# Democracy Enhancing Technologies

Jeremy Clark

A Thesis in
The Concordia Institute for Information Systems Engineering (CIISE)

Presented in Partial Fulfillment of the Requirements
For the Degree of
Doctor of Philosophy
(Information and Systems Engineering)
at
Concordia University
Montréal, Québec, Canada Test

September 2023

# Concordia University

School of Graduate Studies

This is to certify that the thesis prepared

By: **Jeremy Clark**

Entitled: **Title**

and submitted in partial fulfillment of the requirements for the degree of

**Doctor of Philosophy (Information and Systems Engineering)**

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Chair
*Walter Lucia*

External Examiner
*Kaiwen Zhang (ETS)*

Examiner
*Amr Youssef*

Examiner
*M. Mannan*

Examiner
*Carol Fung*

Supervisor
*Jeremy Clark*

Approved by
*Zachary Patterson, Graduate Program Director (CIISE)*

01 Sept 2023
*Mourad Debbabi, Dean (GCS)*

# Abstract

Name: **Jeremy Clark**

Title: **Democracy Enhancing Technologies**

Hello. No more than 250 words.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Diam donec adipiscing tristique risus nec feugiat in fermentum posuere. Et netus et malesuada fames ac turpis. Nullam non nisi est sit. Felis eget velit aliquet sagittis id. Mauris commodo quis imperdiet massa tincidunt. Tellus molestie nunc non blandit massa enim nec. Facilisis mauris sit amet massa. Et molestie ac feugiat sed. Metus vulputate eu scelerisque felis imperdiet proin.

# Acknowledgments

Hello.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Diam donec adipiscing tristique risus nec feugiat in fermentum posuere. Et netus et malesuada fames ac turpis. Nullam non nisi est sit. Felis eget velit aliquet sagittis id. Mauris commodo quis imperdiet massa tincidunt. Tellus molestie nunc non blandit massa enim nec. Facilisis mauris sit amet massa. Et molestie ac feugiat sed. Metus vulputate eu scelerisque felis imperdiet proin.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Hello.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Diam donec adipiscing tristique risus nec feugiat in fermentum posuere. Et netus et malesuada fames ac turpis. Nullam non nisi est sit. Felis eget velit aliquet sagittis id. Mauris commodo quis imperdiet massa tincidunt. Tellus molestie nunc non blandit massa enim nec. Facilisis mauris sit amet massa. Et molestie ac feugiat sed. Metus vulputate eu scelerisque felis imperdiet proin.

Velit laoreet id donec ultrices tincidunt arcu non. Varius vel pharetra vel turpis nunc. Quis enim lobortis scelerisque fermentum dui faucibus in ornare quam. Lacus viverra vitae congue eu consequat ac felis donec. Consectetur adipiscing elit duis tristique sollicitudin nibh. Aliquam malesuada bibendum arcu vitae elementum curabitur vitae nunc. Sit amet venenatis urna cursus. Mi quis hendrerit dolor magna eget est lorem. Et ligula ullamcorper malesuada proin libero nunc consequat interdum varius. Id nibh tortor id aliquet lectus proin. Proin nibh nisl condimentum id venenatis a condimentum vitae sapien. Viverra aliquet eget sit amet tellus cras adipiscing. Bibendum ut tristique et egestas quis ipsum suspendisse ultrices gravida.

Mollis aliquam ut porttitor leo a. Nibh mauris cursus mattis molestie a iaculis at. Ultricies leo integer malesuada nunc vel. Feugiat nibh sed pulvinar proin gravida hendrerit lectus a. Eu facilisis sed odio morbi. Gravida arcu ac tortor dignissim. Nisl pretium fusce id velit ut tortor pretium viverra. Felis bibendum ut tristique et egestas. Turpis egestas pretium aenean pharetra magna ac placerat vestibulum lectus. Arcu non sodales neque sodales ut etiam sit amet nisl. Vulputate enim nulla aliquet porttitor lacus luctus accumsan tortor. Sed risus pretium quam vulputate dignissim suspendisse. Malesuada fames ac turpis egestas maecenas pharetra convallis posuere morbi. Aliquet lectus proin nibh nisl condimentum. Nibh sed pulvinar proin gravida. Quis vel eros donec ac odio tempor orci. Dignissim convallis aenean et tortor. Ac tincidunt vitae semper quis lectus nulla at volutpat. Vulputate odio ut enim blandit.

