

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

## **Title:** PiVer: $\Pi$ Verifiable Secret Sharing Framework

**Subtitle:** A Unified Framework for Computationally Secure Verifiable Secret Sharing in the Synchronous Communication Model

**Version:** 0.1 (2026-01-20)<sup>1</sup>

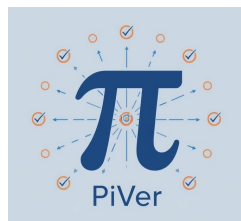
**Team name:** PiVer

**Team members:** Shahla Atapoor, Karim Baghery, Daniele Cozzo, Robin Jadoul, Hossein Moghaddas, Georgio Nicolas, Robi Pedersen, Mahdi Rahimi, Jannik Spiessens, Barry Van Leeuwen

**Abstract:** This proposal introduces *PiVer*, a unified framework for building computationally secure *Verifiable Secret Sharing* (VSS) schemes in the synchronous communication model. PiVer builds upon the  $\Pi$  framework, originally introduced as a general framework for computational VSS in the honest-majority setting. PiVer employs a random oracle and any secure commitment scheme satisfying only hiding and binding—optionally homomorphic, non malleable and/or post-quantum secure. The proposed PiVer framework unifies five major  $\Pi$ -based advances: the original  $\Pi$  VSS protocol (PiVer), a round-optimal variant (Piver-2R); a pre-constructed variant (PiVer-PC) in which the dealer also publishes a commitment to the shared secret; batched and packed extensions (PiVer-Batch) for improved efficiency; and a generalized variant that supports arbitrary  $\mathcal{Q}_2$  access structures (PiVer-Q2). These constructions together enable flexible and efficient design of computational VSS (and Publicly VSS) schemes, including post-quantum instantiations. PiVer provides both discrete-logarithm-based (PiVer-DL) and post-quantum-secure (PiVer-PQ) instantiations, supported by open-source implementations and concrete performance metrics that demonstrate practical viability. This preview submission aligns with the NIST Threshold Call by contributing a modular and extensible framework for threshold cryptography, promoting verifiability, efficiency, and post-quantum readiness in real-world synchronous systems.

**Proposed crypto-systems: PiVer:** A unified framework for constructing multiple variants of computationally secure Verifiable Secret Sharing (VSS) schemes (Category S7).

**Keywords:** Verifiable Secret Sharing; Synchronous Communication; Post-Quantum Security



(Logo AI-generated with Canva)

<sup>1</sup>Preliminary 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:** Shahla Atapoor<sup>i1,a1,†</sup>, Karim Baghery<sup>i2,a1,†</sup>, Daniele Cozzo<sup>i3,a3,†</sup>, Robin Jadoul<sup>i4,a4,‡</sup>, Hossein Moghaddas<sup>i5,a1,\*</sup>, Georgio Nicolas<sup>i6,a1,\*</sup>, Robi Pedersen<sup>i7,a2,†</sup>, Mahdi Rahimi<sup>i8,a1,\*</sup>, Jannik Spiessens<sup>i9,a1,\*</sup>, Barry Van Leeuwen<sup>i10,a1,†</sup>

### Open Researcher and Contributor Identifiers (ORCID):

i1 (0000-0002-6035-9520); i2 (0000-0001-7213-8496); i3 (0000-0001-5289-3769); i4 (0000-0002-5997-9992); i5 (0009-0000-3377-0632); i6 (0000-0002-3240-9009); i7 (0000-0001-5120-5709); i8 (0009-0003-0223-9082); i9 (0009-0005-4738-5758); i10 (0000-0002-3792-4042)

### Affiliations:

<sup>a1</sup> COSIC, KU Leuven, Leuven, Belgium

<sup>a2</sup> Technical University of Denmark, Kongens Lyngby, Denmark

<sup>a3</sup> IMDEA Software Institute, Madrid, Spain

<sup>a4</sup> 3MI Labs, Leuven, Belgium

### Associateship clarifications:

\* Ph.D. Student. † Postdoctoral Researcher. ‡ Ph.D. Researcher.

