COSC-373 Final Report: Bitcoin

Lee Jiaen, Hyery Yoo, and Alexander Lee

May 11, 2022

1 Introduction

2 Preliminaries

To understand Bitcoin, we first need to develop an understanding of the fundamental building-blocks behind it: digital signatures and cryptographic hash functions.

2.1 Digital Signatures

Suppose for example that Alice wants to send a message to Bob over a network. How can Bob ensure that the message he received was from Alice and not from a malicious actor like Eve? What Alice can do is append a *digital signature* to her message and send the message and signature to Bob, which allows Bob to verify that the message he received was indeed from Alice.

The main idea behind digital signatures is as follows. Both Alice and Bob each generate a *private key* and *public key* pair for themselves, where each key is a array of bits and private keys are kept secret (i.e., only Alice knows her private key and only Bob knows his private key). Producing a signature involves a function $\mathsf{sign}(\mathsf{msg}, \mathsf{priKey})$, which takes Alice's message msg and private key priKey as inputs and outputs Alice's signature sig. Similarly, verifying the authenticity of the received message involves a function $\mathsf{verify}(\mathsf{msg}, \mathsf{sig}, \mathsf{pubKey})$, which takes the received message msg, the signature sig, and Alice's public key pubKey as inputs and outputs *true* if the signature was produced using Alice's message and private key, and *false* otherwise. We will not discuss how the $\mathsf{sign}()$ and $\mathsf{verify}()$ functions work in depth since such details are not the focus of this work.

Digital signatures have two required properties. First, the authenticity of Alice's signature generated from her message and private key can be verified easily using her corresponding public key. Secondly, it should be computationally infeasible for someone like Eve to generate a valid signature for Alice without knowing Alice's private key. That is, Eve has no better strategy than guessing and checking if random signatures are valid using Alice's message and public key. Assuming a 256 bit signature, the aforementioned brute force strategy would require Eve to check 2^{256} signatures in the worst case.

2.2 Cryptographic Hash Functions

A *cryptographic hash function* is a hash function that takes as input data of an arbitrary size, known as the *message*, and outputs a bit array of a fixed size, known as the *hash*. In the case of the SHA256 cryptographic hash function, the hash has a length of 256 bits.

Cryptographic hash functions ideally should have the following properties. First, computing the hash of a given message should be quick. Second, generating the message from only the hash should be computationally infeasible. Third, finding two different messages that map to the same hash under the function should also be infeasible. Fourth, slightly changing the message should change the hash so much such that the old and new hashes seem uncorrelated.

3 Bitcoin

The *Bitcoin network* is a randomly connected overlay network of a few thousand nodes, where nodes are controlled by various owners and perform the same operation. In other words, the network is a homogeneous network without central control. Each user of Bitcoin generates a private/public key pair. A user is identified by their *address*, which is derived from their public key and used to receive funds in Bitcoin. All nodes in the Bitcoin network work together to track each address' balance in bitcoins.

4 Transactions

5 Doublespends

6 Proof-of-Work

7 Network

8 Conclusion