Blockchain and Cryptocurrency

#Goalz

What is a blockchain?

How is cryptocurrency “mined”?

What makes cryptocurrency secure?

What is “proof of work”?

The Ledger Analogy

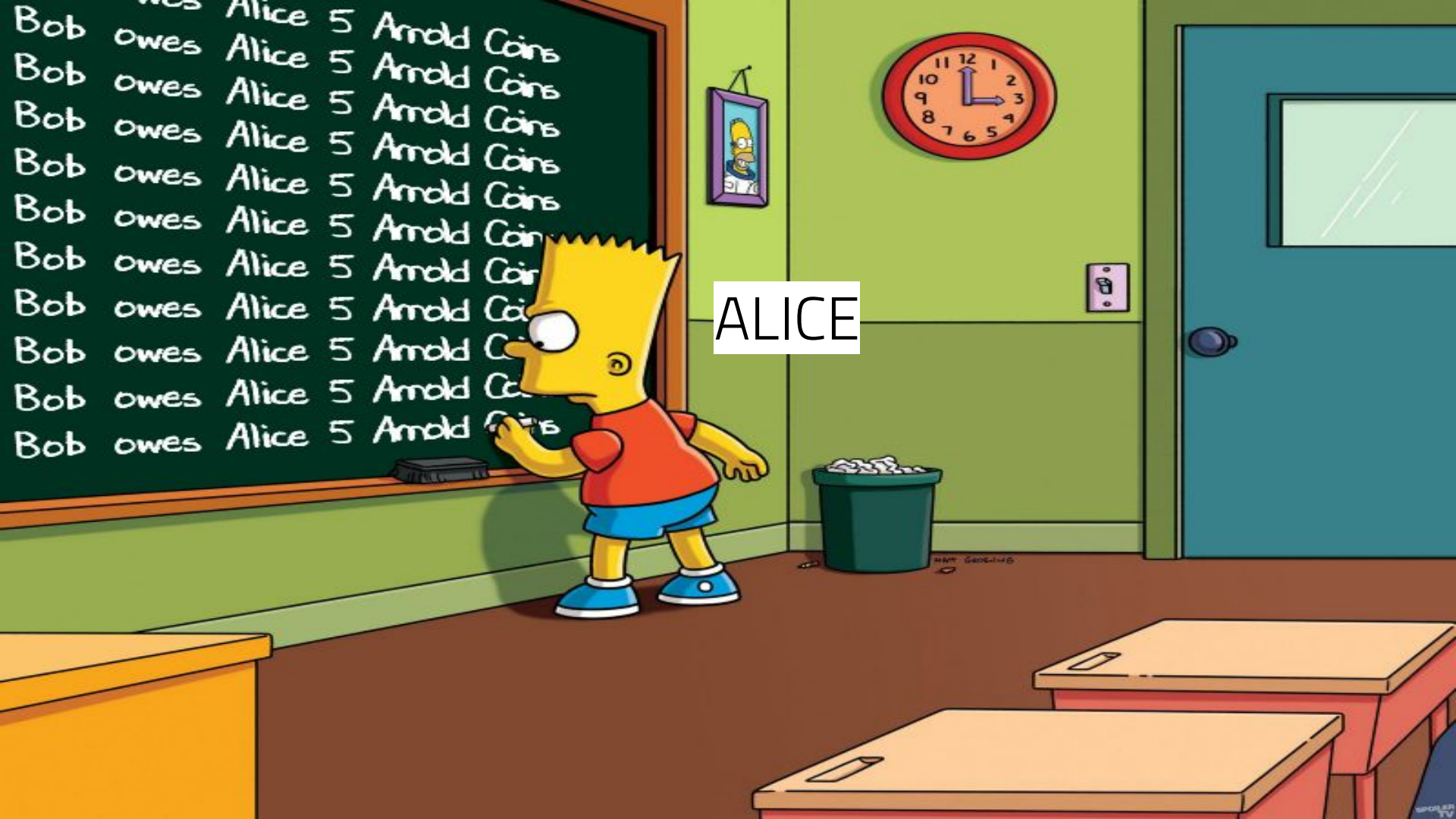
>>> Let's say that you and some friends are constantly spending money (must be nice) and you have a Ledger that you keep track of payments that you are going to make

>>> Let's say that at the end of the month, you all meet up and settle up. If you spend more than receive, put money in the pot. If you receive more than you spend, put money in the pot.

The Ledger Analogy

>>> Who's to stop someone from constantly writing to the ledger that people owe them money?

>>> For example, Alice keeps writing to the ledger that Bob owes her money.


Bob owes Alice 5 Arnold Coins
Bob owes Alice 5 Arnold Coins
Bob owes Alice 5 Arnold Coins
Bob owes Alice 5 Arnold Coins
Bob owes Alice 5 Arnold Coins
Bob owes Alice 5 Arnold Coin
Bob owes Alice 5 Arnold Coin
Bob owes Alice 5 Arnold Coi
Bob owes Alice 5 Arnold C
Bob owes Alice 5 Arnold Co
Bob owes Alice 5 Arnold
ALICE

The Ledger Analogy

>>> Every person in the ledger has a Secret Key and a Public Key

>>> So, when you write a transaction, you (the sender) signs it:

Sign(Message, SK) → Signature

>>> And others can verify the transaction by

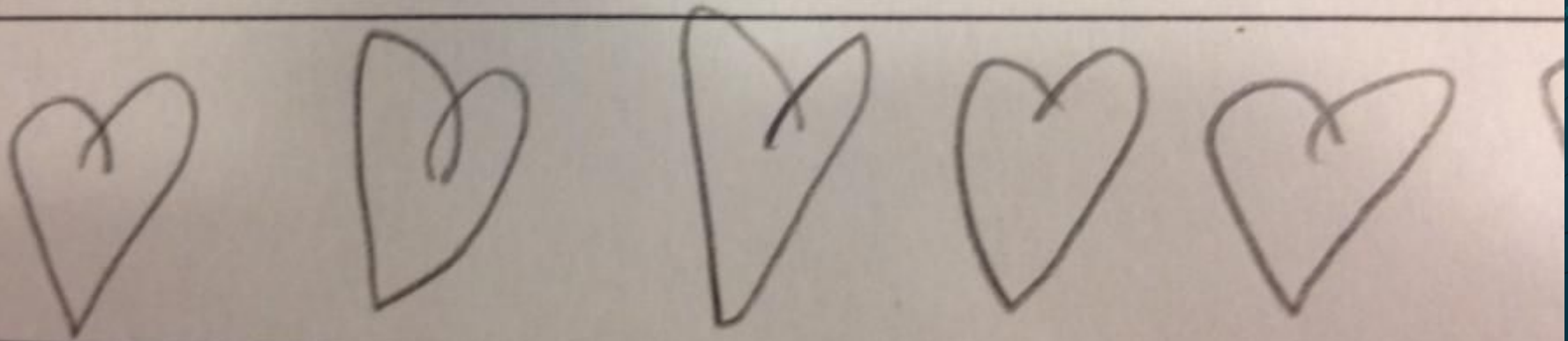
Verify(Message, Signature, PK) → boolean

Bob owes Alice 5 Arnold Coins

Let's say that Alice was trying to forge Bob's signature. She would have to sign the message with her SK. But when the other people in the ledger (including Bob) go to verify that transaction, they will use Bob's public key since he is the sender of the transaction. Verification Failed. Transaction Void.

