# Blockchain Technology and Applications

## CS 731

Consensus in Bitcoin

Dr. Ir. Angshuman Karmakar

IIT Kanpur

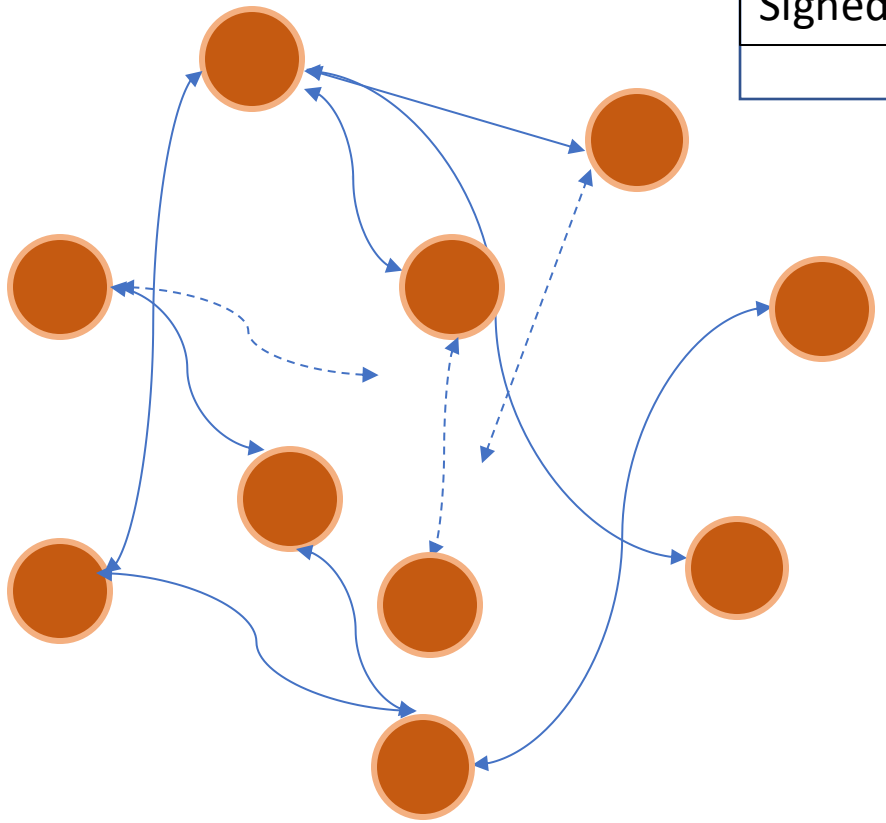
Teaching assistants

- **Sumit Lahiri (**sumitl@cse.iitk.ac.in**)**
- **Chavan Sujeet** (sujeetc@cse.iitk.ac.in)
- **Indranil Thakur** (indra@cse.iitk.ac.in)

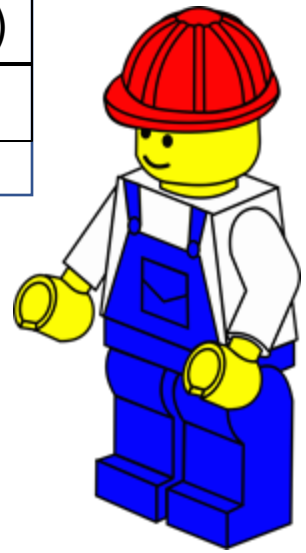
# Consensus

- Broadcast transactions to the p2p network

| PayTo $PK_{Bob}$ : $H(CID_1)$ |
| --- |
| Signed by $SK_{Alice}$ |
| |



| PayTo $PK_{Rusty}$ : $H(CID_3)$ |
| --- |
| Signed by $SK_{Bob}$ |
| |

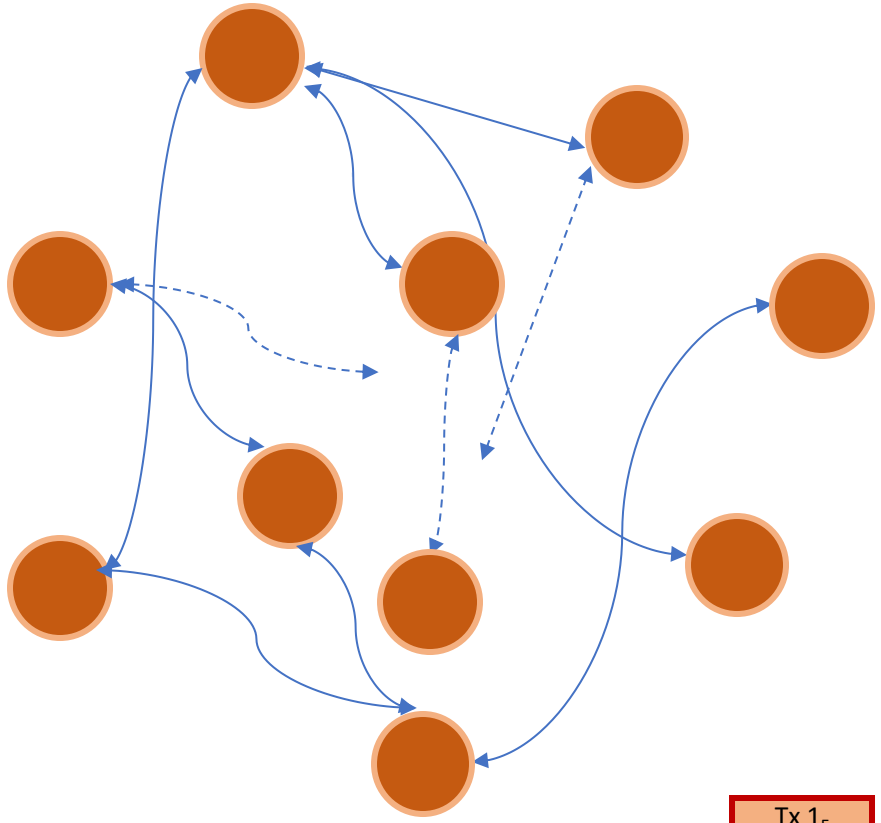


| PayTo $PK_{Rudy}$ : $H(CID_2)$ |
| --- |
| Signed by $SK_{Eva}$ |
| |

# Consensus

- Broadcast transactions to the p2p network

Tx $1_A$
Tx $2_A$
.
.
.
Tx $n_A$



Tx $1_B$
Tx $2_B$
.
.
.
Tx $n_B$

Tx $1_E$
Tx $2_E$
.
.
.
Tx $n_E$

# Consensus in Bitcoin

- A simplified Bitcoin consensus algorithm

1. New transactions are broadcast to all nodes
2. Each node composes blocks with these transactions
3. In each round, a *random* node is selected
   a. This node's block is selected as the next block
4. Other nodes accept this block
   a. Only if all the transactions are valid
   b. Funds are unspent
   c. Signatures are valid, etc.
5. Nodes express their acceptance of the block by including its hash into the next block they create

