

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

Title: Vinaigrette

Subtitle: An MPC Approach for Threshold UOV and MAYO

Version: 1.0 (2026-01-21)¹

Team name: Vinaigrette Team

Team members: Diego F. Aranha, Ward Beullens, Giacomo Borin, Fabio Campos, Sofia Celi, Basil Hess, Matthias J. Kannwischer, Lisa Kohl, Guilhem Niot

Abstract: This document is a preview of our future submission of the *Vinaigrette* framework to the NIST Call for Multi-Party Threshold Schemes. *Vinaigrette* is a framework for threshold signature schemes that can be used on UOV and MAYO (as representatives of Oil-and-Vinegar Signature schemes), which are candidates to the NIST Call for Additional Digital Signature Schemes. Our framework is implemented via secure multiparty computation (MPC) in the dishonest-majority setting and achieves active security. The design supports a distributed key generation (DKG). We adopt an *offline/online* architecture: a preprocessing phase generates the correlated randomness needed for efficient secure multiplication and generation of message independent values, and the message-dependent online phase performs signing using lightweight, information-theoretic checks. The resulting scheme remains compatible with existing UOV and MAYO parameters, and their verification procedure.

Proposed crypto-systems: Vinaigrette (Threshold UOV and MAYO): Categories S1and S4

Keywords: Threshold Cryptography; NIST Threshold Call; UOV; MAYO



Logo designed by Sofia 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: Diego F. Aranha^{i1,a1}, Ward Beullens^{i2,a2}, Giacomo Borin^{i3,a2,a3}, Fabio Campos^{i4,a4}, Sofía Celi^{i5,a5,a6}, Basil Hess^{i6,a2}, Matthias J. Kannwischer^{i7,a7}, Lisa Kohl^{i8,a8}, Guilhem Niot^{i9,a9,a10}

Open Researcher and Contributor Identifiers (ORCID):

i1 (0000-0002-2457-0783); i2 (0000-0003-0888-283X); i3 (0009-0001-7311-3802); i4 (0000-0003-3912-7570); i5 (0000-0002-3333-7764); i6 (0009-0002-7177-4495); i7 (0000-0002-8215-4729); i8 (0009-0007-1870-2181); i9 (0000-0002-2497-8770)

Affiliations:

- ^{a1} Aarhus University @ Denmark
- ^{a2} IBM Research Zurich @ Switzerland
- ^{a3} University of Zurich @ Switzerland
- ^{a4} Darmstadt University of Applied Sciences @ Germany
- ^{a5} Brave Research @ Portugal
- ^{a6} University of Bristol @ UK
- ^{a7} Chelpis Quantum Corp @ Taiwan
- ^{a8} CWI Amsterdam @ Netherlands
- ^{a9} PQShield SAS @ France
- ^{a10} Univ Rennes, CNRS, IRISA @ France

Main contacts:

- **Team mailing list:** <contact@pqvinaigrette.<org>
- **Primary technical contact person:** Sofía Celi, cherenkov@riseup.net

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

We present *Vinaigrette*, a framework that provides threshold variants of post-quantum multivariate-based (MQ) signature schemes based on the Oil-and-Vinegar construction (OV-based), in particular UOV and MAYO (both strong candidates in the NIST Call for Additional Digital Signature Schemes). *Vinaigrette* realizes threshold signing via secure multiparty computation (MPC) in the dishonest-majority setting with active security, aiming for practical threshold signatures that preserve the existing public verification algorithms of UOV and MAYO. Our proposed framework falls within categories S1 and S4 (Key Generation and Signing for a non-NIST Standard, though, we note that OV-based schemes are currently part of the ongoing NIST process).