Parent Signature MOMMY

Any questions or comments:



The Ledger Analogy

>>> Ok, so now we have a place to store transactions, and a way to verify the legitimacy of the transactions.

How Is This Related To BlockChain??

Structure of Blockchain

>>> Blockchain contains a set of Blocks (highest structure)
>>> Blocks contain a set of transactions
>>> Transactions are user-created (lowest structure of blockchain)

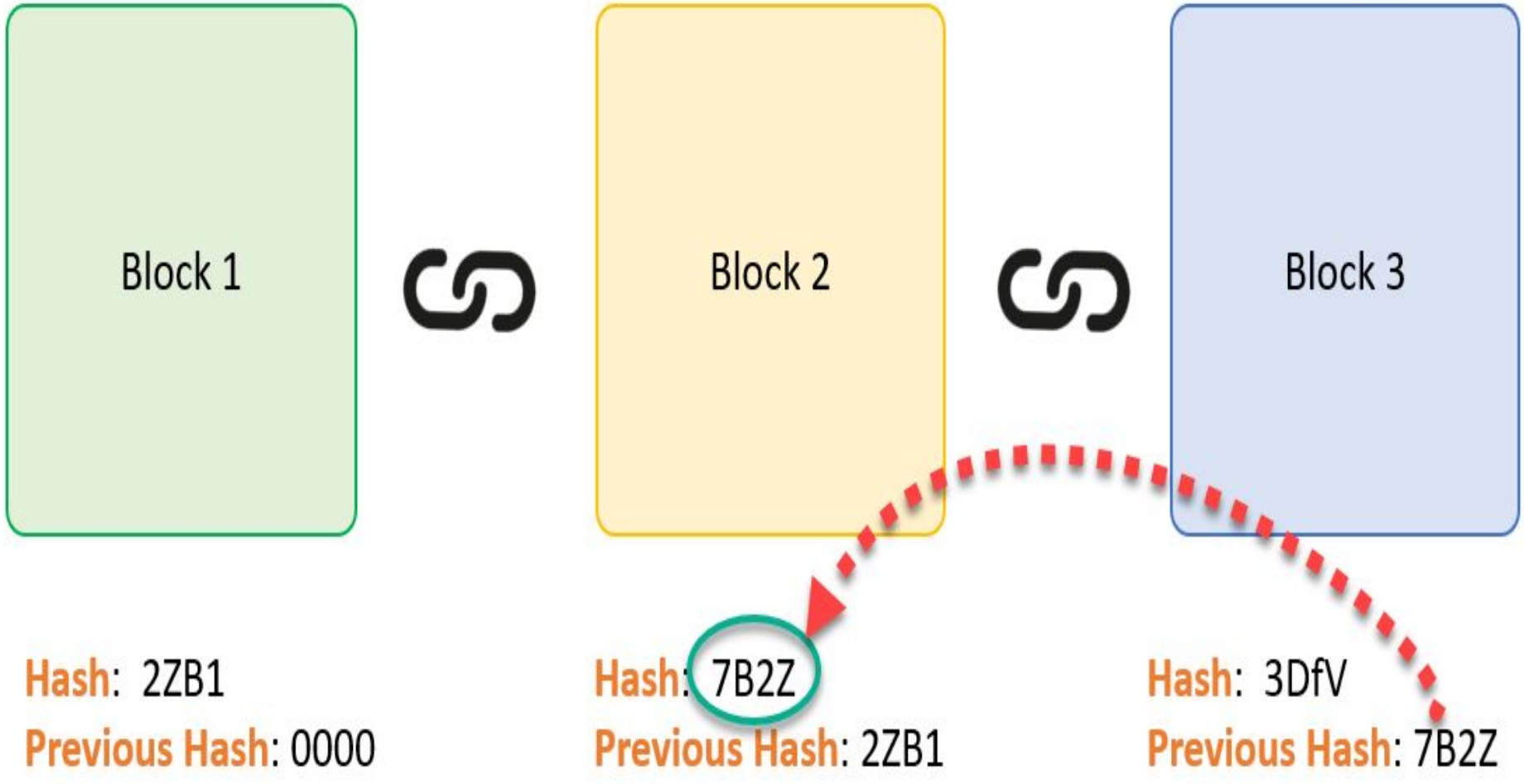
A Top-Down Explanation of Blockchain...

Blockchain

- An immutable collection of ordered blocks.

- The world's most famous linked list

- Why is this important?
  - Every block in the chain depends on the block before it.
  - Every block has a unique hash that is used in the block after it.


Block 1
Block 2
Block 3
Hash: 2ZB1
Previous Hash: 0000
Hash: 7B2Z
Previous Hash: 2ZB1
Hash: 3DfV
Previous Hash: 7B2Z

