Slovak University of Technology in Bratislava
Faculty of informatics and information technologies

Reg. No.:

Lukáš Častven

Enhancing Zero-knowledge proofs in blockchains

Masters's thesis

Thesis supervisor: Ing. Kristián Košťál PhD.

November 2024

Slovak University of Technology in Bratislava
Faculty of informatics and information technologies

Reg. No.:

Lukáš Častven

Enhancing Zero-knowledge proofs in blockchains

Masters’s thesis

Study programme: Intelligent software systems

Study field: Computer Science

Training workplace: Institute of Computer Engineering and Applied Informatics

Thesis supervisor: Ing. Kristián Košťál PhD.

November 2024

Table of contents

Table of contents

Chapter 1

Introduction

Zero Knowledge Proofs (ZKPs) are a cryptographical primitive, with which one party (the prover) can prove to another party (the verifier) that a given statement is true, without revealing any additional information beyond the validity of the statement itself. This property is used for maintaining privacy and security in various digital interactions, where sensitive information must be protected, or a proof of a valid computation is wanted. [1]

The foundation of what later became ZKPs, were introduced in the seminal work by Goldwasser, Micali, and Rackoff in 1985 [2]. They developed a computational complexity theory focusing on what does it mean to know, or possess some information. They described a different kind of proof, an interactive proof. This prove involves two parties (prover and verifier), which create a dialogue proving a truth of a statement. [3]

Main limitation of these proofs is that they require a constant communication between the prover and verifier. This limitation was addressed by Fiat and Shamir in 1986, who proposed a method to transform interactive proofs into non interactive ones using a random oracle [4]. This transformation removed the requirement for the prover and verifier to be simultaneously online, and hence there's no need for an interaction.

Nowadays, ZKPs' main application is scaling blockchain systems, like Ethereum [5]. ZK Rollups are layer 2 scaling solutions that leverage ZKPs to increase the throughput of blockchain while maintaining security and decentralization. They work by bundling multiple transactions offchain and generating a proof that verifies the validity of these transactions, which is verified on Ethereum. Another examples of ZK application are Coin mixers, for instance TornadoCash, or decentralized identity.

Chapter 2

Motivation

Despite their powerful capabilities, modern ZKP systems often face significant practical challenges. Proving systems today typically require high performance computing resources, sometimes even customized ASICs, to achieve reasonable proving times. This reliance on non consumer grage hardware not only raises the cost barrier for participation but also undermines the decentralization ethos central to many cryptographic applications.

The motivation behind this work is to explore avenues for enhancing ZKPs by reducing their hardware requirements. By slightly decreasing the computational demands of proof generation, this work aims to make ZKPs more accessible to individuals using common hardware, akin to how Ethereum staking allows validators to participate without specialized equipment. Lowering the entry barrier will support greater decentralization, enabling a wider community to engage in proof generation and verification processes.

In pursuit of this goal, the potential of utilizing smaller prime fields and binary fields in ZKP systems is investigate. By optimizing the underlying mathematical structures, we hope to enhance efficiency and data density, thereby reducing the need for specialized hardware. This approach could lead the way for more scalable and decentralized cryptographic solutions, benefiting a broad spectrum of applications and users.

Chapter 3

Analysis

This chapter, analyzes shift towards smaller prime fields in ZKP systems to enhance efficiency and data density. Section 3.1 discusses the shift from large fields to smaller ones, highlighting protocols like Plonky2, STWO, and Plonky3 that leverage smaller primes for improved performance. Section 3.2 delves into the use of binary field ($\mathbb{F}_2$), exploring their computational advantages and the protocol proposed by Diamond and Posen [6] that utilizes towers of binary fields. Aim of this chapter is to understand the potential benefits and challenges of adopting smaller and binary fields in ZKP proving systems.

3.1 Smaller prime fields

Today's ZK proving systems work over a large primary fields of bit size 2^{256}. However, the majority of programs use small numbers. Indices of arrays, variables with 64 bit size, or values representing single bit (true or false) use only a fraction of the whole 2^{256} field, thus creating an inefficiency and decreasing the information density.

Current trend and research directions tend towards using smaller prime fields. SNARKs over a elliptic curves become insecure when smaller prime fields are used. On the other hand, STARKs [7] use different approach based on hashing. This make it possible to reduce the size of the field. Plonky2 [8] started this by performing calculation over a 2^{64}, which improved the proof generation performance. Starkware's stwo [9] and Plonky3 [10] shrink the underlying field size further with usage of Mersenne prime $2^{31} - 1$.

