Studying Lattice-Based Zero-Knowlege Proofs: A Tutorial and an Implementation of Lantern

Lena Heimberger, Florian Lugstein, Christian Rechberger

Graz University of Technology

1 Introduction

In a quest towards understanding and implementing zero-knowledge(ZK) proof systems, we are present a tutorial for Lantern [LNP22]. Introduced in 2022, Lantern proves knowledge of Module-Learning-With-Errors(MLWE) secrets and other lattice statements in a few kilobytes, which may be used in quantum-resistant schemes. Lantern's advantage over existing hash-based schemes like Aurora [BCR$^+$19] lies in its efficiency for proving lattice statements.

In our tutorial, we provide a comprehensive explanation of the underlying theory of Lantern and give a full implementation in a Jupyter Notebook using SageMath. To make the scheme more accessible, Lantern's more advanced protocols are split up into multiple smaller subprotocols. For every subprotocol, we provide an explanation of the protocol and an interactive implementation. In the implementation, the user can see all the necessary parameters clearly and change them to their liking. This feature allows users to fully understand the scheme and its building blocks, providing a valuable resource to understand both zero-knowledge proofs and lattice cryptography. The tutorial is available at `https://lattice-zk.iaik.tugraz.at/`.

As an additional result, we report the first implementation of Lantern in Sage. Constructing a Module-LWE proof takes around 35 seconds on a consumer laptop. While not optimized for efficiency, our implementation may serve as a benchmark for future implementations. The proof-of-concept implementation can be accessed at
`https://extgit.iaik.tugraz.at/public-shared/lattice-zk/-/raw/main/code/lattice-zk.py`.

2 Implementation Details and Benchmarks

We provide a general protocol for proving various lattice relations by combining all the techniques from the original paper. The proof toolbox implemented in the abdlop_toolbox function in our code can be instantiated to provide proofs for various schemes like verifiable encryption or group signatures. It allows a user to prove the following statements about a secret:

1. Quadratic relations over $\mathcal{R}_q$ with automorphisms
2. Quadratic relations over $\mathbb{Z}_q$ with automorphisms

3. Shortness in the infinity norm
4. Shortness in the Euclidean norm
5. Approximate Shortness

To prove knowledge of an MLWE secret, we only need the fourth statement. The abdlop_mlwe function implements only this functionality, while the function test_abdlop_mlwe sets up the required matrices and acts as an example for how to construct proofs for a specific application.

2.1 Benchmarks

We benchmarked our implementation on a consumer laptop by measuring cumulative prover and verifier time, and the resulting proof size for various protocols. The results can be found in Table 1.

Proof Type	Time	Size
commitment opening + 1 lin. rel. $\mathcal{R}_q$	390 ms	21.5 kB
commitment opening + 1 lin. rel. $\mathcal{R}_q$ + 1 lin. rel. $\mathbb{Z}_q$	410 ms	23 kB
commitment opening + 1 quad. rel. $\mathcal{R}_q$	510 ms	22 kB
commitment opening + 1 quad. rel. $\mathcal{R}_q$ + 1 quad. rel. $\mathbb{Z}_q$	830 ms	24 kB
Module-LWE secret	35 s	29 kB

Table 1. Lantern implementation benchmarks on a consumer laptop with a 4.1 GHz AMD Ryzen 7 PRO 4750U CPU running Linux 6.6, Python 3.11, and SageMath 10.2.

2.2 Runtime

There are several possible ways to make the protocol more efficient. Since our implementation focuses on verbosity and providing a reference, instead of on performance, both the polynomial arithmetic and the matrix multiplications are relatively slow, which in turn slows down the computation of the commitment matrices ($\mathbf{A}$ in the implementation and the paper). This is especially noticeable in the MLWE secret proof because the approximate shortness technique requires us to compute more than 256 of these matrices.

The authors of the previously shortest proof scheme [ENS20] applied many optimization techniques for lattice computation like AVX2 vectorization and fast polynomial multiplication to their scheme to bring their runtime down to 4ms. Many of these techniques can also be applied to Lantern.

In practice, the protocols are usually used in their non-interactive form by applying the Fiat-Shamir transformation [FFS88]. In that case, it is not necessary to repeat the computation of the expensive commitment matrices when the rejection sampling algorithm rejects. Hence, we measure the runtime of a successful protocol execution without repetitions to give a more accurate picture of the runtime.

2.3 Proof Size

We did not implement Dilithium compression [BG14,LDK+22], which is why our proof sizes exceed the 13 KB size given in the original paper.

2.4 Limitations of our Implementation

Since we use the built-in methods of SageMath, the random sampling procedures of the protocol are not cryptographically secure. In addition, the implementation is also not hardened to prevent side-channel attacks.

2.5 Practical Considerations when using Lantern

There remain some open problems with Lantern. First, the Gaussian sampling required by Lantern is difficult to implement efficiently while protecting against side-channel attacks [HPRR20]. Second, Lantern proofs are not succinct, meaning they grow linearly in the number of committed messages. This means Lantern is not suited for proving large statements like circuit satisfiability. The hash-based alternatives for proving lattice statements are asymptotically succinct but impractical because they are too slow and require infeasible amounts of memory for more advanced proofs [BCOS20].

References

BCOS20. Cecilia Boschini, Jan Camenisch, Max Ovsiankin, and Nicholas Spooner. Efficient post-quantum SNARKs for RSIS and RLWE and their applications to privacy. In Jintai Ding and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020*, pages 247–267. Springer, Heidelberg, 2020.

BCR+19. Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P. Ward. Aurora: Transparent succinct arguments for R1CS. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part I*, volume 11476 of *LNCS*, pages 103–128. Springer, Heidelberg, May 2019.

BG14. Shi Bai and Steven D. Galbraith. An improved compression technique for signatures based on learning with errors. In Josh Benaloh, editor, *CT-RSA 2014*, volume 8366 of *LNCS*, pages 28–47. Springer, Heidelberg, February 2014.

ENS20. Muhammed F. Esgin, Ngoc Khanh Nguyen, and Gregor Seiler. Practical exact proofs from lattices: New techniques to exploit fully-splitting rings. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 259–288. Springer, Heidelberg, December 2020.

FFS88. Uriel Feige, Amos Fiat, and Adi Shamir. Zero-knowledge proofs of identity. *Journal of Cryptology*, 1(2):77–94, June 1988.

HPRR20. James Howe, Thomas Prest, Thomas Ricosset, and Mélissa Rossi. Isochronous gaussian sampling: From inception to implementation. In Jintai Ding and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020*, pages 53–71. Springer, Heidelberg, 2020.

LDK$^+$22. Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Peter Schwabe, Gregor Seiler, Damien Stehlé, and Shi Bai. CRYSTALS-DILITHIUM. Technical report, National Institute of Standards and Technology, 2022. available at `https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022`.

LNP22. Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 71–101. Springer, Heidelberg, August 2022.