

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

Title: Hermine

Subtitle: An Efficient Raccoon-Style Non-Interactive Threshold Signature with Advanced Properties

Version: 1.0 (2026-01-19)¹

Team name: Hermine Team

Team members: Giacomo Borin, Sofía Celi, Rafael del Pino, Thomas Espitau, Shuichi Katsumata, Guilhem Niot, Thomas Prest, Kaoru Takemure

Abstract: This document previews our Hermine submission to the NIST Multi-Party Threshold Cryptography (MPTC) Call. Hermine is a post-quantum lattice-based threshold signature that achieves two-round (partially non-interactive) signing, with non-interactive identifiable aborts (IA), as well as support of Distributed Key Generation (DKG) and proactive key refreshes. Unlike existing post-quantum threshold signatures, Hermine preserves a fully non-interactive online phase even under misbehavior, so identifying faulty signers never triggers extra interaction.

Technically, Hermine combines the two-round lattice-signature paradigm of Espitau, Katsumata and Takemure (Crypto '24) with Vandermonde secret sharing (Asiacrypt '94) to enable efficient advanced properties for medium-size committees (up to 64 signers), matching a key target profile in the NIST MPTC effort. Security is established from standard lattice assumptions (MLWE and MSIS), and the design avoids the use of costly non-interactive zero-knowledge proofs (NIZKs), ensuring simple and efficient implementation.

This document outlines the design paradigm of Hermine, security model, and an implementation plan, including open-source code and benchmarks.

Proposed crypto-systems: Hermine (Categories S1.1 and S4.1)

Keywords: Threshold Cryptography; NIST Threshold Call



Logo designed by Sofía Celi.

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: Giacomo Borin ^{i1,a1,a2}, Sofia Celi ^{i2,a3,a4}, Rafael del Pino ^{i3,a5}, Thomas Espitau ^{i4,a5}, Shuichi Katsumata ^{i5,a5,a6}, Guilhem Niot ^{i6,a5,a7}, Thomas Prest ^{i7,a5}, Kaoru Takemure ^{i8,a5,a6}

Open Researcher and Contributor Identifiers (ORCID):

i1 (0009-0001-7311-3802); i2 (0000-0002-3333-7764); i3 (0009-0001-8638-787X); i4 (0000-0002-7655-9594); i5 (0000-0002-8496-0476); i6 (0000-0002-2497-8770); i7 (0000-0003-1445-6212); i8 (0000-0002-9288-1911)

Affiliations:

^{a1} IBM Research Zurich @ Switzerland

^{a2} University of Zurich @ Switzerland

^{a3} Brave Research @ Portugal

^{a4} University of Bristol @ UK

^{a5} PQShield SAS @ France

^{a6} National Institute of Advanced Industrial Science and Technology (AIST) @ Japan

^{a7} Univ Rennes, CNRS, IRISA @ France

Main contacts:

- **Team mailing list:** contact@hermine-th.org
- **Primary technical contact person:** Guilhem Niot, guilhem@gniot.fr

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

This preview introduces Hermine, a lattice-based threshold variant of the signature scheme Raccoon [PKPR24]. Hermine brings the classical FROST [KG20; BCKMTZ22] desirable properties to the post-quantum setting: two-round online/offline signing, non-interactive identifiable aborts (IA), Distributed Key Generation (DKG) and proactive key refreshes. Our goal is to provide a practical, standard-assumption design that keeps the online round non-interactive even under misbehavior, thereby preserving the latency and robustness benefits of classical threshold schemes.

Hermine targets the Threshold Special Signing track (Category S1) and Special DKG (Category S4) for medium-size committees (up to $N \leq 64$). The protocol is built from standard lattice problems—MLWE and MSIS, and avoids the use of costly non-interactive zero-knowledge proofs (NIZKs).

In the classical setting, FROST is widely adopted thanks to its two-round online/offline efficiency, simple distribution of its key operations, and non-interactive IA. In the post-quantum (PQ) setting, current approaches lack efficient distributed key operations and either (i) achieve two rounds but lack IA (e.g., [EKT24]), or (ii) provide IA but at the expense of an additional round and interactive misbehavior identification (e.g., [PKNRT25]). For deployments where availability and user experience rely on a non-interactive online phase, a PQ analogue of FROST remains desirable and timely.

