithm, and extend the key generation phase with additional commitments. The signing rounds themselves remain unchanged: whether or not the IA extension is enabled, the signers execute the identical signing protocol, and only key generation carries the additional commitments. The IA rounds are executed by the signers of the failed session, and only in the abort case, namely when the aggregated signature fails to verify. Honest executions therefore incur no additional cost, and the critical signing path is unaffected.

Setup. Select additional hash functions $\mathsf{H}_{\mathsf{IA}}, \mathsf{H}_{\mathsf{pk}} : \{0,1\}^* \to \mathbb{Z}_p$, modeled as random oracles. Set up a commitment scheme CS with a committing algorithm Com and an opening algorithm Open. Set up a Non-Interactive Zero-Knowledge (NIZK) proof system Π for the relation $\mathcal{R}$ below, with which each signer proves that its partial signature s_i is well formed, that is, that s_i was computed from the secret key share committed in its partial verification key and from the blinding values committed in its broadcast commitments, relative to its public first-round messages:

$$\mathcal{R} = \left\{ (\mathbb{x}, \mathbb{w}) \mid g^{s_i} D_i^{-1} E_i^{-b} = g^{\mathsf{sk}_i \Lambda_i c} \prod_{j \neq i} g^{\Delta_{i,j} B^s_{i,j}} \wedge \mathsf{pk}_i = \mathsf{C}(\mathsf{sk}_i, \omega_i) \wedge \forall j \neq i : \mathsf{C}_{i,j} = \mathsf{C}(B^s_{i,j}, \omega_{i,j}) \right\},$$

where

$$\mathbb{x} = (i, \mathsf{pm}_i, s_i, b, \mathsf{pk}_i, \{\mathsf{C}_{i,j}\}_{j \neq i \in S}), \quad \text{and} \quad \mathbb{w} = (\mathsf{sk}_i, \{(B^s_{i,j}, \omega_{i,j})\}_{j \neq i \in S}).$$

The three conjuncts state that s_i is computed correctly relative to the public first-round messages (D_i, E_i) and challenges b, c (the first equation, after dividing out D_i and E_i^b), combining the blinding values with the same signs $\Delta_{i,j}$ used in the signing round, that sk_i is the share committed in the partial verification key pk_i, and that each $B^s_{i,j}$ is the value committed in the broadcast commitment $\mathsf{C}_{i,j}$. Crucially, the proof binds s_i only to the *committed* blinding values; it does not certify they were honestly derived as $B^s_{i,j} = \mathsf{H}_s(k_{i,j}, \{D_\ell, E_\ell\}_{\ell \in S}, m, S)$. Closing this gap is the task of the decision algorithm: honest partners produce identical commitments $\mathsf{C}_{i,j} = \mathsf{C}_{j,i}$, so any mismatch is resolved by having the pair open their shared key $k_{i,j}$. The relation can be efficiently instantiated using Schnorr proofs.

Key Generation. In addition to the aforementioned public and private key components, the IA extension of FaFROST requires two further public commitments per signer: a commitment $K_{i,j} \leftarrow \mathsf{H}_{\mathsf{pk}}(k_{i,j})$ to each pairwise shared key $k_{i,j}$, and a Pedersen commitment $\mathsf{Com}(\mathsf{sk}_i, \omega_i)$ to each secret key share sk_i.

First IA Round. Each signer $i \in S$ recomputes $b \leftarrow \mathsf{H}_{\mathrm{non}}(\mathsf{pk}, S, m, \{D_j, E_j\}_{j \in S})$ as well as blinding factors

$$B^s_{i,j} = \mathsf{H}_s(k_{i,j}, \{D_\ell, E_\ell\}_{\ell \in S}, m, S)$$

for $j \in S \setminus \{i\}$. Next, it derives a commitment randomness $\omega_{i,j} = \mathsf{H}_{\mathsf{IA}}(k_{i,j}, \{(\mathsf{pm}_j, s_j)\}_{j \in S})$ for each $j \in S \setminus \{i\}$ which is used to generate commitments $\mathsf{C}_{i,j} = \mathsf{Com}(B^s_{i,j}, \omega_{i,j})$. Finally, it issues a proof $\pi_i \leftarrow \Pi.\mathcal{P}(\mathbb{x}, \mathbb{w})$, with $\mathbb{x}$ and $\mathbb{w}$ as defined above, and outputs $\{\mathsf{C}_{i,j}\}_{j \in S \setminus \{i\}}$ and π_i.

