Bitcoin.

A Thought Experiment In Programmable Money

Introduction

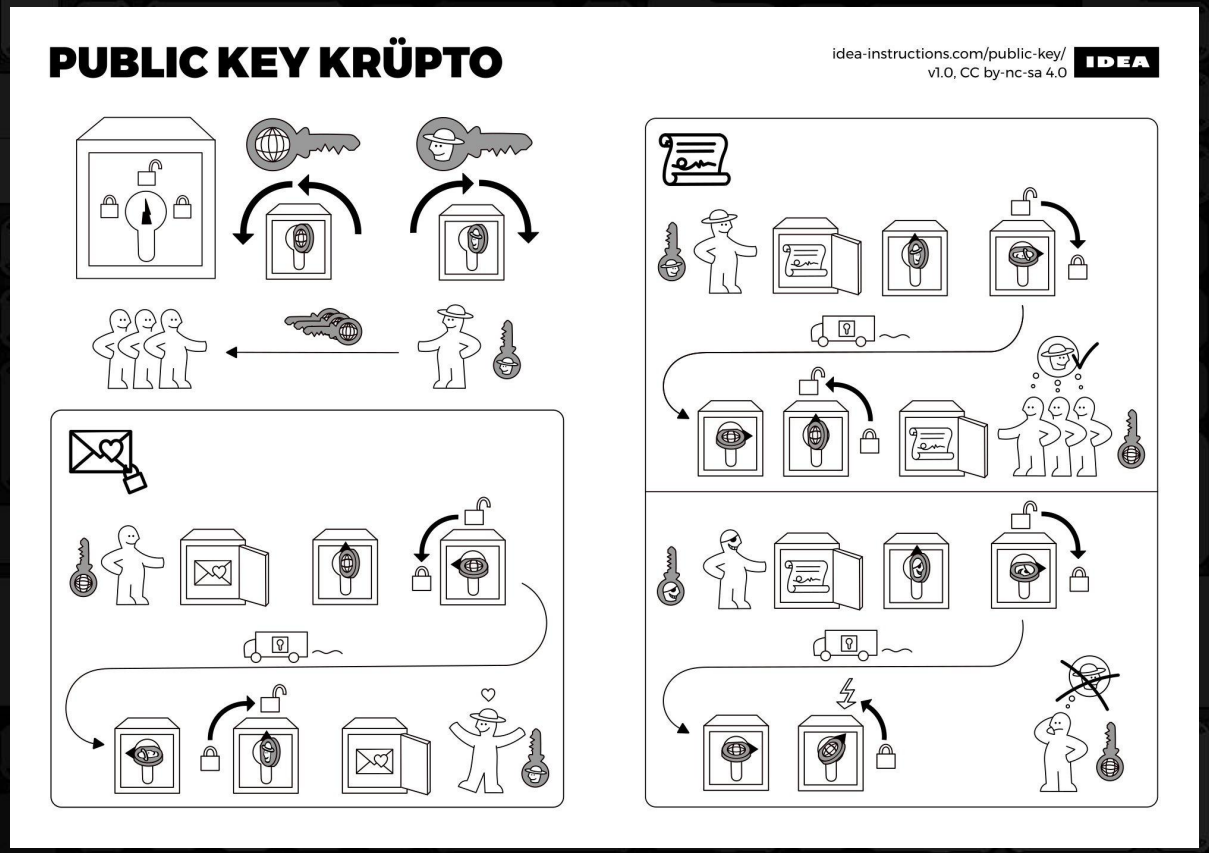
Digital Currency.

Naivecoin.

