

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

Title: FROST: Flexible Round-Optimized Schnorr Threshold Signatures

Subtitle: A Threshold Scheme Interchangeable with EdDSA

Version: 0.1 (2026-01-22)¹

Team name: FROST

Team members: Elizabeth Crites, Conrado Gouvea, Jack Grigg, Ian Goldberg, Jonathan Katz, Chelsea Komlo, Mary Maller, Simon Rastikian, Stefano Tessaro, Nikita Sorokovikov, Denis Varlakov, Chenzhi Zhu

Abstract: This document outlines the Flexible Round-Optimized Schnorr Threshold Signature (FROST) scheme intended for upcoming submission to the NIST Threshold Call. FROST is a threshold signature scheme that thresholdizes the EdDSA signature scheme. FROST signing operations can be performed in two network rounds, or optimized to a single-round variant with preprocessing.

Proposed crypto-systems: FROST: Threshold EdDSA Signing (Category N1.1)

Keywords: Threshold Cryptography; Threshold Signatures; EdDSA; Schnorr Signatures; FROST; NIST Threshold Call

¹Preliminary version submitted to NIST-MPTC for review

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: Elizabeth Crites^{i1,a1}, Conrado Gouvea^{i2,a2}, Jack Grigg^{i3,a3}, Ian Goldberg^{i4,a4}, Jonathan Katz^{i5,a5}, Chelsea Komlo^{i6,a4,a6}, Mary Maller^{i7,a7,a8}, Simon Rastikian^{i8,a6}, Stefano Tessaro^{i9,a9}, Nikita Sorokovikov^{i10,a11}, Denis Varlakov^{i11,a11}, Chenzhi Zhu^{i12,a10}

Open Researcher and Contributor Identifiers (ORCID):

i1 (0000-0001-9992-1771); ; ; i4 (0000-0002-1176-2882); i5 (0000-0001-6084-9303); i6 (0000-0002-2294-2491); ; ; i9 (0000-0002-3751-8546); ; ; i11 (0009-0000-8497-3536); i12 (0000-0002-4276-2797)

Affiliations:

- ^{a1} Web3 Foundation
- ^{a2} Zcash Foundation
- ^{a3} Electric Coin Company
- ^{a4} University of Waterloo
- ^{a5} University of Maryland, Google
- ^{a6} NEAR One
- ^{a7} Ethereum Foundation
- ^{a8} PQShield
- ^{a9} University of Washington
- ^{a10} NTT Research
- ^{a11} Dfns

Main contacts:

- **Team mailing list:** team@frosts signatures.com
- **Primary technical contact person:** Chelsea Komlo, ckomlo@uwaterloo.ca
- **Secondary contact person:** Conrado Gouvea, conrado@zfnd.org

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

In this work, we describe FROST (Flexible Round-Optimized Schnorr Threshold Signatures) [KG20; BCKMTZ22] a two-round threshold signature scheme that thresholdizes the EdDSA signature scheme. The upcoming submission of this threshold scheme fits within Category N1.1 (EdDSA) of the NIST Threshold Call.

FROST addresses the need for efficient threshold EdDSA signing operations, while ensuring strong security properties *without* limiting the parallelism of signing operations. FROST signing operations can be performed in two network rounds, or optimized to a single-round variant with preprocessing.

Here, we describe the two-round version of FROST. However, implementations may perform the first round in a batched setting, allowing the scheme to be used in a manner where online signing requires only a single round of communication.

2. Specification

2.1. System Model

We assume the following when describing FROST:

- **Idealized Key Generation via Shamir Secret Sharing.** We model key generation as an idealized functionality that outputs Shamir secret shares sk_i of a private signing key corresponding to a public key PK, where n total parties receive their respective secret signing key share, and a threshold t parties are required to perform signing. We assume that each signing participant is initialized with their respective secret key share, the public key shares of all other participants, and the joint public key PK representing the group (after key generation has completed).
- **Coordinator Role.** We model message passing between participants via a centralized coordinator. The coordinator is trusted to not perform denial-of-service attacks by dropping messages, but otherwise the coordinator is untrusted.

2.2. Protocol Approach

FROST signing can be performed either in two online rounds, or one online round, after performing the first round using preprocessing. Then, a final stage to perform aggregation is required, at which the joint signature is output. The scheme is defined with respect to the domain-separated hash functions H_{non} and H_{sig} .

Note that while FROST assumes that randomness generated during the first round of signing is used at most once during the second round of signing, it does *not* assume that participants

maintain consistent session identifiers. Moreover, each participant performs the first round of signing independently.

Round One. We next describe the first signing round of FROST as the non-preprocessing variant.