Second IA Round. Each signer $i \in S$ inspects the incoming commitments. For each $j \in S \setminus \{i\}$ it checks that $\mathsf{C}_{i,j} = \mathsf{C}'_{j,i}$, and in case of failure it adds $(j, k_{i,j})$ to a list d_i, initialized to $\emptyset$. Finally, it outputs d_i. The latter represents a list of misbehaving candidates identified by signer i.

Decision. Initialize an empty list of suspicious users $\mathcal{C} = \emptyset$. For each $j \in S$:

1. Verify the proof π_j generated at the end of the first round. In case of failure, add j to $\mathcal{C}$.

2. For each $(\ell, k_{j,\ell}) \in d_j$ check if $K_{j,\ell} = \mathsf{H}_{\mathsf{pk}}(k_{j,\ell})$. If not, add j to $\mathcal{C}$.
3. Next, recompute $\omega_{j,\ell}$ and $B^s_{j,\ell}$, and verify that $\mathsf{C}_{\ell,j} = \mathsf{Com}(B^s_{j,\ell}, \omega_{j,\ell})$. If not, add ℓ to $\mathcal{C}$.

Finally, loop through all the pairs (j, ℓ) such that $j \neq \ell$ and $\mathsf{C}_{\ell,j} \neq \mathsf{C}_{j,\ell}$. If $j \notin d_\ell$, add ℓ to $\mathcal{C}$; otherwise, if $\ell \notin d_j$, add j to $\mathcal{C}$.

Guarantees. We prove two properties of $\mathcal{C}$: no honest signer is ever included, and whenever a session's aggregated signature fails to verify, $\mathcal{C}$ contains at least one corrupted signer. Every signer failing a check is included, so several colluding deviators yield up to t_c entries. The decision is a deterministic computation over the IA messages and the public key-generation commitments, using no secret material; its verdict is therefore publicly recomputable from the authenticated session transcript.

3. Open-Source Implementation

We implemented a prototype of FaFROST [Ger26] in Rust, covering the full protocol stack: dealer-based key generation, two-round signing with pairwise blinding, aggregation, verification, and identifiable aborts. The default instantiation is over edwards25519 with SHA-512 (via `curve25519-dalek`); since signing stays in the prime-order subgroup, an aggregate signature is a valid RFC 8032 signature under any compliant EdDSA verifier, confirming interchangeability with the EdDSA algorithm of FIPS 186-5 [Nat23]. A secp256k1 ciphersuite (via `k256`) targets the Bitcoin/BIP-340 Schnorr variant. Both share the same protocol code, differing only in curve, challenge hash, and wire encoding. Concurrent signing sessions cannot be confused, even when they sign the same message m: each stored nonce pair (d_i, e_i) is bound to its published commitments (D_i, E_i), and every second-round request carries the full commitment list $\{(D_j, E_j)\}_{j \in S}$, which identifies the session uniquely; each nonce pair is used once and then deleted.

4. Experimental Performance Evaluation

FaFROST inherits the communication structure of FROST unchanged, requiring the same number of rounds and messages per signer. The additional computation stems solely from the blinding factors, which are derived locally from Round 1 commitments and require $t - 1$ hash-to-scalar evaluations per signer. Overall, this results in a negligible computational overhead: in absolute terms, signing ranges from about $76\,\mu s$ to $383\,\mu s$, while blinding contributes only 0.8–$16\,\mu s$, below 5% of the total signing time across all evaluated configurations. A FROST implementation without blinding would have essentially identical signing times. The IA protocol is a post-failure diagnostic procedure and is *not on the critical signing path*: it is invoked only when a session has already aborted and the goal is to identify the responsible parties. Among the IA phases, IA1 scales linearly in the number of signers, while the *decide* phase dominates and scales quadratically, since it cross-checks every pair of signers against the opened pairwise keys (from a few milliseconds up to $\approx 0.52\,s$ for $t = 64$). IA2 remains negligible throughout (below $9\,\mu s$). We report detailed runtimes in Table 1.