# Consensus in Bitcoin

## Lack of Identity

- There is no *persistent* identities in Bitcoin
  - Bitcoin uses cryptographic methods as identities
  - Generating public-key is trivial
  - Any node have multiple identities or public-keys
  - Anonymity was a design objective of Bitcoin

# Consensus in Bitcoin

## Lack of Identity

- Building a consensus using identities is simpler
  - Like a lottery system
    - Is not tied to real world identities
    - But there is an identity associated to each individual
  - At each round the node with certain bit arrangement will choose the next block
  - Can be held accountable if something goes wrong
- Owners can prove their ownership of their identities later

# Consensus in Bitcoin

## Sybil attack

- It is trivial and cheap to create multiple identities
- An attacker can create an enormous number of identities
- If we use a simple random node choosing method
  - The attacker will gain disproportionate advantage
  - Can block the other nodes' blocks forever
- This is called *Sybil attack,* and the nodes *Sybil nodes*
- So, the random node choosing algorithm has to be *intelligent*
- Robust against Sybil attack

# Consensus in Bitcoin

- For now let's assume
  - The random node choosing algorithm is intelligent
  - Works without persistent identities
- Lot to assume
  - We will show how these assumptions are satisfied

# Consensus in Bitcoin

- Let’s say the a node or an user Alice is chosen to propose the next block
- Assume Alice is dishonest
- What Alice can do?

# Consensus in Bitcoin

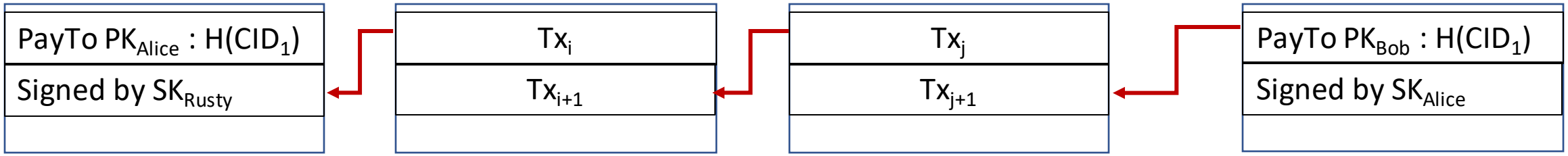
## Dishonest nodes

- Stealing Bitcoins?
- No Secured by digital signatures!
- Denial-of-Service?
- Alice excludes Bob transactions from her block
- Alice may prevent for one or more rounds
- Eventually, an honest node will be picked
- Bob’s transaction will be included in the proposed block
- Minor inconvenience

# Consensus in Bitcoin

## Dishonest nodes

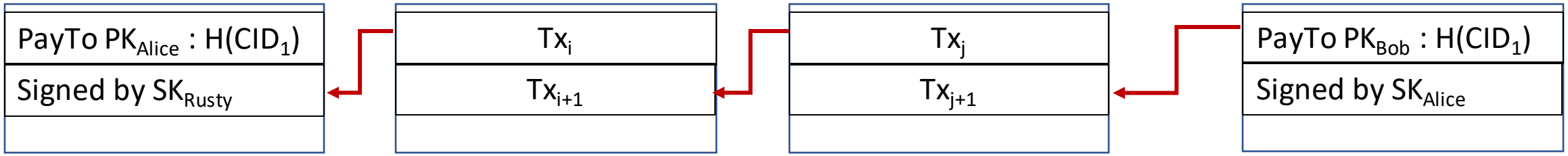
- Double-spend attack?
- Alice purchases service from Bob and pays in coins
- Broadcasts this transaction to the network
- Later, Alice pays the same coin to one of her accounts



# Consensus in Bitcoin

## Double-spend attacks

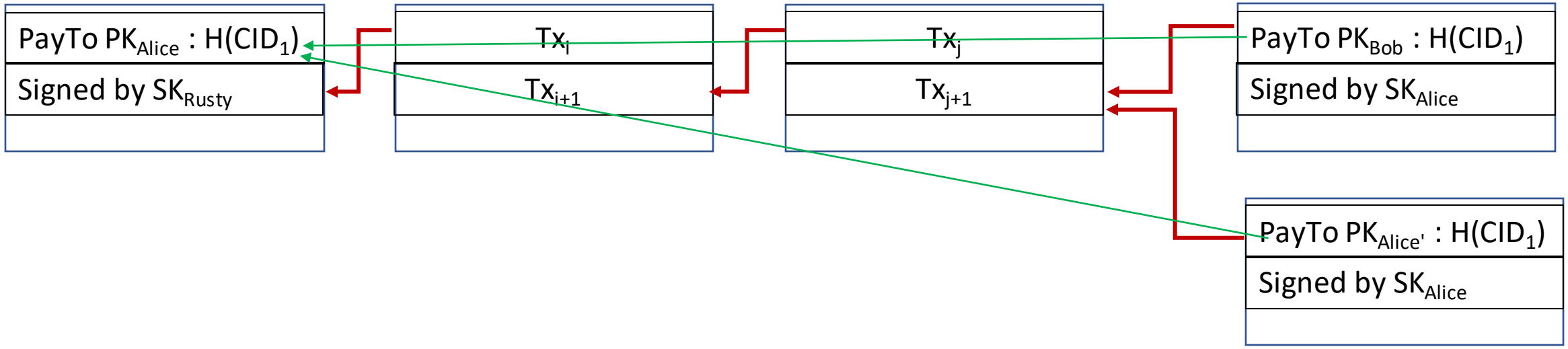
- Zero confirmation transaction
  - Bob ships the item as soon as he sees this transaction
- This can be included by an honest node