**Note on Author Contributions:** Authors are listed in alphabetical order. Karim Baghery serves as the corresponding author.

### Main contacts:

- **Team mailing list:** piver (at) esat (dot) kuleuven (dot) be
- **Primary technical contact person:** Karim Baghery, karim.baghery@kuleuven.be

**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

Verifiable Secret Sharing (VSS) is a fundamental building block of threshold cryptography and secure multi-party computation. It enables a dealer to distribute a secret among multiple parties such that only authorized subsets can reconstruct it, while providing verifiable guarantees that all distributed shares are consistent and correctly formed. This verifiability is crucial for building robust distributed protocols that remain secure in the presence of active and adaptive adversaries. Verifiable secret sharing underpins a wide range of cryptographic applications, including threshold decryption, threshold signatures, Distributed Key Generation (DKG), secure Multi-Party Computation (MPC), randomness beacons, and emerging decentralized and privacy-preserving systems. Consequently, efficient, flexible, and well-specified VSS constructions are essential for the practical deployment and standardization of threshold cryptography.

This submission proposes PiVer, the II Verifiable Secret Sharing Framework: a unified, modular, and extensible package for constructing, specifying, and evaluating computationally secure VSS in the synchronous communication model. The primary objective of PiVer is to provide a principled foundation for the systematic design of efficient VSS protocols, while supporting a wide range of security assumptions, access structures, and deployment settings, including batching and resistance against post-quantum (PQ) adversaries.

PiVer builds upon the II framework [ABCP23; Bag25], originally developed for Shamir-based computational VSS under the honest-majority assumption, and consolidates several significant extensions into a single coherent framework [BBKR25; ABJL25; ABNPS25; BM26]. By unifying these developments, PiVer offers a structured and implementation-oriented view of modern VSS design, emphasizing simplicity, modularity, and practical efficiency.

The framework encompasses the following core components:

- **PiVer**: The unified II framework [ABCP23; Bag25], enabling the construction of simple and practical Shamir-based VSS schemes in both classical and PQ settings;
- **PiVer-2R**: A round-optimal variant minimizing communication rounds and interaction, thereby reducing latency in synchronous deployments [BBKR25];
- **PiVer-PC**: A pre-constructed variant [BM26], in which the dealer additionally publishes a commitment of the shared secret, enabling a more practical, flexible, and versatile reconstruction procedure across higher-level protocols;
- **PiVer-Batch**: Batched and packed extensions that significantly improve amortized efficiency in multi-secret and high-throughput scenarios [ABNPS25];
- **PiVer-Q2**: A generalized formulation supporting arbitrary  $\mathcal{Q}_2$  access structures, extending applicability beyond threshold-only settings [ABJL25].

At its core, PiVer enables the systematic design of practical VSS protocols instantiated from any commitment scheme satisfying standard hiding and binding properties. The framework naturally accommodates both hash-based (non-homomorphic) and homomorphic constructions, and supports instantiations in classical as well as post-quantum settings. This abstraction allows protocol designers to decouple high-level VSS functionality from the choice of underlying cryptographic

assumptions, facilitating future upgrades and performance improvements without altering protocol correctness or security guarantees.

The proposed PiVer package will include formal protocol specifications, reference implementations, and comprehensive performance evaluations across representative instantiations and parameter sets. These artifacts are intended to directly support the goals of the NIST Threshold Cryptography Call by expanding the publicly available body of reference designs, implementation guidance, and comparative benchmarks for threshold primitives. Finally, PiVer demonstrates direct real-world relevance through its applicability to a broad class of decentralized and distributed systems, including threshold decryption, threshold signatures, distributed key generation, and etc. By providing a unified and extensible VSS framework, this submission aims to facilitate both standardization efforts and practical adoption of threshold cryptography.