(t, n)	Commit	Sign	Blind	Agg	IA1	IA2	Decide
$(4, 256)$	16 µs	76 µs	0.80 µs	371 ns	595 µs	369 ns	2.7 ms
$(8, 256)$	16 µs	100 µs	2.27 µs	818 ns	1.16 ms	819 ns	9.3 ms
$(16, 256)$	16 µs	143 µs	4.78 µs	1.71 µs	2.42 ms	1.81 µs	34.1 ms
$(32, 256)$	16 µs	228 µs	7.72 µs	3.50 µs	4.81 ms	3.88 µs	130.9 ms
$(48, 256)$	16 µs	306 µs	14.98 µs	5.36 µs	7.16 ms	6.39 µs	295.0 ms
$(64, 256)$	16 µs	383 µs	15.75 µs	7.38 µs	9.77 ms	8.82 µs	523.7 ms

Table 1: Median runtimes per signer on an Apple M3 Pro (edwards25519, SHA-512). The commit phase is independent of the signer set size, and blinding adds below 5% to the signing time. The IA phases are invoked only upon protocol abort and are not on the critical signing path.

5. Licensing, Patent Claims, and Funding

1. Open-source licenses. The core FaFROST code will be released under the permissive **Apache License 2.0** (with an optional **MIT** dual-license following common Rust-ecosystem practice). All external dependencies (`curve25519-dalek`, `k256`, `sha2`) are distributed under OSI-approved licenses (BSD-3-Clause, MIT/Apache 2.0). No proprietary or non-OSI-approved components are required.

2. Patents. To the best of our knowledge, there are no known patents with claims that cover the contents of this submission and that involve any team member as an inventor, applicant, or assignee, or that are held by an entity employing or sponsoring a team member.

3. Funding. The planned submission is supported in part by research grants to several of the team members. Full funding acknowledgments, including specific grant numbers and agencies, will be listed in the package submission.

6. Comparison with Other Submissions

Table 2 compares the Category N1.1 submission previews across the dimensions most relevant to the NIST call.

Comparison with FROST. FaFROST builds upon FROST [KG20] and shares the same round structure, the message-independence of the first round, and the core signing algorithms. The two schemes differ in key generation (FaFROST adds pairwise shared keys) and in the second signing round (FaFROST uses blinding factors derived from these keys). These additions enable a proof of fully adaptive security under AOMDL alone, whereas FROST requires the LDVR assumption [CKKTZ25], introduced specifically for this analysis, to handle more than $t/2$ corruptions. For signer groups of up to $n = 128$, LDVR provides more than 128 bits of security, so within this range the additional assumption rests on firm ground; for larger groups, its hardness is conjectured rather than established.

	Rounds	Model	Assumption (full adapt.)	IA
FROST [KG20; CKKTZ25]	2	AGM+ROM	AOMDL + LDVR*	✓
Classic Schnorr [AABCDGM26]	3	ROM	UF of Schnorr †	✓
Gargos [BDLR25]	3	ROM	DDH	✓
FaFROST (this work)	**2**	**AGM+ROM**	**AOMDL**	✓

Table 2: Comparison of Category N1.1 submission previews. "Rounds" counts signing rounds; in every scheme the first round is message-independent and can be preprocessed, so the online cost is one round fewer. The "Assumption" column lists the assumption under which *full* adaptive security is attained. *FROST requires the additional LDVR assumption beyond $t/2$ corruptions [CKKTZ25]; LDVR offers more than 128 bits of security for $n \leq 128$ and is a hardness conjecture for larger n. †Classic Schnorr is adaptively secure only if $\binom{n}{t}$ is polynomial in the security parameter.

Comparison with Classic Schnorr. The Fireblocks-3MI team plans to submit the folklore three-round protocol for distributed Schnorr signing, dubbed Classic Schnorr [AABCDGM26]. They state it as secure with abort under adaptive corruptions in the Universal Composability (UC) framework and the global ROM, assuming only the unforgeability of single-signer Schnorr signatures, and offer identification of misbehaving parties as an optional augmentation of the signing phase. Its adaptive security holds only if $\binom{n}{t}$ is polynomial in the security parameter, hence, starting from 128-bit security, it drops below 100 bits already for moderate committees such as $(n, t) = (32, 16)$, where $\binom{n}{t} \approx 2^{29}$.

Comparison with Gargos. Gargos [BDLR25] achieves full adaptive security and identifiable aborts from the DDH assumption in the ROM, but requires three signing rounds. FaFROST reduces the round complexity to two while relying on the AOMDL assumption in the AGM instead of DDH in the ROM.