# Consensus in Bitcoin

## Double-spend attack

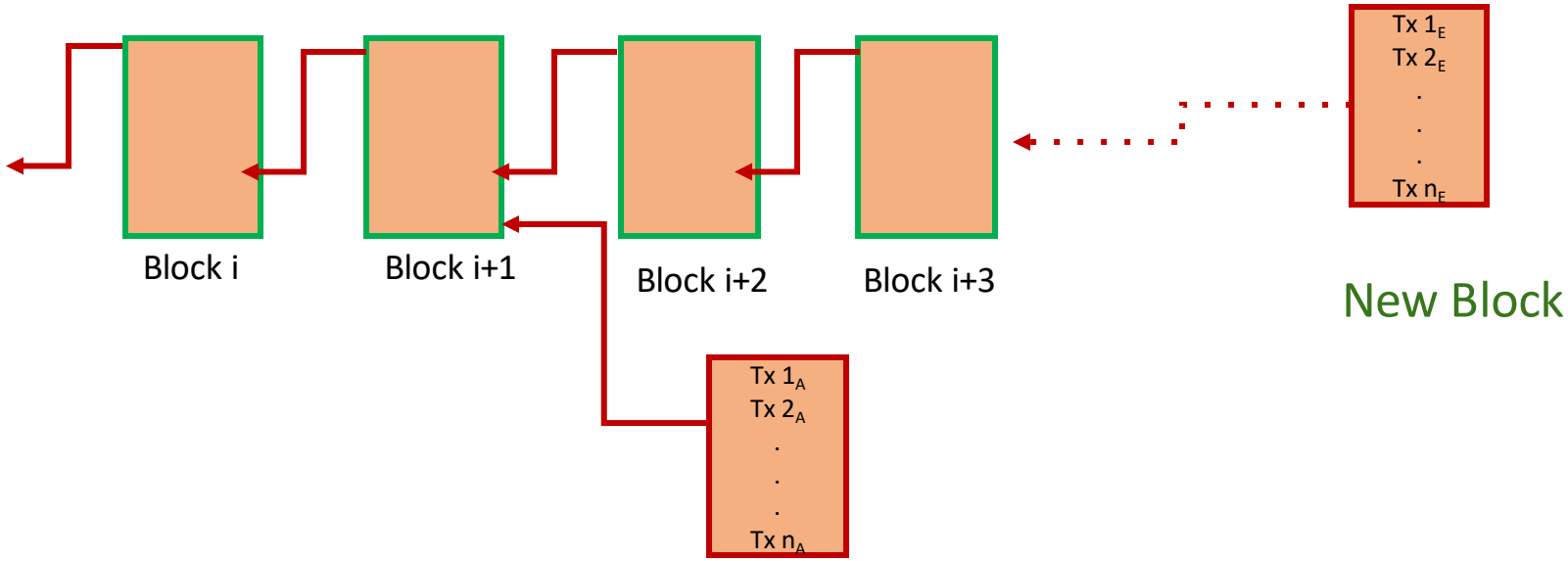
- Suppose Alice is chosen as the next random node
- She creates a payment to herself and ignore the previous node



# Consensus in Bitcoin

## Double-spend attack

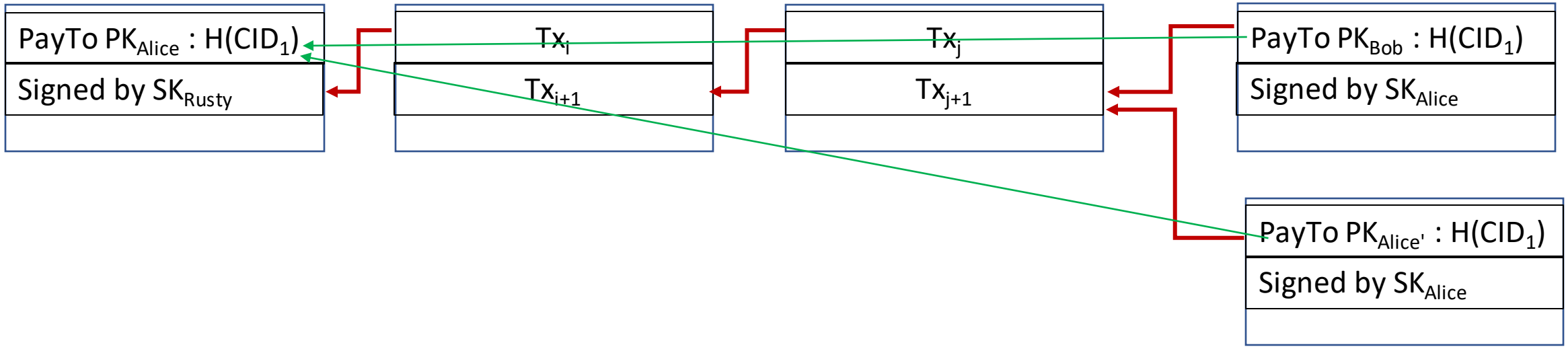
- Principle of "Extending longest valid chain"
- Honest nodes always adds their blocks to the longest valid chain



# Consensus in Bitcoin

## Double-spend attack

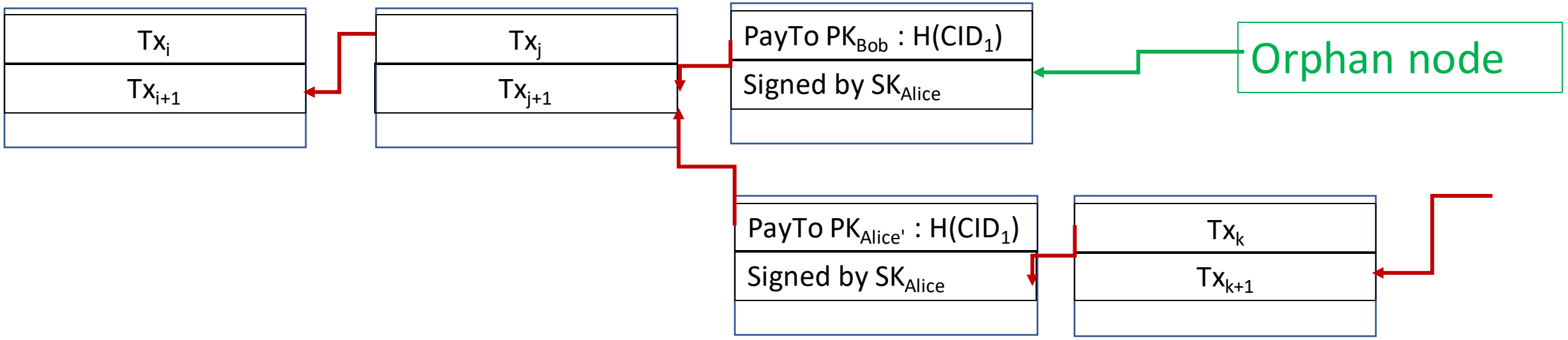
- Technically both chains are valid
- Network latency
  - Some nodes may hear Alice--> Alice transaction before Alice--> Bob transactions