3.2 Binary field

This tendency to shrink underlying field has a logical conclusion, a field over the smallest prime, 2. This field has a beatiful properties when computation is done in it. Addition is a bitwise XOR without the need to carry. Squaring elements is less expensive than multiplying two elements, due to the fact that in this field $(x+y)^2 = x^2+y^2$ (this property can be referred to as "Freshman's dream [11]).

Diamond and Posen in [6] propose a protocol constructed from binary field and binary tower of fields ($F_2 \subset F_{2^2} \subset F_{2^3} \ldots$) [12]. The binary field can be extended as many times as needed [13]. By using the binary field, data of size n will be encoded in n bits, and hence creating a dense encoding.

Multilinear polynomial is commited with a Merkle tree. In order to encode a polynomial representing large set of values, they need to be accessible as evaluations of the polynomial and used field must contain such values. So, the values (the trace of the compuatation) are encoded as points on hypercube $P(x_0, x_1, \ldots, x_k)$. Then to prove evaluations at random points, the data is interpreted as a square, extended with Reed-Solomon encoding. This gives the data redundancy for random Merkle tree queries, so that the evaluation is secure. And thanks to the binary field, the integers produced by extending with Reed-Solomon do not blow up.

The proposed protocol has a $\mathcal{O}(\sqrt{N})$ verification time and for a proof of 2^{32} bits around 11MB is needed.

References

[1] Oded Goldreich, Silvio Micali, and Avi Wigderson. "Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems". In: *Journal of the ACM (JACM)* 38.3 (1991), pp. 690–728.

[2] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. "The Knowledge Complexity of Interactive Proof Systems". In: *SIAM Journal on Computing* 18.1 (Feb. 1989), pp. 186–208. ISSN: 1095-7111. DOI: 10.1137/0218012. URL: http://dx.doi.org/10.1137/0218012.

[3] S Goldwasser and M Sipser. "Private coins versus public coins in interactive proof systems". In: *Proceedings of the eighteenth annual ACM symposium on Theory of computing - STOC '86*. STOC '86. ACM Press, 1986. DOI: 10.1145/12130.12137. URL: http://dx.doi.org/10.1145/12130.12137.

[4] Amos Fiat and Adi Shamir. "How To Prove Yourself: Practical Solutions to Identification and Signature Problems". In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, pp. 186–194. ISBN: 9783540180470. DOI: 10.1007/3-540-47721-7_12. URL: http://dx.doi.org/10.1007/3-540-47721-7_12.

[5] *Ethereum Virtual Machine (EVM) | ethereum.org — ethereum.org*. https://ethereum.org/en/developers/docs/evm/. [Accessed 16-05-2024].

[6] Benjamin E. Diamond and Jim Posen. *Succinct Arguments over Towers of Binary Fields*. Cryptology ePrint Archive, Paper 2023/1784. 2023. URL: https://eprint.iacr.org/2023/1784.

[7] Eli Ben-Sasson et al. *Scalable, transparent, and post-quantum secure computational integrity*. Cryptology ePrint Archive, Paper 2018/046. 2018. URL: https://eprint.iacr.org/2018/046.

[8] *plonky2/plonky2/plonky2.pdf at main · 0xPolygonZero/plonky2 — github.com*. https://github.com/0xPolygonZero/plonky2/blob/main/plonky2/plonky2.pdf. [Accessed 20-11-2024].

[9] Ulrich Haböck, David Levit, and Shahar Papini. *Circle STARKs*. Cryptology ePrint Archive, Paper 2024/278. 2024. URL: https://eprint.iacr.org/2024/278.

[10] *GitHub - Plonky3/Plonky3: A toolkit for polynomial IOPs (PIOPs) — github.com*. https://github.com/Plonky3/Plonky3. [Accessed 20-11-2024].

[11] Wikipedia. *Freshman's dream — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Freshman's%20dream&oldid=1252273320. [Online; accessed 20-November-2024]. 2024.

[12] Wikipedia. *Tower of fields — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Tower%20of%20fields&oldid=1232497196. [Online; accessed 20-November-2024]. 2024.

[13] Doug Wiedemann. *fq.math.ca*. https://www.fq.math.ca/Scanned/26-4/wiedemann.pdf. [Accessed 20-11-2024]. 1986.