

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

Title: Teddy

Subtitle: Teddy: Threshold EdDSA Schemes

Version: Preview Writeup 1.0 (2026-10-06)¹

Team name: Octopus

Team members: Man Ho Au, Robert H. Deng, Liming Gao, Bowen Jiang, Xianhui Lu, Tian Qiu, Guofeng Tang, Ruida Wang, Haiyang Xue, Guomin Yang, Chengru Zhang

Abstract: This document previews the Octopus team’s proposals for the NIST Multi-Party Threshold Cryptography Call, specifying a 2-round t -out-of- n threshold EdDSA scheme.

Our scheme uses a subset-deterministic signing paradigm: rather than mapping a message uniquely to a signature in standard EdDSA, the signature is uniquely determined by the tuple of the message and the active signer subset. It enables the general threshold structures without altering the standard EdDSA verification algorithm. Simultaneously, we leverage an efficient exponent verifiable random function (eVRF) construction to realize verifiable, deterministic distributed randomness generation for threshold EdDSA.

Proposed crypto-systems:

- (I) Teddy: t -out-of- n EdDSA (Categories N1)

Keywords: Threshold Cryptography; NIST Threshold Call; t -out-of- n EdDSA



Document License: [CC BY 4.0 International](https://creativecommons.org/licenses/by/4.0/)

¹Version submitted to NIST-MPTC for publication

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: Man Ho Au ^{i1,a1}, Robert H. Deng ^{i2,a3}, Liming Gao ^{i3,a2*}, Bowen Jiang ^{i4,a3*}, Xianhui Lu ^{i5,a2}, Tian Qiu ^{i6,a4}, Guofeng Tang ^{i7,a3}, Ruida Wang ^{i8,a2}, Haiyang Xue ^{i9,a3}, Guomin Yang ^{i10,a3}, Chengru Zhang ^{i11,a5}

Open Researcher and Contributor Identifiers (ORCID):

i1 (0000-0003-2068-9530); i2 (0000-0003-3491-8146); i3 (0009-0004-0211-4148); i4 (0009-0002-8867-866X); i5 (0000-0001-7091-5810); i6 (0009-0006-2429-8771); i7 (0000-0002-0121-5471); i8 (0009-0002-1276-1299); i9 (0000-0002-0173-7894); i10 (0000-0002-4949-7738); i11 (0009-0001-8755-5206)

Affiliations:

^{a1} Department of Computing, The Hong Kong Polytechnic University

^{a2} Institute of Information Engineering, Chinese Academy of Sciences

^{a3} SCIS, Singapore Management University

^{a4} Digital Trust Centre, Nanyang Technological University

^{a5} Ethereum Foundation

Associateship clarifications:

* Ph.D student (non-employee).

Main contacts:

- **Team mailing list:** octopus_mptc@googlegroups.com
- **Primary technical contact person:** Haiyang Xue, haiyangxue@smu.edu.sg
- **Secondary contact person 1:** Guofeng Tang, tang.guofeng789@gmail.com
- **Secondary contact person 2:** Bowen Jiang, bowen.jiang.2024@phdcs.smu.edu.sg

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

The scope of this preview document is to present our plan for submitting a t -out-of- n threshold EdDSA scheme, denoted by Teddy. The submission focuses on thresholding NIST-standard EdDSA signature [FIPS-186-5, §7] that falls under Category N1: Signing, Subtype: QV, and Specification: EdDSA Sign of the NIST Threshold Call [NIST-IR8214C].

We plan to propose a two-round t -out-of- n threshold EdDSA signature scheme. The core technical contribution lies in a novel, verifiable, and deterministic distributed randomness generation mechanism that seamlessly supports general threshold structures.

2. Specification

2.1. System model

The threshold profile is t -out-of- n . Key generation is executed among any n parties, after which any t of those n parties can cooperatively generate a valid signature.

The parties operate within a synchronous network and are assumed to be connected by authenticated point-to-point channels. We do not assume the availability of a broadcast channel.