# Consensus in Bitcoin

## Double-spend attack

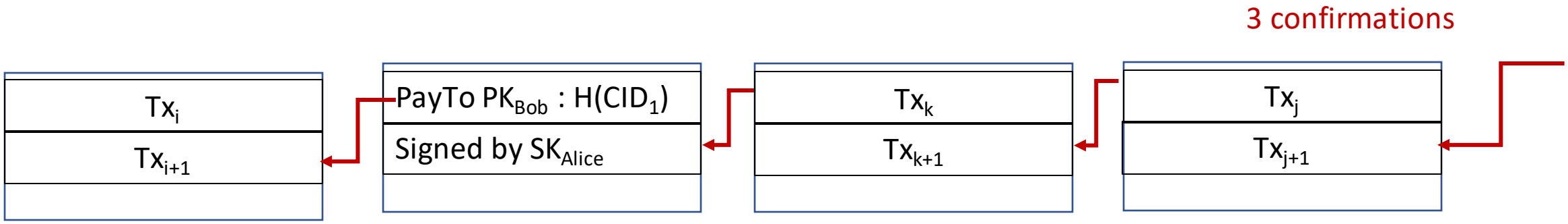
- Honest nodes build on the double-spend branch
- Alice can even bribe the next node
- The double-spend branch gets included in the chain
  - Successful attack



# Consensus in Bitcoin

## Bob's view

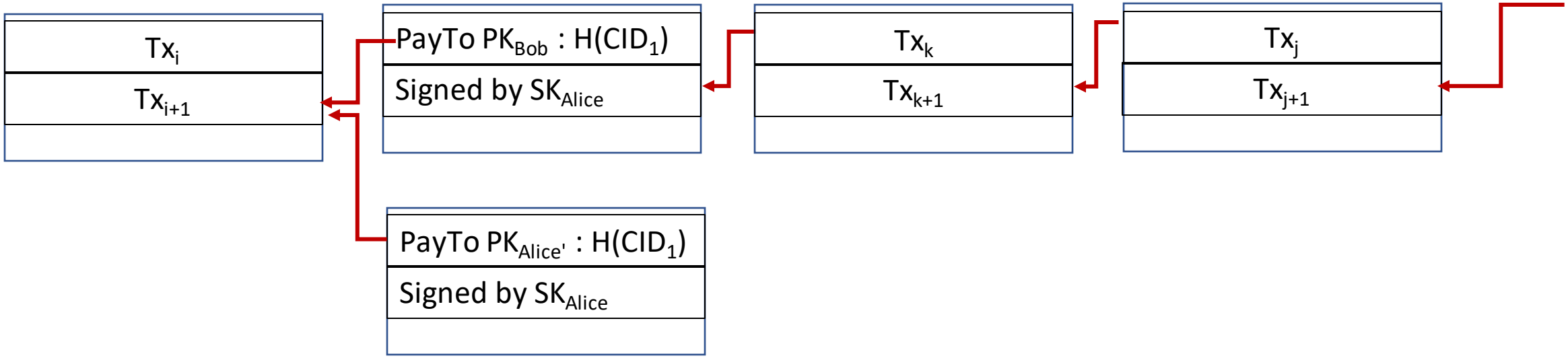
- Bob does not ship the item at the first confirmation
- Waits for more confirmations
- Only ships after enough confirmation
  - *De facto* standard is 6 in Bitcoin



# Consensus in Bitcoin

## Bob's view

- Alice can try to double spend
- Honest nodes will reject the block
- Longest valid chain



# Consensus in Bitcoin

## Bob's view

- Double-spend transaction probability decreases exponentially
  - With each confirmation
- Never 100% chance that the transaction is a success
- Virtually impossible after 6 transactions
  - Takes 1 hour!!!!!

- Recap
  - Protection against invalid transactions are cryptographic
  - Protection against double-spend is purely by consensus

