Bitcoin: A Peer-to-Peer Electronic Cash System

By Dhruv Krishnan and Priya Holani

UCDAVIS

Bitcoin: A Peer-to-Peer Electronic Cash System

By Dhruv Krishnan and Priya Holani

UCDAVIS
UNIVERSITY OF CALIFORNIA

Introduction

UC DAVIS
UNIVERSITY OF CALIFORNIA

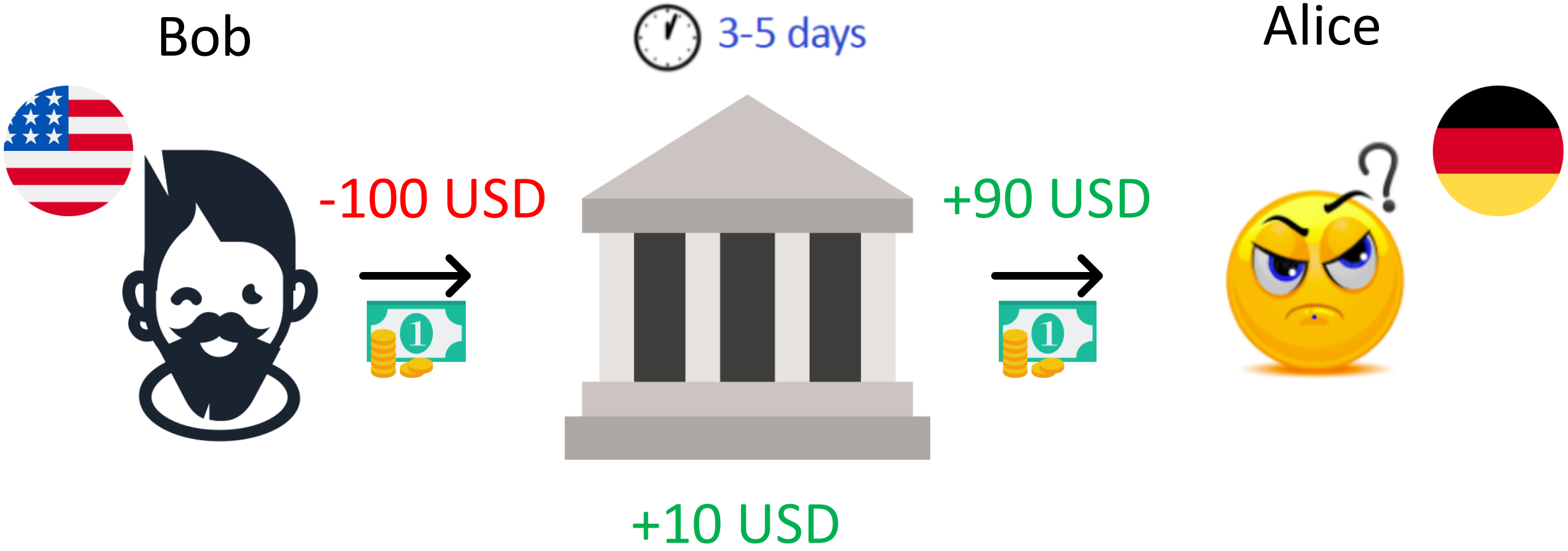
Traditional Transaction Model

Bob

Bank

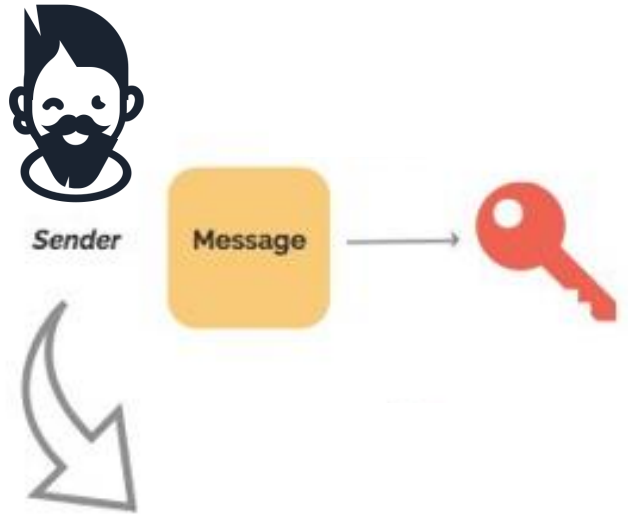
Alice

Problems with the traditional model



Digital Signatures

Bob



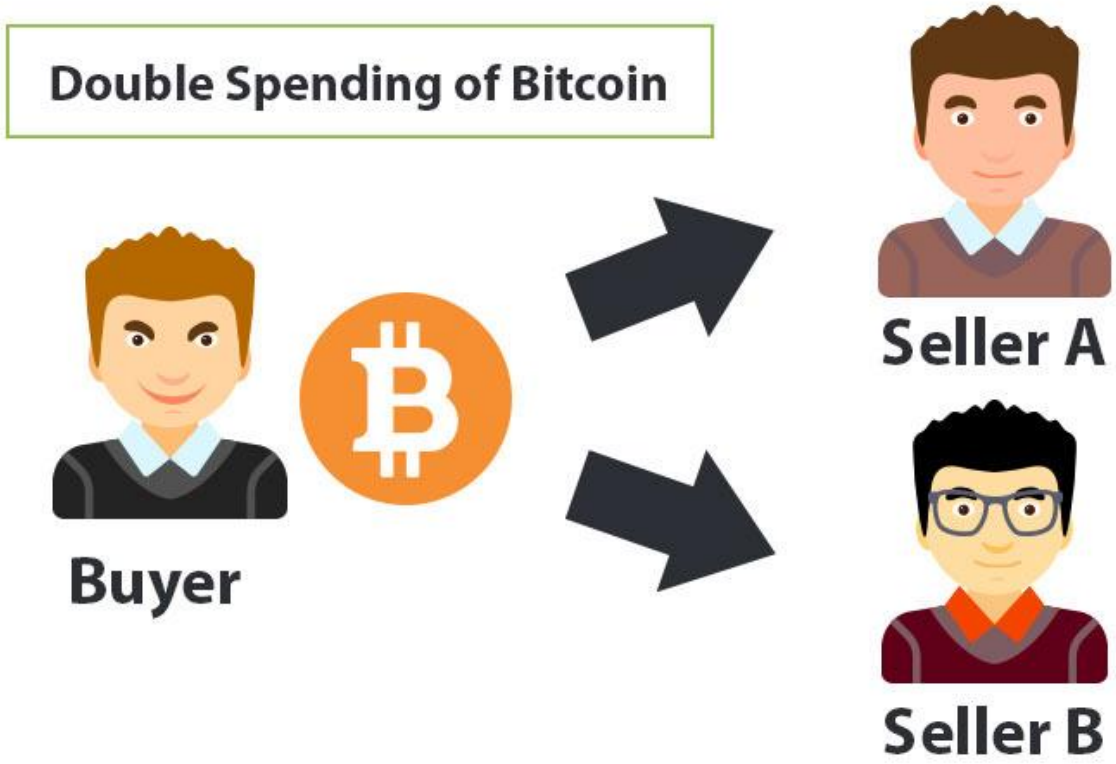
Alice

The message is signed with Bob's private key to prove his identity. This is like logging with your private bank details.

