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

Title: SmallWood

Subtitle: Hash-Based Zero-Knowledge Arguments for Relatively Small Instances

Version: 1.0 (2026-01-20)[1]

Team name: SmallWood Team

Team members: Thibauld Feneuil, Matthieu Rivain

Abstract: The proposed cryptographic system SmallWood is a hash-based zero-knowledge argument of knowledge, designed for efficiently proving statements with witnesses of small to medium size. While existing hash-based arguments such as STARK or Brakedown achieve excellent asymptotic performance for very large instances, and protocols such as VOLE-in-the-Head excel for tiny instances, SmallWood bridges the gap between these extremes. It efficiently handles proofs related to moderate-size statements, such as demonstrating knowledge of a private key, a digital signature, or a hash preimage.

Built entirely upon hash-based primitives, SmallWood provides post-quantum security, as it relies only on assumptions believed to resist quantum attacks. Although its precise application domain is still under exploration, SmallWood already shows promising integration potential within lattice-based cryptosystems and arithmetization-oriented hash constructions, making it a compelling candidate for future threshold cryptographic frameworks.

SmallWood was first introduced in a preprint released in early 2025, accompanied by preliminary proof-of-concept implementations. We plan to refine its design and optimize its implementation for practical use. These developments will be consolidated into a forthcoming submission package to the NIST Threshold Call.

Proposed crypto-systems: SmallWood: A ZKPoK (Category S6).

Keywords: Threshold Cryptography; NIST Threshold Call

(Logo AI-generated with ChatGPT-5)

[1]Version 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: Thibauld Feneuil [i1,a1], Matthieu Rivain [i2,a1]

Open Researcher and Contributor Identifiers (ORCID):

i1 (0000-0001-9342-2859); i2 (0000-0002-9855-4161)

Affiliations:

[a1] CryptoExperts @ Paris, France

Main contacts:

- **Team mailing list:** smallwood@cryptoexperts.com
- **Primary technical contact person:** Matthieu Rivain, matthieu.rivain@cryptoexperts.com
- **Secondary contact person:** Thibauld Feneuil, thibauld.feneuil@cryptoexperts.com

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

Zero-knowledge proofs (ZKPs) are a cornerstone of modern cryptography, enabling powerful privacy-preserving and verifiable computations. In the post-quantum era, hash-based ZKPs have emerged as a promising direction thanks to their conjectured resistance to quantum attacks, as well as their simplicity and efficiency.

On the one hand, MPC-in-the-Head (MPCitH)-based zero-knowledge arguments leveraging GGM seed trees, and in particular the VOLE-in-the-Head (VOLEitH) [BBDKORS23] and Threshold-Computation-in-the-Head (TCitH) [FR25c] frameworks, provide some of the smallest proof sizes for statements involving small witnesses. These protocols are particularly well-suited for proving knowledge of preimages of one-way functions — an archetypal example of compact-witness statements encountered in the design of post-quantum signature schemes.

On the other hand, most existing hash-based polynomial commitment schemes and (zero-knowledge) argument systems focus on achieving asymptotic succinctness (see, e.g., [BBHR18; BBHR19; BCRSVW19; GLSTW23; ACFY24; ZCF24]). In contrast, only a few proposals target the intermediate regime, providing compact proofs for moderately sized witnesses.[2] Notable examples include Ligero [AHIV17; AHIV23] and the Merkle-tree variant of the TCitH framework [FR25c]. Many practical proving applications in threshold cryptography fall within this intermediate regime — for example, proofs of knowledge of a secret key or a signature.

1.1. Overview of the Submission

This submission proposes SmallWood, a hash-based proof system for statements with relatively small witnesses, as presented in our preprint [FR25b]. The design of SmallWood draws inspiration from techniques used in Ligero [AHIV17; AHIV23], Brakedown [GLSTW23], and TCitH [FR25c].

The main features of the proposed scheme are:

- Small transcripts and competitive running times for moderately sized witnesses;
- Plausible quantum resistance, as it relies solely on assumptions believed to remain secure against quantum adversaries (e.g., cryptographic hash functions);
- Transparent setup, without any trusted initialization.

[2] To be specific, the relevant size is that of the extended witness in constraint-system-based representations (such as R1CS). For example, it may include the outputs of all multiplication gates of the arithmetic circuit representing the statement. In the intermediate regime considered here, this extended witness typically ranges from one kilobyte to several dozen kilobytes.

1.2. Relevance and Applications

This submission aligns with Category S6 of the NIST Call for Multi-Party Threshold Schemes. Although its precise application domain is still being explored, preliminary results indicate that SmallWood can efficiently produce proofs of knowledge of:

- Secret keys for ML-KEM (Category N2.2) and ML-DSA (Category N1.5);
- Secret preimages for arithmetization-oriented hash functions (Category S3.3).

The package aims to serve as a flexible tool for other projects related to threshold cryptography, including those submitted to the NIST Threshold call. Further applications — such as in Fully Homomorphic Encryption (FHE) — will also be explored within the project.