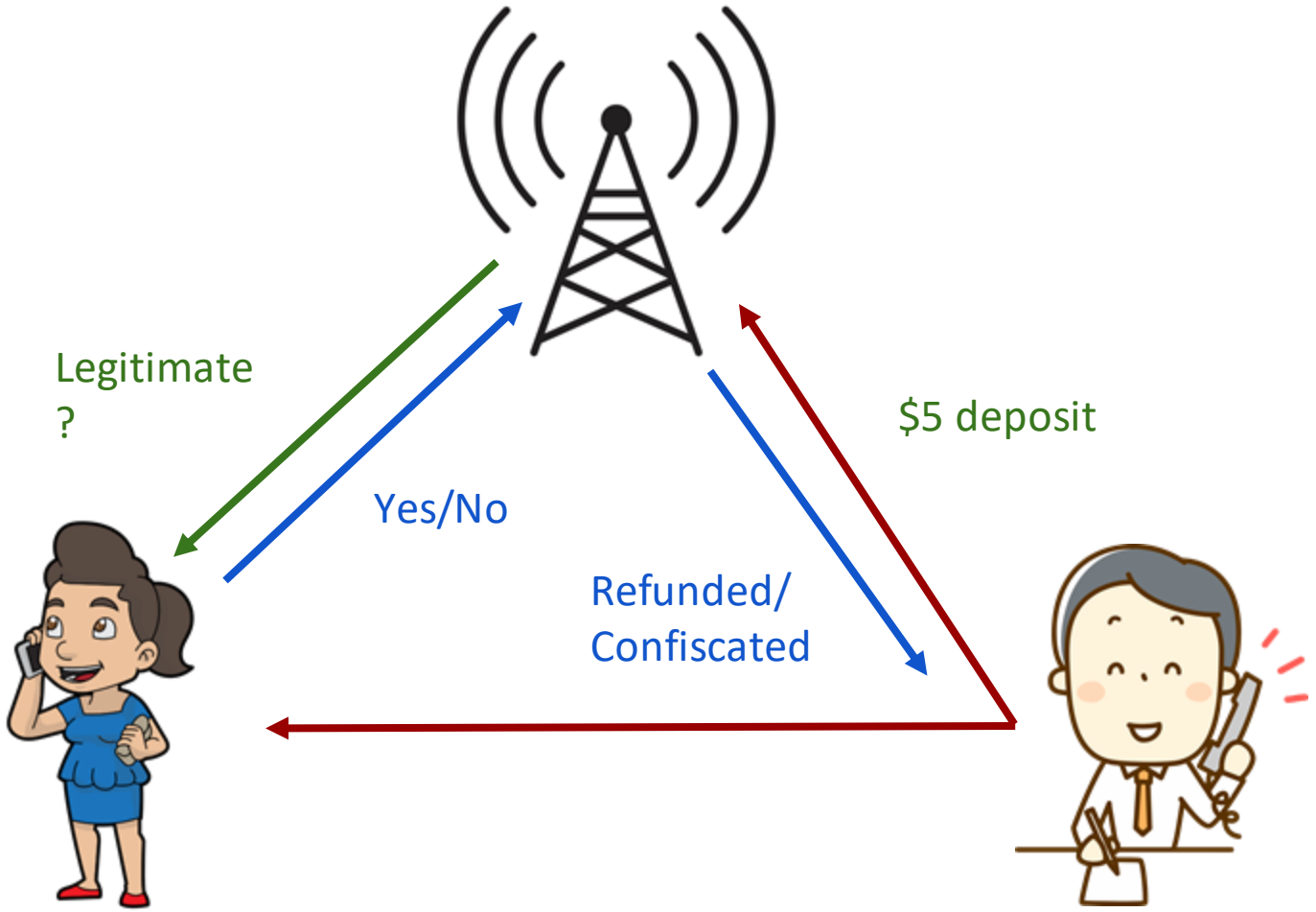
# Consensus in Bitcoin

## Breaking away from traditional assumptions

- How Bitcoin overcomes impossibility results?
- Introduces randomness
  - Choose a node randomly *somehow*
  - No start/end time
  - Consensus happens over a long time
  - Divergence at the end of possible
  - Divergence probability decreases exponentially with time
- Incentives
  - Award/penalty for behaving honestly/dishonestly

# Consensus in Bitcoin

## Blocking Spam Callers



# Consensus in Bitcoin

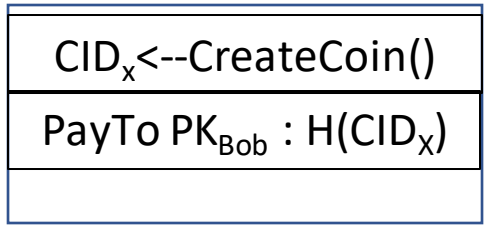
## Incentives

- Our assumptions
- The process can pick a random node
  - At least 50% of the time honest node
- Can we punish the nodes who double-spends?
  - Difficult since there is no identity
- Alternatively, can we reward the honest nodes?
- Currency incentives
  - Possible since Bitcoin is a currency
  - Can award some Bitcoins for honest behaviour

# Consensus in Bitcoin

## Incentives

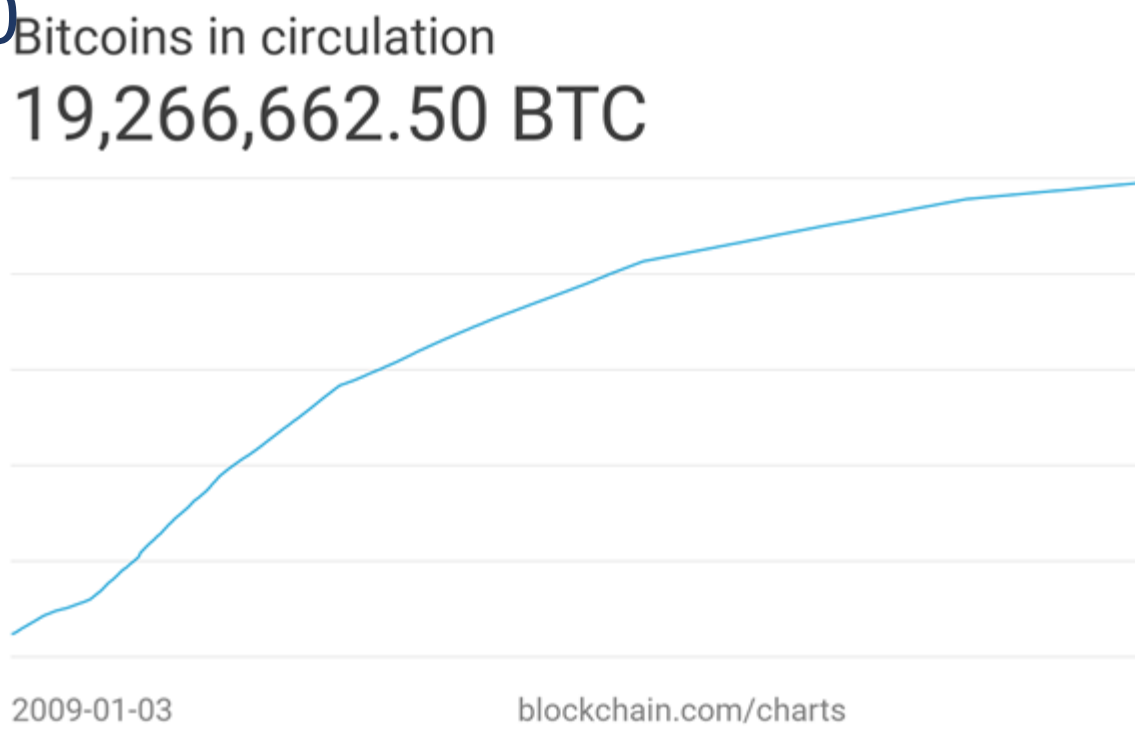
- Block reward
  - A node which creates/proposes a block can add a transaction
  - Awards *certain* number of bitcoins to itself
  - Coinbase transaction



# Consensus in Bitcoin

## Incentives

- 50 BTC in 2009
- Block reward halves every 210k blocks
  - Roughly each 4 years
- Currently 6.25 bitcoins
  - Last halved on May 11, 2020



# Consensus in Bitcoin

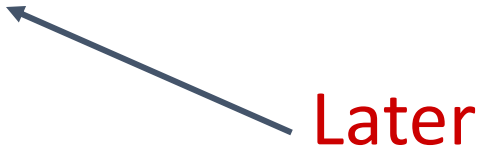
## Incentives

- Total number of bitcoins limited to 21 million
  - To limit the number of coins in circulation
  - Stop inflation
- Finite sum
  - Will end in 2040 approximately
- Then what?
  - Transaction fees
- Nodes may ask for some small fee for a transaction to be included in the block
- Total input Bitcoins > Total output Bitcoins
  - The difference is transaction fee

