AOHashes

AO privacy-friendly primitives testing over Dusk-Plonk ZK library

Filippo Merlo

March 2025

Contents

Chapter 1 Introduction ____ Page 2

Chapter 2 Primitives ____ Page 3

2.1 Implementation 3

Chapter 3 Tests ____ Page 4

3.1 Environment 4

3.2 Constraints 4

3.3 Plain execution 4

3.4 Proof generation 4

3.5 Proof verification 4

Chapter 4 Conclusion ____ Page 5

Chapter 1

Introduction

In few years, the interest in the cryptographic community has grown a lot towards the Zero-Knowledge (ZK) proofs. In fact, the goal of ZK proofs is to provide a way to prove the correctness of a computation without revealing any information about the data involved in the computation itself. Among the types of ZK proofs, there are the non-interactive ones, which are used in environments where prover and verifier are not able to communicate with each other, like in blockchains. In this paper, the ZK system used for the proofs is the PLONK system, which belongs to the SNARK[1] family, and has its own structure for the computation of the proofs based initially on arithmetic circuits, which are translated into constraints ones and finally evaluated as polynomials. This structure allows the use of more powerful gates, while maintaining the computational complexity low. However, having an internal structure like this means that old cryptographic primitives are not suitable and adapted for these computations, leading to huge losses in terms of performance and efficiency. Therefore, this strong interested in ZK has shaped the development of new cryptographic primitives towards the so-called Arithmetization-Oriented (AO) ones, aiming to provide better and more efficient solutions that could exploit as much as possible the power of ZK proofs, by maintaining a complexity as low as possible from a constraints' perspective (i.e. additions and multiplications). For this reason we wrote this paper with the aim of shading some light on the different performances of the latest and most promising AO primitives proposed in these last years.

[1] Succinct Non-interactive ARguments of Knowledge

Chapter 2

Primitives

The primitives that have been implemented and tested in this project are:

- `Poseidon`
- `GMiMC`
- `Rescue`
- `Rescue-Prime`
- `Griffin`
- `Anemoi`
- `Arion`

We can split them into two categories: in the *first category* we found those that have been designed for maintaining the **degree of polynomials** as **low** as possible, while having a high number of rounds to achieve a minimum level of bits security, which are `Poseidon`, `GMiMC`, `Rescue` and `Rescue-Prime`, while in the *second* one has been used the opposite strategy, i.e. achieving efficiency maintaining a **low number of rounds**, but increasing exponentially the polynomial degree with the introduction of multiplicative inverses, and these primitives are `Griffin`, `Anemoi` and `Arion`.

2.1 Implementation

Chapter 3

Tests

3.1 Environment

3.2 Constraints

3.3 Plain execution

3.4 Proof generation

3.5 Proof verification

Chapter 4

Conclusion