

COSC-373: Bitcoin

Lee Jiaen, Hyery Yoo, and Alexander Lee

May 9, 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/public key pair for themselves, where each key is a string 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 `sign(msg, 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 `verify(msg, sig, 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 `sign()` and `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

3 Bitcoin

4 Transactions

5 Doublespends

6 Proof-of-Work

7 Network

8 Conclusion