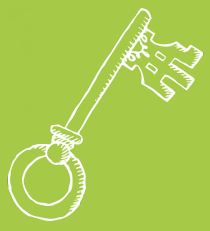


Implementation of AO Hash Functions

Advisor: **Katharina Koschatko**

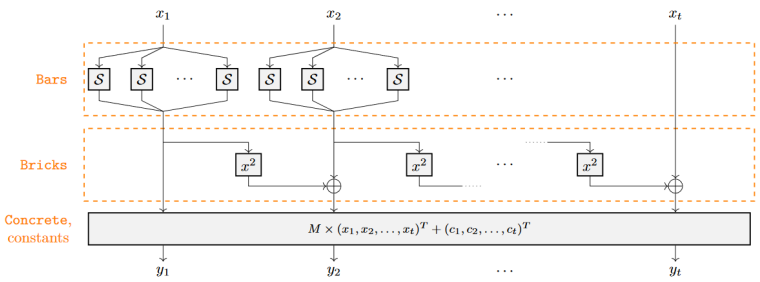
Motivation

Cryptographic hash functions play a vital role in numerous applications, ranging from data integrity verification and digital signatures to securing blockchain systems, cryptographic commitments, and zero-knowledge (ZK) proofs of knowledge. Besides pre-image and collision resistance for cryptographic security, a hash function must also be efficiently computable. More recent ZK-proof systems often additionally require that (1) the hash function works over large prime fields $\mathbb{F}_p$ and (2) can be efficiently integrated into the proof system. This new generation of hash functions is known under the name *arithmetization-oriented* (AO) or *arithmetization-friendly* hash functions.

The goal of this thesis is to implement one or more of these AO primitives in SageMath such that they can be used in a cryptanalysis framework.

Goals and Tasks

- Get familiar with finite fields and their operations.
- Understand the selected AO primitive(s).
- Implement the selected AO primitive(s) in SageMath.



`Monolith` round function [2].

Literature

- R. Walch
  What's the deal with hash functions in Zero Knowledge?
  https://blog.taceo.io/whats-the-deal-with-hashes-in-zk/
- L. Grassi et al.
  Monolith: Circuit-Friendly Hash Functions with New Nonlinear Layers for Fast and Constant-Time Implementations
  ToSC 2024
  https://doi.org/10.46586/tosc.v2024.i3.44-83

Courses & Deliverables

- ☑ **Introduction to Scientific Working**
  Short report on background
  Short presentation
- ☑ **Bachelor Project**
  Project code and documentation
- ☑ **Bachelor's Thesis**
  Project code
  Thesis
  Final presentation

Recommended if you're studying

☑ CS ☑ ICE ☑ SEM

Prerequisites

- Interest in the topic area
- Programming (SageMath)

Advisor Contact

katharina.koschatko@iaik.tugraz.at