In terms of design, *Vinaigrette* starts with an unified description of UOV and MAYO, and from the threshold design for OV-based signatures from [CEN25], with concrete instantiations, and additional optimizations to minimize the number of secure multiparty multiplications (and their depth) and the cost of the online signing phase². We represent the signing computation as arithmetic over a finite field and concretely implement it with authenticated (additive or Shamir) secret sharing [RB89; CFOR12; HO18] and Beaver-matrix triples [CKRRSW20]. The main challenge of the OV-based signature procedure is the inversion of a private matrix representing a system of equations (which we refer to as solving a system of equations obliviously), which we perform efficiently and securely with a masking and reveal technique. Thanks to this procedure, we can safely distribute the computation of the UOV and MAYO signatures, as shown in [CEN25]. However, the scheme presented in [CEN25] is costly (it incurs in around 9 rounds); however, in our current ongoing research, we show that we can reduce the number of rounds to *only 1 online message-dependent round* in the non-active case, with a pre-processing phase. With these optimizations and as a ballpark estimate, threshold signing for MAYO requires approximately $\approx 2.82 \times 10^5$ secure multiplications on \mathbb{F}_{16} and $\approx 4.43 \times 10^5$ for UOV at security level 1.

As noted, the scheme is presented in the standard offline/online split: a preprocessing phase generates the correlated randomness needed for efficient secure multiplications as well as message-independent values, and the (single-round) online phase performs only local linear work (matrix additions) and openings. For flexible threshold profiles, the long-term secret key is stored in a secret-shared way so any subset of size at least t can engage in a signing session; at session start, those shares are converted into the online (authenticated) format.

2. Specification

2.1. Potential Organization

Part I: Foundations & Preliminaries of OV-based Systems. We introduce the theoretical and algebraic foundations that underpin the future specification. It includes:

- **Notation and Models:** common algebraic notation, the UC framework, and the underlying network and trust assumptions.

²The work on this optimization is currently under submission, but we will make public soon.

- **Unified OV Framework:** a cohesive presentation of multivariate quadratic (OV-based) signatures such as UOV and MAYO, highlighting their shared structure and algebraic properties, and where they differ.
- **Security Models:** baseline definitions and adversarial capabilities, focusing on the dishonest-majority setting under active security.

Part II: Core Building Blocks: Secret Sharing and Secure Primitives. The second part introduces the core cryptographic components:

- **Secret-Sharing Techniques:** additive and Shamir variants tailored for OV-based systems, optimized for compactness and composability.
- **Matrix Beaver Triples:** distributed protocols for the secure generation of matrix triplets as the base for linear operations over shared data.
- **Oblivious Linear Solving:** an efficient procedure for solving systems of linear equations over secret shares, forming the algebraic core of threshold signing for OV-based schemes.

Part III: Threshold Functionalities.

- **(D)KeyGen:** distributed or centralized key generation that generates per-signer secret shares and public material.
- **Threshold Signing:** offline precomputations (for matrix beavers and message-independent values, for instance) followed by a message-independent signing phase.
- **Verification and Aggregation:** deterministic aggregation of signature shares and final verification (the same as OV-based *single-party* signatures).

Part IV: Optimizations and Parameters. Algorithmic optimizations are treated separately, allowing efficiency improvements to be studied in a modular manner: this includes a deterministic version of the OV-based signature algorithm (for a fixed salt), and an MPC-friendly representation of OV-based systems. We will specify parameter sets as well.

Part V: Implementation, Evaluation, and Security. This part will include an implementation report and performance evaluation, together with a dedicated cryptanalysis section. Given the algebraic nature of OV-based systems, we will note structural attacks and design subtleties that may arise in the thresholdized setting, specially in regards to revealing values during the solving a system of equations obliviously procedure. Finally, we will formalize our security guarantees: proofs will be structured under composable definitions, emphasizing active security and robustness against dishonest-majority adversaries.

Independent Interest. Our techniques are of independent interest and apply broadly to all OV-based signature schemes (UOV, MAYO, SNOVA), as their core functionality builds on the same OV algorithm. We will discuss matrix Beaver triple generation procedures, which are relevant to any MPC scheme operating over shared matrices, such as secure linear algebra, privacy-preserving

machine learning, or verifiable computation involving matrix operations. Our protocol for secret-shared and oblivious solving of linear systems may be useful in distributed zero-knowledge proofs or privacy-preserving data analysis tasks requiring regression or dimensionality reduction over secret inputs.