2.2. Why Prior Deterministic Schemes Are Limited to n -out-of- n

Recall that an EdDSA signature on a message m under secret key (x, sk) with public key $X = g^x$ is a pair (R, s) with

$$R = g^r, \quad e = H(R \| X \| m), \quad s = r + ex \bmod q,$$

verified by checking $g^s = R \cdot X^e$, where the nonce r is *derived deterministically* from the secret key and the message. This determinism is the defining feature of EdDSA: it removes any dependence on the quality of the signer's randomness.

Carrying determinism into the distributed setting is what forces existing schemes into the n -out-of- n regime. In every such scheme, party P_i derives its nonce share locally, before any interaction, as

$$r_i = F_{\text{sk}_i}(m)$$

for a PRF F evaluated on its own key material (sk_i as share of sk) and the message, and then publishes $R_i = g^{r_i}$ together with a proof that R_i was computed correctly, an exponent-VRF evaluation, in the terminology of [BHLS25]. The parties set $R = \prod_i R_i$ and each replies with its share $s_i = r_i + ex_i$ of the signature. The essential point is that r_i is a function of (sk_i, m) *alone*: at the moment P_i evaluates F , it has not yet spoken to anyone, so it cannot condition r_i on anything the other parties contribute.

The n -out-of- n case is safe. When all n parties must sign, the set of contributors is fixed to $[n]$. For a given message m , the aggregate nonce $R = \prod_{j \in [n]} g^{r_j}$ is therefore a deterministic function of m , and so is the challenge $e = H(R \| X \| m)$. Re-running the protocol on the same m reproduces exactly the same transcript: the adversary learns one linear equation $s_i = r_i + ex_i$ in the two unknowns (r_i, x_i) , and no amount of replaying yields a second, independent one.

A dynamic signing set breaks this. In the t -out-of- n setting the contributing set is no longer fixed, and an adversary may choose it. Suppose $\mathcal{A}$ corrupts $t - 1$ parties and requests a signature on the *same* message m twice, first with a qualified set Q and then with a different qualified set $Q' \neq Q$, where an honest party P_i belongs to both. The two executions produce

$$R_Q = \prod_{j \in Q} g^{r_j}, \quad R_{Q'} = \prod_{j \in Q'} g^{r_j}, \quad R_Q \neq R_{Q'},$$

and hence two distinct challenges $e_Q = H(R_Q \| X \| m)$ and $e_{Q'} = H(R_{Q'} \| X \| m)$. Party P_i , however, derives the *same* $r_i = F(\text{sk}_i, m)$ in both executions, since its derivation does not see Q . It therefore publishes

$$s_i^Q = r_i + e_Q \lambda_{i,Q} x_i, \quad s_i^{Q'} = r_i + e_{Q'} \lambda_{i,Q'} x_i,$$

where $\lambda_{i,Q}$ is the Lagrange coefficient for i with respect to Q . These are two independent linear equations in the two unknowns r_i and x_i , so the adversary simply solves

$$x_i = \frac{s_i^Q - s_i^{Q'}}{e_Q \lambda_{i,Q} - e_{Q'} \lambda_{i,Q'}},$$

the denominator being nonzero except with probability $1/q$. Note that the Lagrange coefficients offer no protection: they are public and known to $\mathcal{A}$. Having recovered x_i , the adversary holds t shares of x , the $t - 1$ it corrupted plus P_i 's, and reconstructs the full signing key. We stress that $\mathcal{A}$ never deviates from the protocol; it only exercises the freedom that a t -out-of- n access structure is supposed to grant.

2.3. Key idea

The failure above is due to the fact that r_i ignores the signing set, while the challenge e does not. Therefore, our signing protocol enables each party to *deterministically* derive its local randomness r_i as a function of its secret key share sk_i , the (hashed) message m , and the active signer set S , i.e., $r_i = F_{\text{sk}_i}(S \| m)$, which is a key mechanism that empowers general threshold structures. Each party then broadcasts $R_i = g^{r_i}$ alongside a zero-knowledge proof of correct computation. Under the hood, we leverage an exponent verifiable random function (eVRF) construction, as formalized by Boneh et al. [BHLS25]. Specifically, an eVRF allows one to persuade the verifier the honest evaluation of a pseudorandom function (PRF) $v = F_k(u)$, without revealing both the PRF key k and the evaluation result v , but only g^k and g^v . This property of eVRF makes it an ideal solution to threshold EdDSA, and now each party only needs to prove the following relation:

$$\mathcal{R} = \{((\text{pk}_i, m, R_i); \text{sk}_i) : \text{pk}_i = g^{\text{sk}_i} \wedge R_i = g^{F_{\text{sk}_i}(S \| m)}\}$$

We enhance this design by introducing a streamlined eVRF scheme for several candidates of F , such as Poseidon, AES, SHA256, etc.