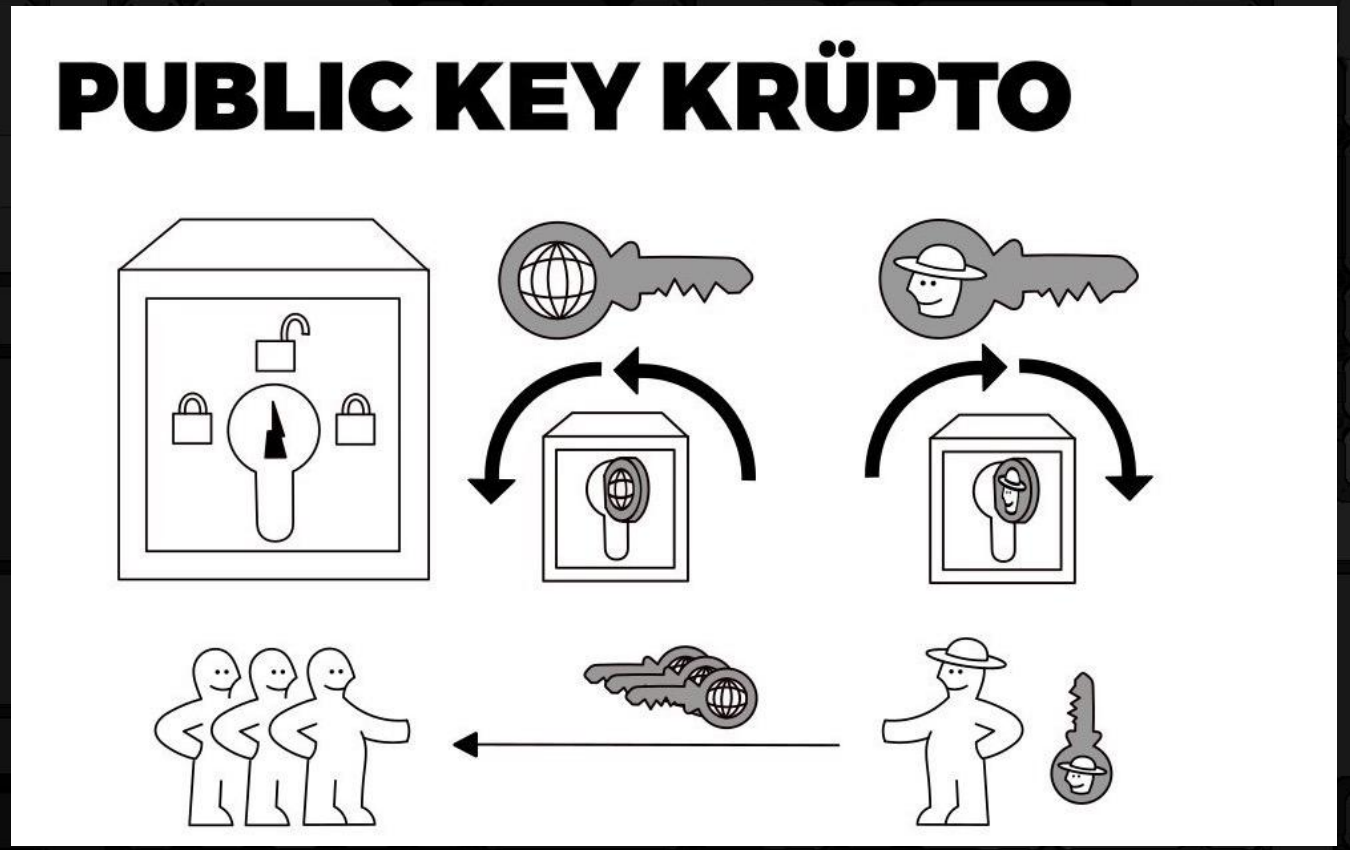
Naivecoin: Transactions

- Alice wants to pay Bob.
- Alice declares, “I, Alice, am giving Bob one Naivecoin” and then signs the message.