1.3. Current Status and Future Work

The planned package submission will be based on the article [FR25b], but design refinements are ongoing. Open-source proof-of-concept implementations of the SmallWood proof system already exist, and we plan to optimize them for practical use in future work.

2. Specification

The future specification of the project will be organized as follows:

- **High-level description:** A section presenting the overall design and intuition behind the scheme.
- **Low-level specifications:** A section providing a detailed technical description of the proof system, including algorithmic pseudo-code and implementation-level conventions.
- **Security analysis:** A section detailing the security properties of the scheme, including at least *soundness* and *zero-knowledge*.
- **Instantiations:** A section describing concrete instantiations of the proof system, including a methodology for selecting secure parameters. This section will also specify how to instantiate the symmetric primitives (PRG, XOF) used in the scheme, with the possibility of relying on *arithmetization-oriented hash functions* to enable recursive constructions.
- **Evaluation and comparison:** A section presenting performance evaluations of the proof system in various contexts, along with comparisons to the state of the art. The evaluation will cover both:
  - General statements, formalized as arithmetic circuits, R1CS constraints, etc.;
  - Specific use cases, including at least proofs of knowledge of secret keys for ML-DSA/ML-KEM, and proofs of knowledge of secret preimages of arithmetization-oriented hash functions.

3. Open-Source Implementation

Some open-source proof-of-concept implementations of the SmallWood proof system already exist:

- A Python3/SageMath implementation is available at

  https://github.com/CryptoExperts/smallwood-python.

- A C implementation is available at

  https://github.com/CryptoExperts/smallwood.

While the Python implementation is more flexible and suitable for experimentation, the C library is significantly more efficient. The goal of the submission package is to provide at least two implementations:

- a Python version, extending the existing code to make it more flexible and user-friendly;
- a C version (or Rust version), optimized for efficiency and targeting specific use cases.

Although the core functionalities are already implemented, the final library will incorporate the design improvements proposed during the project. Furthermore, while the current version of SmallWood naturally supports the Parallel and Aggregated Constraint Systems (PACS) formalism [FR25b], efforts will be made to implement efficient arithmetization interfaces from Rank-1 Constraint Systems (R1CS) and other constraint systems.

4. Experimental Performance Evaluation

SmallWood has already been used in two contexts. The first application consists in proving exact knowledge of the secret in Learning With Errors (LWE) instances [FR25b] and can be used to prove knowledge of secret key for ML-KEM and ML-DSA. The corresponding proof sizes are reported in Table 1. No implementation is currently available for this application, but preliminary results indicate that, for the first security level, the proving and verification times would be on the order of a few tens of milliseconds and a few milliseconds, respectively, when executed on a modern laptop using a non-optimized C implementation (using a single thread).

The second application consists in proving preimages of arithmetization-oriented hash functions [FR25a]. The corresponding performance results are presented in Table 2, assuming that we have to support efficient proof recursivity (*ie.* all the hash functions involved in the proof system are arithmetization-oriented ones).

We believe that SmallWood can be applied to many other use cases, although we do not have additional concrete performance results at this stage.

LWE Instance	LWE Parameters				SmallWood
	q	n	m	β	
ML-KEM-512	3329	2×256	2×256	3	14 115 B
ML-KEM-768	3329	3×256	3×256	2	15 004 B
ML-KEM-1024	3329	4×256	4×256	2	16 455 B
ML-DSA-44	8380417	4×256	4×256	2	17 514 B
ML-DSA-65	8380417	5×256	6×256	4	22 076 B
ML-DSA-87	8380417	7×256	8×256	2	22 700 B

Table 1: Proof sizes in bytes (B) obtained by SmallWood for proving knowledge of secret keys for ML-KEM and ML-DSA. The sizes obtained for ML-DSA assume that the vector t_0 that correspond to the public key is publicky known. q is the field size, n is the secret length, m is the noise length, and β is the infinity norm of each small coefficients.

Permutation Family	Trade-off	Proof Size	Proving Time	Verif. Time
Anemoi-5 (BN254's field)	Short	9 092 B	9900 ms	29 ms
	Default	11 486 B	2460 ms	29 ms
	Fast	12 337 B	710 ms	41 ms
Anemoi-7 (Goldilocks field)	Short	9 084 B	394 ms	1.2 ms
	Default	9 715 B	101 ms	1.4 ms
	Fast	10 729 B	29 ms	1.6 ms

Table 2: Proof sizes in bytes (B) obtained by SmallWood for proving knowledge of secret preimages of a single call of the arithmetization-oriented permutation Anemoi. The running times are obtained while all the hash functions involved in SmallWood are instantiated with Anemoi itself, to support *proof recursivity*. The benchmark has been performed on an AMD Ryzen Threadripper PRO 7995WX (using a single thread).