# Consensus in Bitcoin

## Proof-of-work

- Incentives solves the problem of reward/penalty
- How to pick a random node?
- Incentives introduces more problem
- Everybody wants to run Bitcoin nodes and get reward
- Sybil attacks!!
- Select nodes in proportion of ownership of resources
  - Hard to monopolize
- Computing resource --> Proof-of work
- Ownership of coins --> Proof-of-stake



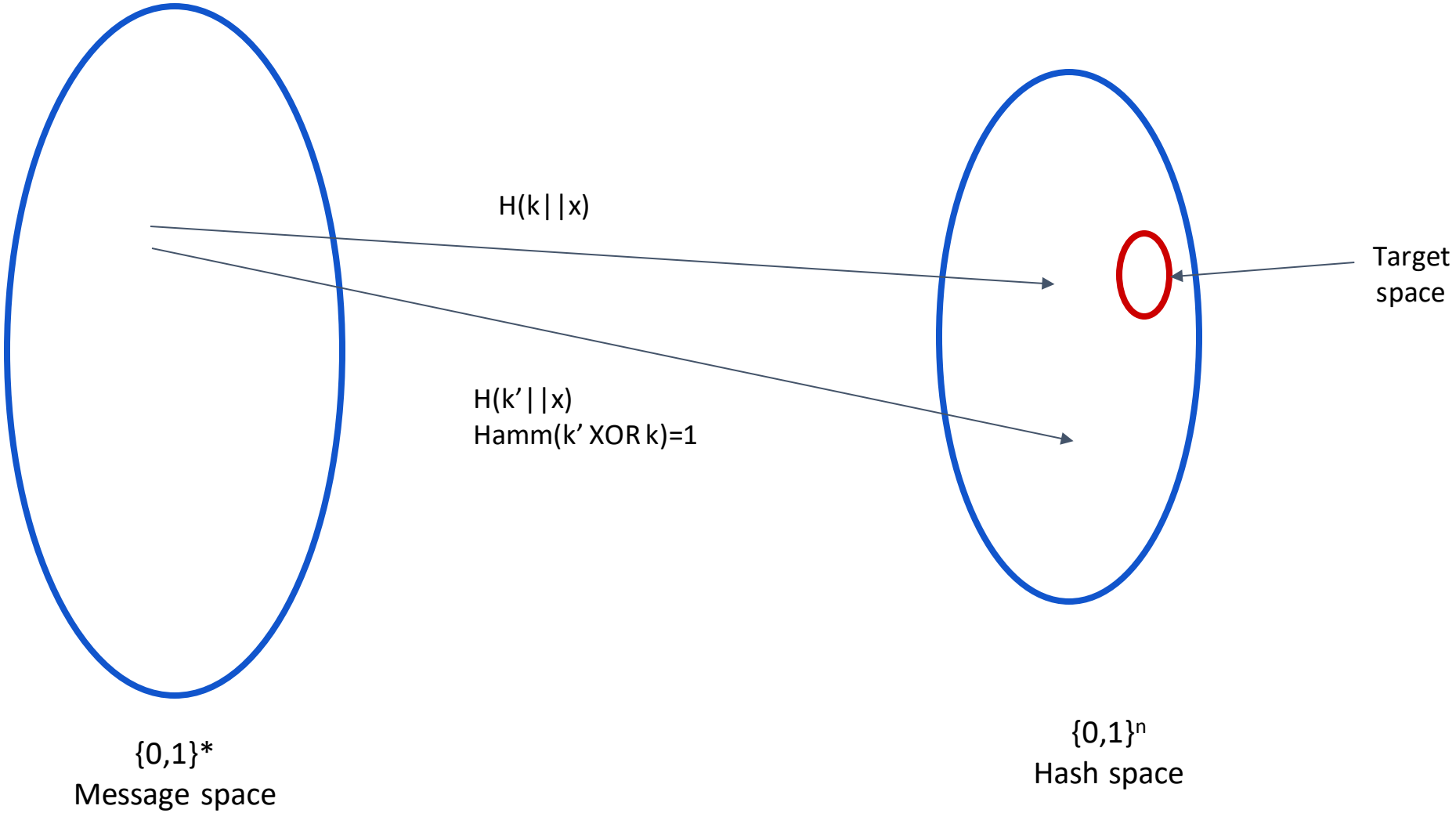
# Consensus in Bitcoin

## Proof-of-work

- Hash puzzles
- For fixed, x and find k (also called nonce), such that,
  - The hash i.e. H(k||x) has some specific values
  - E.g. t number of bits out of n bits are zero

# Consensus in Bitcoin

## Proof-of-work



# Consensus in Bitcoin

## Proof-of-work

- As an example
- Let's assume the hash space has only 16 bits
  - We set our target circle with those hashes where the first 4 bits are set to 0 or 1
  - The probability of finding such values is $2^{12}/2^{16}$
  - If the hash function is puzzle friendly we have to try $2^{16}/2^{12}$ different values of k in the worst case to find a hash in the target space
  - Or in other words, we have to perform this much hashes or work to find a hash in the target area
- The amount of hashes to perform to find the desired nonce is also called *difficulty* of the puzzle

# Consensus in Bitcoin

## Proof-of-work

- In Bitcoin, proof-of-work is achieved by solving hash puzzles
- Hashing requires computation power
  - Often requires powerful GPUs, ASICs, etc.
  - Requires significant time and monetary investment
- Difficult to monopolize
  - For a node to be selected with more than .5 probability the node has to control 50% of the global hashing power
- Apart from the incentive to behave honestly,
  - If a node behaves dishonestly all the work done to solve the puzzle and therefore the money required to do that is wasted
  - So it also serves as a penalty for dishonest behaviour