Tellus at urna condimentum mattis pellentesque id nibh tortor. Convallis posuere morbi leo urna molestie. Quis commodo odio aenean sed adipiscing diam donec. Iaculis eu non diam phasellus vestibulum. Tempus urna et pharetra pharetra massa massa ultricies mi. Faucibus turpis in eu mi bibendum neque egestas. Molestie ac feugiat sed lectus vestibulum mattis. Est ullamcorper eget nulla facilisi. In nibh mauris cursus mattis molestie a iaculis at. Velit ut tortor pretium viverra suspendisse potenti. Cum sociis natoque penatibus et magnis dis parturient montes. Orci nulla pellentesque dignissim enim sit amet. Dui nunc mattis enim ut.

Placerat orci nulla pellentesque dignissim enim sit amet venenatis. Feugiat in ante metus dictum at tempor. Interdum posuere lorem ipsum dolor sit amet consectetur adipiscing. Viverra vitae congue eu consequat ac felis donec et odio. Augue neque gravida in fermentum et sollicitudin ac. Justo eget magna fermentum iaculis eu non diam phasellus vestibulum. Imperdiet dui accumsan sit amet nulla. Sed id semper risus in hendrerit gravida rutrum quisque non. Turpis egestas pretium aenean pharetra. Pulvinar elementum integer enim neque. Amet purus gravida quis blandit turpis cursus in. In hac habitasse platea dictumst quisque sagittis purus. Amet nulla facilisi morbi tempus iaculis urna. Bibendum at varius vel pharetra vel turpis nunc. Consectetur lorem donec massa sapien. Mauris vitae ultricies leo integer malesuada.

**Scope.** Blah blah blah.

**Contributions.** Our primary contributions are as follows.

1. Blah blah blah.

2. Blah blah blah.

3. Blah blah blah.

# Chapter 2

# Background

This chapter introduces to bitcoin, the world of cryptocurrency and marketplaces. It describes as well what is a proof of solvency, and some of its evolution and current state.

## 2.1 Bitcoin

Bitcoin is recognized as the world's first successful cryptocurrency and decentralized digital currency. The goal of Bitcoin is to allow financial transactions to be settled without the need of a financial institution. Transactions can occur within 2 participants of the network in realtime, without any middleman. All transactions are settled on a public blockchain, which means that everything can be verified by everyone.

### 2.1.1 Transactions

For every participant of the network, there is a public key, a private key and a wallet address. The public key is derived from the private key using elliptic curve multiplication, and the wallet address is derived from the public key using a hashing function. Both are one way function, meaning you cannot derived the other way around. The wallet address can be seen as a bank account number. When you send bitcoin to someone, you send it to their wallet address. To be able to send some bitcoin, you need to sign your transaction. Since transactions are sent on the network, we need to make sure a transaction originates from the sender. The way to do that is to sign your transaction. The digital signature is created from the transaction data and the private key, which is only known by the owner of the address. The public key is then used to make sure that the signature originates from the right private key. Sending a transaction is the easiest problem to solve. The real challenge is to keep track of who owns what, and to avoid the double spending problem. The way to do that is to keep the history of every single transactions. Bitcoin is a blockchain. The blockchain is made of blocks, and the transactions are filling these blocks.

### 2.1.2 Network

The challenge of the network, is to have every single node agree on the transaction history. Nodes are computers connected to the network, working on publishing new blocks. The nodes work together to agree on the order of transactions. Every new transaction is broadcaseed to all nodes. The nodes puts the transactions into a block, and try to publish that block. In order to publish a block, each node need to solve a proof-of-work challenge. When a node solves the challenge, it broadcasts the block to every nodes. The nodes accepts the block if all transactions are valid. Their is no formal way of approving a new block. A node show its acceptance by starting to work on a new block using the hash of the accepted block as previous hash. Some nodes might accept different blocks, depending on what time they received new blocks. To solve the issue of multiple chains, the longest chain is consideredto be the correct one. If two chains have the same length, nodes keep working on their respective chains untill one of the chains receive a new block, breaking the tie.

### 2.1.3 Proof-of-work

In order to submit a new block, a node have to find a hash with x number of leading 0 bits. It is exponentially more difficult every time you add a zero. The way to have different hash values, is to change the block timestamp, and the nonce value. The nonce value is there solely for that purpose. Once a block is published, you cannot change any value inside of it because the hash value would change. You would need to redo all the work to find a new good hash. Older blocks are even more secured, because in order to change the 2nd to last block, you would need to redo the work for the 2 latest blocks. This is the same for every block down the road. The longest chanin is determined by the greatest proof-of-work invest in it. If their is a majority of honest nodes, that chain will grow up the fastest. The difficulty of the new block is determined by an average, in order to generate blocks at a steady pace.

