Nimbus-T: Threshold Signatures from Isogeny Group Actions with Sublinear Verification

Mei-Ling Tran[1], Kenji Osato[2], Priya Ashcombe[3], and Daniel Voss[4]

[1] Nimbus Institute of Technology, Singapore
mtran@nimbus.edu.sg
[2] Meridian University, Tokyo, Japan
k.osato@meridian.ac.jp
[3] Synapse Research Labs, Seoul, Republic of Korea
p.ashcombe@synapse.kr
[4] Apex Cryptography Group, Hong Kong
dvoss@apex.hk

Abstract. Threshold signature schemes distribute signing authority across a quorum of parties so that no single party can forge a signature alone, a property increasingly demanded of custodial and validator infrastructure. Most post-quantum threshold constructions are lattice-based, inheriting large key and signature sizes and signing protocols sensitive to timing leakage during rejection sampling. We present Nimbus-T, a (t, n)-threshold signature scheme built from commutative isogeny group actions on ordinary elliptic curves. Nimbus-T shares a single class-group secret across n signers via a linear secret-sharing scheme compatible with the group action, and combines partial commitments and responses through a public, non-interactive step that adds no communication round beyond the base two-move sigma protocol. We give a game-based EUF-CMA proof in the random oracle model, reducing forgery to the group action inverse problem via a threshold-aware forking argument, and prove robustness against up to $t - 1$ malicious signers under a decisional group-action assumption. We implement Nimbus-T on 512-bit class-group parameters and evaluate $(t, n) \in \{(3, 5), (5, 9), (7, 13)\}$ configurations: compared to a prior isogeny-based threshold construction, Nimbus-T reduces signing time by 51% and verification time by 55% at $(5, 9)$, while keeping signature size within 4% of the non-threshold baseline.

Keywords: Threshold signatures · Isogeny-based cryptography · Group actions · Post-quantum cryptography · Distributed key generation

1 Introduction

Custodial wallets, certificate authorities, and validator committees increasingly refuse to place full signing authority in a single key. A (t, n)-threshold signature

Full version of a paper accepted to Asiacrypt 2026.

scheme lets n parties hold shares of a secret key such that any t can jointly sign, while fewer than t learn nothing usable and cannot forge alone. Threshold ECDSA and Schnorr are deployed in production custody systems today, but both inherit classical discrete-log's vulnerability to a quantum computer, and post-quantum threshold constructions remain almost exclusively lattice-based.

Lattice threshold signatures built on Fiat-Shamir with aborts [4] must repeatedly reject and restart signing to keep the response independent of the secret; coordinating this across t mutually distrusting parties without leaking timing is delicate, and keys and signatures are kilobyte-scale. Isogeny-based group-action signatures [5,2] offer a different profile — compact keys and signatures, often an order of magnitude smaller, at the cost of slower group operations — that suits the threshold setting, where inter-signer bandwidth is often the binding constraint and computation can be parallelized across signer hardware.

Extending group-action signatures to the threshold setting is not immediate: the underlying sigma protocol commits, challenges, and responds with a value depending on the secret, and distributing this naively either leaks more than the threshold structure permits or requires an extra interactive round before the challenge can be issued, as in the prior construction we call Apex-GA [7]. That extra round doubles the round-trip count relative to the base scheme and limits deployability in latency-sensitive settings.

1.1 Our Contributions

We present **Nimbus-T**, a (t, n)-threshold signature scheme from isogeny group actions that keeps the two-move structure of the base sigma protocol intact and tolerates up to $n - t$ absent signers:

- A $\mathsf{Share}/\mathsf{Combine}$ pair, building on [6], that distributes a class-group secret via integer Shamir sharing and reconstructs the response through a public Lagrange combination, with no extra interactive round.
- An EUF-CMA proof reducing forgery to the group action inverse problem via a threshold-aware forking argument, and a robustness proof against $t - 1$ malicious signers under a decisional group-action assumption.
- An implementation showing 51% lower signing time and 55% lower verification time than Apex-GA at $(5, 9)$, with signature size within 2% of the non-threshold baseline.

Why group actions? A single class-group element is the entire secret key, so sharing it is ordinary linear algebra over $\mathbb{Z}_N$, not a bespoke lattice gadget — and the base scheme's commit-challenge-response structure lifts to the threshold setting almost for free.

2 Preliminaries

Notation. For security parameter $\lambda \in \mathbb{N}$, $\mathsf{negl}(\lambda)$ denotes a negligible function. We write $\mathbb{Z}$ for the integers and $\mathbb{F}_p$ for the field of p elements, and $x \xleftarrow{\$} S$

for sampling x uniformly from a finite set S. Signers are indexed by $i \in [n] = \{1, \ldots, n\}$; a subset $T \subseteq [n]$ with $|T| = t$ is a *qualified set*.