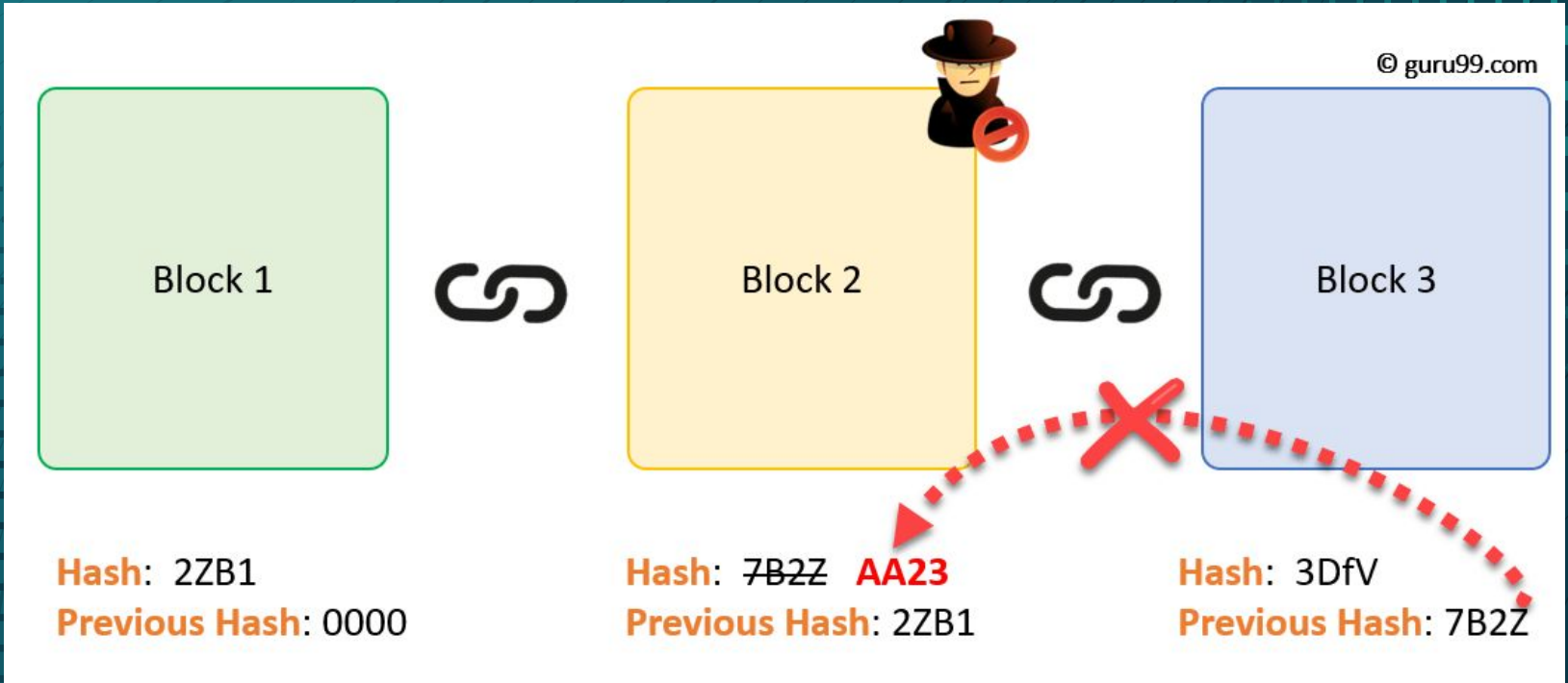
Blocks

- Blocks are a collection or group of transactions.
- The hash of the current block contains:
  - the hash of the previous block
  - a nonce value (a calculated random number)
  - Timestamp
  - Hash of all transactions (Merkle Root)

Blocks

- Changing a transaction or any information in the block would result in completely different hash value for the block. How does this affect the chain?
- Effectively, the rest of the chain becomes invalid from the point the previous hash is invalid.

The rest of the chain from block 2 is invalid



Blocks

- So how do we know that the start of the chain isn't invalid? How do we know a hacker can't go and change that?
- In comes the "Genesis Block"

Genesis Block

- The first block in the blockchain
- Has a predefined value instead of a hash of a previous block.
  - The values of the Genesis are known to everyone, and so, can be verified by everyone to have a very specific hash value.
- All valid blocks can trace back to the Genesis.

Bitcoin's Genesis Value

- Typically coins will use a headline from the day of launch.
- This is proof of when the block was established as the headline would not have been known prior to the date and shows that there would not have been blocks created before that date.

Bitcoin's Genesis Value

```
*/
static CBlock CreateGenesisBlock(uint32_t nTime, uint32_t nNonce, uint32_t nBits, int32_t nVersion, const CAmount& ge
{
    const char* pszTimestamp = "The Times 03/Jan/2009 Chancellor on brink of second bailout for banks";
    const CScript genesisOutputScript = CScript() << ParseHex("04678afdb0fe5548271967f1a67130b7105cd6a828e03909a67962
    return CreateGenesisBlock(pszTimestamp, genesisOutputScript, nTime, nNonce, nBits, nVersion, genesisReward);
}


/**
```

Transactions

-The movement of digital currency from one address to another.

-How do we know if both parties approve of a transaction?

Transactions

Core Requirements of a Transaction:

- A message
- A digital signature
- hash
- Transaction fees*

Transactions

Message

- Amount
- Source Address -> Destination Address

Problem: Anyone can forge a transaction if they know the address.

Solution: Digital Signatures!

Transactions

Digital Signatures

- Every transaction is digitally signed by the sender
- Everyone has a public and a private key.

Sign(Message, PrivateKeyOfSender) -> DigitalSignature #256 bits

Verify(Message, DigitalSignature, PublicKeyOfSender) -> T/F

Transactions

Question: What prevents an attacker from using the same message twice? (Replay attack)

Solution: Only allow addresses to send currency once.
-If you use an address to send currency, it cannot be used again.
-An attacker would have to use a different address than the one used.
-A different address means, a different message which would also change the digital signature.

Transactions

What makes a transaction valid?

-Verify(Message, Digital Signature, Public Key) -> Must equal true.

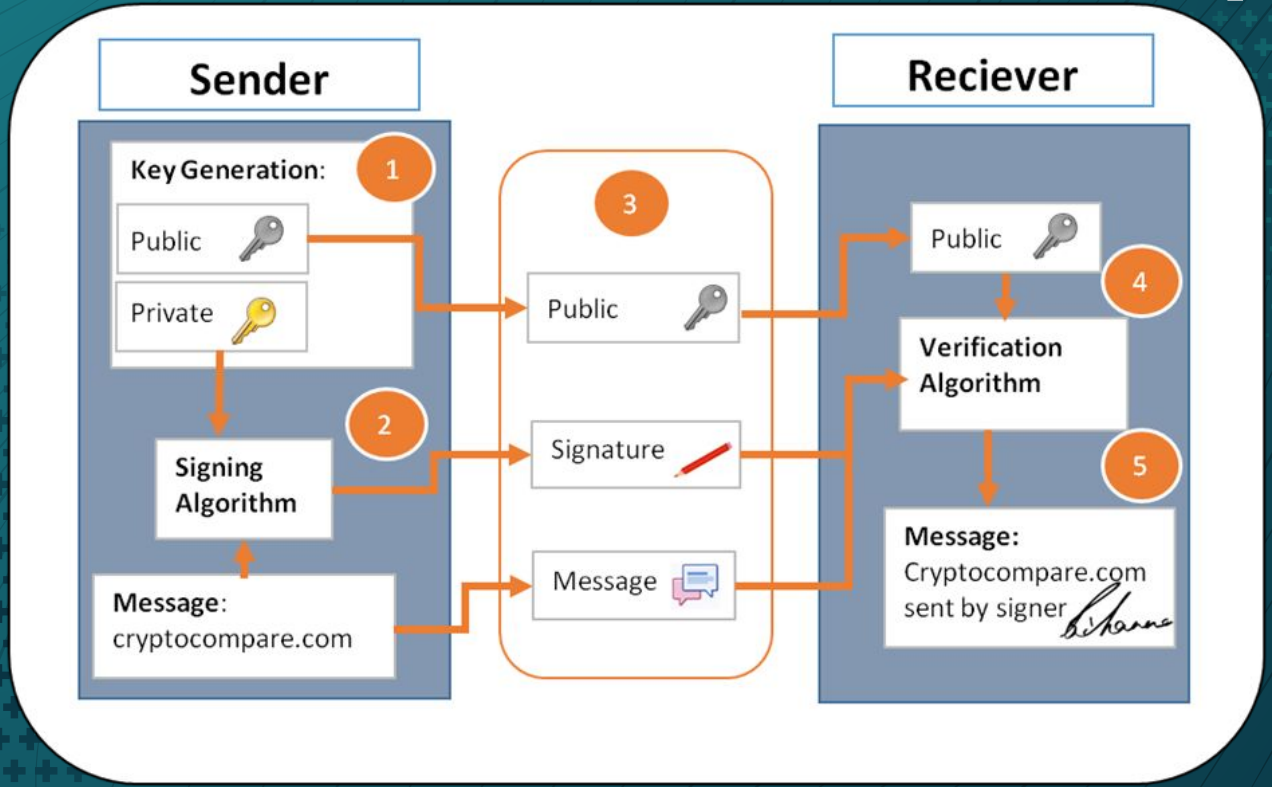
Transactions

What makes a transaction confirmed?