## 2. Specification

This section specifies the scope, system model, security goals, and internal organization of the PiVer package. The schemes in PiVer, henceforth PiVer schemes, form a family of verifiable secret sharing protocols obtained by instantiating a unified framework under different assumptions, access structures, and efficiency requirements.

### 2.1. Foundations and System Model

The PiVer Verifiable Secret Sharing framework is founded on the notion of *Threshold Zero-Knowledge* (TZK) proofs, introduced by Boneh et al. [BBCG19]. TZK proofs enable soundness and zero-knowledge guarantees for statements over secret-shared data in the synchronous honest-majority setting. All protocols specified in this submission follow this system model, except for packed variants, which relax the honest-majority assumption as discussed later. Similar to the classic Feldman (and Pedersen) VSS schemes, we work in a synchronous network of  $n$  parties, of which up to  $t < n$  may be actively corrupted by a polynomial-time adversary. The dealer communicates with the parties over private authenticated channels, while all parties have access to a reliable broadcast channel. No trusted setup phase is required. In particular, running the protocol with a fresh set of parties only requires publishing new public keys and agreeing on an instantiation of the random oracle, making PiVer suitable for dynamic and real-world deployments.

### 2.2. PiVer: The Unified Framework

The PiVer framework builds on the II framework [ABCP23; Bag25] as a general construction for secure computational Verifiable Secret Sharing in the synchronous communication model, and ensures the following two core properties: (i) *Reconstructability*: any authorized subset of at least  $t + 1$  (honest) parties can reconstruct a unique secret; and (ii) *Unpredictability/ Simulatability*: an adversary corrupting up to  $t$  parties cannot predict the secret; in stronger variants, its entire view can be efficiently simulated, implying that it learns no information about the secret beyond

what is revealed by the public data. These properties are essential for threshold cryptographic applications such as DKG, threshold signatures, and threshold decryption.

A key feature of the II framework is its modularity: it can be instantiated with *any* secure commitment scheme satisfying standard hiding and binding properties—optionally homomorphic, non-malleable, and/or post-quantum secure. In particular:

- Discrete Logarithm (DL)-based commitment schemes yield efficient classical VSS constructions that achieve computational reconstructability together with either computational unpredictability or perfect simulatability. The latter corresponds to the simulatability guarantees provided by Pedersen-style commitments.
- Hash-based (non-homomorphic) commitments result in lightweight VSS schemes that can achieve computational reconstructability and simulatability against quantum adversaries.

The hash-based variant of PiVer provides post-quantum security, making it directly applicable to lattice- and isogeny-based threshold cryptography. Unlike classic homomorphic-commitment-based VSS constructions such as Feldman and Pedersen [Fel87; Ped91], PiVer can be instantiated with any commitment scheme. This flexibility enables lightweight, hash-based VSS instantiations while also permitting DL-based instantiations that, in general, attain superior asymptotic performance compared to traditional homomorphic-commitment-based approaches. Moreover, we show that the same design principles can be extended to *Publicly Verifiable Secret Sharing* (PVSS), where verification can be performed by any external party using public information alone.

### 2.3. PiVer-2R: Round-Optimal Variant of PiVer

PiVer also includes a round-optimal, two-round variant [BBKR25], which minimizes interaction while preserving security in both honest and adversarial settings. In contrast, the original PiVer framework requires only a single round in the honest case but may require up to three rounds in general adversarial scenarios. As shown in [BKP11], two rounds are optimal for computational VSS in the synchronous model, assuming the existence of a reliable broadcast channel between the dealer and the parties. Like the base framework, this variant remains compatible with any commitment scheme, supporting both classical and post-quantum instantiations, and is particularly well suited for scenarios where minimizing communication rounds is critical.