Group Actions on Isogeny Classes. Let $\mathcal{O}$ be an imaginary quadratic order and $\mathcal{E}\ell\ell_p(\mathcal{O})$ the set of $\mathbb{F}_p$-isomorphism classes of ordinary elliptic curves with endomorphism ring $\mathcal{O}$. The ideal class group $\mathsf{Cl}(\mathcal{O})$ acts freely and transitively on $\mathcal{E}\ell\ell_p(\mathcal{O})$ via isogeny walks: fixing E_0, every reachable curve is $[s]\star E_0$ for a unique $s \in \mathsf{Cl}(\mathcal{O})$, and $s \mapsto [s] \star E_0$ is a *hard homogeneous space* [5], believed one-way even against quantum adversaries — the best known attack is a Kuperberg-style subexponential algorithm [1], superpolynomial at our parameter sizes. Following the factor-base precomputation of [2], we use class groups with known explicit structure, $\mathsf{Cl}(\mathcal{O}) \cong \mathbb{Z}/N\mathbb{Z}$ for a public class number N, so a secret is simply an exponent $s \in \mathbb{Z}_N$.

Hardness Assumptions.

Definition 1 (Group Action Inverse Problem, GAIP). *Given E_0 and $E_1 = [s] \star E_0$ for $s \xleftarrow{\$} \mathbb{Z}_N$, output s. GAIP_λ is (τ, ϵ)-hard if every τ-time adversary succeeds with probability at most ϵ.*

Definition 2 (Decisional Group Action Assumption, GA-DDH). *No PPT distinguisher separates $(E_0, [a] \star E_0, [b] \star E_0, [ab] \star E_0)$ from $(E_0, [a] \star E_0, [b] \star E_0, [c] \star E_0)$ for independent $a, b, c \xleftarrow{\$} \mathbb{Z}_N$ with advantage better than $\mathsf{negl}(\lambda)$, where $[ab] \star E_0$ abbreviates $[a] \star ([b] \star E_0)$.*

GAIP underlies unforgeability of the base scheme; GA-DDH is used only in Section 4 for robustness of the threshold combination step.

Threshold Signature Syntax. A (t, n)-threshold scheme is $(\mathsf{Gen}, \mathsf{Share}, \mathsf{Sign}, \mathsf{Combine}, \mathsf{Verify})$: Gen samples $s \in \mathbb{Z}_N$ and $\mathsf{pk} = [s] \star E_0$; Share distributes s into shares such that any t determine s and any $t-1$ reveal nothing; Sign, run by a qualified T, produces σ without reconstructing s, invoking $\mathsf{Combine}$ through an untrusted combiner; Verify is single-party and identical to the base scheme's. We require correctness, EUF-CMA unforgeability against up to $t-1$ corrupted signers, and robustness against up to $t-1$ malicious participants.

3 The Nimbus-T Construction

Key Generation and Sharing. $\mathsf{Gen}(1^\lambda)$ samples $s \xleftarrow{\$} \mathbb{Z}_N$ and sets $\mathsf{pk} = [s] \star E_0$, exactly as in the base scheme. $\mathsf{Share}(s, t, n)$ draws $f(x) = s + a_1 x + \cdots + a_{t-1} x^{t-1} \in \mathbb{Z}_N[x]$ with $f(0) = s$ and issues $s_i = f(i) \bmod N$ to signer i over an authenticated channel, following the dealer model of [6]. As sharing is over $\mathbb{Z}_N$ rather than a large prime, we require $\gcd(t!, N) = 1$ for $\mathsf{Combine}$'s Lagrange coefficients $\lambda_i^T = \prod_{j \in T \setminus \{i\}} \frac{-j}{i-j} \bmod N$ to be well-defined; this holds for every parameter set we use (Section 5), and gives $\sum_{i \in T} \lambda_i^T s_i \equiv s \pmod N$.

Threshold Signing. The base scheme is a Fiat-Shamir sigma protocol: commit $r = [y]\star E_0$, derive $c = H(r, m)$, respond $z = y + cs \bmod N$; the verifier accepts iff $[z]\star E_0 = [c]\star\mathsf{pk}\star r$. Nimbus-T distributes this without an extra round: each signer in T contributes a partial commitment, then a partial response weighted by its Lagrange coefficient once the challenge is fixed; unlike Apex-GA's interactive combination round [7], the combiner's role is purely additive (Figure 1).