The message is sent to Alice

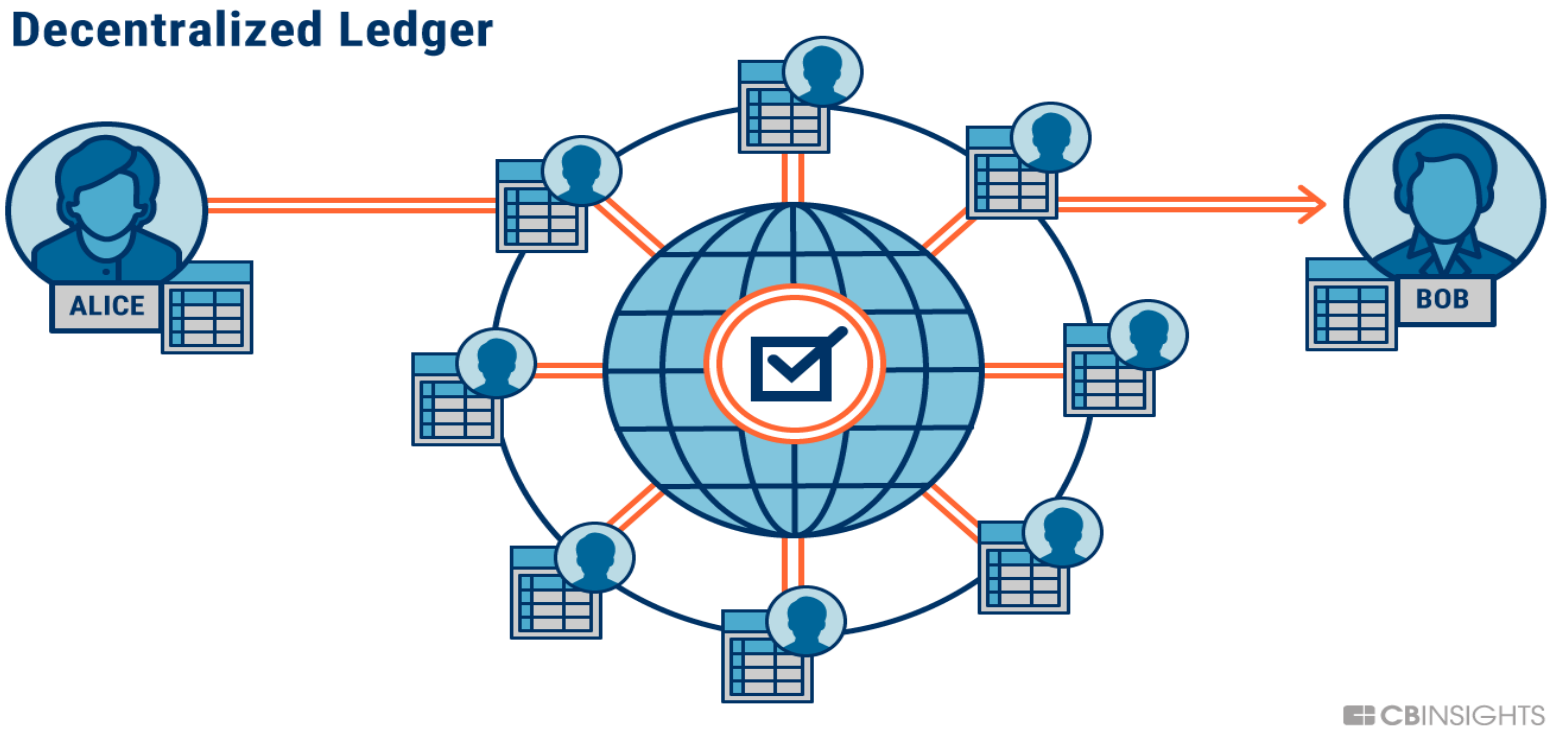
Bob's public key is attached to the message, so Alice knows it is really him

The Double Spending Problem



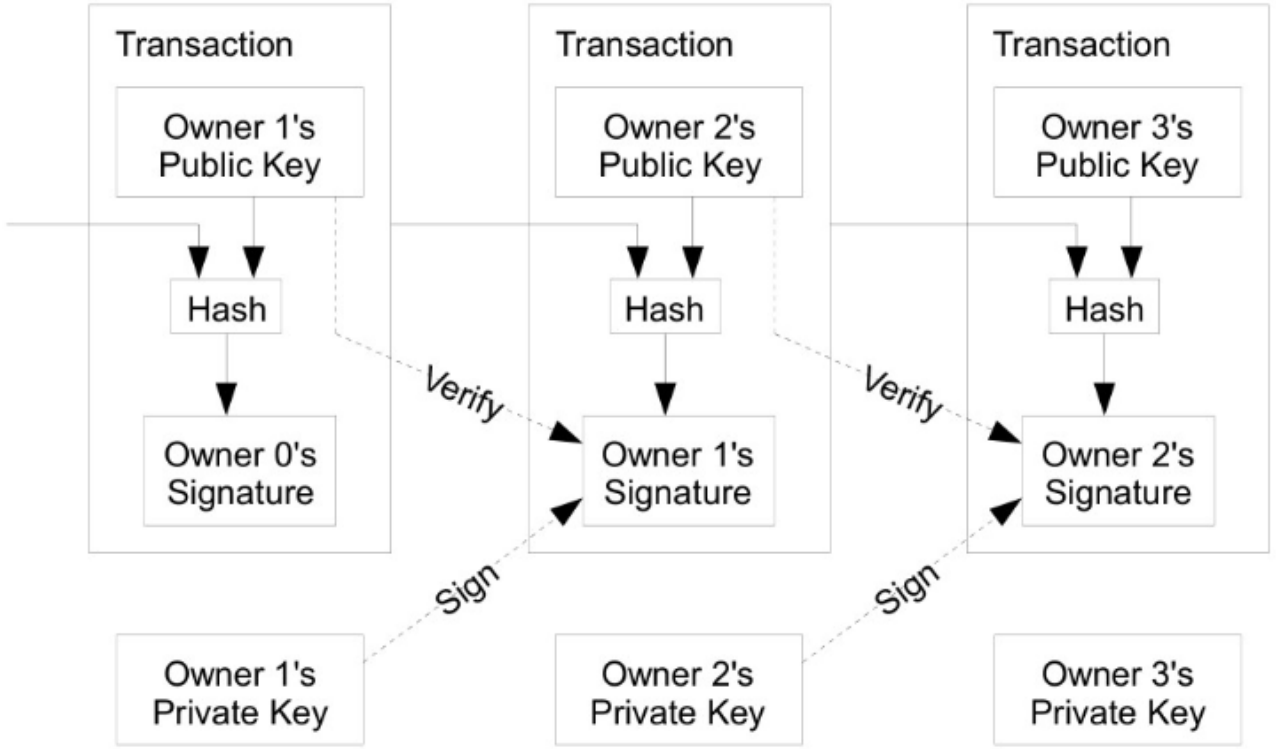
90's problems

Bitcoin Solution



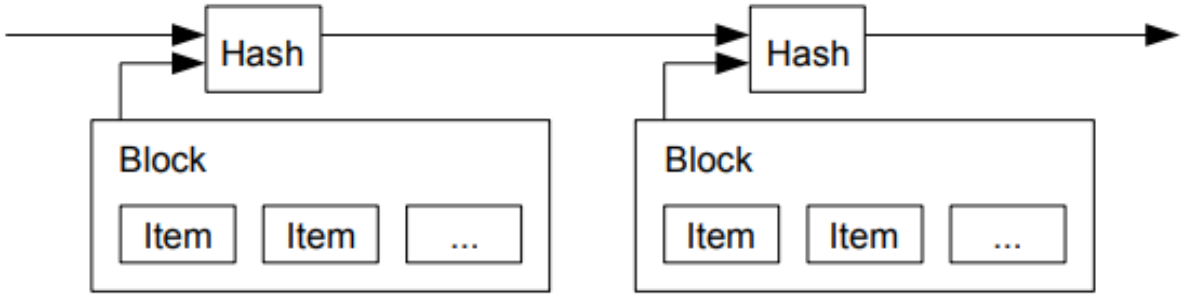
Transactions

- Electronic coin - chain of digital signatures.
- Each owner :
  1. Digitally signs a hash of the previous transaction and the public key of the next owner
  2. Adds these to the end of the coin.
- A payee can verify the signatures to verify the chain of ownership.