-The transaction must be added to valid block in the longest blockchain in order to be confirmed.

- A transaction can be valid but unconfirmed for a while. The majority of miners must agree that the block that transaction is in is a “confirmed block” and is added to the main chain.

Transactions Example



Ok, so how do we confirm transactions?

Proof of Work

Proof of Work

>>> Up until now, we have been thinking about a centralized blockchain.

>>> Cryptocurrencies use a decentralized blockchain.

>>> Everyone must have a copy of the blockchain.

>>> Everyone must have a way to update their copy.

>>> Everyone must be able to add valid transactions and blocks.

>>> All nodes must arrive at a consensus on which blockchain to follow. Consensus is typically the chain with the most computation put into it.

Decentralization

>>> Blockchain does not store any of its information in a central location.

>>> The blockchain is copied and spread across a network of computers.

>>> Whenever a new block is added to the blockchain, every computer on the network updates its blockchain to reflect the change, via consensus

Proof of Work

- algorithm used to confirm transactions and produce new blocks to the chain.

- How does the algorithm work?

Proof of Work

- We know the details of a block (in its' simplest form)
  - The hash of the previous block
  - A set of transactions (hashed)
  - And something called a nonce value (we'll get to this later)

THE BITCOIN TRANSACTION LIFE CYCLE

The Ledger Analogy

>>> Trust the ledgers with the most work put into it.

>>> Everyone will listen to the transactions, and accept the ledgers into their pile of ledgers (chain of ledgers) that have the most computational work put into it.

>>> Fraud ⇔ Computationally infeasible. Everyone follows the same protocol.

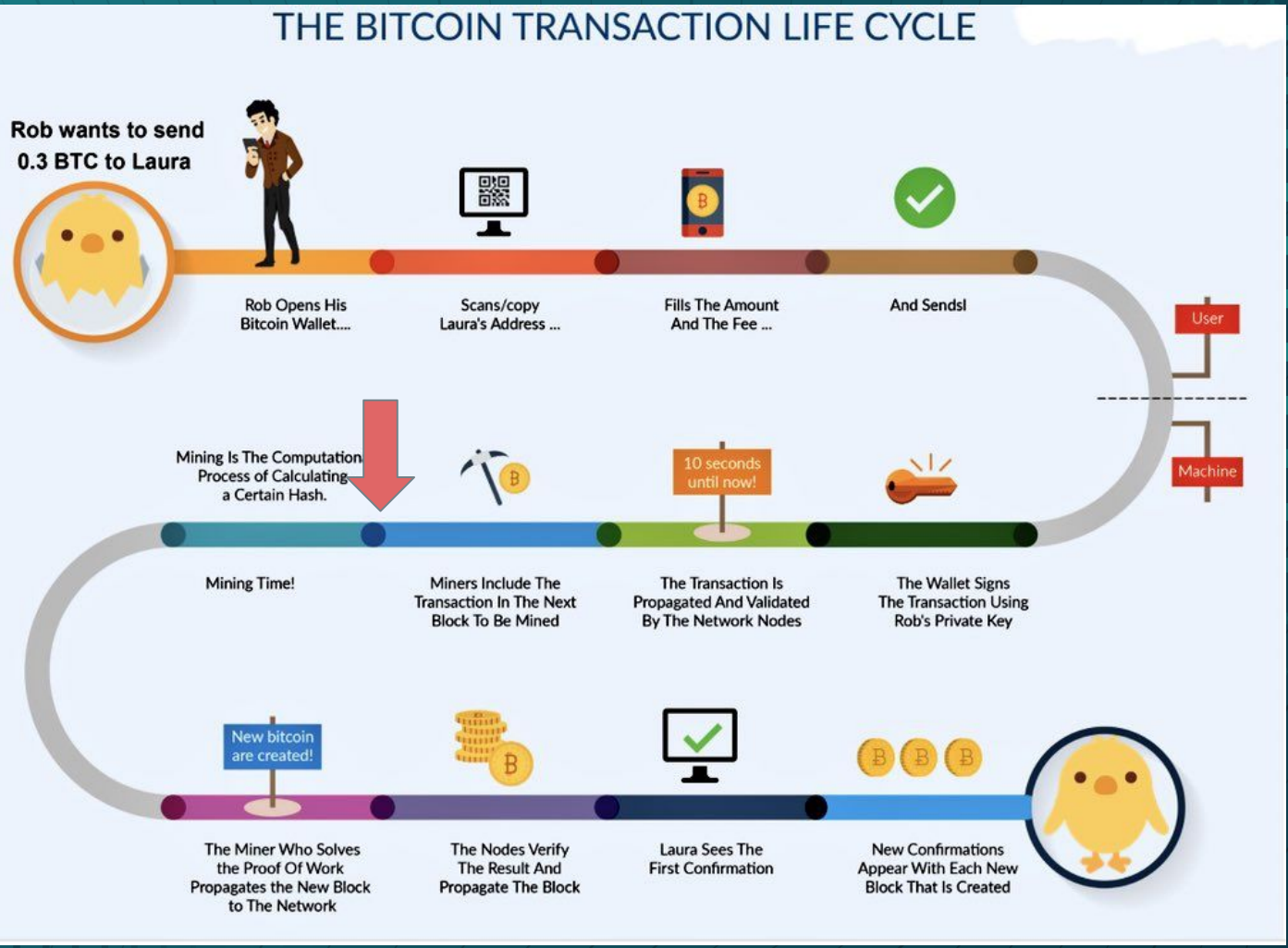
Mempool

>>> Transactions are broadcasted by other nodes and added into a “mempool” -- A collection of all unconfirmed transactions

>>> All nodes listen for these broadcasted transactions and keep a copy of this list(not ordered)

>>> Miners add transactions from the mempool to blocks. Once that transaction is added to a block and confirmed, it is removed from the mempool

THE BITCOIN TRANSACTION LIFE CYCLE



Proof of Work - Nonce

- The nonce value is (basically) proof of work, in its' simplest form.
- The nonce value is a calculated string that when added to the block, the hash of the block has a certain number of leading 0's (bitcoin)

Proof of Work - Nonce

- How many 0's?
  - This is going to be the "difficulty of the chain". It can vary, depending on how quickly the cryptocurrency wants blocks mined.
    - Bitcoin is roughly every 10 mins

Proof of Work - Nonce

- How do I find this magical value?
- In short, a lot of computation power.
- The value can be anything, so you had better be ready to try everything. As a miner, your job is to get that block into the chain

Proof of Work - Miner

- Why should a miner go through all that effort?
  - Every transaction has a reward associated with it
  - Every block has a block reward (cumulative transaction reward)
  - Once block is mined, that reward goes to the miner!