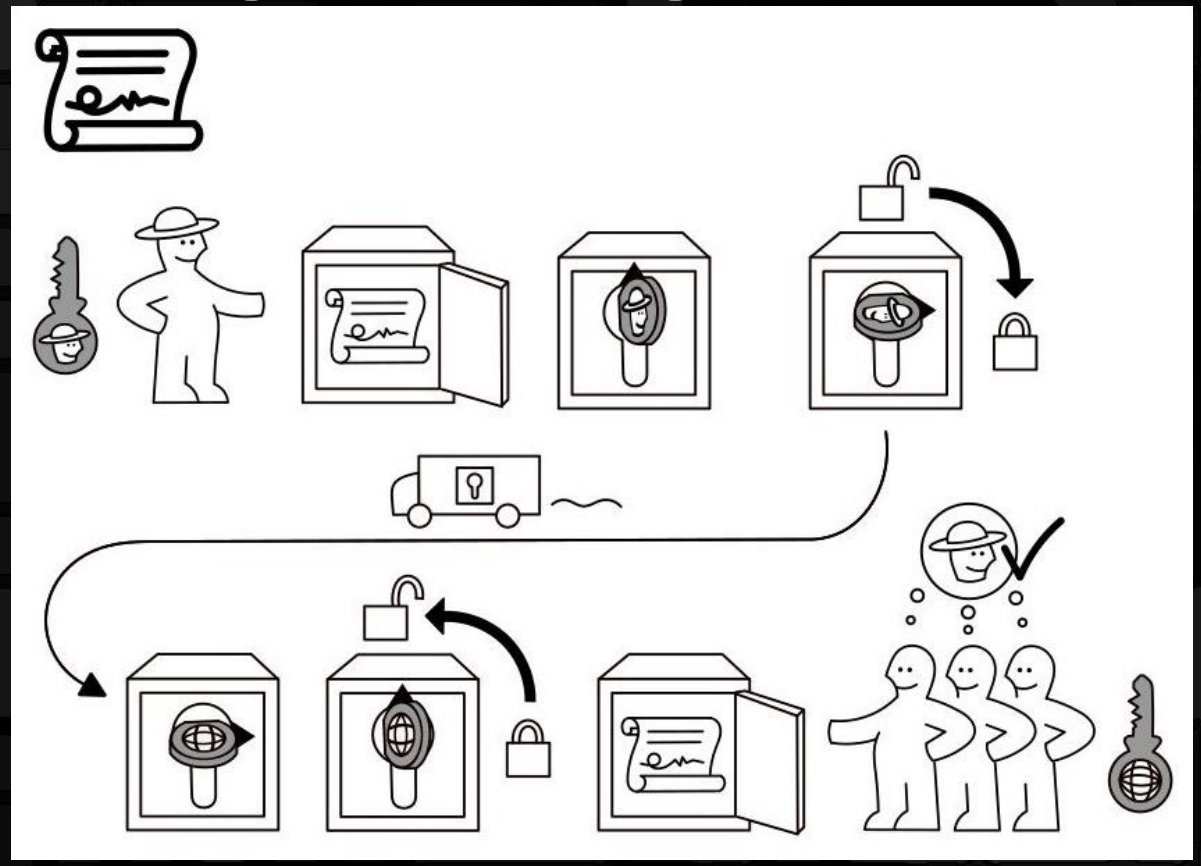
Naivecoin: Digital Signatures



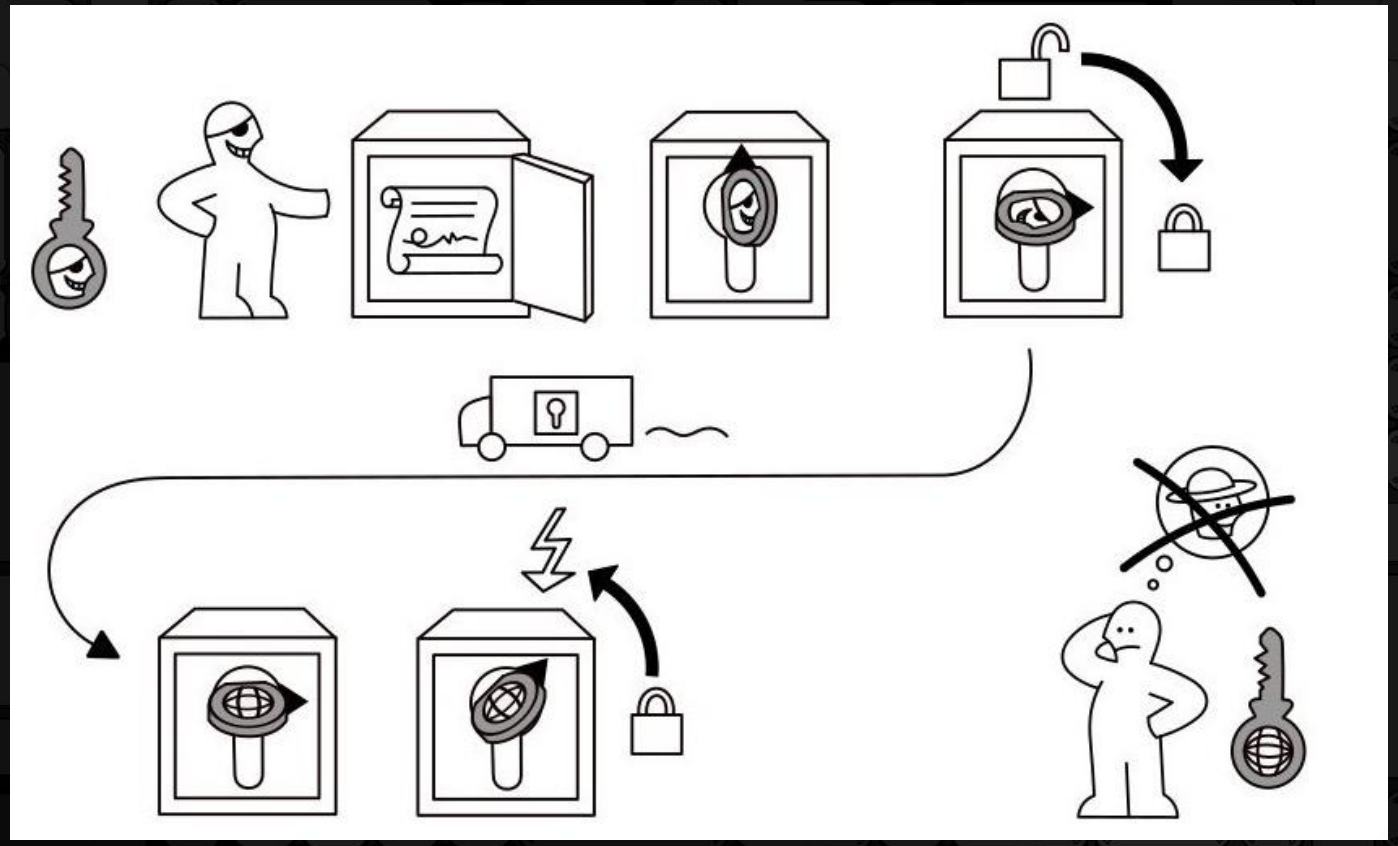
Naivecoin: Digital Signatures



Naivecoin: Digital Signatures



Naivecoin: Digital Signatures



Naivecoin: Transactions

- Alice wants to pay Bob.
- Alice declares, “I, Alice, am giving Bob one Naivecoin” and then signs the message.

Naivecoin: Digital Signatures

- Intent.
- Limited Forgery Protection.

Naivecoin: Multiple Transactions