2.2. System Model

In this section, we define the system model within which our *Vinaigrette* framework is designed and analysed. The model specifies the computational assumptions, network infrastructure, and threshold access structures required for our constructions. Since our focus is on threshold signatures, we adopt a (t, n) access structure, where any subset of at least t parties can jointly produce a valid signature. We emphasize composability and modularity by working in the Universal Composability (UC) framework, which ensures that each sub-protocol can be analysed in isolation and securely composed into the overall system.

2.2.1. High-level view of the scheme

For an integer $N > 0$, we note $[N] = \{0, \dots, N - 1\}$. Vectors are noted with small bold letters (i.e. \mathbf{x}), and we write (x_0, \dots, x_{n-1}) for the coordinates of $\mathbf{x} \in \mathbb{F}_q^n$. We denote by $\mathbb{F}_q^{m \times n}$ the set of (zero-indexed) matrices over \mathbb{F}_q with m rows and n columns.

Network model. We assume a synchronous network, as already assumed and required by modern threshold signature schemes [LN18; GG18; HO18]. We also consider that the adversary can observe any message sent, and choose to deliver or block messages at will on this network.

The UC Framework: Model and Functionalities. For our ideal functionalities, adversarial model, and proofs, our framework instantiations rely on the UC framework introduced by Canetti [Can01]. This framework allows functionalities to be implemented independently and then securely composed, and is based on *Probabilistic Polynomial-Time (PPT) Interactive Turing Machines (ITMs)* that are used to model parties, adversaries, and simulators.

Within the UC framework, we work directly with **secret-sharing-based** schemes, instantiated using either *additive* or *Shamir* secret sharing. We denote by $\llbracket x \rrbracket$ a secret-shared value of x , distributed among the participating parties. To achieve active security under a dishonest-majority setting, we employ *authenticated secret sharing* arithmetic operations on shared values following the standard paradigm: additions, subtractions, and multiplications by public constants can be performed locally, while multiplications of secrets require an interactive protocol.

2.2.2. Thresholdising OV-based schemes

Oil and Vinegar-based (OV-based) schemes (like UOV and MAYO) rely on a trapdoor function constructed using multivariate quadratic equations. Specifically, the function $P : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$ consists of m homogeneous quadratic polynomials in n variables over a small finite field \mathbb{F}_q . The assumption underlying these schemes is that finding preimages (solving the system of quadratic equations) is computationally hard. However, if one possesses additional structural information, known as the trapdoor, it becomes efficient to find preimages for any output. The trapdoor, or secret key, in these schemes is a linear subspace $\mathcal{O} \subset \mathbb{F}_q^n$ of dimension o , such that for all

vectors $\mathbf{o} \in \mathcal{O}$, the function P vanishes: $P(\mathbf{o}) = 0$. Knowledge of this trapdoor allows for the creation of the signatures by collapsing the procedure to solving a system of equations and using the “Hash-and-Sign” approach. In the case of OV-based schemes, this subspace \mathcal{O} is structured such that $o \approx m$, and the quadratic map P can be used to verify the validity of a signature for a message msg using the public key P .

The majority of operations on OV-based schemes can be easily thresholdised using generic MPC techniques: hence our *Vinaigrette* framework is straightforward in this regard, assuming a secret-sharing and triplets generation procedure. Note also that, due to this, our framework does not need a trusted setup. However, to enable our threshold signing, we require a core operation: solving a system of linear equations in which both the input matrix and the target vector remain secret (hence, solving a system of equations in an oblivious manner): $\mathbf{A} \cdot \mathbf{b} = \mathbf{x}$. We will provide in our specification the description of an enhanced version of that algorithm based in [CEN25].

The rest of the operations needed to thresholdise OV-based schemes (and realise the full *Vinaigrette* framework) are given in [CEN25]. There, we also describe a verification operation that can be thresholdised in case applications require it. In our full package, we will provide a complete description of each procedure: key generation (via Distributed Key Generation), signing, and verification (with its threshold version). At present, we achieve active security through the use of message authentication codes (MACs), though we are also exploring more advanced and potentially more efficient approaches.