### 2.4. PiVer-PC: Pre-Constructed Variant of PiVer

PiVer also includes Pre-Constructed variants [BM26], which extend the original framework by requiring the dealer to broadcast an additional commitment to the shared secret. These variants not only make the resulting VSS schemes more versatile for protocol design but also enable a more efficient reconstruction procedure, referred to as optimistic reconstruction [BKNR25].

## 2.5. PiVer-Batch: Batched and Packed Variant of PiVer

PiVer also includes a Batched and Packed variant, referred to as PiVer-Batch [ABNPS25], which is designed for efficiently sharing and verifying multiple secrets simultaneously. In this variant, *batching* is parameterized by  $k$ , where the dealer shares  $k$  independent secrets using  $k$  independent polynomials while keeping the original number of parties  $n$  and corruption threshold  $t$  unchanged. *Packing* is parameterized by  $\ell$ , where  $\ell$  secrets are encoded into a single polynomial to reduce per-secret overhead. This comes at the cost of increasing the polynomial degree, which either requires more parties to reconstruct the secrets or reduces the maximum number of corrupted parties that can be tolerated. By combining batching and packing, PiVer-Batch achieves significant efficiency gains and allows the dealer to share  $k \times \ell$  secrets in a single protocol execution while maintaining standard verifiability guarantees and remaining compatible with any commitment scheme. These parameters are tunable, allowing designers to balance efficiency, robustness, and corruption tolerance, making the variant particularly suitable for DKG, threshold signatures, and secure multiparty computation.

## 2.6. PiVer-Q2: PiVer with Generalized Access Structures

Beyond threshold-only settings, PiVer supports general access structures satisfying the  $\mathcal{Q}_2$  condition [ABJL25]. Recall that an access structure is said to be  $\mathcal{Q}_2$  if no union of two unqualified sets covers the entire set of parties; equivalently, the complement of any unqualified set is necessarily qualified. This property can be viewed as a natural generalization of the honest-majority assumption in threshold schemes. Supporting  $\mathcal{Q}_2$  access structures enables fine-grained authorization policies in which participants may have different roles or levels of authority, more accurately reflecting practical deployment requirements than traditional threshold VSS. As a result, PiVer naturally supports a broad class of distributed protocols based on generalized access structures, including replicated and weighted secret sharing, in addition to standard threshold-based protocols.

The generalized framework preserves the reconstructability and unpredictability (or simulatability) guarantees of classical VSS while remaining commitment-agnostic. Unlike prior work, it is also compatible with post-quantum secure instantiations. This flexibility positions PiVer as a unifying reference framework for a wide range of real-world threshold protocols.

## 2.7. Package Organization and Security

**Organization:** The PiVer package will be organized modularly, with separate specification components for the base PiVer framework and its different variants. Core building blocks—such as commitment abstractions and random oracles—are specified independently to facilitate analysis and alternative instantiations.

**Security Goals and Assumptions:** All PiVer schemes aim to provide correctness, reconstructability, and unpredictability (or simulatability) against active adversaries under standard cryptographic assumptions. Security proofs rely on the security properties of the underlying commitment scheme,

and the Random Oracle model. Post-quantum security is achieved by selecting commitment schemes based on PQ-hard assumptions, without altering the high-level protocol design.

**Security Strength Estimation:** The specification will include parameter recommendations and performance benchmarks for representative classical and post-quantum instantiations. Security levels will be calibrated to align with standard symmetric-key security targets (e.g., AES-128 equivalence), following best practices for threshold cryptographic primitives.

### 3. Open-Source Implementation

We implemented our different schemes in pure Rust with minimal dependency on external crates. We include benchmarking code, end-to-end integration testing, and unit tests. At the moment, we assume reliable transmission as we run our tests under a virtual harness on one machine. However, our library has extensive error handling capabilities and would handle an unreliable network. We plan to add such tests in the future.

