

Blockchain Security Attacks

Yue Duan

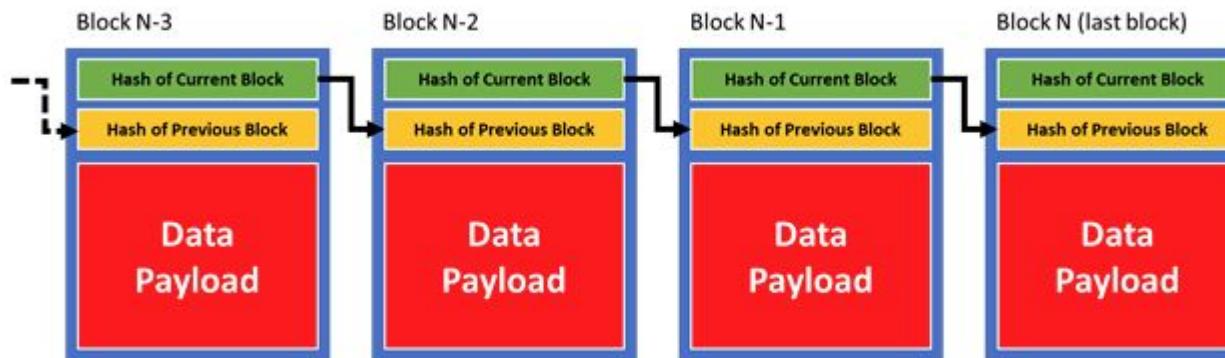
based on slides from Cristina Nita-Rotaru, David Evans, Aviv Zohar, Lefteris Kokoris-Kogias, Ittay Eyal

Outline

- Blockchain basics recap
- 51% attack
- Research paper:
 - Majority is not Enough: Bitcoin Mining is Vulnerable
 - Eclipse Attacks on Bitcoin's Peer-to-Peer Network

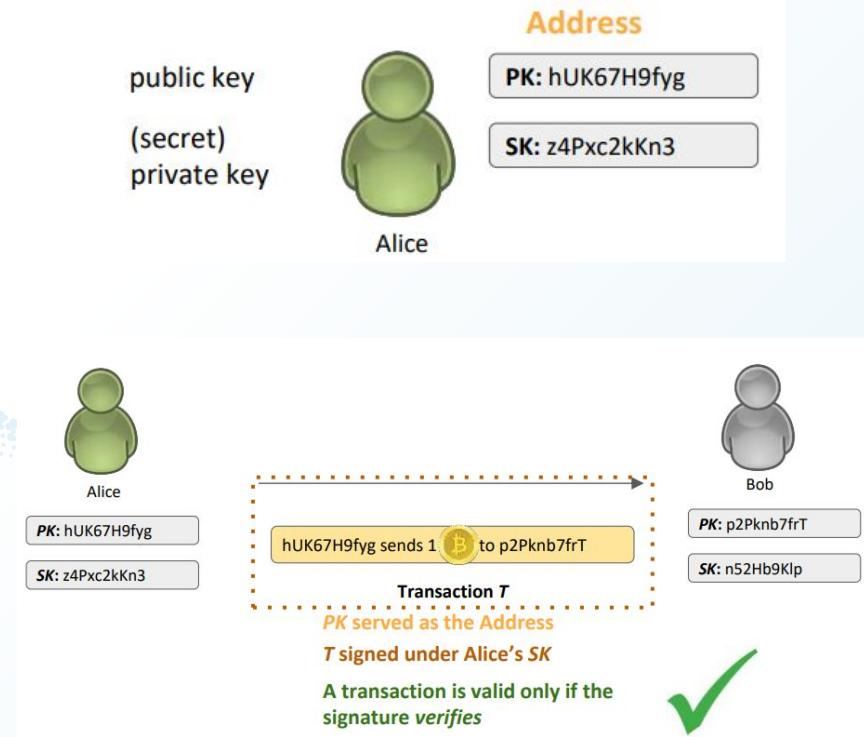
Blockchain Basics

- A peer-2-peer network
 - decentralized system
 - no central node or central government
 - based on ledger to store information



Blockchain Basics

- Bitcoin users
 - permissionless: everyone can join, just generates a key pair
- Bitcoin transactions
 - use digital signatures (use hash function)