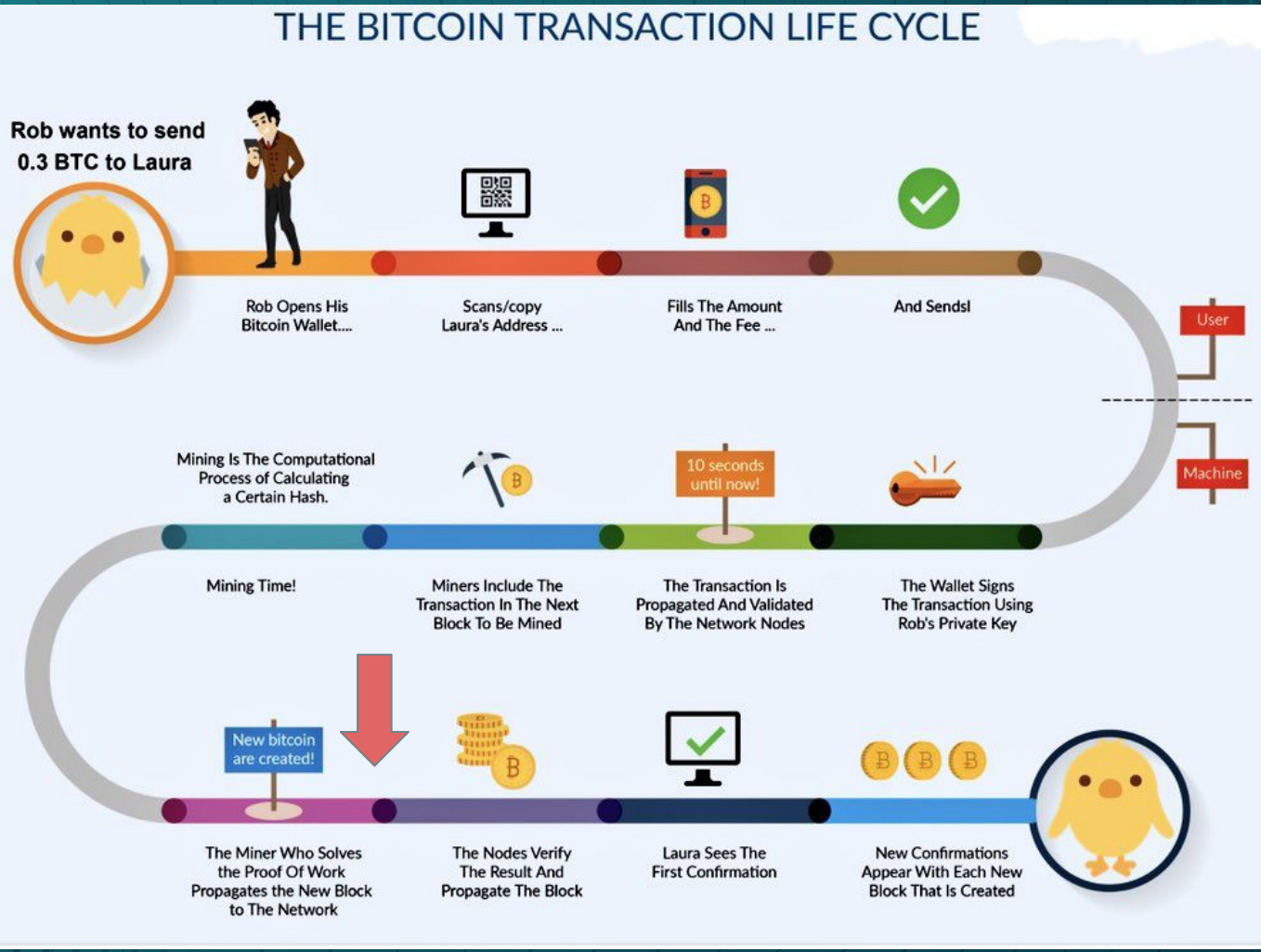
Proof of Work -- Nonce

127.0.0.1 - - [10/Mar/2020 11:39:59] "POST /new_transaction HTTP/1.1" 201 -
127.0.0.1 - - [10/Mar/2020 11:39:59] "GET /chain HTTP/1.1" 200 -

abdb51df025a1647f975a9e264ffc90b6478c5456d2c073074dd79d309f44324
242941715a21494feeca1a64049ec7d79c5e0f274801fbd1fe1c3eb79cfae0ec
46aa5777fcf80d345f911180ac3122144e659f546fa7e26942b8e4778fdc4e4a
c77866a032c59a3599ae03bac81b98691d314546b9bbea5e700e4edae99d3be7
478da4c2095b9e5bdee31f907ee3046acf0c11e52f9994b1391d182464da03aa
878a6b3855881244eea58f1d136a27ed092a49ab7190475ea85c36ff94104472
0000bdf746bad70af379100f17e179e87221405ee5f00859cb82b9a8cbba2ec2

FOUND NONCE

THE BITCOIN TRANSACTION LIFE CYCLE

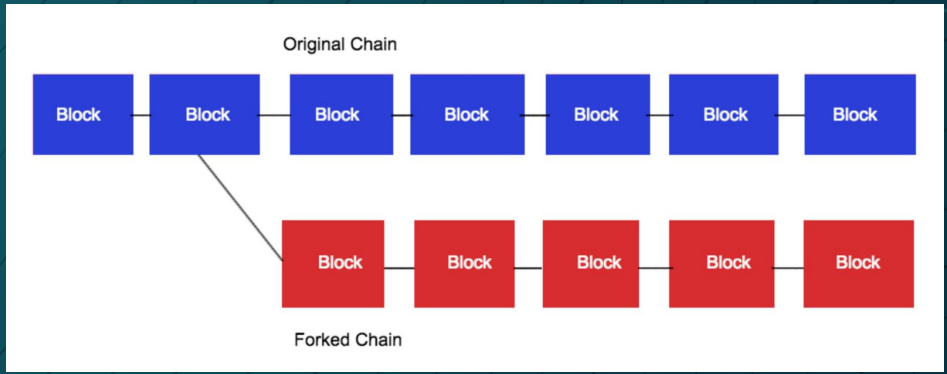


Proof of Work -- Forks

What if different miners created different valid blocks and they both attempt to add it to their subscribed blockchains?

Create a fork. Two seperate blockchains.

Which is the real one?