The major goal of our construction is to improve both the proving and verification efficiency of $\mathcal{R}$. While it is possible to encode $\mathcal{R}$ as a monolithic circuit and prove it via a proof system (specifically, a zkSNARK), this is suboptimal in terms of performance. Note that $\mathcal{R}$ consists of two components: the evaluation of F that involves bit operations (when instantiated with AES, SHA3, etc.) or field operations (when instantiated with circuit-friendly hash functions such as Poseidon), and the computation of g^x that involves group operations over twisted Edwards curves. Consequently, we must *emulate* at least one type of these operations via foreign-field arithmetic, over the field that the circuit is defined upon. This introduces typically one to two orders of magnitude overhead per multiplication.

Inspired by [KMN23; OKMZ25], our construction no longer depends on a single circuit for the entire $\mathcal{R}$. Instead, we handle joint hash and elliptic-curve group operations efficiently by leveraging specialized solutions for each type of operations and connecting them via commit-and-prove. Specifically:

- $pk_i = g^{sk_i}$ and $R_i = g^{r_i}$ on edwards25519 are proven by a Schnorr-style Sigma protocol outside of a circuit.
- $r_i = F_{sk_i}(S || m)$ is proven in-circuit by a proof system that can efficiently handle Poseidon/AES/SHA256/SHA3.
- The linkage of sk_i and r_i between these two components is proven by a link argument, tying their commitments together.

To summarize, our technical contributions are twofold: the general t -out-of- n access structure for threshold EdDSA, and optimised zero-knowledge proofs for exponent-VRF computation across several PRF candidates, including Poseidon, AES and SHA256 or SHA3.

2.4. Security

Our scheme is proven unforgeable in a game-based security model, offering resistance against adversaries corrupting up to $t - 1$ parties. Remarkably, we guarantee strong unforgeability: the adversary cannot forge a fresh signature σ^* for any message m^* , even if m^* was previously signed during the query phase, provided that (m^*, σ^*) was never returned as a pair.

3. The Signing Protocol

The digest. Every party in S computes the same public value

$$h = H_{\text{non}}(\text{dst} || H_{\text{pp}}(pp) || \text{ep} || \langle S \rangle || X || m) \in \{0, 1\}^{256}, \quad (1)$$

where dst is a domain separator naming the PRF. The indicator encoding $\langle S \rangle$ has fixed length n and one value per set, so neither an ordering convention for S nor the absence of a separator

between S and m can make two distinct pairs (S, m) collide. The digest is a full 256 bits: with 128 bits, a birthday search would find $(S, m) \neq (S', m')$ giving an honest party the same nonce share in two sessions with different challenges. Since h is public and known *before* the first round, it is computed in the clear and never inside a circuit, and F processes a single 256-bit input, independent of n and of $|m|$.

The nonce relation.

$$\mathcal{R}_{\text{nonce}} = \left\{ \left((\text{pk}, h, R); \text{sk} \right) : 0 \leq \text{sk} < q \wedge \text{pk} = g^{\text{sk}} \wedge R = g^{\text{red}_q(F_{\text{sk}}(h))} \right\}. \quad (2)$$

We write $r(\text{sk}, h) = \text{red}_q(F_{\text{sk}}(h))$ and let $(\text{NP.Prove}, \text{NP.Verify})$ be a NIZK for $\mathcal{R}_{\text{nonce}}$ — the *nonce proof* — constructed in Section 4.

Signing: two rounds, point-to-point, no broadcast

Input to P_i , $i \in S$: (pp, x_i, sk_i) , signer set S with $|S| = t$, message m . **Common:** h as in (1).

Round 1 (Sign_1).