### 2.1.4 Merkle Tree

Only the merkle root is stored in the block header. When a block is enough in the past, nodes start to only keep block headers in memory. They do not keep the rest of the block. This is why the hash of a block is the hash of the block header, and not the whole block. To keep the integrity of the chain. The merkle root is the top of the Merkle Tree. A merkle tree is a tree where the parent node is the hash of the child nodes.

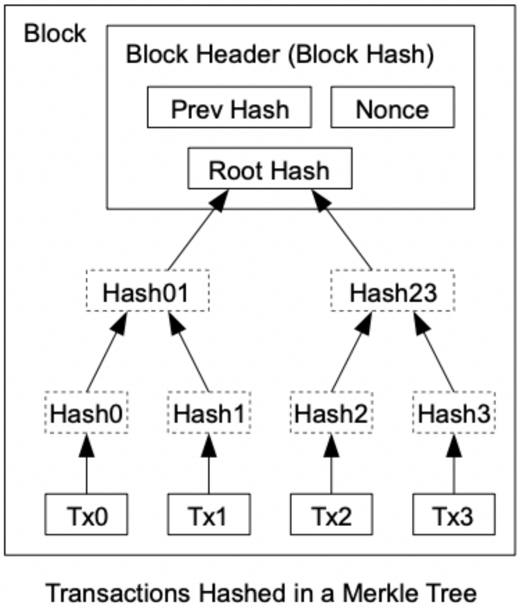


Figure 2.1: Bitcoin Merkle Tree [1]

## 2.2 Marketplaces

The best way to buy bitcoin for the first time is thourgh marketplaces. Marketplaces allow you to exchange money for bitcoin, but the bitcoin you acquire in the process is not linked to a wallet address you own. It is in custody of the marketplace. In order to have possession of the bitcoins, you need to enter a transfer request, where the bitcoin you own is sent to your wallet. Once you have bitcoins, you can transact on the network without needing a 3rd party, unless you are running a node, you will need to trust a 3rd party, wheter it is a marketplace, or over the counter.

While the bitcoin you own is in custody of the marketplace, there is no way to see your bitcoin onchain. Marketplaces have many wallets, some are made public and some are not. This is an inherent contradiction. Bitcoin was built to be transparent and public, without the need of a 3rd party to do a transaction. The problem with no being able to track your bitcoin in a marketplace, is that you have no proof that the marketplace has enough bitcoin to reimburse every client who has bitcoin in the marketplace. However, this problem is being actively worked on. Marketplaces have begin to use proof of solvency (or proof of reserve) to show that they are solvent, but there is still a long way to go [2].

## 2.3 Zero Knowledge

Zero knowledge proofs is a proof that something is known, without revealing any informations. For instance, the classic way of proving that you know the solution to an equation f(x), is to give the x that solves the equation. However, with zero knoledge you are able to prove that you know the solution, without giving x. To construct a zero knowledge proof, you need to construct a proof that is sound and compelte. You also need your proof to be zero knowledge[3].

- **Completeness**: If the statement is true, an honest verifier will be convinced by an honest prover.
- **Soundness**: If the statement is false, no dishonest prover can convince the honest verifier (except with some infinitysimal probability).
- **Zero-Knowledge**: If the statement is true, a verifier learns nothing other than the fact that the statement is true. [4]

### 2.3.1 Non interactive proofs

Zero knowledge proofs were originaly designed as interactive, that is multiple rounds of interaction between the prover and the verifier [5]. leading to what are called interactive zero-knowledge proofs. An alternative model was then proposed where the verifier and prover use a reference string that is shared during a trusted setup. Once we have the reference string, a single message is needed between the prover and the verifier. No rounds of interactions are needed. These are called noninteractive zero knowledge proofs. [6] [7]

### 2.3.2 SNARKS

One of the recent advancement for non-interactive proofs is what is known as SNARK (non-interactive argument of knowledge). This means a proof that is:

- **Succinct**: the size of the proof is very small compared to the size of the witness.

- **Non-interactive**: No rounds of interactions between the prover and the verifier.

- **Argument**: Secured only for provers with bounded computational ressources, that is a dishonest prover with unlimited computational power could prove a wrong statement.

- **Knowledge-sound**: If the statement is true, a verifier learns nothing other than the fact that the statement is true. [9]

A SNARK can also be zero-knowedge. We call such proof a zk-SNARK.

### 2.3.3 Arithmetic circuit

SNARKs use arithmetic circuits. An arithmetic circuit is a set gates, each assigned a distinct set of inputs corresponding to the numbers to be processed. These gates are configured to execute arithmetic operations such as addition, subtraction, multiplication, or division. The outputs of the gate circuit represent the digits of the resulting computation.