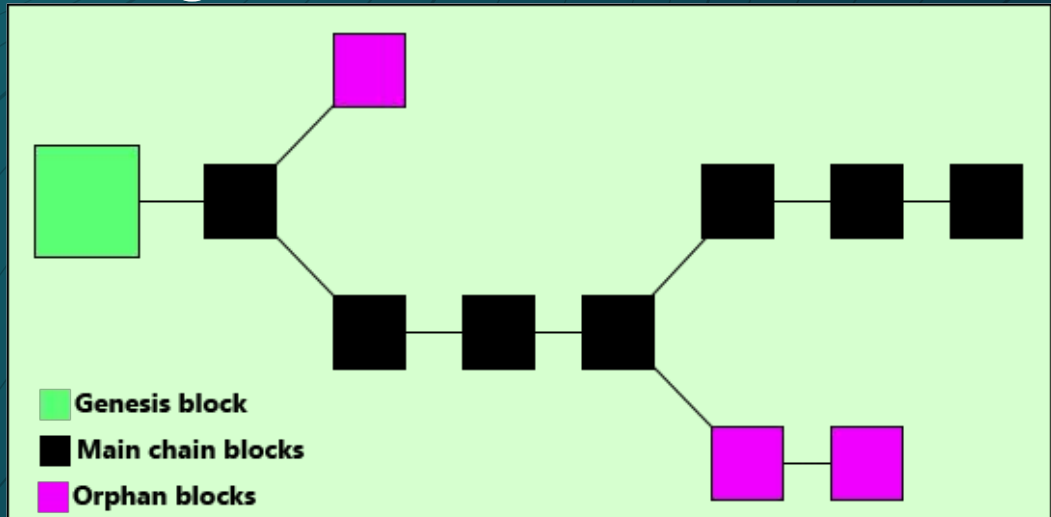
Proof of Work

Nodes must arrive at a consensus on which chain to follow.

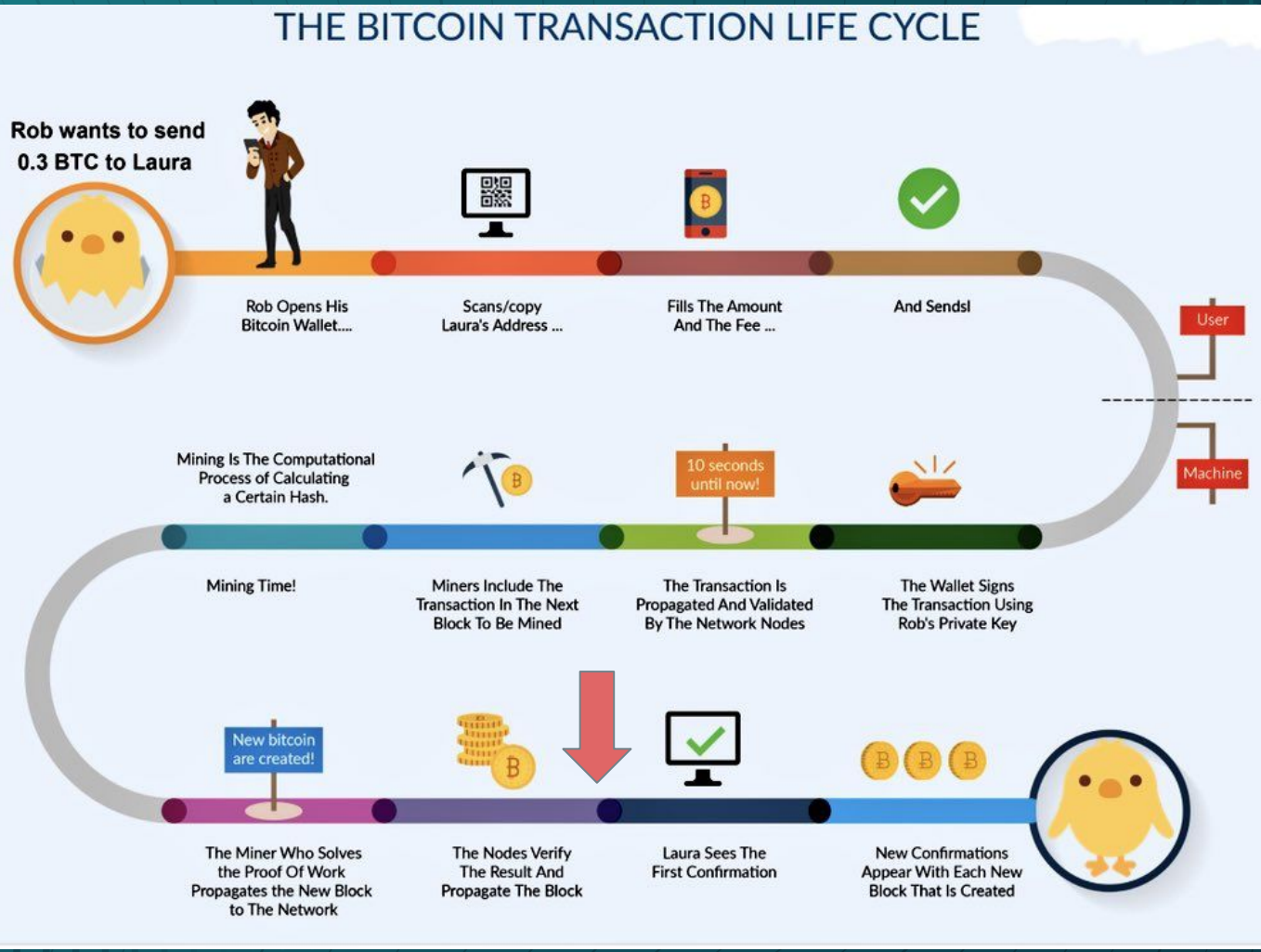
For POW, trust the longest blockchain or the blockchain with the most work(Hence the name).

Proof of Work -- Tree Reorganization

- Miners continue to add blocks to forked chains. Whichever chain becomes the largest, will become the legitimate blockchain.