We will provide an open source implementation along with build instructions on:

- <https://www.esat.kuleuven.be/cosic/sites/nist-mptc-piver/implementations/>

This implementation will build on the current Rust implementation provided by [ABNPS25] and will provide implementations for the different instantiations. The implementation will allow both testing on the protocol level as well as benchmarking each employed low-level operation.

### 4. Experimental Performance Evaluation

The performance of our schemes depends on several factors, including the choice of commitment scheme, the number of participating parties, the number of secrets being shared, and the targeted security level. We report benchmark results for various DL-based and hash-based commitment schemes [ABNPS25] and for different numbers of random secrets sampled from a prime-order field of size 252 bits. The results, shown in Figure 1, correspond to an illustrative setting with 256 parties, a threshold of 127, and a target security level of 128 bits.

The benchmarks are taken from [ABNPS25] and were conducted on a MacBook Pro equipped with an M4 Pro CPU (up to 14 cores) and 24GB of RAM, with all measurements performed without parallelization. Results are reported for sharing 1, 10, 50, and 100 random secrets (field elements). We compare the basic scheme Piver (II) and its batched variant ( $BII$ ) against standard Shamir secret sharing. The VSS instantiations  $\mathbf{PiVer-DL} := (\Pi_P, \Pi_P^+, \Pi_F)$  as their batched variants are based on DL and serve as alternatives to the classic Pedersen and Feldman schemes [Bag25; Fel87; Ped91], while  $\mathbf{PiVer-PQ} := (\Pi_{LA}, BII_{LA})$  are hash-based constructions that can also be viewed as synchronous counterparts of the lightweight asynchronous VSS scheme of Shoup and Smart [SS24]. We observe that verification costs are negligible across all evaluated parameters, remaining below 1ms in all cases considered in Figure 1.