5. Licensing, Patent Claims, and Funding

The authors of the project do not hold, and do not intend to hold, any patent or patent application containing a claim — or that could be amended to include a claim — that may cover the cryptosystem or its implementations.

The content of the submitted package is intended to be open-source. To the best of their knowledge, the authors certify that there are no existing patents or patent applications that may cover the proposed cryptosystem or its source codes.

References

[ACFY24] Gal Arnon, Alessandro Chiesa, Giacomo Fenzi, and Eylon Yogev. "STIR: Reed-Solomon Proximity Testing with Fewer Queries". In: *CRYPTO 2024, Part X*. Ed. by Leonid Reyzin and Douglas Stebila. Vol. 14929. LNCS. Springer, Cham, August 2024, pp. 380–413. DOI: 10.1007/978-3-031-68403-6_12. Also at ia.cr/2024/390.

[AHIV17] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. "Ligero: Lightweight Sublinear Arguments Without a Trusted Setup". In: *ACM CCS 2017*. Ed. by Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu. ACM Press, October 2017, pp. 2087–2104. DOI: 10.1145/3133956.3134104. Also at ia.cr/2022/1608.

[AHIV23] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. "Ligero: lightweight sublinear arguments without a trusted setup". In: *DCC* 91.11 (2023), pp. 3379–3424. DOI: 10.1007/s10623-023-01222-8. Also at ia.cr/2022/1608.

[BBDKORS23] Carsten Baum, Lennart Braun, Cyprien Delpech de Saint Guilhem, Michael Klooß, Emmanuela Orsini, Lawrence Roy, and Peter Scholl. "Publicly Verifiable Zero-Knowledge and Post-Quantum Signatures from VOLE-in-the-Head". In: *CRYPTO 2023, Part V*. Ed. by Helena Handschuh and Anna Lysyanskaya. Vol. 14085. LNCS. Springer, Cham, August 2023, pp. 581–615. DOI: 10.1007/978-3-031-38554-4_19. Also at ia.cr/2023/996.

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. "Fast Reed-Solomon Interactive Oracle Proofs of Proximity". In: *ICALP 2018*. Ed. by Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella. Vol. 107. LIPIcs. Schloss Dagstuhl, July 2018, 14:1–14:17. DOI: 10.4230/LIPIcs.ICALP.2018.14.

[BBHR19] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. "Scalable Zero Knowledge with No Trusted Setup". In: *CRYPTO 2019, Part III*. Ed. by Alexandra Boldyreva and Daniele Micciancio. Vol. 11694. LNCS. Springer, Cham, August 2019, pp. 701–732. DOI: 10.1007/978-3-030-26954-8_23.

[BCRSVW19] Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P. Ward. "Aurora: Transparent Succinct Arguments for R1CS". In: *EUROCRYPT 2019, Part I*. Ed. by Yuval Ishai and Vincent Rijmen. Vol. 11476. LNCS. Springer, Cham, May 2019, pp. 103–128. DOI: 10.1007/978-3-030-17653-2_4. Also at ia.cr/2018/828.

[FR25a] Thibauld Feneuil and Matthieu Rivain. *CAPSS: A Framework for SNARK-Friendly Post-Quantum Signatures*. Cryptology ePrint Archive, Report 2025/061. 2025. URL: https://eprint.iacr.org/2025/061.

[FR25b] Thibauld Feneuil and Matthieu Rivain. *SmallWood: Hash-Based Polynomial Commitments and Zero-Knowledge Arguments for Relatively Small Instances*. Cryptology ePrint Archive, Report 2025/1085. 2025. URL: https://eprint.iacr.org/2025/1085.

[FR25c] Thibauld Feneuil and Matthieu Rivain. "Threshold Computation in the Head: Improved Framework for Post-Quantum Signatures and Zero-Knowledge Arguments". In: *Journal of Cryptology* 38.3 (July 2025), p. 28. DOI: 10.1007/s00145-025-09543-8. Also at ia.cr/2023/1573.

[GLSTW23] Alexander Golovnev, Jonathan Lee, Srinath T. V. Setty, Justin Thaler, and Riad S. Wahby. "Brakedown: Linear-Time and Field-Agnostic SNARKs for R1CS". In: *CRYPTO 2023, Part II*. Ed. by Helena Handschuh and Anna Lysyanskaya. Vol. 14082. LNCS. Springer, Cham, August 2023, pp. 193–226. DOI: 10.1007/978-3-031-38545-2_7. Also at ia.cr/2021/1043.

[ZCF24] Hadas Zeilberger, Binyi Chen, and Ben Fisch. "BaseFold: Efficient Field-Agnostic Polynomial Commitment Schemes from Foldable Codes". In: *CRYPTO 2024, Part X*. Ed. by Leonid Reyzin and Douglas Stebila. Vol. 14929. LNCS. Springer, Cham, August 2024, pp. 138–169. DOI: 10.1007/978-3-031-68403-6_5. Also at ia.cr/2023/1705.

[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.