THE BITCOIN TRANSACTION LIFE CYCLE

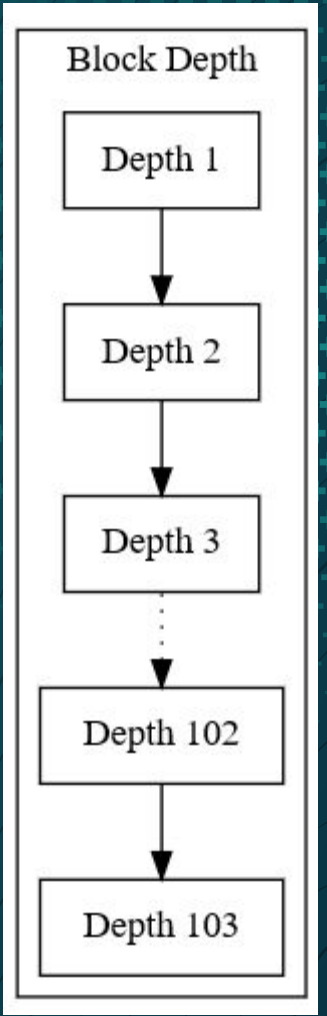


Proof of Work

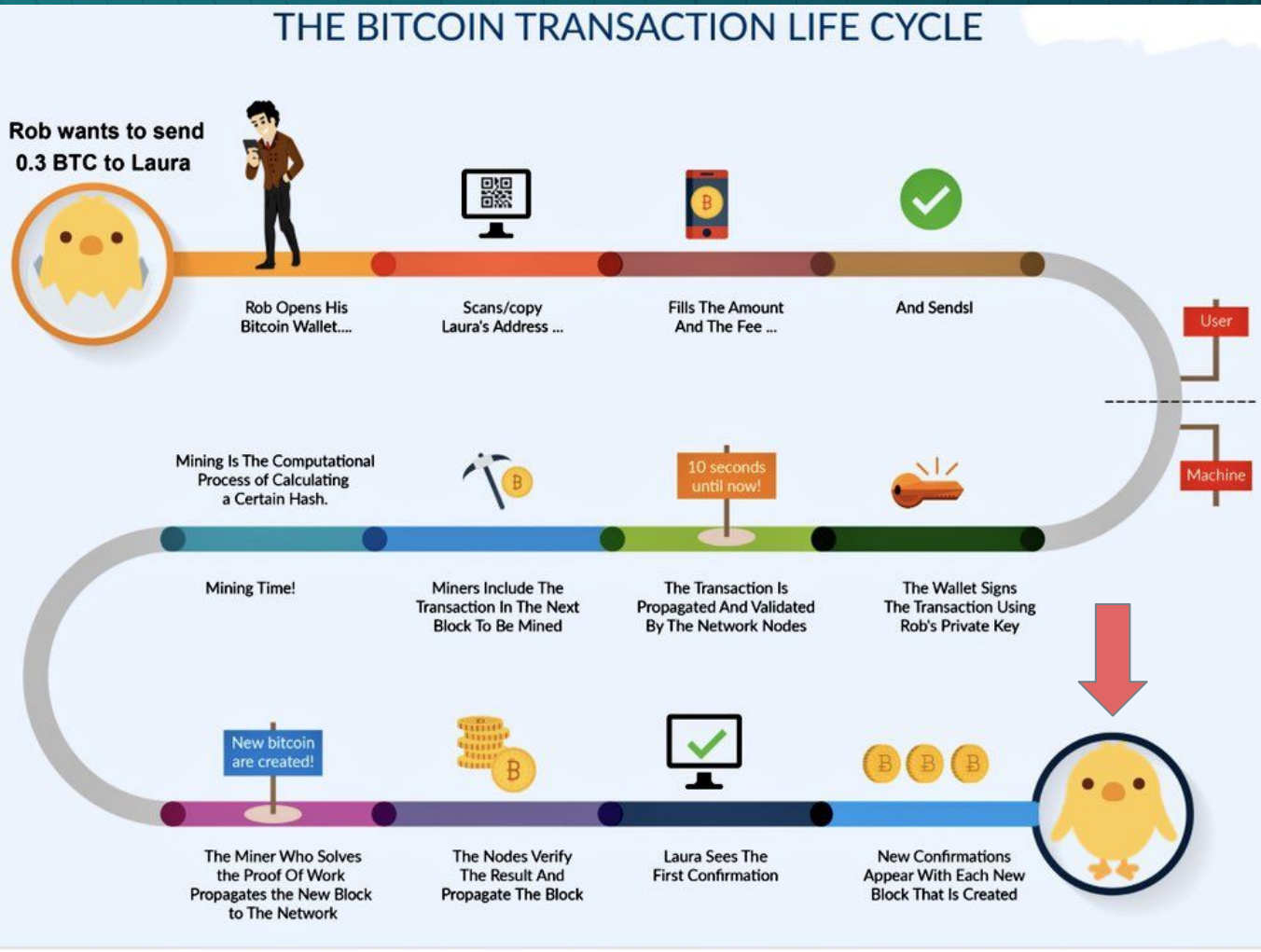
-Transactions are only confirmed after that threshold has been reached.

-Exchanges or other services that accept cryptocurrency can set their own thresholds. The bigger the threshold, the longer it takes to confirm a transaction.

-e.g If the threshold is 6 blocks, there must be 6 more successfully mined blocks appended to the blockchain after that transaction's block.



THE BITCOIN TRANSACTION LIFE CYCLE



Proof of Work

Why is this necessary?

-Ensure that everyone can have the same copy of the blockchain.

-If everyone can produce blocks very quickly, there would be many conflicts among mining nodes as they try to make their chain the longest.

Proof of Work

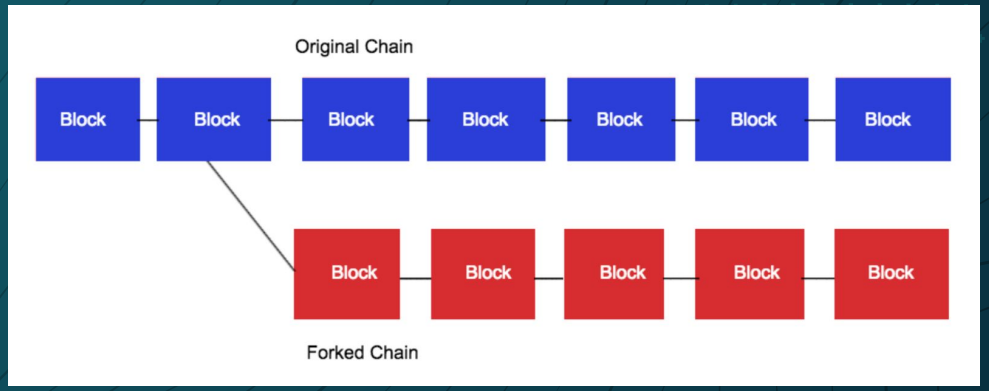
Pros:
-Less hash power is wasted. More nodes working on one chain rather than attempting to build their own.
-Better security.
-Less forks and less re-organization time as collisions are resolved faster.
-Less bandwidth and node to node communication required as there are less collisions.
-Less physical memory required. More blocks would mean, more transactions, more memory.

51 Percent Attack

- Allows a group controlling 51% of block hashing power control over the blockchain
- Allows the controlling group to re-write the blockchain.
- Uses the group's ability to produce blocks faster than all other nodes.
- Allows the malicious group to “double spend”
  - Spend currency and then reverse the transaction after goods are acquired.