- “I, Alice, am giving Bob one Naivecoin”
- “I, Alice, am giving Bob one Naivecoin”
- “I, Alice, am giving Bob one Naivecoin”
- “I, Alice, am giving Bob one Naivecoin”
- “I, Alice, am giving Bob one Naivecoin”
- “I, Alice, am giving Bob one Naivecoin”
- “I, Alice, am giving Bob one Naivecoin”
- “I, Alice, am giving Bob one Naivecoin”

Serialcoin?

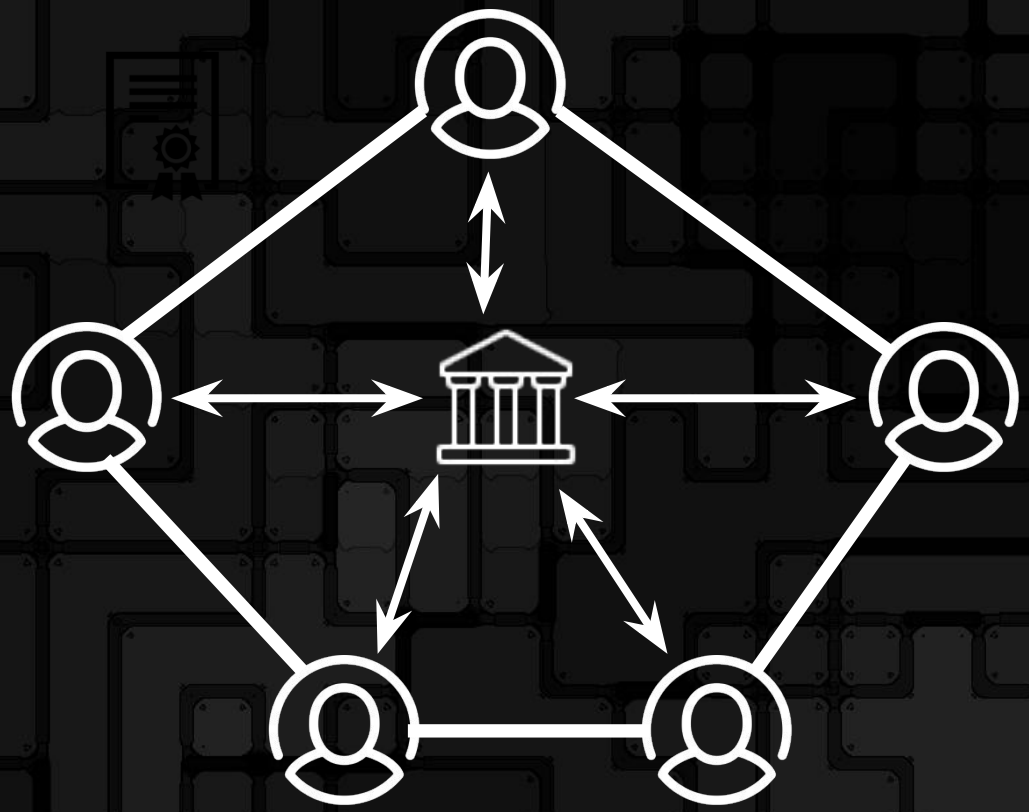
Serialcoin: Serial Numbers

”I, Alice, am giving Bob one Serialcoin” with serial number 789.