2.3. Security

Our security analysis is formulated in the UC framework, comparing real-world executions of our protocols with ideal functionalities. The security goals are threefold: (i) correctness, (ii) unforgeability under chosen-message attacks, and (iii) resilience against leakage arising from retries and rank deficiencies. We consider an active adversarial model: dishonest participants are unable to forge signatures on their own, even if they actively deviate from the protocol.

Our assumptions build on the hardness of the underlying OV problems (as in UOV, MAYO, and related), the soundness of the secret-sharing, and the security of message authentication codes (MACs) used to protect against forgeries in the dishonest-majority setting. Leakage is explicitly modeled by a parameter, capturing the rank defects observed during the oblivious solving of linear systems: this leakage is formally bounded and conjectured not to weaken unforgeability.

Recall that UOV and MAYO follow the *Hash-and-Sign with Retry* paradigm: given a fixed hash digest and a choice of private coins, it may not be possible to invert the trapdoor. Concretely, this means that a private matrix depending on the private coins and the secret oil space may be rank-deficient. From a security point of view, the main difference between *Vinaigrette* and the centralized schemes (UOV, MAYO) lies in the leakage of a randomized function of the rank of this matrix, due to the distributed nature of our protocol. This leakage resembles the information given to an adversary in a side-channel scenario. In the specification, we prove that *Vinaigrette* UC-realizes the centralized signature schemes with this additional leakage, and we provide concrete bounds and parameter choices showing that security remains within the targeted levels.

3. Open-Source Implementation

1. **Code structure.** The core implementation is written in Go. The main modules include: `mpc/` (protocol logic), `model/` (types and parameters), `rand/` (PRNG helpers), `flags/` (CLI configuration), and `mock/` (test scaffolding), alongside `main.go` and `go.mod`. Compilation is done with the standard Go toolchain (`go build`, `go run`), with parameter sets selected via build tags. Benchmarking is supported through a command-line flag that repeatedly executes protocol runs. Dependencies are managed via Go modules: current code relies primarily on the standard library.
2. **Code progress and availability.** The code is openly available in a public Git repository at <https://github.com/AU-HC/mayo-threshold-go>. Development is active, with a working command-line interface supporting DKG, signing, and verification. The current version focuses on correctness and clarity (passive security), with active security explored in a separate branch (branch `active`) via MACs.
3. **Implementation of the networking model.** Networking functionalities (broadcast, reliable transmission) are currently mocked for local testing. The baseline platform intends to implement networking through Go's native concurrency and channels, with external libraries such as `gRPC` or `libp2p` as likely candidates for future extensions.
4. **Testing:** Our implementation provides unit-test for local operations (share generation, reconstruction, matrix algebra), but protocol-level testing must account for malicious parties and adverse network conditions. We plan tests that simulate inconsistent shares or dropped messages to verify correct abort/detection, and reproducible regression tests with fixed randomness to handle the scheme's probabilistic components.

We also plan to extend the implementation to C for broader performance testing.

4. Experimental Performance Evaluation

We provide a preliminary performance evaluation below, evaluating the performance of our proof-of-concept Go implementation of our framework. We report the computational times in fig. 1 for both UOV and MAYO at security level 1: all reported timings correspond to wall-clock time measured using Go's standard timing facilities. The reported timings correspond to the mean of 100 (local) signing successful attempts. All experiments were run for multiple combinations of (N, T) , although our scheme does not impose an *a priori* bound on these parameters. All experiments were conducted on macOS running on an Apple M3 machine with 24GB of memory. The implementation is sequential and does not exploit parallelism; in particular, we forced single-core execution by setting `GOMAXPROCS=1`.