# Consensus in Bitcoin

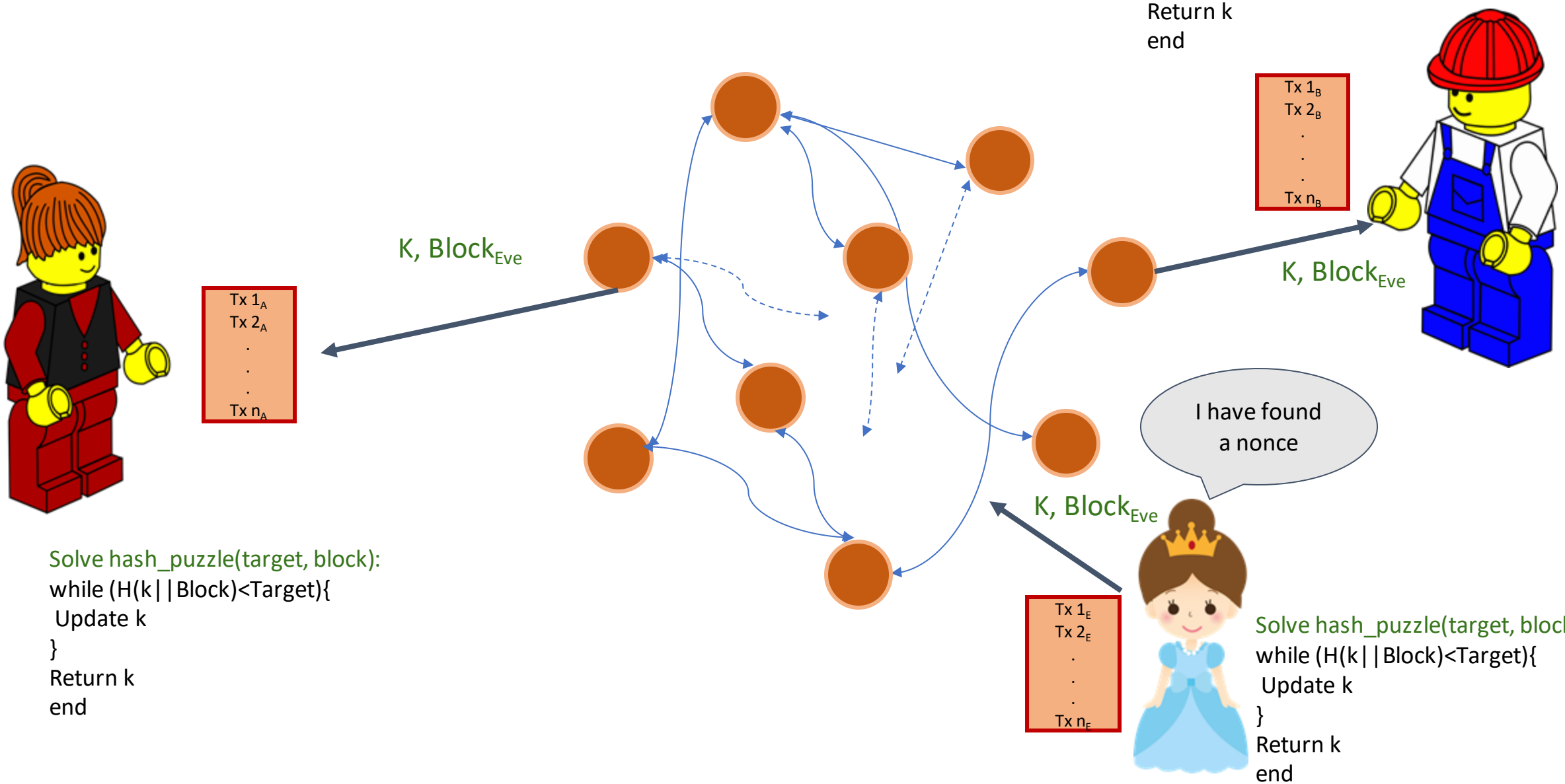
## Proof-of-work

- A block has many data
  - Prev hash
  - Root hash
  - Data related to transactions
- Node must find a *nonce* such that
  - H(nonce|| tx||prev_hash||root_hash||...)< target
- Recall, for puzzle-friendly hash functions only way to find the nonce is by *brute-force*
- *Trivial* to verify

# Consensus in Bitcoin

## Proof-of-work

Solve hash_puzzle(target, block):
while (H(k||Block)<Target){
Update k
}
Return k
end




Solve hash_puzzle(target, block):
while (H(k||Block)<Target){
Update k
}
Return k
end

Solve hash_puzzle(target, block):
while (H(k||Block)<Target){
Update k
}
Return k
end

# Consensus in Bitcoin

## Proof-of-work

- We are allowing nodes to compete
- The nodes will be selected in the proportion of their %age of total global computing power
  - It is difficult to monopolize the total hash rate
- Sybil attack resistant
  - Nodes can still create many new identities
  - No extra advantage unless the hash rate is increased
  - Financial restriction
- Upholds our assumption on choice of random nodes

# Consensus in Bitcoin

## Proof-of-work

- Cost : the probability a miner is going to get the next block is proportional to the fraction of global hash power it controls
  - Alice controls .1% of hash power, so Alice is going to win roughly one block out of 1000 blocks
  - Parameterizable cost
- To sum up everything,
  - Roughly the profit from mining
  - Mining reward – mining cost
  - Mining reward = block reward + transaction fees
  - Mining cost = hardware cost + electricity + cooling + real estate, etc.

# Consensus in Bitcoin

## Proof-of-work

- Solving this hash puzzle is also called mining
- They adjust the difficulty such that considering the total hash power of the network
  - The block finding time follows a Poisson distribution
  - It takes 10 minutes to find a new block on an average
  - Some blocks can be found earlier and some can be found later