Timestamp Server

- The only way to confirm the absence of a transaction is to be aware of all transactions that happened before.
- Need a system for the participants to agree on a single history of the order in which they were received.
- A Hash of a block is taken to be timestamped and widely published.
- Each timestamp includes the previous timestamp in its hash, forming a chain, with each additional timestamp reinforcing the ones before.



Proof of Work

- Consensus protocol in order to decide what the next block in the chain is.
- The node that solves a complicated cryptographic puzzle g ets to decide the next block.
- Nodes provide consensus by verifying this block, generating a proof-of-work for a new block and attaching it to the current chain.

Proof of Work

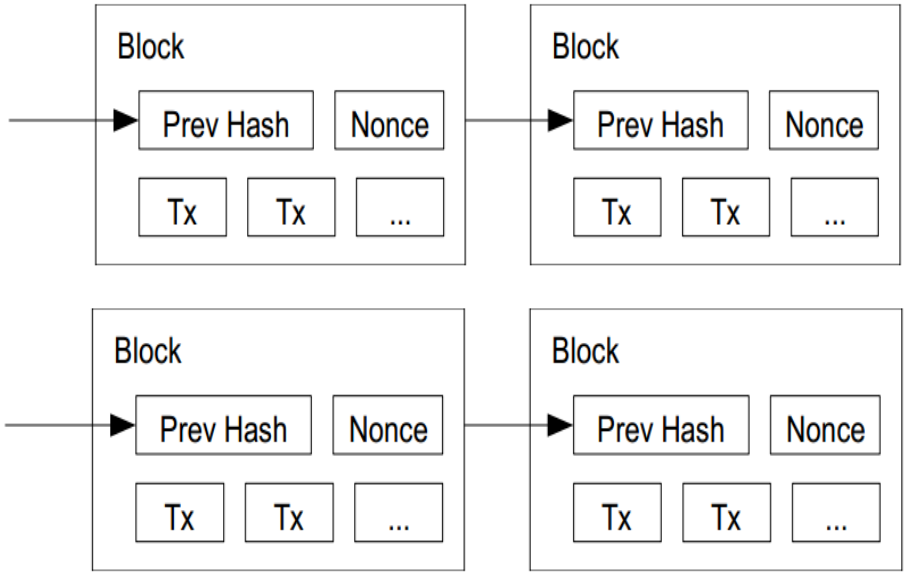
- Solves the problem of determining representation in majority decision making - **One-CPU-one-vote**.
- POW requires that **honest nodes possess the majority of the computational power in the network.**
- **Proof of work difficulty** is determined by the avg number on blocks you want to produce per hour – the difficulty of the puzzle depends on whether the number of blocks produced per hour is more than the average. If yes, then it means that the difficulty needs to be increased.

UCDAVIS

Tie breaking

- Two nodes may find a correct block simultaneously.
  - Keep both and work on the first one
  - If one grows longer than the other, take the longer one

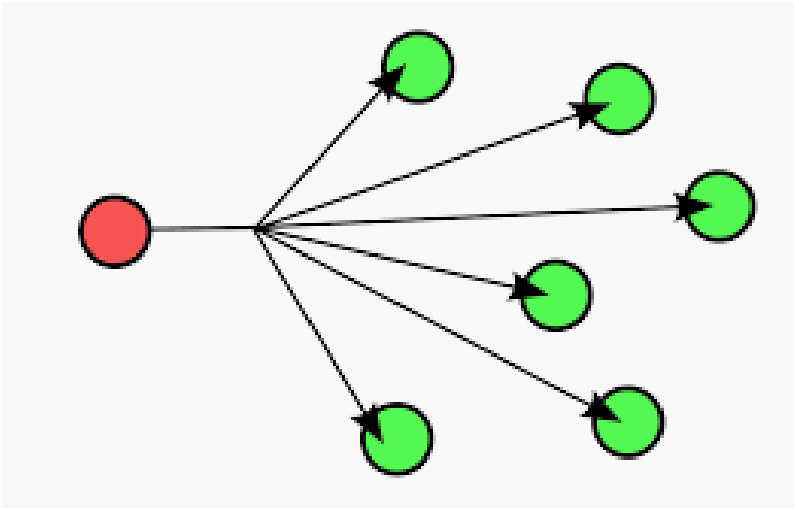
Two different block chains (or blocks) may satisfy the required proof-of-work.



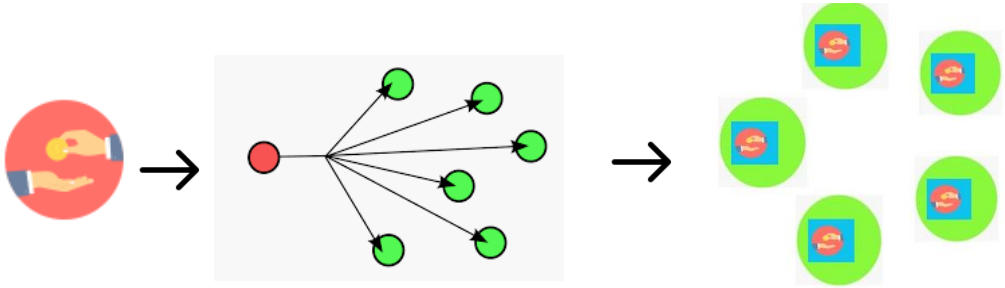
How does the bitcoin Network actually run?

Network

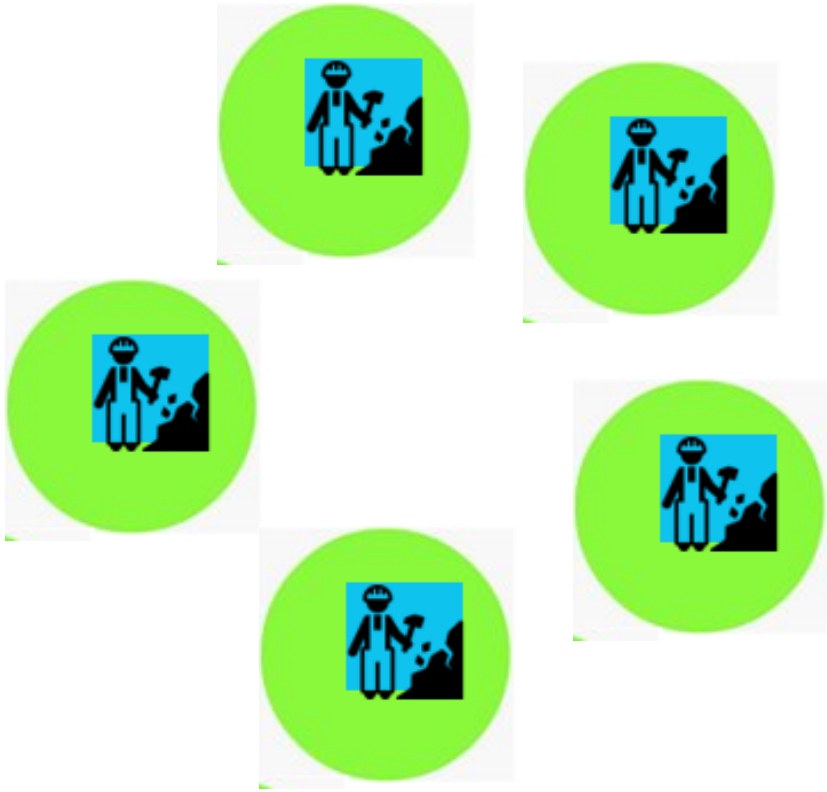
1) New transactions are broadcast to all nodes.