References

[AABCDGM26] Michael Adjedj, Tomer Ashur, Amit Singh Bhati, Geoffroy Couteau, Cyprien Delpech de Saint Guilhem, Michael Gutkin, and Nikos Makriyannis. *Distributed Schnorr Signatures: Classic Schnorr*. Preview writeup, submitted to the NIST First Call for Multi-Party Threshold Schemes. 2026. URL: https://csrc.nist.gov/csrc/media/Projects/threshold-cryptography/documents/TCall-1/Fireblocks-3MI-a-PW01.pdf.

[BDLR25] Renas Bacho, Sourav Das, Julian Loss, and Ling Ren. "Adaptively Secure Three-Round Threshold Schnorr Signatures from DDH". In: *Advances in Cryptology – CRYPTO 2025*. Vol. 15827. Lecture Notes in Computer Science. Springer, 2025, pp. 390–422. DOI: 10.1007/978-3-032-01887-8_13. Also at ia.cr/2025/1009.

[BGCRSY25] Ruben Baecker, Paul Gerhart, Davide Li Calsi, Luigi Russo, Dominique Schröder, and Arkady Yerukhimovich. *Fully Adaptive FROST in the Algebraic Group Model From Falsifiable Assumptions*. Cryptology ePrint Archive, Paper 2025/1950. 2025. URL: https://eprint.iacr.org/2025/1950.

[BGJNRS26] Ruben Baecker, Paul Gerhart, Stanislaw Jarecki, Phillip Nazarian, Daniel Rausch, and Dominique Schröder. *Adaptive Distributed Key Generation for Discrete-Log Cryptosystems*. Cryptology ePrint Archive, Paper 2026/892. 2026. URL: https://eprint.iacr.org/2026/892.

[BR93] Mihir Bellare and Phillip Rogaway. "Random Oracles are Practical: A Paradigm for Designing Efficient Protocols". In: *ACM CCS 93*. ACM Press, 1993, pp. 62–73. DOI: 10.1145/168588.168596.

[BTZ22] Mihir Bellare, Stefano Tessaro, and Chenzhi Zhu. *Stronger Security for Non-Interactive Threshold Signatures: BLS and FROST*. Cryptology ePrint Archive, Paper 2022/833. 2022. URL: https://eprint.iacr.org/2022/833.

[CKKTZ25] Elizabeth Crites, Jonathan Katz, Chelsea Komlo, Stefano Tessaro, and Chenzhi Zhu. "On the adaptive security of FROST". In: *Annual International Cryptology Conference*. Springer. 2025, pp. 480–511. DOI: 10.1007/978-3-032-01887-8_16. Also at ia.cr/2025/1061.

[CKM21] Elizabeth Crites, Chelsea Komlo, and Mary Maller. *How to Prove Schnorr Assuming Schnorr: Security of Multi- and Threshold Signatures*. Cryptology ePrint Archive, Paper 2021/1375. 2021. URL: https://eprint.iacr.org/2021/1375.

[FKL18] Georg Fuchsbauer, Eike Kiltz, and Julian Loss. "The Algebraic Group Model and its Applications". In: *Advances in Cryptology – CRYPTO 2018*. Vol. 10992. Lecture Notes in Computer Science. Springer, 2018, pp. 33–62. DOI: 10.1007/978-3-319-96881-0_2.

[Ger26] Paul Gerhart. *FaFrost*. https://github.com/pGerhart/FaFrost. GitHub repository, accessed 2026-05-13. 2026.

[KG20] Chelsea Komlo and Ian Goldberg. "FROST: Flexible round-optimized Schnorr threshold signatures". In: *International Conference on Selected Areas in Cryptography*. Springer. 2020, pp. 34–65. DOI: 10.1007/978-3-030-81652-0_2. Also at ia.cr/2020/852.

[Nat23] National Institute of Standards and Technology. *Digital Signature Standard (DSS)*. (U.S. Department of Commerce) Federal Information Processing Standards (FIPS) 186-5. 2023. DOI: 10.6028/NIST.FIPS.186-5.

[NRS21] Jonas Nick, Tim Ruffing, and Yannick Seurin. "MuSig2: Simple Two-Round Schnorr Multi-Signatures". In: *Advances in Cryptology – CRYPTO 2021*. Vol. 12825. Lecture Notes in Computer Science. Springer, 2021, pp. 189–221. DOI: 10.1007/978-3-030-84242-0_8.

[NIST-IR8214C] Luís T. A. N. Brandão and René Peralta. *NIST First Call for Multi-Party Threshold Schemes*. (National Institute of Standards and Technology) NIST Internal Report (NISTIR) 8214C. 2026. DOI: 10.6028/NIST.IR.8214C.