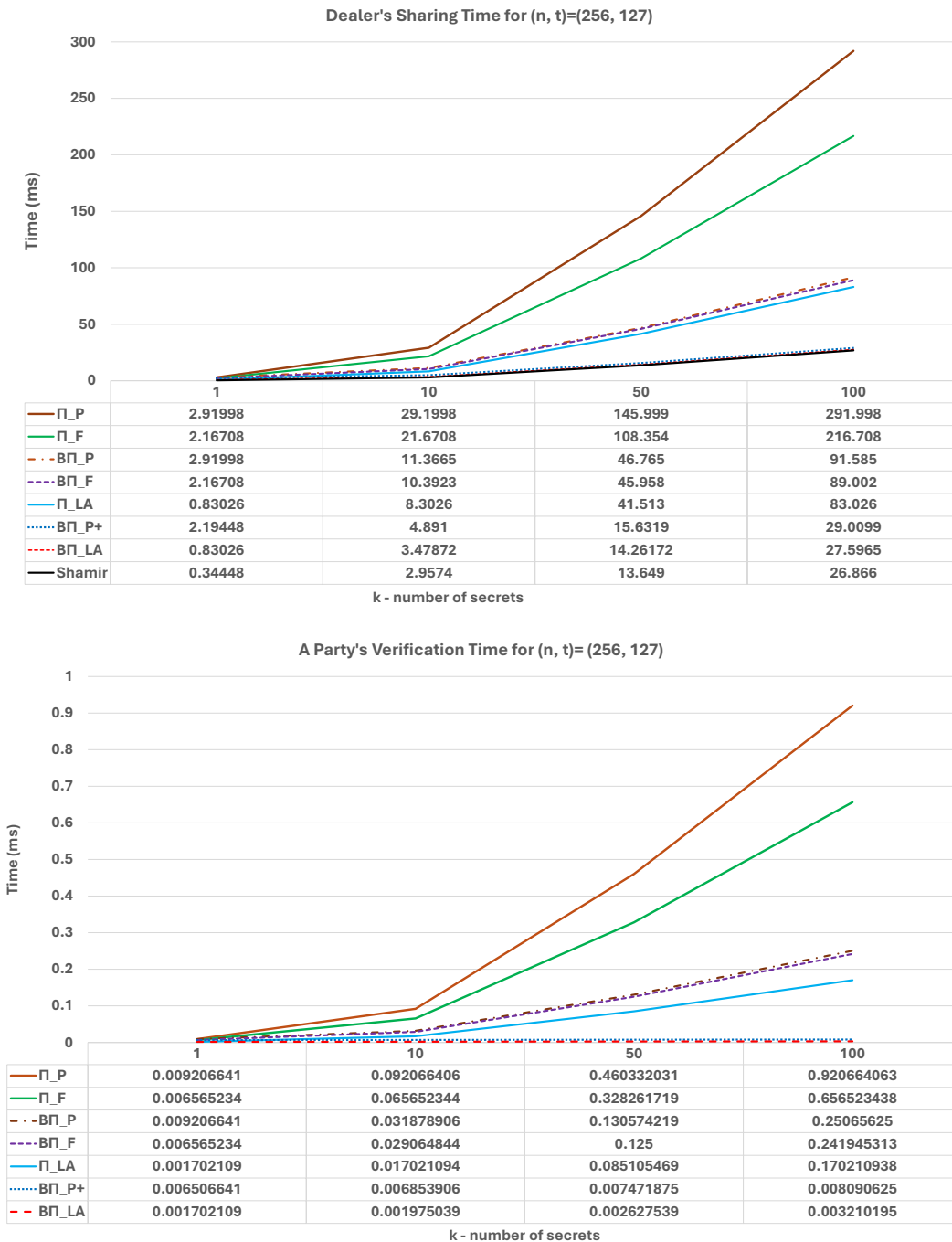


Figure 1: Sharing and verification times for the dealer and the parties across different instantiations of our VSS schemes with 256 parties and a threshold of 127.

## 5. Licensing, Patent Claims, and Funding

All source code will be released under open-source licenses compatible with those of eventual bundled dependencies. There is no known patent that would cover the contents of this submission. The works done in this submission were supported by the Flemish Government through the Cybersecurity Research Program with grant number VOEWICS02.