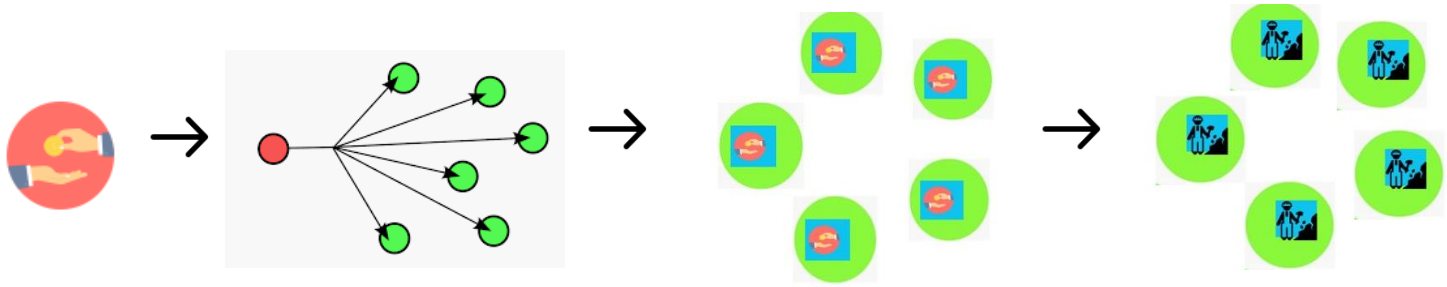


2) Each node collects new transactions into a block.

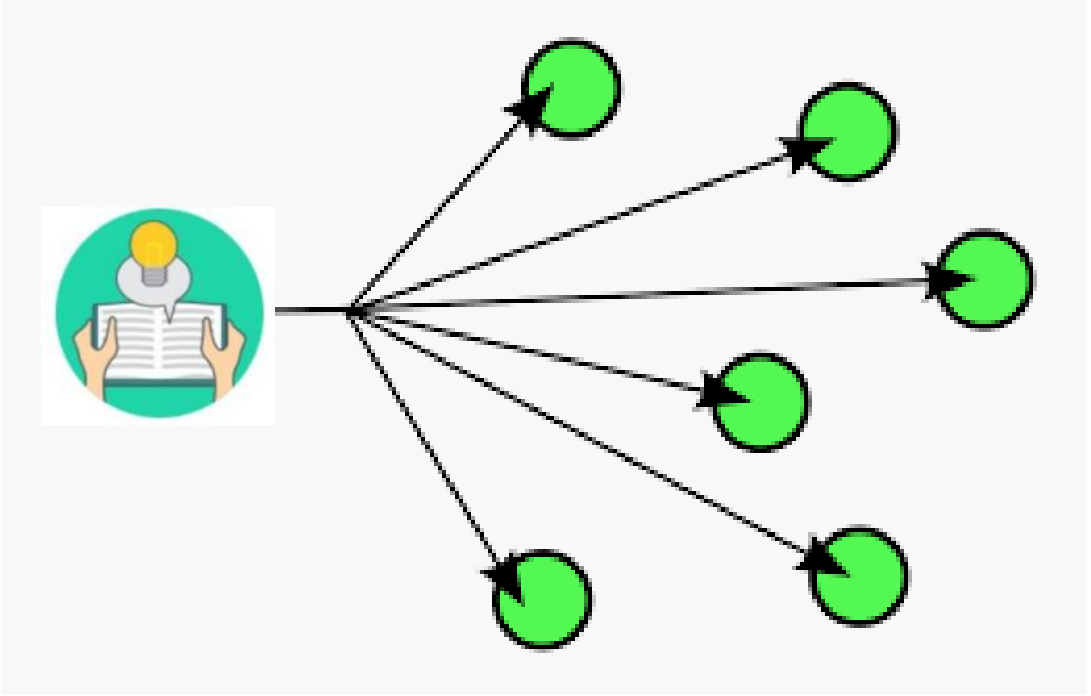


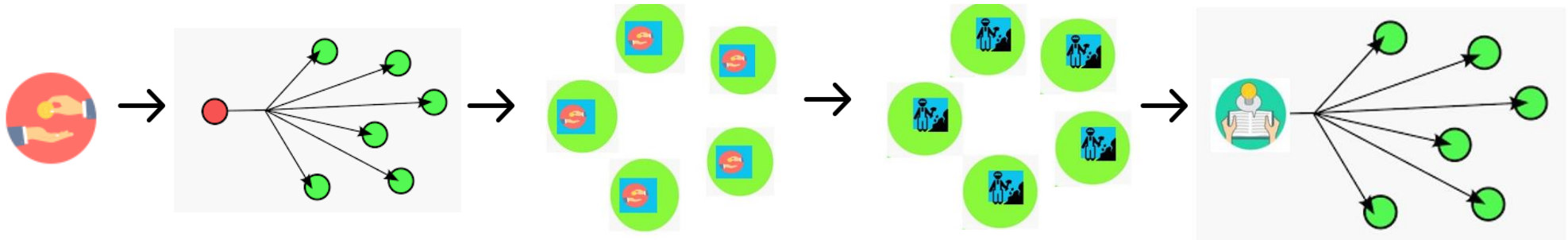
3) Each node works on finding a difficult proof-of-work for its block.



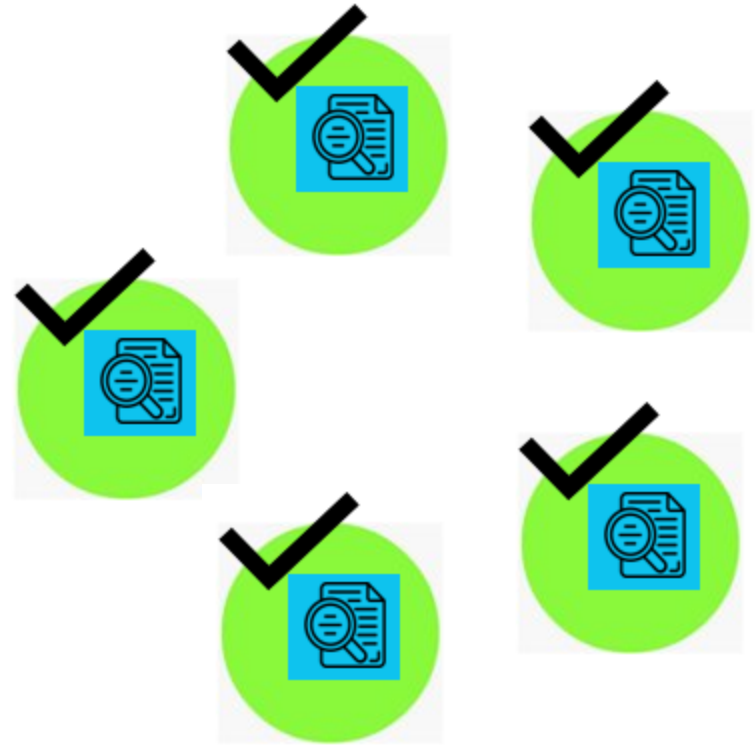


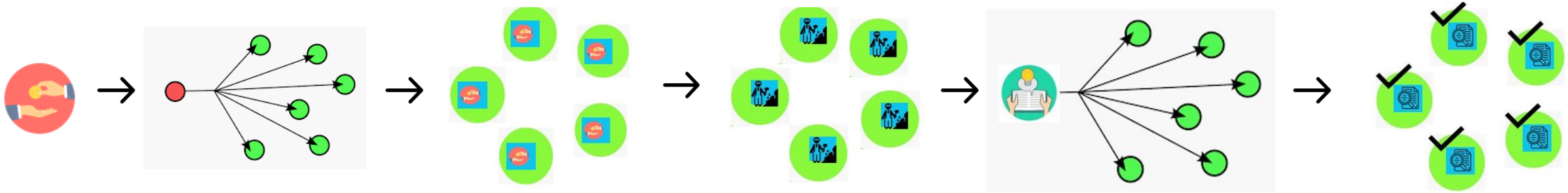
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.