Each party with identifier $k \in \{1, \dots, n\}$, where n is the total number of parties, samples two nonces $(r_k, s_k) \xleftarrow{\$} \mathbb{Z}_p^2$ uniformly at random, and then derives the corresponding commitments $R_k \leftarrow g^{r_k}$, $S_k \leftarrow g^{s_k}$, where g is the generator. They store (r_k, R_k, s_k, S_k) in their internal state, and output (R_k, S_k) .

Round Two. All participants in a signing coalition $C \subseteq [n]$, $|C| \geq t$, accept as input a message m , the coalition C , and a tuple of commitments $\mathcal{S}_{\text{com}} := \{(i, R_i, S_i)\}_{i \in C}$ from parties in the coalition C .

Each party $k \in C$ retrieves (r_k, R_k, s_k, S_k) from their internal state. (The party aborts if such (R_k, S_k) does not match any given as input.) Then, each party derives binding factors $\rho_i \leftarrow H_{\text{non}}(i, \text{PK}, m, \mathcal{S}_{\text{com}})$ for each party $i \in C$. Each party then derives $R \leftarrow \prod_{i \in C} R_i \cdot S_i^{\rho_i}$, and the challenge $c \leftarrow H_{\text{sig}}(R, \text{PK}, m)$. Finally, each party outputs their signature share as $z_k \leftarrow r_k + s_k \cdot \rho_k + c \cdot \text{sk}_k \cdot \lambda_k$, where λ_k is the Lagrange coefficient for the coalition C for party k , and deletes (r_k, s_k) .

Combine. The coordinator derives $z = \sum_{i \in C} z_i$, and the group commitment R as explained above. The output signature $\sigma = (R, z)$ is a standard EdDSA signature, and verifies under the (single-party) EdDSA verification algorithm.

2.3. Security Properties

A threshold signature scheme is considered secure if the scheme is unforgeable, assuming the adversary is able to corrupt fewer than a threshold number of participants. FROST achieves a strong notion of both static and adaptive unforgeability in the dishonest majority setting, under the algebraic one-more discrete logarithm (AOMDL) assumption in the random oracle model (ROM) [BCKMTZ22; CKKTZ25]. In the adaptive setting, the security of FROST additionally requires the low-dimensional vector representation (LDVR) assumption.

Intuitively, unforgeability in a threshold signature setting guarantees that an adversary which controls the coordinator and up to $t-1$ signers cannot generate a valid signature for any message m . Note the coordinator in this model is trusted to relay messages for liveness, but not unforgeability.

As demonstrated by Bellare et al. [BCKMTZ22], there are different conditions to declare m to be valid (signed), which gives different security levels for partially non-interactive schemes such as FROST.

TS-UF-0. The starting condition, which gives the “standard” unforgeability for a threshold signature scheme, we refer to as TS-UF-0. This condition considers that m was signed as long as at least one honest party generated a signature share for m . In other words, for a TS-UF-0-secure

scheme, the adversary cannot forge a valid signature for m if no honest party outputted a signature share z_i for m . TS-UF-0 however does not consider the more general setting case when the adversary corrupts *fewer* than $(t - 1)$ parties, i.e., it assumes that the number of corrupted parties is exactly $f = t - 1$.

TS-UF-1. The next level of security, TS-UF-1, generalizes the above condition to requiring that at least $(t - f)$ honest parties generated signature shares for m , such that the allowable f is generalized such that $f \leq t$. I.e., the adversary corrupts fewer parties than the maximum allowed $t - 1$. More precisely, for a TS-UF-1-secure scheme, a corrupted coordinator can send different combinations of commitments to different combinations of honest parties for signing m in the (online) second round, and as long as the total number of honest parties responded is at least $(t - f)$, the adversary might be able to compute a valid signature for m .

TS-UF-2. Such a malicious behavior is prevented by TS-UF-2, where we consider m to be signed only if at least $(t - f)$ honest parties generated signature shares for m and they received the same commitment combination when generating the shares.

TS-UF-3. Bellare et al. [BCKMTZ22] showed that under the AOMDL assumption, FROST achieves the next level of security, TS-UF-3, in the ROM, where the above condition is further strengthened: we declare m to be signed only if there exists a coalition C and $\mathcal{S}_{\text{com}} = \{(i, R_i, S_i)\}_{i \in C}$ such that not only $(t - f)$ honest parties, but also all honest parties $i \in C$ with *correct* (R_i, S_i) , generated signature shares for the same second-round input $(m, \mathcal{S}_{\text{com}})$, where we say (R_i, S_i) is correct if and only if it was output by party i in Round 1.²

TS-UF-4. The strongest notion of security defined by Bellare et al. [BCKMTZ22] is TS-UF-4, which requires that there is a coalition C and $\mathcal{S}_{\text{com}} = \{(i, R_i, S_i)\}_{i \in C}$ such that all honest parties in C generated signature shares z_i for $(m, \mathcal{S}_{\text{com}})$. TS-UF-4 is a stronger condition because the size of C is at least t and thus the number of honest parties in C is at least $(t - f)$. One means to achieve TS-UF-4 security is by ensuring authenticity of participants' messages, to prevent an adversary from performing integrity attacks that may result in a valid output signature but where participants' views are inconsistent during the protocol execution.

