Excel@FIT 2025
http://excel.fit.vutbr.cz

Proof-of-Useful Work Consensus Protocol for Computing zk-SNARK Proofs

Bc. Richard Gazdík*

Abstract

The increasing demand for efficient and meaningful blockchain consensus protocols has highlighted the limitations of traditional Proof-of-Work (PoW), which consumes substantial computational resources without providing valuable outputs. This paper presents an alternative approach based on Proof-of-Useful-Work (PoUW) for computing zk-SNARK proofs while ensuring that the secret witness remains confidential. Building upon previous research, we design and implement a blockchain-based zero-knowledge proof marketplace that extends the PoUW framework to support private inputs. By integrating secure outsourcing mechanisms and cryptographic protections, the system enables the generation and verification of zk-SNARK proofs without revealing sensitive data. The proposed solution successfully incorporates privacy-preserving outsourcing into the decentralized environment with minimal overhead.

*xgazdi05@stud.fit.vut.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

The wide adoption of blockchain systems has intensified the focus on optimizing their consensus protocols for greater efficiency and meaningful utility. Traditional Proof-of-Work (PoW) systems, while effective at securing blockchains, waste vast amounts of computational resources on calculations with no intrinsic value. In contrast, Proof-of-Useful Work (PoUW) aims to align the security needs of blockchain with solving real-world problems, such as generating zero-knowledge proofs (zk-SNARKs). However, practical deployments of PoUW protocols are still in their infancy, especially when considering the critical privacy requirements inherent to zk-SNARK generation. Protecting the secret witness during outsourced computation is essential for enabling decentralized proof marketplaces without compromising user privacy.

The core problem addressed is the secure outsourcing of zk-SNARK proof generation in a decentralized PoUW blockchain while ensuring the confidentiality of the private witness. A proper solution must guarantee that the workers performing the proof generation cannot learn the sensitive witness. At the same time, it must maintain the key properties of PoUW: non-interactiveness (the client who requested the proof generation can go offline), open proving eligibility (any worker can participate), and minimal overhead compared to traditional zk-SNARK proving.

Previous research by Ing. Samuel Olekšák introduced a zk-SNARK proof marketplace that uses the PoUW consensus protocol [1], being one of the first to integrate zk-SNARK computations directly into the consensus layer of a blockchain. In contrast, other existing approaches, such as those based on Multi-Party Computation (MPC), Fully Homomorphic Encryption (FHE), or Trusted Execution Environments (TEE), primarily focus on outsourcing zk-SNARK generation at the application layer and are not tightly coupled with consensus protocols. MPC-based solutions [2, 3, 4, 5], while offering strong privacy guarantees, typically require significant interactivity and introduce trust assumptions incompatible with decentralized consensus. FHE-based approaches [6, 7], although offering strong theoretical privacy, suffer from extreme computational overhead, making them impractical for efficient generation of zk-SNARK. TEE-based designs are based on hardware trust and currently lack practical zk-SNARK provers adapted to secure enclaves. Recently, the scheme by Nakamura et al. proposed a lightweight and privacy-preserving outsourcing protocol suitable for zk-SNARKs with a trusted setup [8], providing strong confidentiality guarantees through randomized witness encryption with minimal computational overhead.

Our work proposes extending the zk-SNARK marketplace introduced by Olekšák to incorporate privacy-preserving outsourcing based on the scheme of Nakamura et al. Specifically, we design a system where the prover encrypts the witness using simple, efficient randomization techniques before submitting the task to the marketplace. The worker then computes the zk-SNARK proof on the transformed data, ensuring that the private inputs remain confidential.

The main contributions of this work are the integration and adaptation of Nakamura et al.'s encryption-based outsourcing scheme to the decentralized marketplace context. Our system advances the practical deployment of useful and privacy-respecting computations in decentralized blockchain environments.

2. Architecture

The proposed system integrates privacy-preserving zk-SNARK proof generation directly into the Proof-of-Useful-Work (PoUW) consensus protocol. It consists of four main components: provers, workers, setup nodes, and the blockchain network.

Provers create proof generation tasks by defining zk-SNARK circuits and encrypting sensitive private inputs (witnesses) using randomized parameters based on Nakamura et al.'s scheme. The encrypted witness, public inputs, and circuit metadata are packed into a ProofTransaction and submitted to the blockchain mempool.