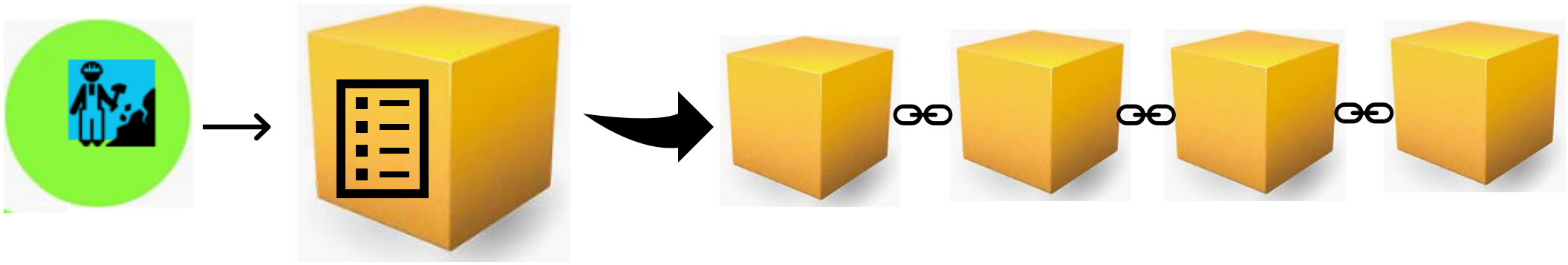


5) Nodes accept the block only if all transactions in it are valid and not already spent.





6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.



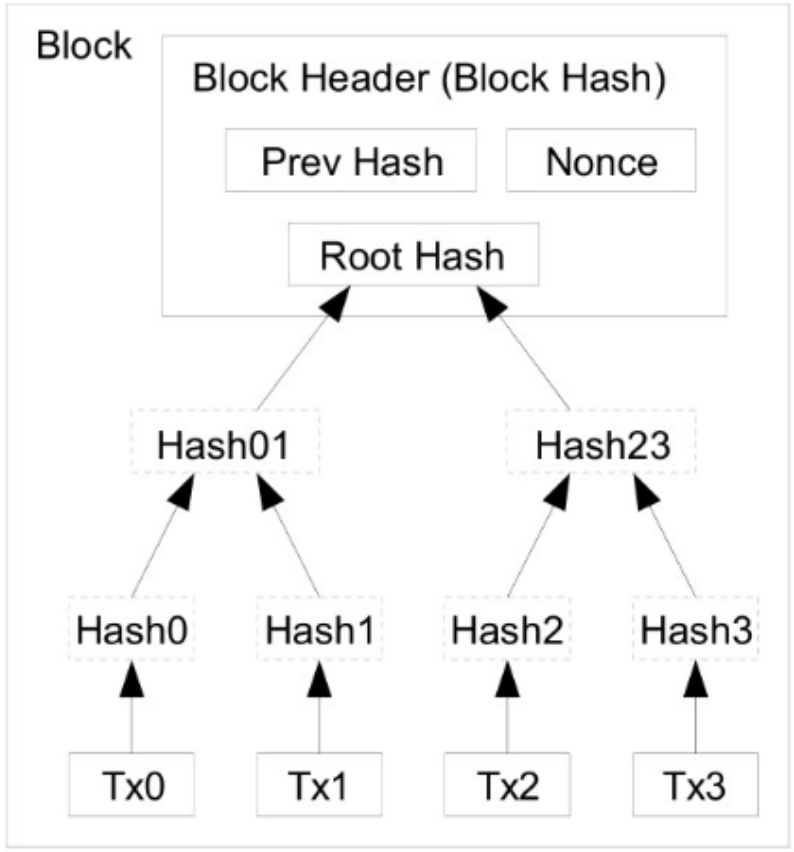
Block creation using
Hash of Previous
Accepted block

Why would nodes support the network?

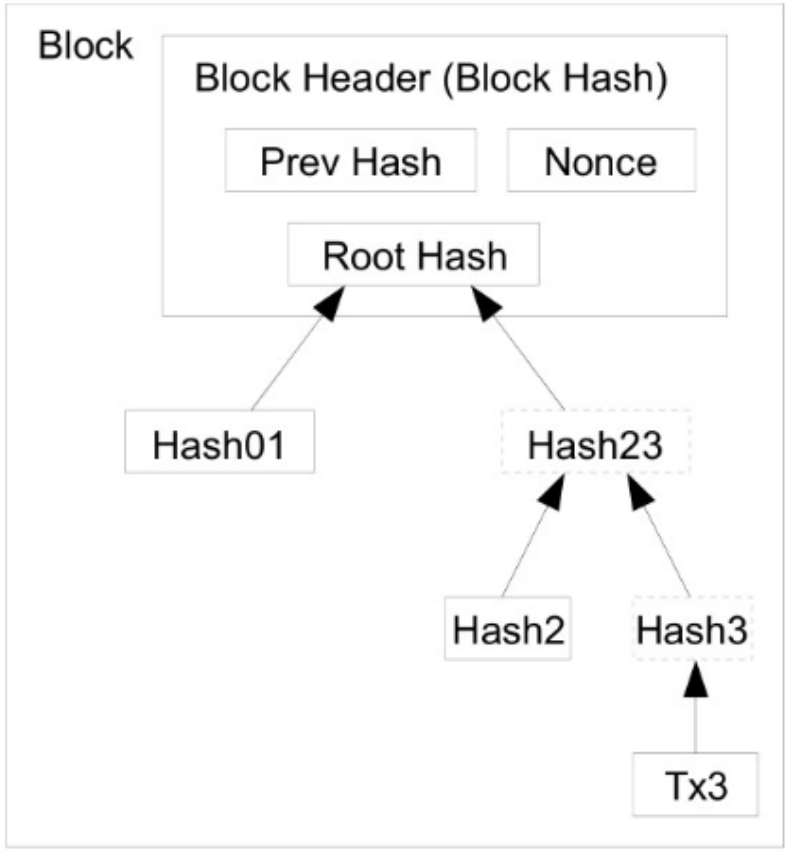
Incentives

1) The first transaction in a block is a special transaction that starts a new coin owned by the creator of the block.
   - This provides an incentive for nodes to support the network by distributing coins into circulation.
2) A transaction can include a transaction fees which incentivizes a miner to include it into the block they are mining.