We compare three variants of the threshold signing algorithm, and, here, we only show the numbers for the non-active version. The *Standard* variant corresponds to the baseline threshold signing algorithm from [CEN25], moving message-independent computations to a preprocessing phase. The *Explicit-Standard* variant goes further by reordering computations to extend the preprocessing strategy and move almost all computation to an offline phase, leaving only a lightweight online signing step. It further pre-samples a salt before knowing the message to be

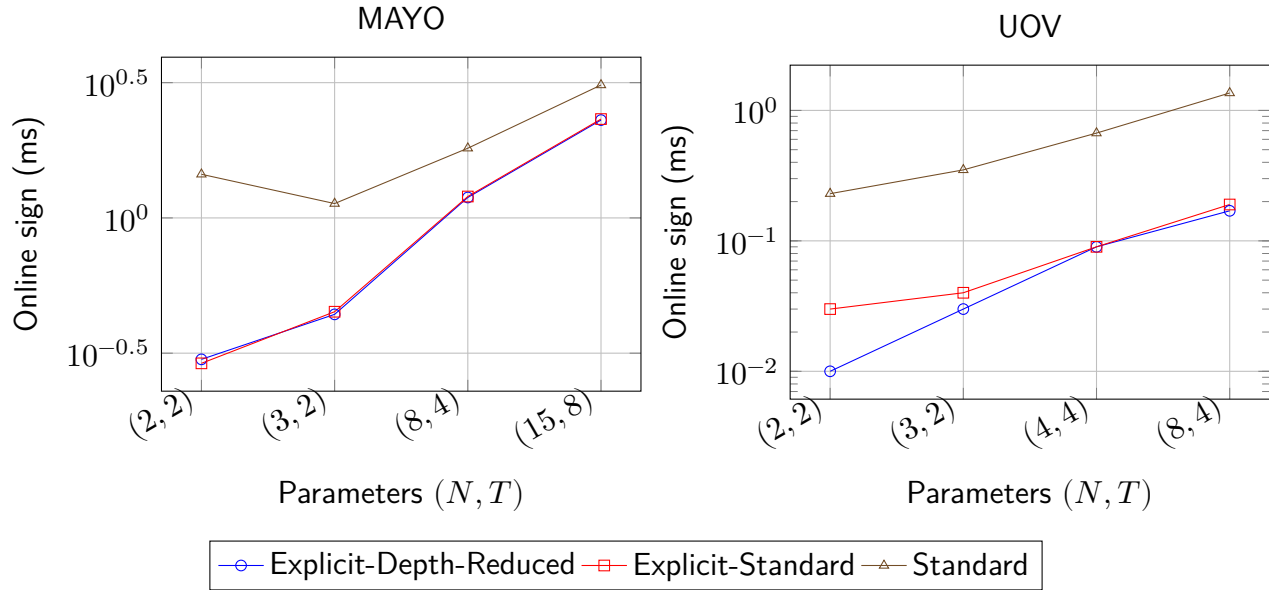


Figure 1: Mean online signing time at security level 1 for non-active across different parameter choices (N, T) , comparing the different variants. Left: MAYO1. Right: UOVIP.

signed, allowing to attain a single online round. Finally, the *Explicit-Depth-Reduced* variant is based on a new interpretation of OV-based signature algorithms that is specifically designed to reduce multiplication depth and efficient multi-party computation. These optimizations are part of ongoing work and will be described in full in a forthcoming submission, where they will be made public. As shown by our results, these optimizations substantially reduce online signing costs and yield an online phase consisting of a single message-dependent round. Finally, we note that the core computations in our scheme consist solely of matrix–matrix multiplications over finite fields. As a result, the implementation does not rely on platform-specific instructions and can be deployed across a wide range of architectures and execution environments [GG08].

5. Licensing, Patent Claims, and Funding

The core implementation is open source and licensed under the terms of the permissive Apache License 2.0. It does not have external dependencies. We are not aware of any known patent nor plan to submit for one.

Giacomo Borin is supported by *CryptonIs*, *SNSF Consolidator Grant 213766*, (<https://data.snf.ch/grants/grant/213766>). Lisa Kohl is supported by NWO Talent Programme Veni (VI.Veni.222.348) and by NWO Gravitation project QSC.