Workers are decentralized nodes monitoring the mempool for pending ProofTransactions. Upon retrieving a task, a worker uses the public data and encrypted witness to generate a zk-SNARK proof without accessing the original inputs. Proof generation is performed using Circom and snarkJS, and the proof is submitted to the blockchain.

Setup nodes, selected through proof of stake, collaboratively generate trusted setup parameters for each circuit using a Multi-Party Computation (MPC) protocol. As long as at least one participant behaves honestly and discards their randomness, the trusted setup remains secure. After completion of the setup, the nodes must securely delete any sensitive intermediate data, especially the compiled circuit file (.r1cs).

The blockchain network manages the mempool, stores ProofTransactions, verifies zk-SNARK proofs, and ties successful proof generation to the eligibility for block production, embedding useful computation into the consensus process.

The system operates as follows:

- Setup nodes generate a trusted setup collaboratively via MPC for every circuit.
- Prover encrypts the private witness and submits a ProofTransaction.
- Worker retrieves the ProofTransaction and generates a zk-SNARK proof.
- Worker submits the proof to the blockchain.
- Blockchain verifies the proof and rewards the worker.

3. Conclusions

We successfully designed and implemented a Proof-of-Useful-Work consensus protocol for decentralized zk-SNARK proof generation while maintaining zero-knowledge and non-interactiveness from the prover's side. Our approach uses lightweight witness encryption, achieving almost no overhead compared to standard zk-SNARK workflows, and solving the privacy challenges that other heavier technologies introduce. I am proud that we managed to integrate useful computation directly into the consensus layer without sacrificing scalability or security.

4. Future Work

Future work will focus on optimizing proof generation time to further improve efficiency. We also plan to simplify the trusted setup process and reduce associated complexity. Additionally, the system will be extended to support a wider range of zk-SNARK circuits and proof systems.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Ing. Martin Perešíni, for his professional guidance, valuable feedback, and continuous support throughout the development of this work. His insights and encouragement were instrumental in shaping the direction and quality of the project.

I would also like to thank my consultant, Ing. Samuel Olekšák, for his expert advice, inspiring ideas, and helpful suggestions, which significantly contributed to the technical depth and refinement of the thesis.

References

[1] Samuel Olekšák. The analysis of cryptographic techniques for offloading computations and storage in blockchains. Master's thesis, Brno University of Technology, Faculty of Information Technology, 2024.

[2] Alex Ozdemir and Dan Boneh. Experimenting with collaborative zk-SNARKs: Zero-knowledge proofs for distributed secrets. Cryptology ePrint Archive, Paper 2021/1530, 2021.

[3] Alessandro Chiesa, Ryan Lehmkuhl, Pratyush Mishra, and Yinuo Zhang. Eos: Efficient private delegation of zkSNARK provers. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 6453–6469, Anaheim, CA, August 2023. USENIX Association.

[4] Sanjam Garg, Aarushi Goel, Abhishek Jain, Guru-Vamsi Policharla, and Sruthi Sekar. zkSaaS: Zero-knowledge SNARKs as a service. Cryptology ePrint Archive, Paper 2023/905, 2023.

[5] Yunbo Yang, Yuejia Cheng, Kailun Wang, Xiaoguo Li, Jianfei Sun, Jiachen Shen, Xiaolei Dong, Zhenfu Cao, Guomin Yang, and Robert H. Deng. Siniel: Distributed privacy-preserving zkSNARK. Cryptology ePrint Archive, Paper 2024/1803, 2024.

[6] Sanjam Garg, Aarushi Goel, and Mingyuan Wang. How to prove statements obliviously? Cryptology ePrint Archive, Paper 2023/1609, 2023.

[7] Mariana Gama, Emad Heydari Beni, Jiayi Kang, Jannik Spiessens, and Frederik Vercauteren. Blind zkSNARKs for private proof delegation and verifiable computation over encrypted data. Cryptology ePrint Archive, Paper 2024/1684, 2024.

[8] Masayuki Nakamura, Takashi Miyamae, and Masaru Morinaga. A privacy-preserving outsourcing scheme for zero-knowledge proof generation. *Journal of Information Processing*, 30:151–154, 2022.