1. $r_i \leftarrow \text{red}_q(F_{\text{sk}_i}(h))$, $R_i \leftarrow g^{r_i}$.
2. $\pi_i \leftarrow \text{NP.Prove}((\text{pk}_i, h, R_i), \text{sk}_i)$ for $\mathcal{R}_{\text{nonce}}$.
3. Send $\alpha_{1,i} = (R_i, \pi_i)$ to every P_j , $j \in S$.

Round 2 (Sign_2). On receiving $(R_j, \pi_j)_{j \in S}$:

1. Recompute r_i and R_i from (sk_i, h) (*no state is kept between rounds*).
2. For each $j \in S \setminus \{i\}$: if R_j does not decode to a point of $\mathbb{G}$, or $\text{NP.Verify}((\text{pk}_j, h, R_j), \pi_j) = 0$, output (abort, j).
3. $R \leftarrow \prod_{j \in S} R_j$, $e \leftarrow H_{\text{sig}}(R \parallel X \parallel m)$, $s_i \leftarrow r_i + e \lambda_{i,S} x_i \bmod q$.
4. Send $\alpha_{2,i} = s_i$ to every P_j , $j \in S$.

Combine (public, deterministic). Recompute R and e . For each $j \in S$, if $g^{s_j} \neq R_j \cdot X_j^{e \lambda_{j,S}}$, output (abort, j). Otherwise output $\sigma = (R, \sum_{j \in S} s_j \bmod q)$.

Verify. Standard Ed25519 verification of σ under X .

Set-and-message determinism. For fixed pp and ep , the nonce shares, the partial responses and the signature are determined by (S, m) ; only the proofs π_i are randomized. The scheme therefore outputs at most one signature per signer set and message — in practice one per operational quorum — each of them a standard Ed25519 signature. No party keeps per-session secret state.

4. The Nonce Proof

The relation $\mathcal{R}_{\text{nonce}}$ of (2) mixes

- an *algebraic* part — two exponentiations $\text{pk} = g^{\text{sk}}$ and $R = g^r$ on edwards25519; and
- a *non-algebraic* part — the PRF evaluation $r = \text{red}_q(F_{\text{sk}}(h))$.

A monolithic circuit must emulate one of the two. Group operations over a circuit field different from the base field of edwards25519 need foreign-field arithmetic (one to two orders of magnitude per multiplication); bit-oriented PRFs over a prime field must be bit-decomposed. We therefore keep the group equations *outside* the circuit and let the circuit see only scalars.

Following Sigmabus, the exponentiations are checked by a Σ -protocol outside the circuit, and the circuit only checks its *response*. Unusually, a *single* Σ -protocol covers both exponents:

Prover

$k \xleftarrow{\$} \mathbb{Z}_q$, $K = g^k$
 commit to $(k, \text{sk}, \text{PRF witness})$
 $\rightarrow K, \text{cm}$

Verifier

$$c = H_{\text{fs}}((\text{pk}, h, R), \text{cm}, K)$$

$z = k + c \cdot \text{sk} + c^2 \cdot r \bmod q$
 $\rightarrow z, \pi_{\text{circ}}$

Outside the circuit: $g^z \stackrel{?}{=} K \cdot \text{pk}^c \cdot R^{c^2}$

Inside π_{circ} : cm opens to $(k, \text{sk}, \dots)$, $0 \leq \text{sk} < q$, $z = k + c \text{sk} + c^2 \text{red}_q(F_{\text{sk}}(h))$.

Let k^*, sk^*, r^* be the discrete logarithms of K, pk, R , and let $\tilde{k}, \tilde{\text{sk}}, \tilde{r}$ be the committed values. The group equation gives $z = k^* + c \text{sk}^* + c^2 r^*$ and the circuit gives $z = \tilde{k} + c \tilde{\text{sk}} + c^2 \tilde{r}$, so c is a root of

$$\Delta(c) = (k^* - \tilde{k}) + (\text{sk}^* - \tilde{\text{sk}})c + (r^* - \tilde{r})c^2,$$

whose coefficients are fixed *before* c is derived (the commitment is hashed into c). If $\Delta \neq 0$ a uniform c is a root with probability $\leq 2/q$; if $\Delta = 0$ the statement is true. Equivalently, three transcripts with distinct challenges and the same first message extract sk and r .