At a high-level, Hermine combines the two-round lattice-signing paradigm of [EKT24] with Vandermonde secret sharing [DDB95] to realize efficient distributed key operations and non-interactive IA. The Vandermonde structure enables simple, partial signature verifications that suffice to pinpoint faulty signers without triggering extra interaction during the online round. The resulting secret sharing is practical for $N \leq 64$, matching a key NIST MPTC parameter profile. Concretely, it achieves signature sizes comparable to Raccoon and other two-round lattice-based threshold schemes [EKT24; BKLMTT25], while providing the added benefits of non-interactive IA and efficient DKG/refresh. These ideas were drafted in our works [PENP25; BCdENP25].

We plan to submit: (i) a self-contained specification of Hermine (algorithms, system model); (ii) an open-source reference implementation with benchmarks and reproducible scripts; and (iii) an evaluation of communication/computation costs under the NIST "Medium" profile.

2. Specification

2.1. Organization

The Hermine specification targets the Threshold Special Signing (S1) and Special DKG (S4) categories. The document will be organized in two parts:

Part I: Foundation — Raccoon signature scheme. We begin by recalling the Raccoon signature scheme [PKPR24], which serves as the underlying primitive for Hermine, with parameters adapted to support the threshold functionality. This part specifies:

- The algebraic setting (rings, moduli, degrees) we target in Hermine.

- Security levels and parameter sets that we will adapt to the threshold setting. These will be focused on the public parameters used for public key encoding and for the signature verification.
- The hash functions used (e.g., SHAKE256) and their roles (domain separation, challenge derivation, commitment hashing).
- A high-level overview of Raccoon key generation, signing, and a detailed specification of its verification.

Part II: Threshold variant — Hermine. Building on Part I, we introduce the threshold variant with a clear separation between an offline precomputation phase and a non-interactive online phase. The interfaces are:

- (D)KeyGen: either centralized or distributed key generation, exporting per-signer secret shares and public verification material (partial public keys); supports identifiable aborts and proactive refreshes.
- Preprocess(): offline generation of commitments, independent of message and signer set.
- SignOnline(m, S, C): one-shot, non-interactive per-signer contribution on message m and signer set S using the commitments C produced by the committee; emits a signature share.
- Aggregate(S, shares): combine signature shares from a threshold subset into a final Raccoon signature.
- Verify(pk, m, σ): standard Raccoon verification (from Part I).
- IdentifyAbort($S, \text{PKs}, \text{transcript}$): non-interactive identification of misbehaving signers from their partial public keys and the transcript of the failed session.

Modular building blocks. Components are modularized for reuse and analysis: (i) the Raccoon core, (ii) a Vandermonde secret-sharing layer with short shares enabling partial signature verification, with both a distributed key generation protocol and proactive refreshes, and (iii) a two-round signing protocol with a single online round, and non-interactive identifiable aborts. We will provide parameter sets for the NIST “Medium” profile with $N \leq 64$.

2.2. System model

We consider a pessimistic system model, with security against active adversaries, and support of a dishonest majority, i.e. for any signing threshold $T \leq N$, up to $T - 1$ parties may be corrupted. The number of parties is limited to $N \leq 64$ due to the scalability of the secret sharing we use.

For communications, we model the presence of an aggregator that collects and broadcasts the messages of all the parties. This aggregator is untrusted for the unforgeability of the threshold scheme. For abort identification, it is assumed to implement a reliable network, as it is otherwise trivial to drop or alter user messages to have them marked as misbehaving.

The protocol key material can be setup distributively, and we provide a refresh mechanism (operated by one party) that ensures that as long as at most $T - 1$ parties are corrupted between two honest

refreshes, the scheme remains secure, allowing to recover from past corruptions. We assume that key refreshes are much less frequent than signing, and only support a bounded number of refreshes $\ll 2^{64}$ while maintaining security and identifiable aborts.

2.3. Security

Formulation and security goals. We analyze the security of both the distributed key generation (DKG) and the signing protocol. For the DKG, we consider a simulatability notion called functional simulatability, which is defined under a specific distribution of a simulated verification key. This notion ensures the existence of a simulator which simulates honest parties such that all parties obtain this simulated verification key as a result of the protocol *while allowing a partial leakage of the secret*. The goal of an adversary in this security game is to distinguish the real from the simulated game.