References

- [Can01] Ran Canetti. “Universally Composable Security: A New Paradigm for Cryptographic Protocols”. In: *42nd Annual Symposium on Foundations of Computer Science*. Las Vegas, NV, USA: IEEE Computer Society Press, October 2001, pp. 136–145. DOI: [10.1109/SFCS.2001.959888](https://doi.org/10.1109/SFCS.2001.959888). Also at ia.cr/2000/067.
- [CEN25] Sofía Celi, Daniel Escudero, and Guilhem Niot. “Share the MAYO: Thresholdizing MAYO”. In: *Post-Quantum Cryptography - 16th International Workshop, PQCrypto 2025, Part I*. Ed. by Ruben Niederhagen and Markku-Juhani O. Saarinen. Taipei, Taiwan: Springer, Cham, Switzerland, April 2025, pp. 165–198. DOI: [10.1007/978-3-031-86599-2_6](https://doi.org/10.1007/978-3-031-86599-2_6). Also at ia.cr/2024/1960.
- [CFOR12] Alfonso Cevallos, Serge Fehr, Rafail Ostrovsky, and Yuval Rabani. “Unconditionally-Secure Robust Secret Sharing with Compact Shares”. In: *Advances in Cryptology – EUROCRYPT 2012*. Ed. by David Pointcheval and Thomas Johansson. Vol. 7237. Lecture Notes in Computer Science. Cambridge, UK: Springer Berlin Heidelberg, Germany, April 2012, pp. 195–208. DOI: [10.1007/978-3-642-29011-4_13](https://doi.org/10.1007/978-3-642-29011-4_13). URL: <https://web.cs.ucla.edu/~rafail/PUBLIC/136.pdf>.
- [CKRRSW20] Hao Chen, Miran Kim, Ilya P. Razenshteyn, Dragos Rotaru, Yongsoo Song, and Sameer Wagh. “Maliciously Secure Matrix Multiplication with Applications to Private Deep Learning”. In: *Advances in Cryptology – ASIACRYPT 2020, Part III*. Ed. by Shiho Moriai and Huaxiong Wang. Vol. 12493. Lecture Notes in Computer Science. Daejeon, South Korea: Springer, Cham, Switzerland, December 2020, pp. 31–59. DOI: [10.1007/978-3-030-64840-4_2](https://doi.org/10.1007/978-3-030-64840-4_2). Also at ia.cr/2020/451.
- [GG08] Kazushige Goto and Robert A. van de Geijn. “Anatomy of high-performance matrix multiplication”. In: *ACM Trans. Math. Softw.* 34.3 (May 2008). DOI: [10.1145/1356052.1356053](https://doi.org/10.1145/1356052.1356053). URL: https://www.cs.utexas.edu/~flame/pubs/GotoTOMS_revision.pdf.
- [GG18] Rosario Gennaro and Steven Goldfeder. “Fast Multiparty Threshold ECDSA with Fast Trustless Setup”. In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. CCS '18. Toronto, Canada: Association for Computing Machinery, 2018, pp. 1179–1194. DOI: [10.1145/3243734.3243859](https://doi.org/10.1145/3243734.3243859). URL: <https://doi.org/10.1145/3243734.3243859>.
- [HO18] Brett Hemenway and Rafail Ostrovsky. “Efficient robust secret sharing from expander graphs”. In: *Cryptography Commun.* 10.1 (January 2018), pp. 79–99. DOI: [10.1007/s12095-017-0215-z](https://doi.org/10.1007/s12095-017-0215-z). URL: <https://doi.org/10.1007/s12095-017-0215-z>.
- [LN18] Yehuda Lindell and Ariel Nof. “Fast Secure Multiparty ECDSA with Practical Distributed Key Generation and Applications to Cryptocurrency Custody”. In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. CCS '18. Toronto, Canada: Association for Computing Machinery, 2018, pp. 1837–1854. DOI: [10.1145/3243734.3243788](https://doi.org/10.1145/3243734.3243788). URL: <https://doi.org/10.1145/3243734.3243788>.

- [RB89] Tal Rabin and Michael Ben-Or. “Verifiable Secret Sharing and Multiparty Protocols with Honest Majority (Extended Abstract)”. In: *21st Annual ACM Symposium on Theory of Computing*. Seattle, WA, USA: ACM Press, May 1989, pp. 73–85. DOI: [10.1145/73007.73014](https://doi.org/10.1145/73007.73014). URL: <https://www.cs.umd.edu/~gasarch/TOPICS/secretsharing/rabinVSS.pdf>.
- [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).