Bankcoin

- Issues serial numbers.
- Tracks who owns which serial numbers.
- Verify transactions are legitimate.

Bankcoin



Identitycoin.

Identitycoin: We Are The Bank

Everyone using Identitycoin keeps a complete record of the Identitycoins that belong to each person.

Identitycoin: Transactions

1. Alice signs the message “I, Alice, am giving Bob one Identitycoin, with serial number 789.”
2. Alice gives the message to Bob.
3. **Bob checks using his copy of the blockchain.**
4. If correct, he broadcasts Alice’s message and his acceptance of the transaction.
5. Everyone else updates their copy of the blockchain.

Identitycoin: Problems

- Serial Numbers
- Double Spend

Identitycoin: Solution

Outsource verification to the network.

Identitycoin: Transactions

1. Alice signs the message “I, Alice, am giving Bob one Identitycoin, with serial number 789.”
2. Alice gives the message to Bob.
3. Bob checks using his copy of the blockchain.
4. Bob broadcasts Alice’s message to the network.
5. If she owns the coin, they broadcast “Yes, Alice owns Identitycoin 789.”
6. If correct, he broadcasts Alice’s message and his acceptance of the transaction.
7. Everyone else updates their copy of the blockchain.

Identitycoin: Sybil Attack

- What happens if Alice creates a lot of identities?

Identitycoin: Problems

- Serial Numbers
- Double Spend
- Sybil Attack

Bitcoin.

Bitcoin: Proof of Work

- Let's make it **expensive** to validate transactions.
- Let's **reward** users for trying to help validate transactions.

Bitcoin: Proof of Work

1. Alice broadcasts to the network the news that “I, Alice, am giving Bob one Bitcoin, with serial number 789.”

Bitcoin: Proof of Work

1. Alice broadcasts to the network the news that “I, Alice, am giving Bob one Bitcoin, with serial number 789.”
2. David (a miner) will add her message to a transaction queue of transactions not yet approved.

Transaction List

I, Bob, am giving Charlie one Bitcoin, with serial number 818.

I, David, am giving Mallory one Bitcoin, with serial number 313.

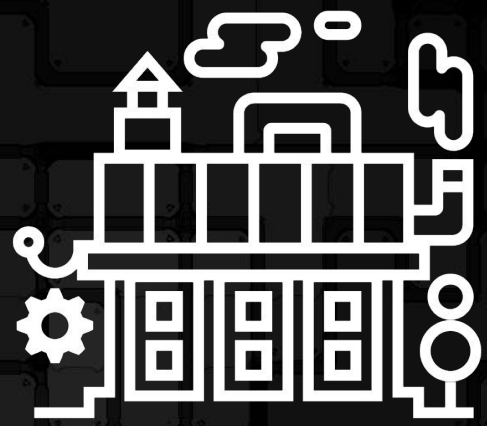
I, Eve, am giving Alice one Bitcoin, with serial number 929.

Bitcoin: Proof of Work

1. Alice broadcasts to the network the news that “I, Alice, am giving Bob one Bitcoin, with serial number 789.”
2. David (a miner) will add her message to a transaction queue of transactions not yet approved.
3. David checks the transactions in his queue are valid and wants to broadcast them.

Bitcoin: Functions

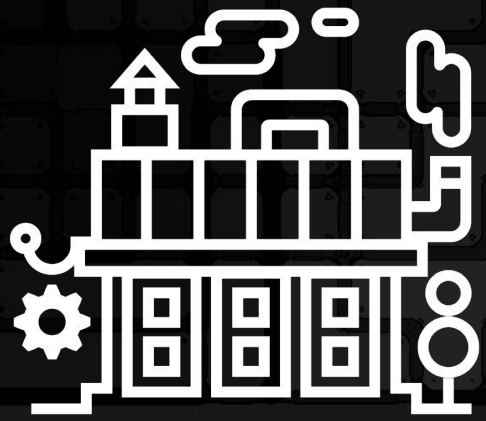
$$y = x + 3$$

1 →  → 4

Bitcoin: Hash Functions

Hello, World!0

SHA-256



1312af178c253f8402
8d480a6adc1e25e81
caa44c749ec819761
92e2ec934c64

Bitcoin: Hash Function

Hash(“Hello, World!0”) =
1312af178c253f84028d480a6adc1e25e81caa44c749ec81976192e2ec934c64

Hash(“Hello, World!1”) =
e9afc424b79e4f6ab42d99c81156d3a17228d6e1eef4139be78e948a9332a7d8

Bitcoin: The Puzzle

1. Let’s grab a bunch of random transactions.
2. Now we are going to start a counter from zero. We’ll call it a nonce.
3. Let’s append the nonce at the end of the transaction.
4. Now we’ll hash the whole thing.
5. Check if it satisfies the puzzle’s requirements.
6. If not, increment the nonce and go back to Step 3.