We consider two security notions for the signing protocol: unforgeability and abort identifiability. We adopt game-based security notions for both. The goal of an adversary in the unforgeability game is to produce a valid signature on a message *which has never been signed*. A adversary is allowed to execute the signing protocol with honest parties by adaptively choosing a message and participants, and *maliciously* behaving as the corrupted parties and the aggregator. In the security game for identifiable abort, an adversary executes the signing protocol with honest parties and an *honest* aggregator. Then the goal of the adversary is to cause two undesirable events: (i) an honest party is detected as misbehaving by IdentifyAbort, and (ii) the aggregated signature is invalid despite IdentifyAbort detecting no party as a misbehaving signer. Note that both security notions are defined under the successful termination of the DKG.

Model of corruption and refresh. We consider a semi-adaptive corruption model. In this model, an adversary first chooses the initial corruption set at the beginning of the security game (i.e., before starting the key generation) and is allowed to re-choose corrupted parties in the next epoch before starting the refresh protocol. Note that an epoch is the period between key generation and the first key refresh or between two key refreshes. We believe the scheme remains secure even under adaptive corruptions, where the adversary can choose corrupted parties at any time during the security game, but as for prior works, it appears challenging to prove this stronger notion in the lattice setting without sacrificing efficiency.

We assume the existence of a trusted dealer during the refresh protocol who generates update tokens for each party *without reconstructing the actual secret*. Each new secret key share in our refresh protocol is derived from the share in the current epoch and the corresponding update token. In the security game, an adversary is given such updated shares for a new corrupted set which are derived from *untampered* shares in the current epoch and honestly generated update tokens.

Assumptions and security strength estimation. We will show the security of our DKG and signing protocol under the MLWE and MSIS assumptions via the AOM-MISIS assumption [ZT25]. Also we will provide parameters for each security levels, e.g., 128-bit, 192-bit, and 256-bit security, based on cryptanalysis in [EKT24].

3. Open-Source Implementation

3.1. Code structure and language

We plan a reference implementation in C, with a small, portable codebase suitable for auditing and benchmarking. The implementation will expose a clear API corresponding to the elements of the protocol: verification, secret sharing, offline/online signing, aggregation, and IA verification. It will keep dependencies minimal.

3.2. Repository and availability

We will publish a public Git repository alongside our specification document. The repository will include straightforward build scripts and reproducible instructions for Linux and macOS. It will be made available under the Github organization [PQ-Hermine](#).

3.3. Networking model

We will include a mocked networking with a simple local runner that simulates broadcast with a centralized aggregator.

3.4. Testing and reproducibility

We will provide unit tests to validate share generation, aggregation, signature verification, and IA checks on small instances. Tests will be deterministic via seedable randomness. We will additionally include minimal benchmarking scripts to report signing and aggregation latencies.

4. Experimental Performance Evaluation

4.1. Scope, metrics, and methodology

We focus on the NIST “Medium” profile ($N \leq 64$). We will report the offline/online latency of our scheme, communication per signer, public key and signature sizes, and precomputation storage. Benchmarks will use deterministic, seedable randomness and our chosen parameter sets.

4.2. Platform and comparisons

Our implementation will target the NIST baseline platform (single machine, 16 cores, 64 GB RAM). We will compare our approach against other lattice-based two-round threshold signature schemes, such as [EKT24; BKLMTT25]. We expect better computational efficiency as we avoid the need for zero-shares to randomize Shamir secret shares. The added cost of the Vandermonde secret sharing is mainly in the storage of several shares per party, but does not impact the signing latency.

5. Licensing, Patent Claims, and Funding

The core implementation is open source and licensed under the terms of the permissive Apache License 2.0. The initial implementation builds on [Lattigo](#) to abstract ring operations, which is licensed under the Apache License 2.0 as well.

We are not aware of any known patent applicable to this submission nor plan to submit for one.

Giacomo Borin is supported by *CryptonIs*, *SNSF Consolidator Grant 213766*, (<https://data.snf.ch/grants/grant/213766>).

Rafael del Pino, Thomas Espitau, Guilhem Niot and Thomas Prest are supported by the French National Research Agency (ANR), through the project RELATE (reference: ANR-25-CE39-4214-01).