Reclaiming Disk Space (Transactions hashed in a Merkle Tree)



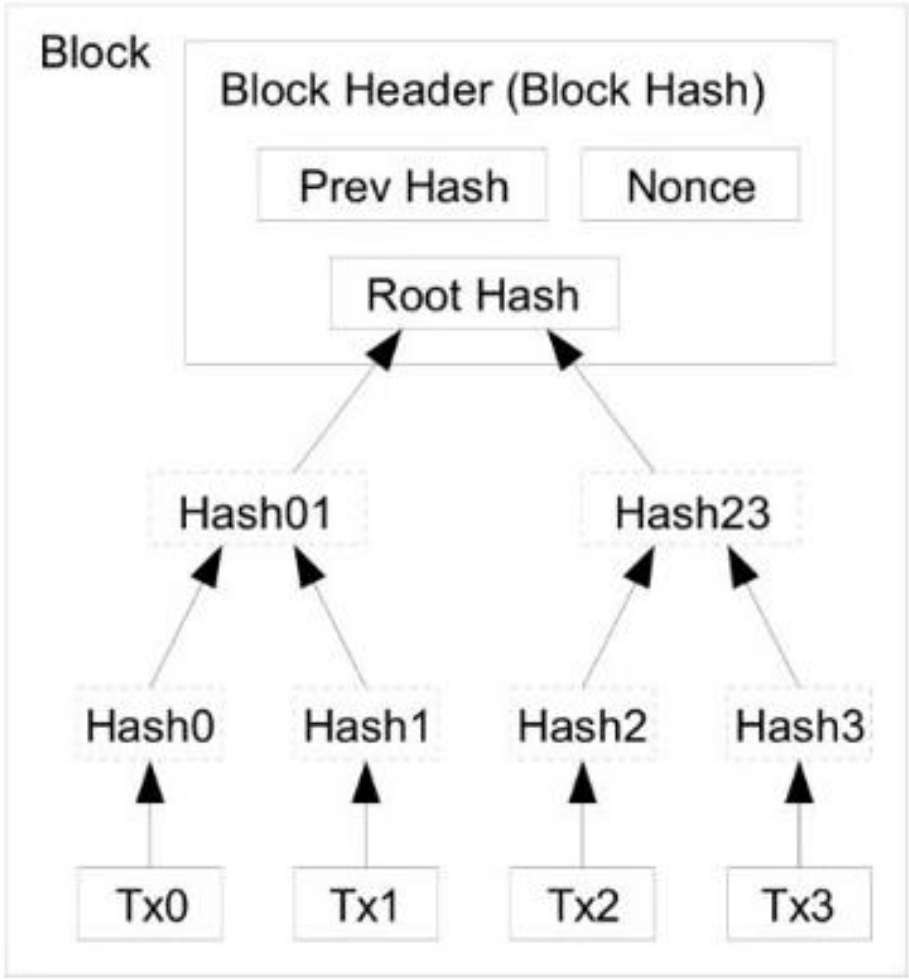
Transactions Hashed in a Merkle Tree



After Pruning Tx0-2 from the Block

Possible Savings

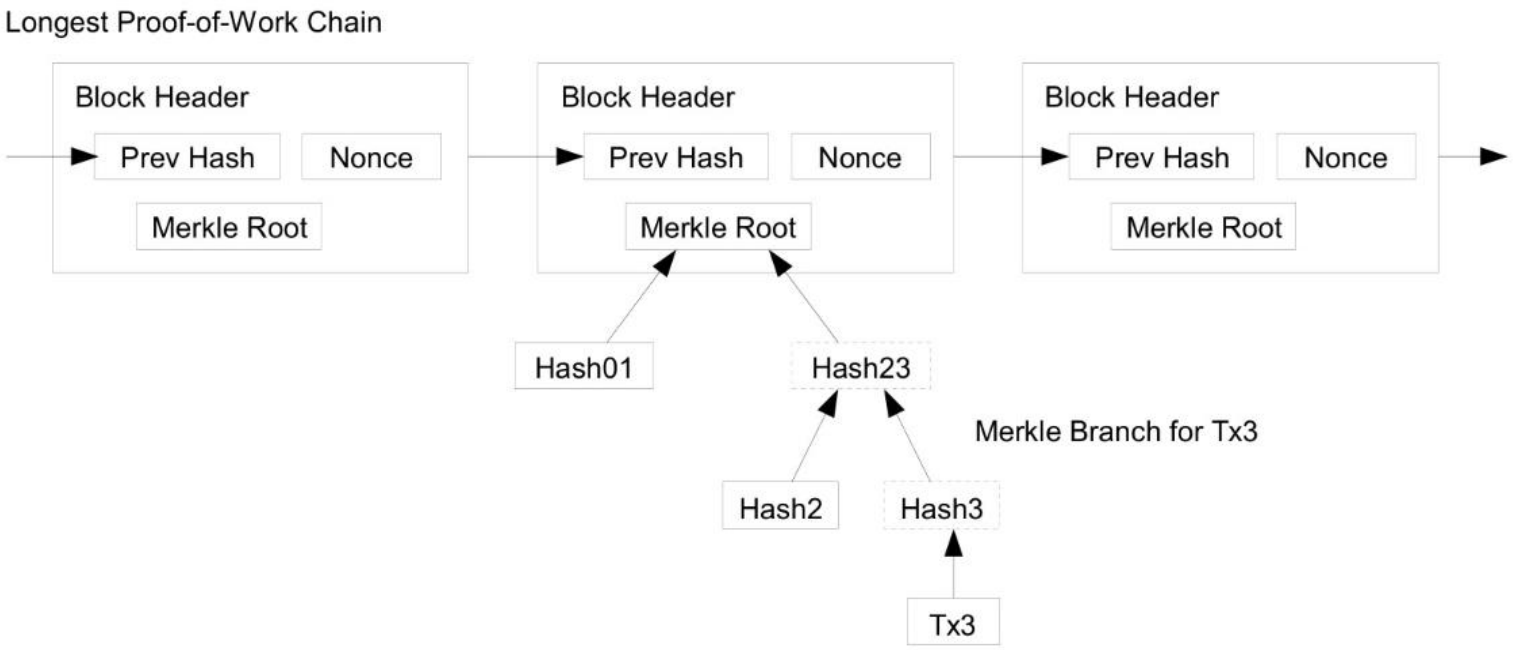
- Average size of each block (storing whole transactions):
  0.97 MB
- Size if only root hash is used to store transaction list:
  80 Bytes