Candidate Instantiations. The commit-and-prove SNARK could be *Groth16* extended by a commitment to part of the witness in the style of BSB22 and LegoSNARK: constant proofs below 1 KB and a verifier of 0.8–3.1 ms, which matters because each party verifies $t - 1$ proofs per session and Groth16 proofs batch into one final exponentiation. Its circuit-dependent setup is acceptable, since a deployment fixes one circuit and can run a one-time ceremony. Bit operations become table lookups, checked inside the circuit with *LogUp* or outside it with *Locq*, taking the SHA-512 circuit from 67,313 constraints to 23,120 or 3,018. Two PRFs are instantiated: $F(h) = \text{SHA-512}(\|h)$, one compression with a 512-bit output, and the Poseidon permutation with width 5 at 128-bit security. AES stays under investigation: lookup-friendly, but modes with a 128-bit bottleneck admit a nonce collision after about 2^{64} messages. Security holds in the algebraic group model under d -DLOG, with the PRFs modelled as keyed random oracles.

Our preliminary benchmark for SHA-256 and Poseidon is given in Table 1.

Table 1: Costs of nonce proofs in deterministic multi-party Schnorr signing.

Scheme	Nonce derivation	Prove	Proof	Verify
Teddy	SHA-512, Groth16 with Locq	32ms	512B	1.85ms
Teddy	Poseidon, Groth16	5.5ms	256B	0.83ms

5. Open-Source Implementation

1. **Code structure:** We will provide a Rust reference implementation of Teddy. It will implement a complete protocol that includes distributed key generation and two-round signing. The core components involve elliptic-curve operations, secret sharing, and the selected eVRF construction. Toolchain and dependency versions will be pinned, third-party components will be documented, and the implementation will be built and tested on the NIST baseline platform.
2. **Code progress and availability:** The protocol and candidate eVRF instantiations are being refined, and the reference implementation is under development. The implementation will be audited against its protocol specification. At submission, the repository will be made public at a fixed commit with instructions for building, running, testing, and benchmarking the code.
3. **Implementation of the networking model:** Each party will run as a separate process and communicate over TCP using authenticated messages. Point-to-point messages will be signed by their senders and bound to the protocol and session in which they are sent. Configuration files will specify party identities, public keys, network addresses, threshold parameters, and session information.
4. **Testing:** Testing will cover primitives, protocol, and multiprocess executions of DKG and signing. We will introduce malicious deviations and network disruptions at different protocol stages, and verify that honest parties either complete correctly or terminate without accepting invalid results.

6. Licensing and Patent Claims

1. **Licensing:** The Rust code developed for this submission will be released under MIT OR Apache-2.0. All Bundled Dependencies and External Dependencies will use licenses as required by the NIST Threshold Call. The package will record licenses using SPDX expressions, preserve required notices, identify the license applicable to each component, and verify that the compiled reference implementation can be redistributed under an allowed license.
2. **Patent claims:** To our knowledge, no patent has or could have claims covering the contents of our submission.

References

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

[BHLS25] Dan Boneh, Iftach Haitner, Yehuda Lindell, and Gil Segev. “Exponent-VRFs and Their Applications”. In: *EUROCRYPT*. Vol. 15607. Springer, 2025, pp. 195–224. DOI: [10.1007/978-3-031-91098-2_8](https://doi.org/10.1007/978-3-031-91098-2_8). URL: <https://eprint.iacr.org/2024/397>.

[KMN23] George Kadianakis, Mary Maller, and Andrija Novakovic. “Sigmaplus: Binding Sigmas in Circuits for Fast Curve Operations”. In: *IACR Cryptol. ePrint Arch.* 2023 (2023), p. 1406. URL: <https://eprint.iacr.org/2023/1406>.

[OKMZ25] Michele Orrù, George Kadianakis, Mary Maller, and Greg Zaverucha. “Beyond the circuit: How to minimize foreign arithmetic in zkp circuits”. In: *IACR Communications in Cryptology 2* (2025), p. 1. DOI: [10.62056/an-4c3c2h](https://doi.org/10.62056/an-4c3c2h). URL: <https://eprint.iacr.org/2024/265>.

[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).