CS291D Final Report: a Basic Zcash Implementation

Gwyneth Allwright, Karl Wang, Dewei Zeng

December 5, 2020

Abstract

In this project, we attempt a basic implementation of Zerocash [1] in Python. Zerocash is a ledger-based digital currency that makes use of zero-knowledge Succinct Non-Interactive Arguments of Knowledge (zk-SNARKs) to provide stronger privacy guarantees than currencies such as Bitcoin [2] and Zerocoin [3]. This functionality is provided through a decentralized anonymous payment (DAP) scheme that hides a transaction's origin, destination and amount. We follow [1] to implement the following core functions: `Setup`, `CreateAddress`, `Receive`, `Mint`, `VerifyTransaction` and `Pour`, which form the foundations of Zerocash.

Contents

1 Introduction

Data on blockchains such as Bitcoin is public, including the sender, receiver and the amount of money transferred in a payment. While Bitcoin users often utilize different identities to disguise their transactions, it is possible to gain access to both the structure of the transaction graph and the values and dates of transactions. Zerocoin, a cryptographic extension to Bitcoin, aims to introduce better privacy guarantees without requiring new trusted parties, but it still cannot hide the location that money is sent to, as well as the amount of money that is transferred [3]. In addition, it lacks some features of fully-fledged cryptocurrencies, such as payments of exact values.

In order to solve these problems with Bitcoin and Zerocoin, a new digital currency known as Zerocash was devised. Zerocash makes the sender, receiver and amount of money transferred in a payment anonymous, while also improving on the efficiency of Zerocoin [1]. These outcomes are achieved with the help of zk-SNARKs, which are efficient variants of zero-knowledge proofs of knowledge. Zero-knowledge proofs allow the prover of a certain statement to demonstrate that the statement in question is true without revealing additional information about the statement that could result in a compromise of privacy.

One of the primary objectives of this project is to explore zk-SNARKs and their potential applications in the world of blockchains and cryptocurrencies. To achieve this, we use existing zk-SNARK tooling to implement a minimal version of Zerocash in Python with the purpose of gaining a better understanding of Zerocash's theoretical underpinnings. This would be a first step towards demonstrating that zk-SNARKs are a feasible method of enhancing the privacy and performance of transactions on a simple blockchain. Next steps would include benchmarking and comparisons to a blockchain with similar functionality that does not make use of zk-SNARKs.

2 Problem Definition

We wish to understand how to incorporate zk-SNARKs into a basic blockchain in order to improve the blockchain's privacy guarantees. The setup of our scheme must not require any trust beyond a one-time trusted setup of public parameters. The implementation needs to support the minting, merging and splitting of coins without exposing the identities of the users who perform the transactions and the amounts of the currency involved. These objectives should be achieved through the use of zk-SNARKs.

3 Solution

As part of our DAP scheme, we implement the following core functions: `Setup`, `CreateAddress`, `Receive`, `Mint`, `VerifyTransaction` and `Pour`, which are described in [1]. In the sections that follow, we provide an overview of the DAP scheme and describe the important functions that form the the foundations of its implementation.

3.1 DAP Scheme

3.2 Important Functions

3.2.1 `Setup`

3.2.2 `CreateAddress`

3.2.3 `Receive`

3.2.4 `Mint`

3.2.5 `VerifyTransaction`

4 Related Work

5 Evaluation

References

[1] Eli Ben-Sasson et al. "Zerocash: Decentralized Anonymous Payments from Bitcoin". In: *IEEE Symposium on Security and Privacy* (2014), pp. 459–474.

[2] Satoshi Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System.* URL: `https://bitcoin.org/bitcoin.pdf`.

[3] Ian Miers et al. "Zerocoin: Anonymous Distributed E-Cash from Bitcoin". In: *IEEE Symposium on Security and Privacy* (2013), pp. 397–411.