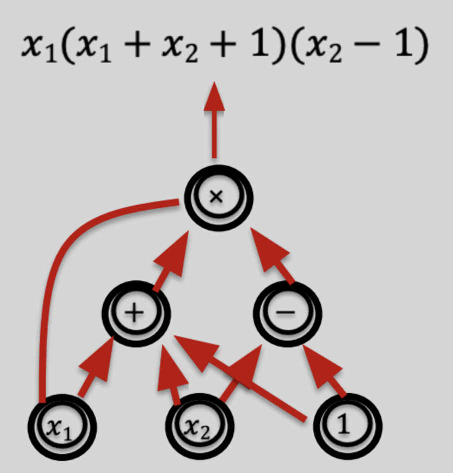
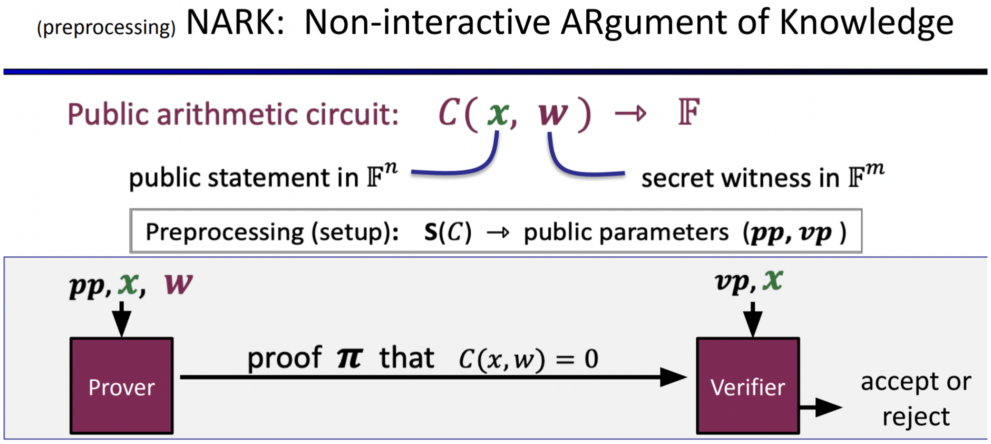


Figure 2.2: General Arithmetic circuit [8]

For SNARKs, the prover can create a proof, using the setup parameter, a private witness, and public input, showing that the arithmetic circuit is equal to 0. That proof can be verified by the verifier using the stup parameter and the public input.



(preprocessing) NARK: Non-interactive ARgument of Knowledge

Public arithmetic circuit: $C(x, w) \to \mathbb{F}$

public statement in $\mathbb{F}^n$

secret witness in $\mathbb{F}^m$

Preprocessing (setup): $\mathsf{S}(C) \to$ public parameters $(pp, vp)$

Figure 2.3: Arithmetic circuit for SNARK [8]

- **S(C)**: Public parameters (pp,vp) for prover and verifier

- **P(pp,x,w)**: Proof $\pi$

- $\mathbf{V(vp,x,}\pi\mathbf{)}$: Accept or reject

- **C(x,w)**: Arithmetic circuit

- **w**: Private witness

- **x**: Public input

## 2.4 Proof of solvency

# Chapter 3

# Recursion proofs

# Chapter 4

# Proof of liabilities

# Bibliography

[1] S. Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. Technical report, Bitcoin.org, 2008.

[2] Binance. Proof of Reserves. `https://www.binance.com/en/proof-of-reserves`.

[3] Boaz Barak. Lecture 14: Zero knowledge proofs. `https://www.boazbarak.org/cs127spring16/chap14_zero_knowledge.html`.

[4] LCX Team. Introduction to Zero-Knowledge Proofs. November 3, 2023. `https://www.lcx.com/introduction-to-zero-knowledge-proofs/`.

[5] Goldwasser, S., Micali, S., Rackoff, C. The Knowledge Complexity of Interactive Proof Systems. *SIAM Journal on Computing*, 18(1), 186–208. 1989.

[6] Blum, M., Feldman, P., Micali, S. Non-Interactive Zero-Knowledge and Its Applications. *STOC '88: Proceedings of the twentieth annual ACM symposium on Theory of computing*, 103–112. 1988.

[7] Goldreich, O., Micali, S., Wigderson, A. Interactive and Noninteractive Zero Knowledge are Equivalent in the Help Model? *STOC '91: Proceedings of the twenty-third annual ACM symposium on Theory of computing*, 113–131. 1991.

[8] Dan Boneh, Shafi Goldwasser, Dawn Song, Justin Thaler, Yupeng Zhang. Zero Knowledge Proofs: Introduction to Modern SNARKs. 2023.

[9] Nitulescu, Anca. *zk-SNARKs: A Gentle Introduction.* 2020.