Bellare et al. [BCKMTZ22] showed that if we assume authenticated network channels, which guarantee that a corrupt coordinator cannot forward incorrect commitments to honest parties in the online round, then FROST achieves TS-UF-4. However, because implementations may wish to define authenticated channels in a manner specific to their setup, we do not define this authentication layer within this specification. As such, we allow implementations to choose if they wish to achieve TS-UF-4 security, and simply recommend that FROST be performed over authenticated channels to do so.

Strong Unforgeability. Moreover, Bellare et al. [BCKMTZ22] showed that FROST is *strongly* unforgeable, referred to as TS-SUF-3 (or TS-SUF-4 assuming authenticated channels), which, analogous to the strong unforgeability of signature schemes, guarantees that an adversary cannot

²Since the coordinator was corrupted, the commitment (R_i, S_i) might not be one of the commitments output by honest party i in Round 1.

forge a message-signature pair (m, σ) that is not considered authorized. Also, it is guaranteed that there is at most one signature σ that can be issued for each second-round input $(m, \mathcal{S}_{\text{com}} = \{(i, R_i, S_i)\}_{i \in C})$, and (m, σ) is considered issued only if there are *sufficiently many* honest parties that generated signature shares for $(m, \mathcal{S}_{\text{com}})$. In particular, for TS-SUF-3, “sufficiently many honest parties” includes all honest parties $i \in C$ with *correct* (R_i, S_i) , and the total number of honest parties must be at least $(t - f)$; for TS-SUF-4, “sufficiently many honest parties” refers to all honest parties in C .

Adaptive Security. Crites et al. [CKKTZ25] analyze the adaptive security of FROST, which means that the adversary can adaptively choose which signers to corrupt even after learning the public key and participating in signing interactions. Upon corrupting a signer, the adversary learns all of the signer’s internal state, including its signing key share and any nonces it has generated (we do not assume secure deletion for our proof). They show that FROST achieves adaptive TS-UF-0 under the AOMDL assumption in the ROM when the maximum number of corrupted signers satisfies $f = t/2$. Furthermore, when $f > t/2$, they prove that FROST remains adaptively TS-UF-0-secure in the algebraic group model (AGM) under the additional assumption of the hardness of the low-dimensional vector representation (LDVR) problem, which is a new problem introduced in the paper. They also show that, in certain parameter regimes, the LDVR problem is unconditionally hard.

Disclosure. All technical content was produced by the team. The text integrates some text-formatting improvements suggested by GenAI (and reviewed by the team).

References

- [BCKMTZ22] Mihir Bellare, Elizabeth C. Crites, Chelsea Komlo, Mary Maller, Stefano Tessaro, and Chenzhi Zhu. “Better than Advertised Security for Non-interactive Threshold Signatures”. In: *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part IV*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13510. Lecture Notes in Computer Science. Springer, 2022, pp. 517–550. DOI: [10.1007/978-3-031-15985-5_18](https://doi.org/10.1007/978-3-031-15985-5_18).
- [CKKTZ25] Elizabeth C. Crites, Jonathan Katz, Chelsea Komlo, Stefano Tessaro, and Chenzhi Zhu. “On the Adaptive Security of FROST”. In: *CRYPTO 2025 - 45th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2025, Proceedings, Part VI*. Ed. by Yael Tauman Kalai and Seny F. Kamara. Vol. 16005. Lecture Notes in Computer Science. Springer, 2025, pp. 480–511. DOI: [10.1007/978-3-032-01887-8_16](https://doi.org/10.1007/978-3-032-01887-8_16). Also at ia.cr/2025/1061.
- [KG20] Chelsea Komlo and Ian Goldberg. “FROST: Flexible Round-Optimized Schnorr Threshold Signatures”. In: *SAC 2020, Halifax, NS, Canada (Virtual Event), October 21-23, 2020*. Ed. by Orr Dunkelman, Michael J. Jacobson Jr., and Colin O’Flynn. Vol. 12804. LNCS. Springer, 2020, pp. 34–65. DOI: [10.1007/978-3-030-81652-0_2](https://doi.org/10.1007/978-3-030-81652-0_2). Also at ia.cr/2020/852.
- [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](https://doi.org/10.6028/NIST.IR.8214C).