## References

- [ABCP23] Shahla Atapoor, Karim Baghery, Daniele Cozzo, and Robi Pedersen. “VSS from Distributed ZK Proofs and Applications”. In: *Advances in Cryptology - ASIACRYPT 2023 - 29th International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, December 4-8, 2023, Proceedings, Part I*. Ed. by Jian Guo and Ron Steinfeld. Vol. 14438. Lecture Notes in Computer Science. Springer, 2023, pp. 405–440. DOI: [10.1007/978-981-99-8721-4\\_13](https://doi.org/10.1007/978-981-99-8721-4_13). Also at [ia.cr/2023/992](https://ia.cr/2023/992).
- [ABJL25] Shahla Atapoor, Karim Baghery, Robin Jadoul, and Barry van Leeuwen. *On Computational VSS for General Access Structures*. Cryptology ePrint Archive, Paper 2025/2001. 2025. URL: <https://eprint.iacr.org/2025/2001>.
- [ABNPS25] Shahla Atapoor, Karim Baghery, Georgio Nicolas, Robi Pedersen, and Jannik Spiessens. *Batched and Packed (Publicly) Verifiable Secret Sharing: A Unified Framework and Applications*. Cryptology ePrint Archive, Paper 2025/2018. 2025. URL: <https://eprint.iacr.org/2025/2018>.
- [Bag25] Karim Baghery. “II: A Unified Framework for Computational Verifiable Secret Sharing”. In: *Public-Key Cryptography - PKC 2025 - 28th IACR International Conference on Practice and Theory of Public-Key Cryptography, Røros, Norway, May 12-15, 2025, Proceedings, Part IV*. Ed. by Tibor Jager and Jiaxin Pan. Vol. 15677. Lecture Notes in Computer Science. Springer, 2025, pp. 110–142. DOI: [10.1007/978-3-031-91829-2\\_4](https://doi.org/10.1007/978-3-031-91829-2_4). Also at [ia.cr/2023/1669](https://ia.cr/2023/1669).
- [BBCGI19] Dan Boneh, Elette Boyle, Henry Corrigan-Gibbs, Niv Gilboa, and Yuval Ishai. “Zero-Knowledge Proofs on Secret-Shared Data via Fully Linear PCPs”. In: *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part III*. Ed. by Alexandra Boldyreva and Daniele Micciancio. Vol. 11694. Lecture Notes in Computer Science. Springer, 2019, pp. 67–97. DOI: [10.1007/978-3-030-26954-8\\_3](https://doi.org/10.1007/978-3-030-26954-8_3). Also at [ia.cr/2019/188](https://ia.cr/2019/188).
- [BBKR25] Karim Baghery, Navid Ghaedi Bardeh, Shahram Khazaei, and Mahdi Rahimi. “On Round-Optimal Computational VSS”. In: *IACR Communications in Cryptology 2.2* (July 7, 2025). DOI: [10.62056/a0zo-4tw9](https://doi.org/10.62056/a0zo-4tw9). Also at [ia.cr/2025/1246](https://ia.cr/2025/1246).
- [BKNR25] Karim Baghery, Noah Knapen, Georgio Nicolas, and Mahdi Rahimi. “Pre-constructed Publicly Verifiable Secret Sharing and Applications”. In: *Applied Cryptography and Network Security - 23rd International Conference, ACNS 2025, Munich, Germany, June 23-26, 2025, Proceedings, Part I*. Ed. by Marc Fischlin and Veelasha Moonsamy. Vol. 15825. Lecture Notes in Computer Science. Springer, 2025, pp. 89–119. DOI: [10.1007/978-3-031-95761-1\\_4](https://doi.org/10.1007/978-3-031-95761-1_4). Also at [ia.cr/2025/576](https://ia.cr/2025/576).

- [BKP11] Michael Backes, Aniket Kate, and Arpita Patra. “Computational Verifiable Secret Sharing Revisited”. In: *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*. Ed. by Dong Hoon Lee and Xiaoyun Wang. Vol. 7073. Lecture Notes in Computer Science. Springer, 2011, pp. 590–609. DOI: [10.1007/978-3-642-25385-0\\_32](https://doi.org/10.1007/978-3-642-25385-0_32). Also at [ia.cr/2011/281](https://ia.cr/2011/281).
- [BM26] Karim Bagheri and Hossein Moghaddas. *Fully Secure DKG Protocols for Discrete Logarithm Revisited*. Cryptology ePrint Archive, Paper 2026/042. 2026. URL: <https://eprint.iacr.org/2026/042>.
- [Fel87] Paul Feldman. “A Practical Scheme for Non-interactive Verifiable Secret Sharing”. In: *28th Annual Symposium on Foundations of Computer Science, Los Angeles, California, USA, 27-29 October 1987*. IEEE Computer Society, 1987, pp. 427–437. DOI: [10.1109/SFCS.1987.4](https://doi.org/10.1109/SFCS.1987.4).
- [Ped91] Torben P. Pedersen. “Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing”. In: *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*. Ed. by Joan Feigenbaum. Vol. 576. Lecture Notes in Computer Science. Springer, 1991, pp. 129–140. DOI: [10.1007/3-540-46766-1\\_9](https://doi.org/10.1007/3-540-46766-1_9).
- [SS24] Victor Shoup and Nigel P. Smart. “Lightweight Asynchronous Verifiable Secret Sharing with Optimal Resilience”. In: *J. Cryptol.* 37.3 (2024), p. 27. DOI: [10.1007/S00145-024-09505-6](https://doi.org/10.1007/S00145-024-09505-6). Also at [ia.cr/2023/536](https://ia.cr/2023/536).
- [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).