Bitcoin: Hash Function

Hash(“Hello, World!0”) =

1312af178c253f84028d480a6adc1e25e81caa44c749ec81976192e2ec934c64

…

Hash(“Hello, World!4250”)

0000c3af42fc31103f1fdc0151fa747ff87349a4714df7cc52ea464e12dcd4e9

Bitcoin: Difficulty

- REMEMBER: Hash function is inherently random.
- Automatically retargets to 10 minute block times.
- Real Bitcoin protocol uses a puzzle that requires the hash to be below a certain number (the ”target”)

Bitcoin: Mining

- Why do miners mine? What incentivises them?

Bitcoin: Minting Schedule

- Initial reward of 50 BTC per block.
- Every 210 000 blocks, this is halved (~4 years).
- Currently at 12.5 BTC per block.

Bitcoin: UTXOs

- You can only spend Bitcoin if you can prove ownership of it.
- When Bitcoin is minted, it is minted as a UTXO that belongs to the miner's address.
- To spend, you must consume an existing UTXO as an input and generate a new UTXO as an output.
- Can get rid of serial numbers.

Bitcoin: UTXOs

Transaction View information about a bitcoin transaction

63fce5b4b8ba32676532731c10b7d8110b7d7c70ac8e728ce8f19afd1f25f08d

1NgQQqEfkgpDpKQsGk3eFZMFPgdAWZVcif	12dUgRybQCUvs1E1x6dbk4XwhH18qAJ7xB	0.02706676 BTC
	3Ahjcmz5wUADwMddsvBmbabve4JYT6vuBA	0.09289327 BTC

1 Confirmations | 0.11996003 BTC

Summary	
Size	223 (bytes)
Weight	892
Received Time	2018-03-18 22:19:48
Lock Time	Block: 514149
Included In Blocks	514151 (2018-03-18 22:24:13 + 4 minutes)
Confirmations	1 Confirmations
Visualize	View Tree Chart

Inputs and Outputs	
Total Input	0.12164003 BTC
Total Output	0.11996003 BTC
Fees	0.00168 BTC
Fee per byte	753.363 sat/B
Fee per weight unit	188.341 sat/WU
Estimated BTC Transacted	0.02706676 BTC
Scripts	Show scripts & coinbase

Bitcoin: Fees

- No concept of change in Bitcoin.
- Fees is difference between total inputs and total outputs.

Bitcoin: Proof of Work

1. Alice broadcasts to the network the news that “I, Alice, am giving Bob one Bitcoin, with serial number 789.”
2. David (a miner) will add her message to a transaction queue of transactions not yet approved.
3. David checks the transactions in his queue are valid and wants to broadcast them.
4. David finds the right nonce. (hooray!)

Bitcoin: Proof of Work

5. David broadcasts the block of transactions to the network.
6. Other participants can verify the block.
7. David is given some bitcoins for his trouble.

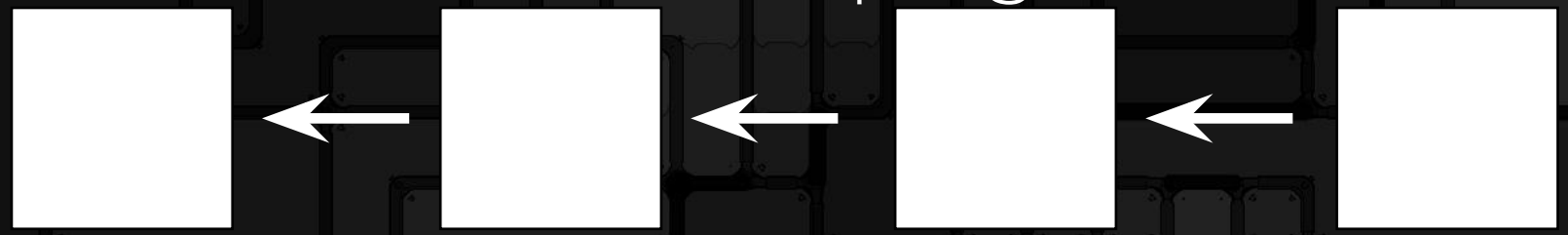
Bitcoin Consensus.