References

- [BCdENP25] Giacomo Borin, Sofía Celi, Rafaël del Pino, Thomas Espitau, Guilhem Niot, and Thomas Prest. *Threshold Signatures Reloaded: ML-DSA and Enhanced Raccoon with Identifiable Aborts*. Cryptology ePrint Archive, Report 2025/1166. 2025. URL: <https://eprint.iacr.org/2025/1166>.
- [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: *CRYPTO 2022, Part IV*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13510. LNCS. Springer, Cham, August 2022, pp. 517–550. DOI: [10.1007/978-3-031-15985-5_18](https://crypto.iacr.org/2022/papers/538806_1_En_18_Chapter_OnlinePDF.pdf). URL: https://crypto.iacr.org/2022/papers/538806_1_En_18_Chapter_OnlinePDF.pdf.
- [BKLMTT25] Cecilia Boschini, Darya Kaviani, Russell W. F. Lai, Giulio Malavolta, Akira Takahashi, and Mehdi Tibouchi. “Ringtail: Practical Two-Round Threshold Signatures from Learning with Errors”. In: *2025 IEEE Symposium on Security and Privacy*. Ed. by Marina Blanton, William Enck, and Cristina Nita-Rotaru. IEEE Computer Society Press, May 2025, pp. 149–164. DOI: [10.1109/SP61157.2025.00070](https://doi.org/10.1109/SP61157.2025.00070). Also at ia.cr/2024/1113.
- [DDB95] Yvo Desmedt, Giovanni Di Crescenzo, and Mike Burmester. “Multiplicative Non-abelian Sharing Schemes and their Application to Threshold Cryptography”. In: *ASIACRYPT’94*. Ed. by Josef Pieprzyk and Reihaneh Safavi-Naini. Vol. 917. LNCS. Springer, Berlin, Heidelberg, November 1995, pp. 21–32. DOI: [10.1007/BFb0000421](https://doi.org/10.1007/BFb0000421).
- [EKT24] Thomas Espitau, Shuichi Katsumata, and Kaoru Takemure. “Two-Round Threshold Signature from Algebraic One-More Learning with Errors”. In: *CRYPTO 2024, Part VII*. Ed. by Leonid Reyzin and Douglas Stebila. Vol. 14926. LNCS. Springer, Cham, August 2024, pp. 387–424. DOI: [10.1007/978-3-031-68394-7_13](https://doi.org/10.1007/978-3-031-68394-7_13). Also at ia.cr/2024/496.
- [KG20] Chelsea Komlo and Ian Goldberg. “FROST: Flexible Round-Optimized Schnorr Threshold Signatures”. In: *SAC 2020*. Ed. by Orr Dunkelman, Michael J. Jacobson Jr., and Colin O’Flynn. Vol. 12804. LNCS. Springer, Cham, October 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.
- [PENP25] Rafael del Pino, Thomas Espitau, Guilhem Niot, and Thomas Prest. *Simple and Efficient Lattice Threshold Signatures with Identifiable Aborts*. Cryptology ePrint Archive, Paper 2025/871. 2025. URL: <https://eprint.iacr.org/2025/871>.
- [PKNRT25] Rafaël Del Pino, Shuichi Katsumata, Guilhem Niot, Michael Reichle, and Kaoru Takemure. “Unmasking TRaccoon: A Lattice-Based Threshold Signature with An Efficient Identifiable Abort Protocol”. In: *CRYPTO 2025, Part VI*. Ed. by Yael Tauman Kalai and Seny F. Kamara. Vol. 16005. LNCS. Springer, Cham, August 2025, pp. 423–456. DOI: [10.1007/978-3-032-01887-8_14](https://doi.org/10.1007/978-3-032-01887-8_14). Also at ia.cr/2025/849.

- [PKPR24] Rafaël del Pino, Shuichi Katsumata, Thomas Prest, and Mélissa Rossi. “Raccoon: A Masking-Friendly Signature Proven in the Probing Model”. In: *CRYPTO 2024, Part I*. Ed. by Leonid Reyzin and Douglas Stebila. Vol. 14920. LNCS. Springer, Cham, August 2024, pp. 409–444. DOI: [10.1007/978-3-031-68376-3_13](https://doi.org/10.1007/978-3-031-68376-3_13). Also at ia.cr/2024/1291.
- [ZT25] Chenzhi Zhu and Stefano Tessaro. “The Algebraic One-More MISIS Problem and Applications to Threshold Signatures”. In: *CRYPTO 2025, Part I*. Ed. by Yael Tauman Kalai and Seny F. Kamara. Vol. 16000. LNCS. Springer, Cham, August 2025, pp. 548–581. DOI: [10.1007/978-3-032-01855-7_18](https://doi.org/10.1007/978-3-032-01855-7_18). Also at ia.cr/2025/436.
- [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).