

Zero-Knowledge Shuffle Improvement in Ethereum Single Secret Leader Election

Anders Malta Jakobsen*, Oliver Holmggaard†

Abstract—This is the abstract Zero-Knowledge Proof (ZKP) [1].

Index Terms—Ethereum, Proof of Shuffle, Distributed Systems, Inner Product Arguments, Zero-Knowledge Proof

1 INTRODUCTION

This is the introduction

Related Work

This is related work

2 RELATED WORK

2.1 Whisk

Ethereum currently has an improvement proposal suggesting the implementation of a protocol called Whisk [2]. Whisk is a zero-knowledge Single secret leader election (SSLE) system that Through a zero-knowledge argument called curdleproofs [3] allows for the verification of the correctness of a shuffle without revealing the input or output. It is based on the concept of inner product arguments and does not require a honest setup. It uses elliptic curve cryptography based on the BLS12-381 curve to achieve its goals.

Whisk is designed to be efficient and scalable, making it suitable for use in Ethereum and other blockchain systems. Whisk is one of the suggested solutions to attacks on the Ethereum network targeting block proposers. With the help of the zero-knowledge proofs, Whisk helps making the previous public proposer list and order into a system where only the proposer can see if it is their turn to propose a block, and they can proof that to be the case.

2.2 Shuffling algorithm

The shuffling algorithm used in curdleproofs has gone through many iterations and improvements in order to increase speed and reduce the size the proof. This is because the proposer has a limited amount of time to propose a block in each slot, and the addition of the proof to the protocol

increases the size of the block the proposers have to create. This is the reason why the current implementation of curdleproofs has chosen the shuffling algorithm [4] proposed by Larsen et al.

The way the shuffle works is by selecting 2 days' worth of proposers, and then shuffling the proposers over one day's worth of slots to create a new list of proposers for the following day. In each slot a subset of the proposers are shuffled, and the rest are left unchanged.

Though experiments Larsen et al. has shown that after enough shuffles becomes secure even in adversarial environments. They also suggests that their may be room to lower the size of the subsets chosen in each lot without losing the security of the shuffle. Thereby increasing the speed of the shuffle and reducing the size of the proof being added to the blockchain.

2.3 Bulletproofs

A big inspiration for the curdleproofs protocol is the use of bulletproofs [5]. Bulletproofs is a type of range proof that uses inner product arguments to prove that a committed value is within a certain range without revealing the value itself. Bulletproofs is in itself not a zero-knowledge proof system, but with the help of Fiat Shamir [5] it can be used to create a zero-knowledge proof. Bulletproofs also has had a few iterations and improvements to increase the speed and reduce the size of the proof since it was used in curdleproofs. One of these is Bulletproofs+ [6] which is a new version of bulletproof that uses a weighted inner product argument instead of the standard inner product argument to achieve a better performance. Bulletproofs+ is also different because it is zero-knowledge proof by itself unlike the original bulletproofs. A third version of the bulletproofs is Bulletproofs++ [7] which is a even newer version of bulletproofs that uses a new type of argument called the norm argument to achieve a better performance. Unlike the two other proofs Bulletproofs++ is a binary range proof, which means that even if it is the fastest proof it is not suitable for the curdleproofs protocol due to the binary nature of the bulletproofs++.

• All authors are affiliated with the Dept. of Computer Science, Aalborg University, Aalborg, Denmark
 • E-mails: *amja23, †oholmg20@student.aau.dk

3 BACKGROUND

3.1 Notation

3.2 Zero-knowledge proofs

Curdleproofs is a zero-knowledge proof system, which means that it allows a prover to convince a verifier that they know a secret without revealing the secret itself. within the context of Ethereum it could be the ability to convince someone that a transaction is valid without revealing information about the transaction such as the value of it.

3.3 Springproofs

Springproofs [8] is an inner product argument that aims to allow a more flexible and efficient way of creating zero-knowledge proofs by avoiding the need for padding when working with inputs that are not of the size of power of 2.

Currently, the way to work with inner product arguments is to either only work with input sets that have the size of a power of 2, or to pad the input to the size of the next power of 2. This leads to either forcing the prover to work with regied sizes of input sets, or to pad the input with zeros slowing down the process and forcing the prover to work with larger sets than necessary.

Springproofs is a new type of inner product argument that allows for the use of arbitrary sized input sets without the need for padding.

4 APPROACH

4.1 Shuffle security

The shuffle method proposed by Larsen et al. [4] that was used in curdleproofs is based on the idea of shuffling a list of proposers over a set of slots. The shuffle itself however is not too complex. A formal definition of the shuffle is given in section 4.1.

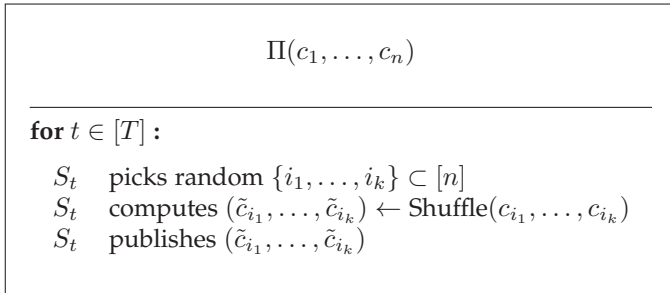


Fig. 1. Distributed shuffling protocol.

Here the set (c_1, \dots, c_n) is a set of ciphertexts that are shuffled over T slots.

4.2 Springproofs

4.3 implementation

5 EXPERIMENTAL PROTOCOL

This is the experiment.

6 RESULTS

These are the results

7 DISCUSSION

This is the discussion

8 CONCLUSION

This is the conclusion

9 FUTURE WORK

This is the future work.

10 ACKNOWLEDGEMENTS

We want to express our sincere gratitude to Daniele Dell'Aglia and Michele Albano for their supervision and guidance throughout this thesis.

We also acknowledge the usage of AI tools such as ChatGPT, GitHub Copilot, and Grammarly. These have been used for clarification and implementation purposes.

REFERENCES

- [1] G. D. Greenwade, "The Comprehensive Tex Archive Network (CTAN)," *TUGBoat*, vol. 14, no. 3, pp. 342–351, 1993.
- [2] G. Kadianakis, "Whisk: A practical shuffle-based ssle protocol for ethereum," 2024, Accessed: 22-10-2024.
- [3] T. E. F. C. R. Team, "Curdleproofs," Accessed: 24-04-2025.
- [4] K. G. Larsen, M. Obremski, and M. Simkin, *Distributed shuffling in adversarial environments*, Cryptology ePrint Archive, Paper 2022/560, 2022.
- [5] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, "Bulletproofs: Short proofs for confidential transactions and more," in *2018 IEEE symposium on security and privacy (SP)*, IEEE, 2018, pp. 315–334.
- [6] H. Chung, K. Han, C. Ju, M. Kim, and J. H. Seo, "Bulletproofs+: Shorter proofs for a privacy-enhanced distributed ledger," *Ieee Access*, vol. 10, pp. 42 081–42 096, 2022.
- [7] L. Eagen, S. Kanjalkar, T. Ruffing, and J. Nick, "Bulletproofs++: Next generation confidential transactions via reciprocal set membership arguments," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2024, pp. 249–279.
- [8] J. Zhang, M. Su, X. Liu, and G. Wang, "Springproofs: Efficient inner product arguments for vectors of arbitrary length," in *2024 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2024, pp. 3147–3164.

APPENDIX A

APPENDIX

This is the appendix