Transactions Hashed in a Merkle Tree

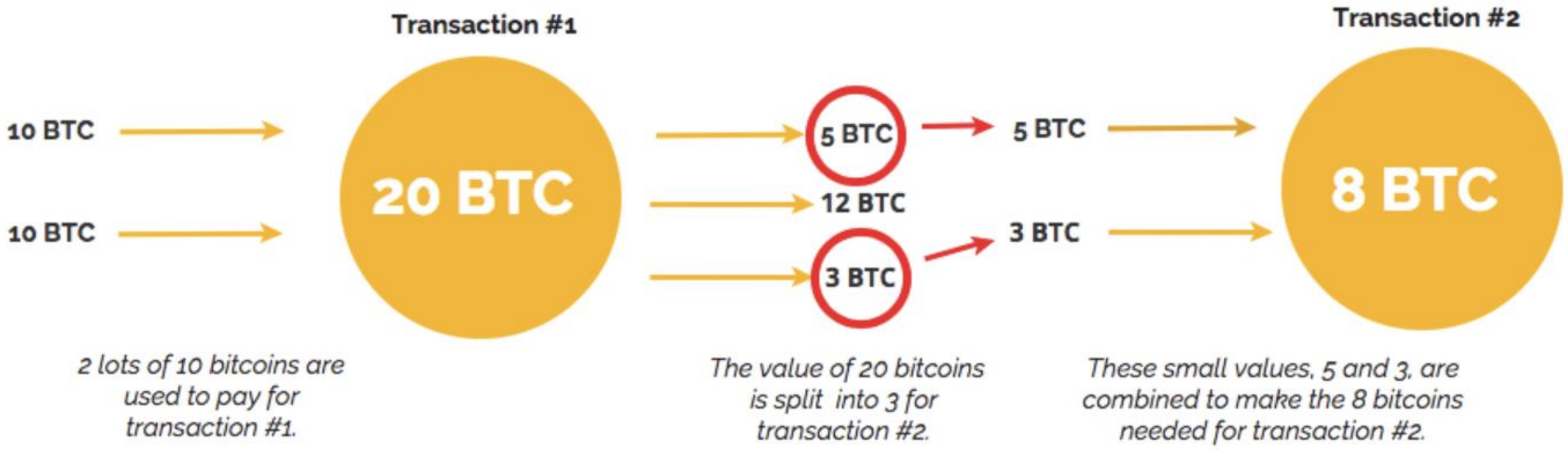
Simplified Payment Verification

- A node can verify a transaction by querying from other nodes.
- First, get the longest proof-of-work chain.
- Query the block that the transaction to be verified (Tx3) is in.
- Only need Hash01 and Hash2 to verify; not all the transactions.



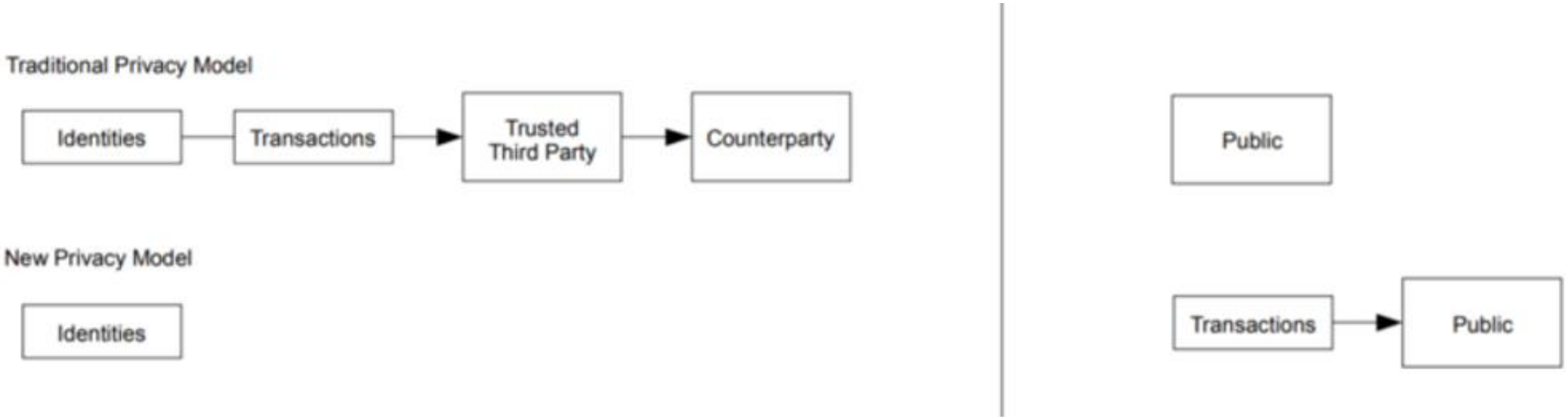
Combining and Splitting Values

Splitting a group bitcoin into smaller values. This allows all transactions to be processed in the blockchain, no matter how small the value.

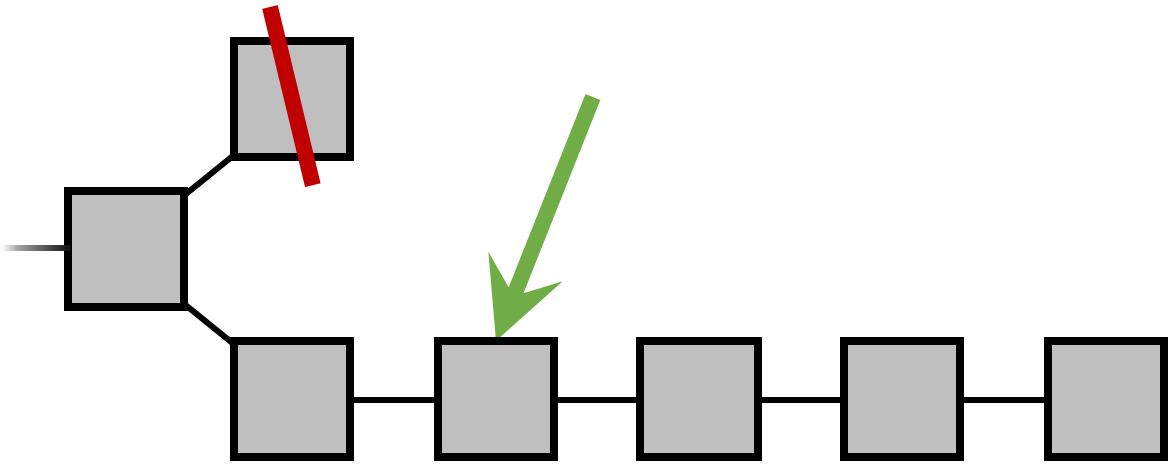


Privacy

- Pseudo-anonymous: Transactions can be tracked using public key.
- Can create new key-pairs for new transactions to prevent this.



How much time will it take to confirm a transaction?

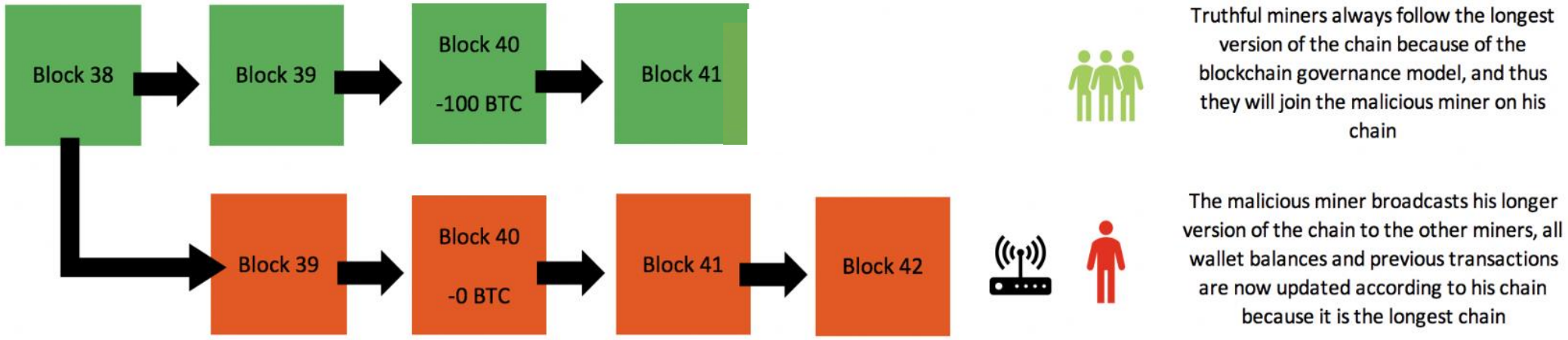


A transaction is **confirmed** when it is **buried** deep enough

Nodes Verifying Bitcoin Transactions

Double Spending by forking

- Consider the scenario of an attacker trying to fork an alternate chain faster than the honest chain.
- The attacker will have succeeded in modifying a block if the alternate chain becomes longer than the honest chain.