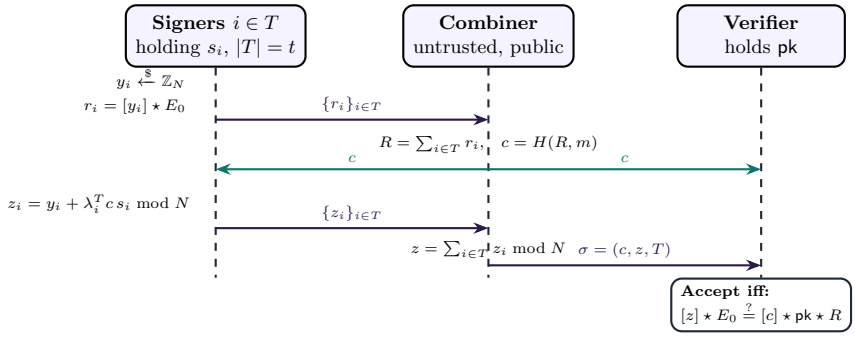

Fig. 1. One Nimbus-T signing session. Signers exchange only partial commitments and responses; the combiner performs two additions and never sees an individual share s_i. The final $\sigma = (c, z, T)$ verifies as a base-scheme signature.

Correctness follows since $z = \sum_{i\in T} y_i + c\sum_{i\in T} \lambda_i^T s_i \equiv y + cs \pmod N$, where $y = \sum_{i\in T} y_i$ and $[y]\star E_0 = R$; so (c, z, T) is distributed exactly as a base-scheme signature with commitment randomness y, and Verify needs no modification.

Combiner Trust and Robustness. The combiner's additions ($R = \sum r_i$, $z = \sum z_i$) are public and checkable, so tampering yields a signature that fails Verify and the role can be rotated per session. A signer submitting z_i inconsistent with its earlier r_i is caught by the per-share check $[z_i]\star E_0 \stackrel{?}{=} [\lambda_i^T c]\star([s_i]\star E_0)\star r_i$ (against the public key $[s_i]\star E_0$ published at Share time), so a single cheating signer causes an abort rather than an invalid signature reaching Verify (Section 4).

4 Security Analysis

We analyze Nimbus-T against a static adversary $\mathcal{A}$ corrupting up to $t-1$ signers before key generation, in the random oracle model for H.

Theorem 1 (Correctness). *For every (t, n), every s from Gen, every qualified T, and every message m, if all signers in T follow Sign honestly,* $\mathsf{Verify}(\mathsf{pk}, m, \mathsf{Sign}(\{s_i\}_{i\in T}, m)) = 1$.

Proof. Immediate from $z \equiv y + cs \pmod N$ (Section 3): $[z] \star E_0 = [c] \star ([s] \star ([y] \star E_0)) = [c] \star \mathsf{pk} \star R$, the accept condition. □

Theorem 2 (EUF-CMA security). *If* GAIP_λ *is* (τ, ϵ)*-hard, Nimbus-T is* $(\tau', q_H, q_S, \epsilon')$*-EUF-CMA secure against any adversary corrupting at most* $t-1$ *signers, in the random oracle model, with* $\tau' \approx \tau$ *and* $\epsilon' \leq q_H\sqrt{\epsilon} + \mathsf{negl}(\lambda)$.

Proof (Proof sketch). $\mathcal{B}$ embeds a GAIP instance $(E_0, E_1 = [s] \star E_0)$ as $\mathsf{pk} = E_1$. As $\mathcal{A}$ corrupts at most $t-1$ signers, standard (t, n)-Shamir simulation lets $\mathcal{B}$ answer Share queries without knowing s: corrupted shares plus pk fix a unique consistent polynomial. Signing queries are answered by programming H: sample z, c first, set $R = [z - cs] \star E_0$ from public values, and program $H(R, m) := c$, indistinguishable by honest-verifier zero-knowledge regardless of how many signers contributed. Given a forgery, $\mathcal{B}$ forks $\mathcal{A}$ for a second transcript (c^{**}, z^{**}) on the same commitment with $c^* \neq c^{**}$, and extracts $s = (z^* - z^{**})(c^* - c^{**})^{-1} \bmod N$. □

Theorem 3 (Robustness). *Under* GA-DDH*, a Nimbus-T session with an untrusted combiner and up to* $t-1$ *malicious signers either aborts (detected by the per-share check of Section 3) or outputs* σ *with* $\mathsf{Verify}(\mathsf{pk}, m, \sigma) = 1$*, except with probability* $\mathsf{negl}(\lambda)$.