# Consensus in Bitcoin

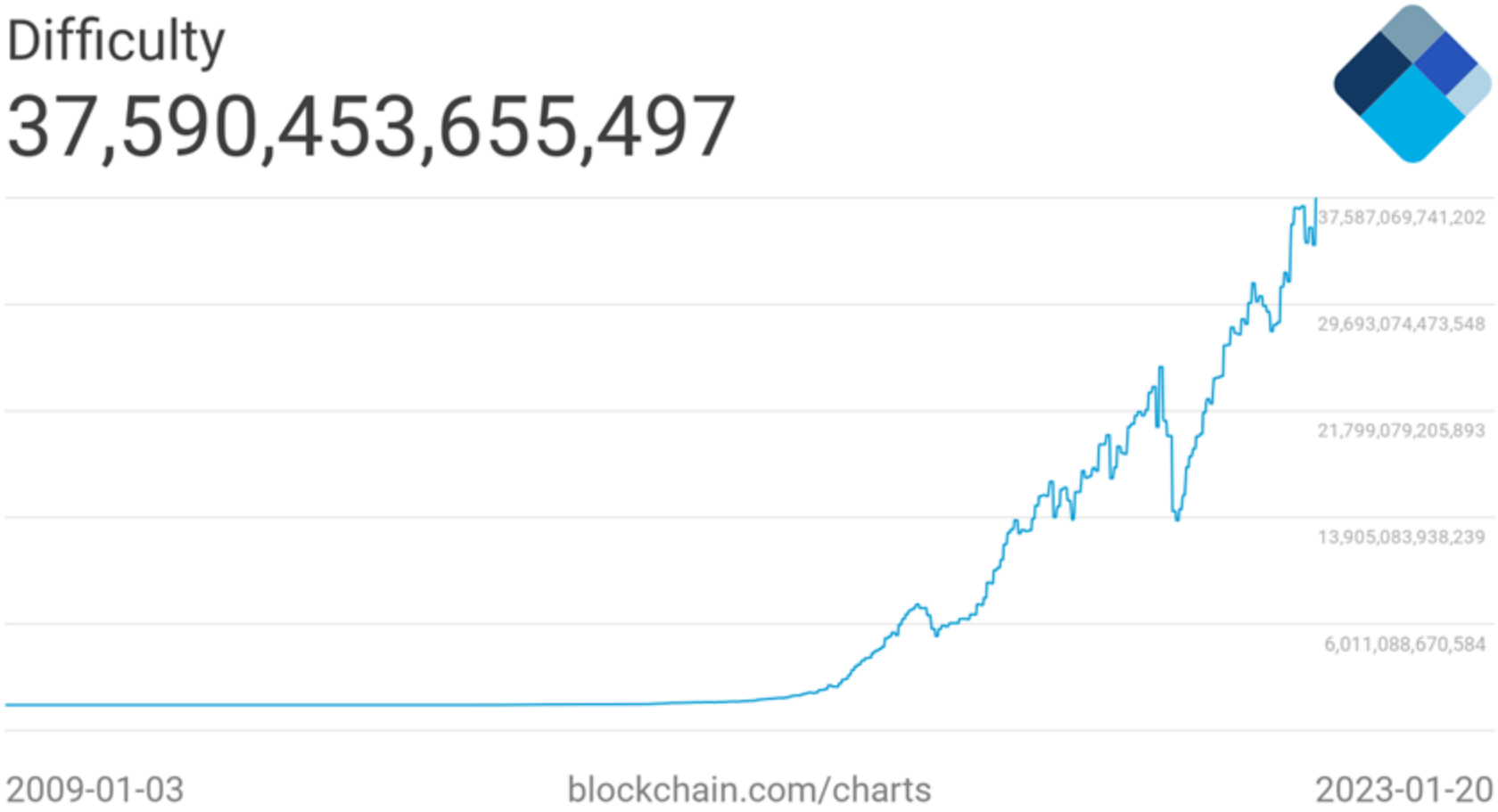
## Proof-of-work

- Nodes can leave or join network
- Hash rate of the network increases or decreases
- The p2p network automatically chooses new target
- So the difficulty should be adjusted to keep
  - To maintain the 10 minutes average block time
  - Every 2016 blocks
  - Every two weeks
- Why?
  - Caching
    - Can store the previous nonces
  - Global hash rate increases or decreases

# Consensus in Bitcoin

## Proof-of-work

- Hash puzzles are difficult to compute



# Consensus in Bitcoin

## Proof-of-work

- Hashing requires computation power
  - Often requires powerful GPUs, ASICs, etc.
  - Requires significant time and monetary investment
- The cost of being dishonest is too high
- Proof-of-work + cost of mining ensures
  - Majority of miners, weighted by hash powers is honest
  - Random selection in the proportion of hash power will pick honest nodes
- It creates a stable equilibrium that nobody can get paid higher by being dishonest
- Still an active area of research

# Consensus in Bitcoin

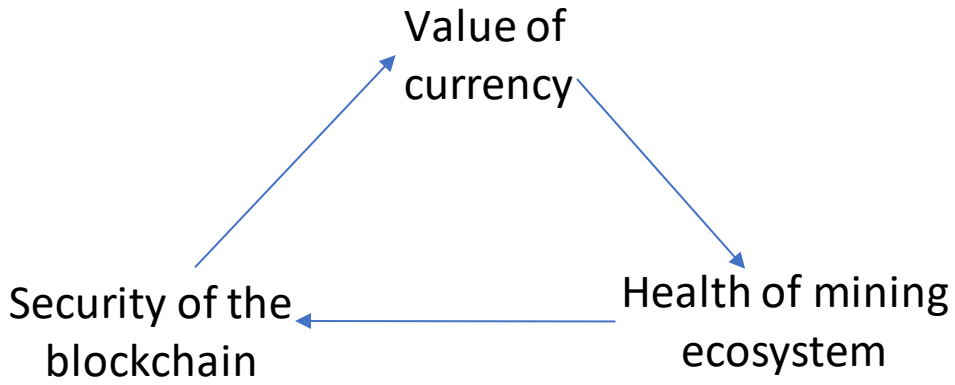
## Breaking away from traditional assumptions

- Works in practice
- No theoretical explanation!!
  - Catching up fast
- Necessary for security and reliability

# Consensus in Bitcoin

## Bootstrapping

- The value of a cryptocurrency depends on three ideas
  - Value of the currency
  - Security of the blockchain
  - Health of mining eco-system
- Each one of these is dependent on the others




- Crucial for a new cryptocurrency to succeed
- Bootstrapping

# 51% attack

- Steal coins from existing address? ✗
- Suppress some transactions?
  - From the block chain ✓
  - From the P2P network ✗
- Change the block reward? ✗
- Destroy confidence in Bitcoin? ✓✓

# Bitcoin : Fully decentralized?

- Designed to be fully decentralized.
- Anyone can join and leave
  - No central authority
- Mining
  - Technically open for everyone
  - High capital cost
  - Concentrated in few regions
- Rules and updates
  - Few trusted groups/users
- Hybrid
- Another example : SMTP

# Bitcoin : Fully decentralized?

- Based on data provided by World Population Review, the current hash rates of the leading countries in Bitcoin mining, as of 2023, are as follows:
- United States: 35.4%
- Kazakhstan: 18.1%
- Russia: 11.23%
- Canada: 9.55%
- Ireland: 4.68%
- Malaysia: 4.58%
- Germany: 4.48%
- Iran: 3.1%

# The End !!