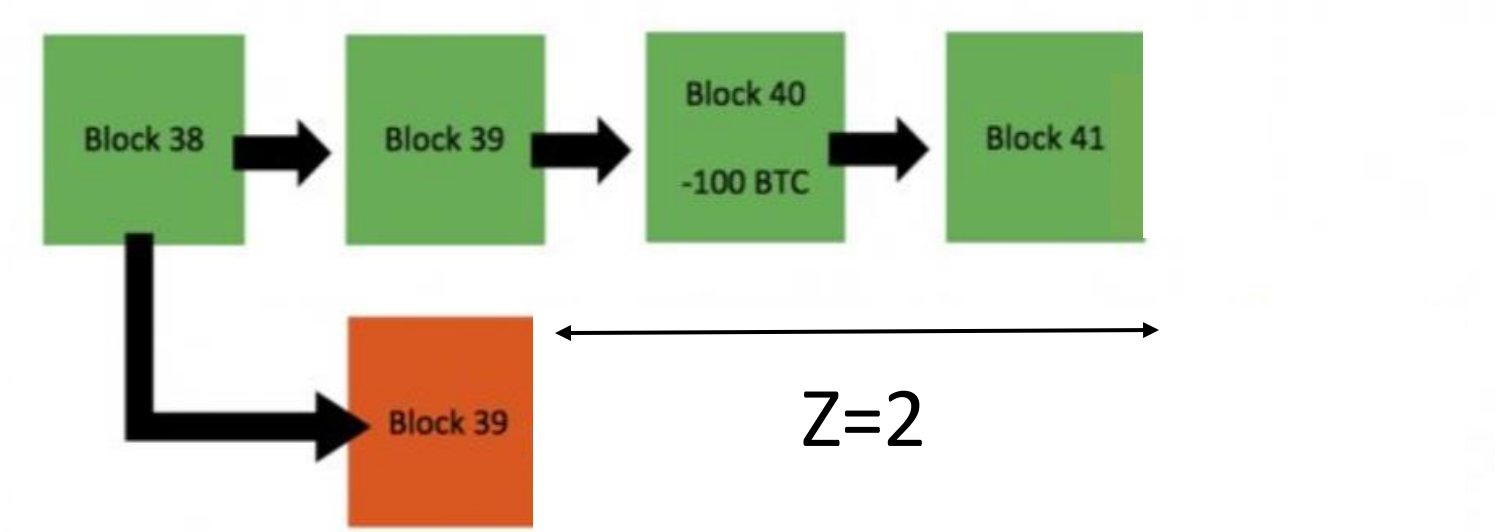


The corrupt miner broadcasts its chain to the rest of the network once it is longer (heavier) than the original chain.

UCDAVIS

Calculation

If the alternate chain is 'z' blocks behind the honest chain, assume that:
P = Probability that the attacker catches up with the honest chain from 'z' blocks behind.
q= % of computational power of the network held by the attacker.



Probabilities when attacker owns 10% of the network

q=10%

(Attacker owns 10% of the network)

z=0 -> P=1.0000000

z=1 -> P=0.2045873

z=2 -> P=0.0509779

z=3 -> P=0.0131722

z=4 -> P=0.0034552

z=5 -> P=0.0009137

z=6 -> P=0.0002428

q-> % of computational power of the network that the attacker owns.

z-> Number of blocks to wait to confirm the transaction.

P-> Probability that an attacker will be able to create a longer chain.

Choice of 'z' based on probability

P < 0.001

q=10% -> z=5

q=15% -> z=8

q=20% -> z=11

q=25% -> z=15

q=30% -> z=24

q=35% -> z=41

q=40% -> z=89

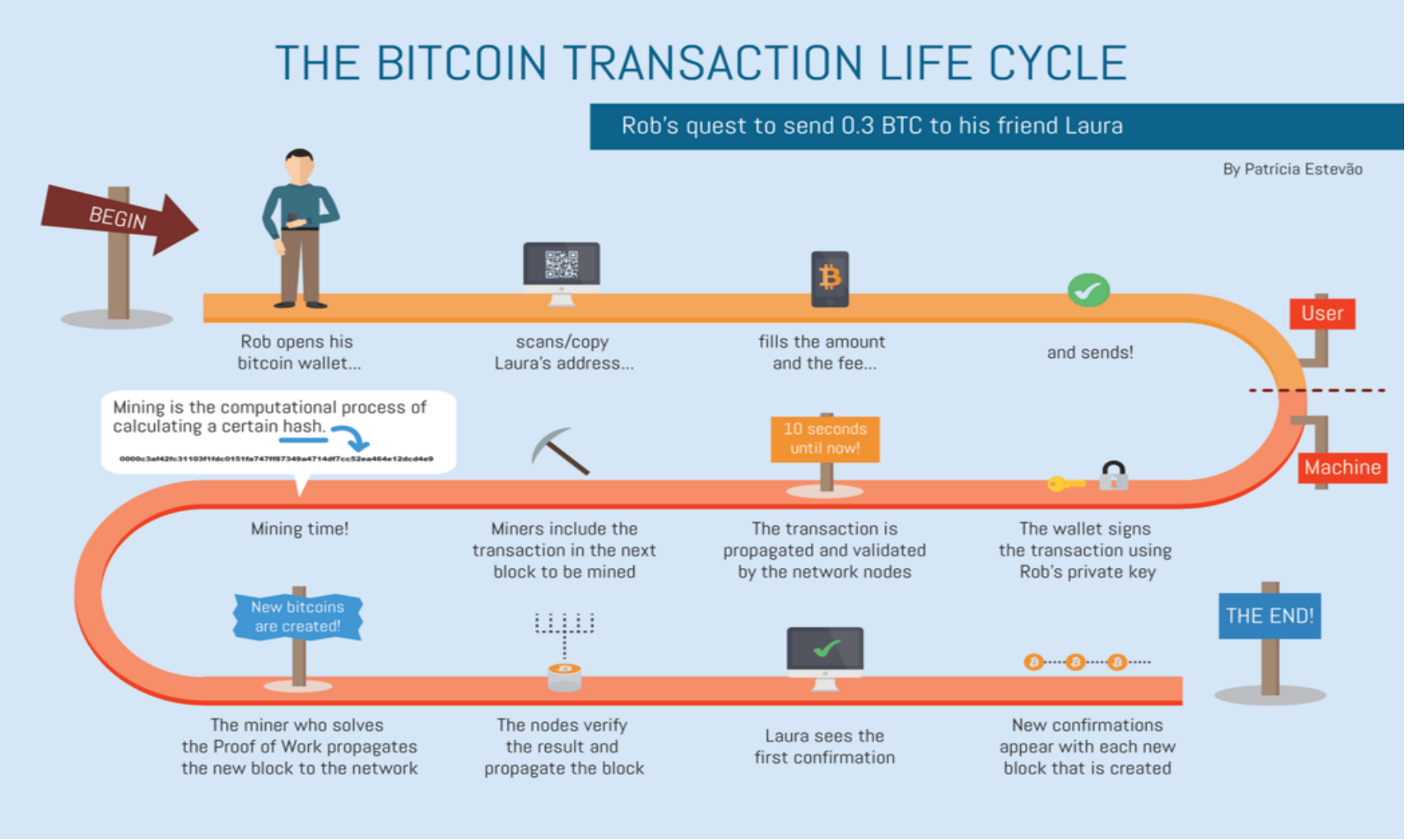
q=45% -> z=340

q-> % of computational power of the network that the attacker owns.

z-> Number of blocks to wait to confirm the transaction.

P-> Probability that an attacker will be able to create a longer chain.

The complete picture.



UCDAVIS

The paper is an awesome read!
https://bitcoin.org/bitcoin.pdf

Scan me