Bitcoin: Ordering

How do we order blocks on the blockchain?

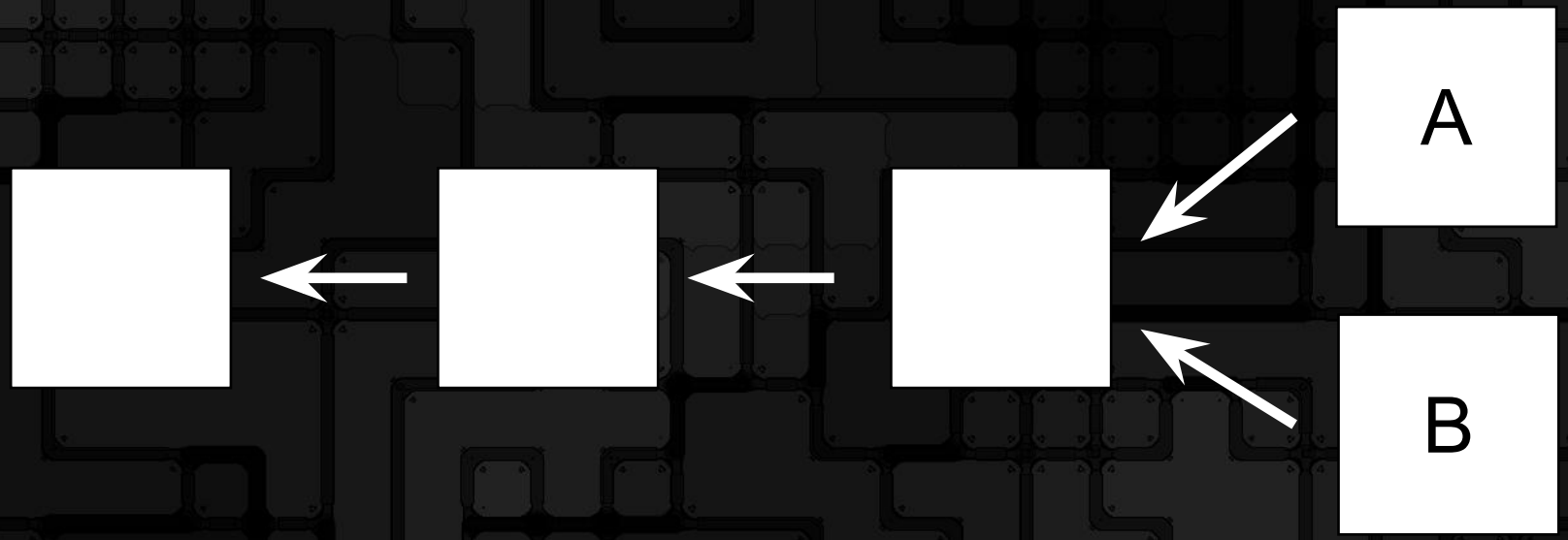
Bitcoin: Ordering

- Each block points to the previous block.
- In Bitcoin, this is done by storing the hash of the previous block.
- This also acts as a timestamping mechanism.