51 Percent Attack

- Requires a majority of block hashing power(still possible with less but not guaranteed)
- Malicious nodes will NOT broadcast their newly mined blocks.
- They keep building a seperate blockchain



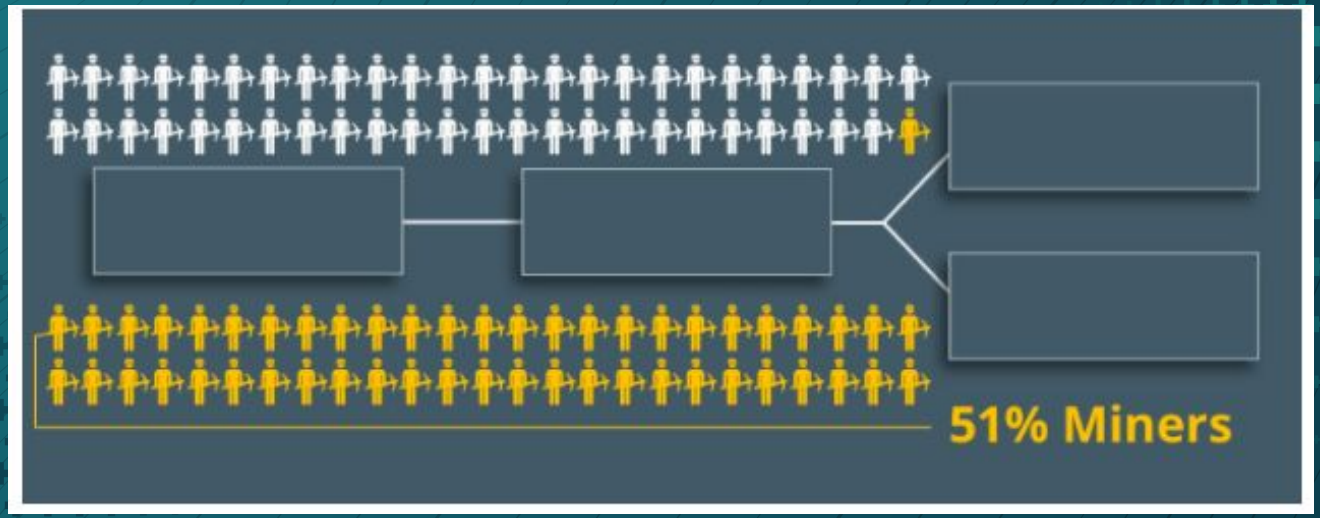
51 Percent Attack

- The malicious blockchain will eventually become longer than the original.
- When that happens, the malicious nodes will broadcast their chain.
- Since the longest blockchain is a valid blockchain, the original is no longer valid.
- All transactions and blocks in the original blockchain are no longer valid.
- Those transactions are effectively reversed.
- All legitimate nodes would be subscribed to the malicious blockchain given that the blocks and transactions within it are valid.
  - If this is the case, how can the malicious body profit?
    - Answer: The Double Spend

51 Percent Attack

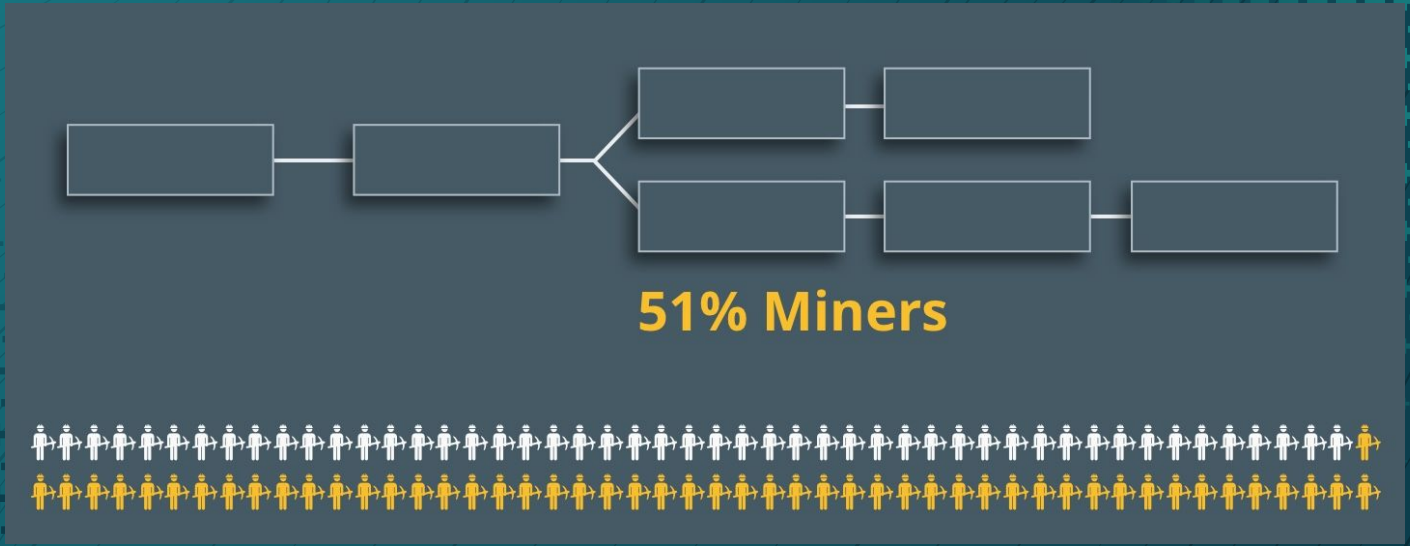
1	Start by having your malicious nodes create their own chain without broadcasting blocks.
2	Purchase a material good(preferably Money) with Cryptocurrency and broadcast your transaction (This part is legitimate).
3	Wait for your transaction to be added to the legitimate blockchain and wait until it is verified and confirmed(past the block threshold).
4	Once your transaction is approved, and you effectively legally own the goods you purchased, your malicious nodes then broadcast their longer blockchain.
5	Your transaction is effectively reversed and your spent Cryptocurrency is back in your account while you still own your goods.

51 Percent Attack



IF YOU OWN 51% OF THE HASHING POWER...

51 Percent Attack



IF YOU OWN 51% OF THE HASHING POWER...
YOUR CHAIN GETS ACCEPTED BY MAIN CHAIN

51 Percent Attack

51 Percent Attack

Do they actually happen? Yes.

- Ethereum Classic 51 Percent Attack
  - https://bravenewcoin.com/insights/etc-51-attack-what-happened-and-how-it-was-stopped
  - Attacker successfully executed different attacks amounting in over $1 million in loss for different cryptocurrency exchanges.
  - The neither the money or ETC were recovered.
  - Actions taken by exchanges limited and prevented further attacks.
    - Increased confirmation threshold
    - Blacklisted addresses/wallets

51 Percent Attack

- Bitcoin Cash 51 Percent Attack
  - https://www.coindesk.com/bitcoin-cash-miners-undo-attackers-transactions-with-51-attack
  - An unknown miner took advantage of bug to mine empty blocks and taking coins that they were not supposed to have access to.
  - Miners from BTC.com and BTC.top combined their mining resources to undo that miner's transactions.

51 Percent Attack

Prevention? Not clear cut.

Proposed solutions have pros and cons.

>>> Use a greater threshold for confirming transactions.

>>> Blacklist malicious addresses and wallets.

>>> Alternate consensus algorithms.

>>> Invalidate deep re-orgs.

Key Takeaways

- A Blockchain is made of blocks, which is made of transactions
- The blocks in the chain are linked based on the previous hash of the n-1 block
- Blockchain is “decentralized”
- Transactions are confirmed when blocks are mined (added to chain)
- PoW -- add a block to a chain. Need to find the right “calculated random nonce” value to match the difficulty setting of the chain
- Main chain is updated based on the length of the “merging chain” -- takes longest sub chain

Why Use Cryptocurrency?

- Anonymity
  - Only requires a digital wallet.
- Decentralized
  - All nodes keep copy of all transactions.
  - Don’t need to trust central authority.
- Alternative to Legal Tender