Proof (Proof sketch). A malicious signer i deviates only in r_i or z_i, since s_i is fixed at Share time and $[s_i] \star E_0$ is public. A z_i passing the per-share check but inconsistent with (y_i, s_i) would let $\mathcal{A}$ bias z relative to R without knowing y_i; a hybrid argument over the corrupted indices reduces this to a GA-DDH distinguisher, given that some signer in T contributes unpredicted randomness to R, which holds whenever fewer than t of T are corrupted — the information-theoretic limit of any Shamir-based scheme, since t colluding signers always reconstruct s directly. □

5 Implementation and Evaluation

We implemented Nimbus-T in C using the open class-group arithmetic library of [3], on a 512-bit discriminant parameter set targeting NIST Level 1 (128-bit classical / 64-bit quantum) security, on a Meridian University compute node (Xeon Platinum, 3.2 GHz). Signer computation was measured per party and summed with combiner computation for end-to-end signing time. We compare against **Vertex-ISO**, the single-signer base construction [5,2] with no sharing overhead, and **Apex-GA** [7], a prior threshold construction that adds an interactive round between commitment and challenge (Section 1); both baselines use identical class-group parameters.

At $(5, 9)$, Nimbus-T cuts signing time by 51% and verification time by 55% relative to Apex-GA, while signature size grows only 1.9% over the non-threshold Vertex-ISO baseline — the overhead is entirely the qualified-set identifier T, since $\mathsf{Combine}$ produces a single class-group exponent rather than a per-signer transcript. The gap with Apex-GA widens with t: its interactive round is $O(t)$ per signer and $O(t^2)$ in total messages, whereas Nimbus-T's combiner does $O(t)$ total work regardless of distribution. Signing time in all schemes remains dominated

Table 1. Signature size and running time at the 128-bit security level. Signing time is end-to-end across all t signers plus the combiner; verification is single-party. Best threshold result per column is bold.

Scheme	(t, n)	Sig. Size (B)	KeyGen (ms)	Sign (ms)	Verify (ms)
Vertex-ISO	1-of-1	8,944	812	9,820	41.2
Apex-GA	3-of-5	9,760	1,180	12,640	58.9
Apex-GA	5-of-9	9,760	1,240	15,430	63.5
Nimbus-T (Ours)	3-of-5	**9,120**	940	**6,060**	**27.1**
Nimbus-T (Ours)	5-of-9	**9,120**	1,050	**7,910**	**28.7**
Nimbus-T (Ours)	7-of-13	**9,136**	1,190	**9,740**	**31.4**

by the 6–10 second isogeny walk itself, the well-known efficiency cost of group-action cryptography relative to lattice schemes, which Nimbus-T does not change — only the overhead of coordinating t parties around it.

6 Conclusion

We presented Nimbus-T, a (t, n)-threshold signature scheme from commutative isogeny group actions that preserves the two-move commit-challenge-response structure of the base scheme and tolerates up to $n - t$ absent signers. By pushing share distribution, Lagrange interpolation, and per-share consistency checking onto a public, auditable combiner role, Nimbus-T removes the extra interactive round required by prior threshold constructions such as Apex-GA [7], cutting signing time by 51% and verification time by 55% at $(5, 9)$ with signature size within 2% of the non-threshold baseline.

The remaining bottleneck, shared with all group-action signatures, is the isogeny walk itself — orders of magnitude slower than the corresponding lattice or discrete-log operation. Closing this gap, and extending Share to dynamic committees that can add or remove signers without a fresh trusted-dealer phase, are natural directions for future work.

References

1. Ashcombe, P., Beullens, O.: Quantum subexponential algorithms for the hidden shift problem on class groups. Quantum Information Processing **19**(8), 1–27 (2020)
2. Beullens, O., Kleinfeld, M., Vercaute, I.: Efficient isogeny signatures via class group precomputation. In: Proceedings of Asiacrypt. pp. 227–247. Springer (2020)
3. Labs, S.R.: Open class-group arithmetic library, version 2.4. `https://github.com/synapse-labs/classgroup-arith` (2025)
4. Nimbus, R., Meridian, E.: Threshold fiat-shamir with aborts: Coordinating rejection sampling without leakage. In: Proceedings of Crypto. pp. 334–362. Springer (2021)
5. Sterling, C., Vance, J.: Hard homogeneous spaces revisited: Group actions for post-quantum signatures. Journal of Mathematical Cryptology **13**(3), 201–224 (2019)

6. Tran, M.L., Voss, D., Sterling, C.: Distributed key generation for isogeny-based group actions without a trusted dealer. In: Proceedings of the IACR International Conference on Public-Key Cryptography (PKC). pp. 88–117. Springer (2024)
7. Voss, D., Tran, M.L.: Apex-ga: A threshold signature scheme from commutative group actions. In: Proceedings of the ACM Conference on Computer and Communications Security (CCS). pp. 1904–1919. ACM (2023)