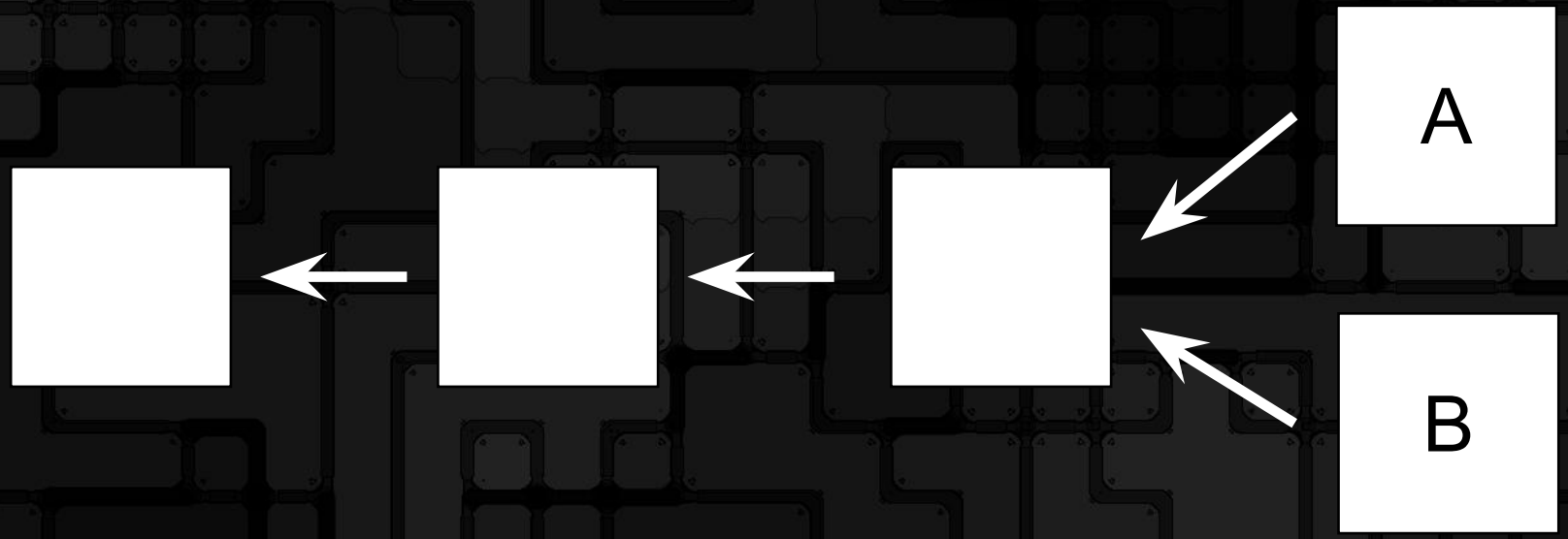
Bitcoin: Fork Resolution

- What happens if two blocks are found at the same time?



Bitcoin: Fork Resolution

- Miners will always work on the longest chain.
- Miners will always work on the block they are given first.



Bitcoin: Transaction Confirmation

- Must be in the longest fork.
- Must have at least 5 blocks that follow it (6 confirmations).

Bitcoin: Double Spend

- Impossible to double spend based on this game theory.
- Many attacks on Bitcoin are expensive or game theoretically unfavourable.

Questions.

Double Spend Appendix.

Bitcoin: Double Spend Attempt #1

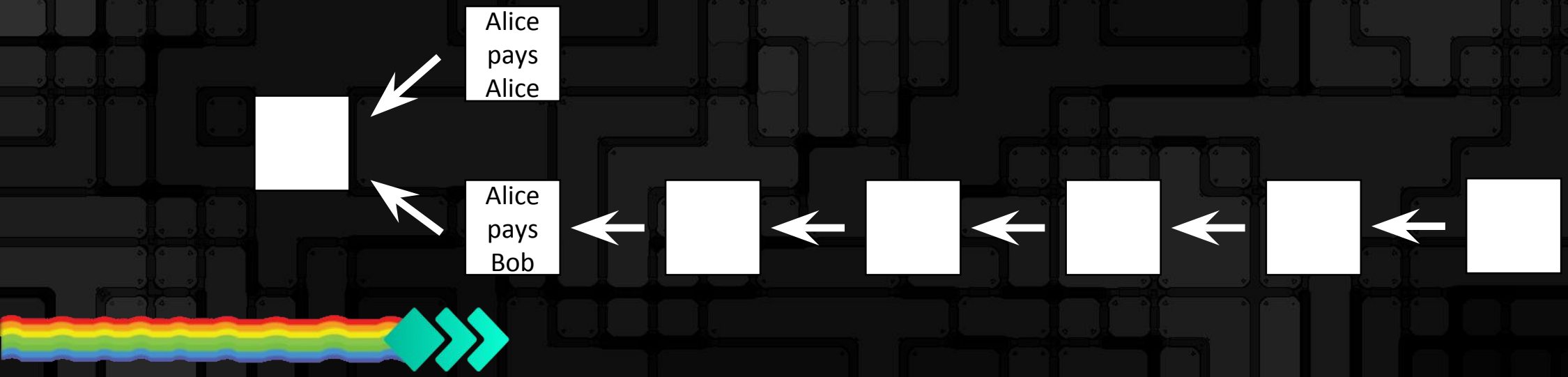
- Alice tries to send money to Bob & Charlie.
- She participates as a miner.
- She processes a block containing transaction of her sending same Bitcoin to both Bob & Charlie.

Bitcoin: Double Spend Attempt #2

- Alice can broadcast a transaction sending money to Bob to one subset of miners and a transaction sending money to Charlie to another subset of miners.
- Two blocks may form at the same time.

Bitcoin: Double Spend Attempt #3

- Alice pays Bob and waits until Bob accepts the transactions (6 confs.)
- Alice will then attempt to fork the chain before she paid Bob, adding a block where she pays herself.



Bitcoin: Double Spend Attempt

#3Q`1ÅC#

- No one will help Alice.
- She has to work as fast as everyone else in the network to catch up.
- It is now very hard to catch up.