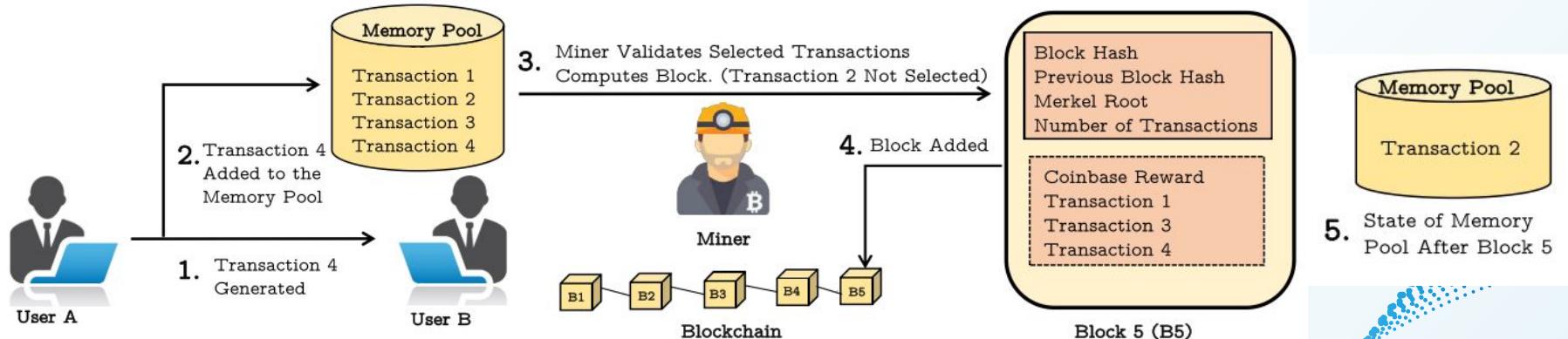
Consensus Mechanisms

- Two major consensus protocols
 - Proof of Work (PoW)
 - most popular
 - used by bitcoin, ethereum, etc
 - Proof of Stake (PoS)
 - promising
 - could become dominant in the future

Proof of Work

- Miners solve puzzles to mine blocks
 - block contains a sequence of transactions
 - puzzle:
 - hard to solve (difficulty is dynamically adjusted)
 - easy to verify
- When a solution is found
 - miner broadcasts a new block to the network for verification
 - append the block to the main chain

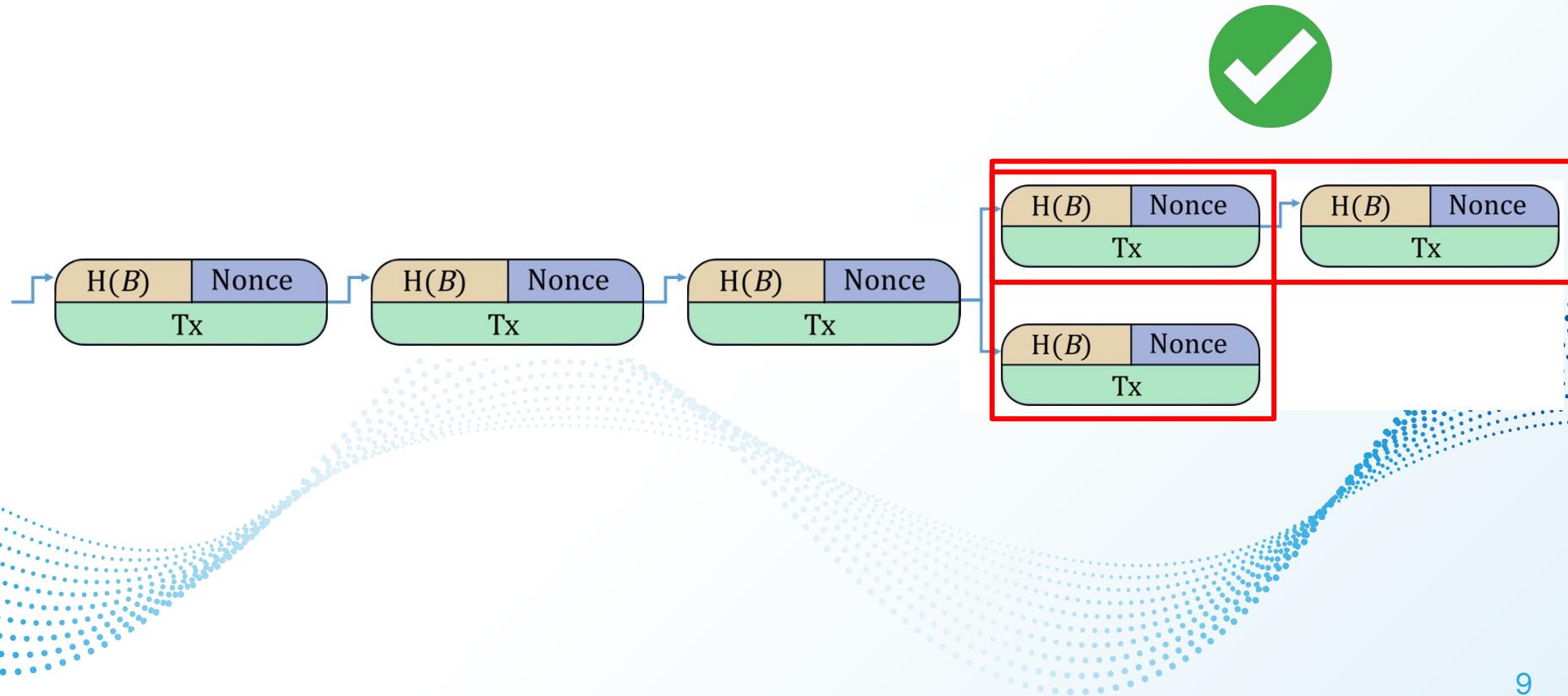
Proof of Work



Bitcoin Protocol

- Each P2P node runs the following algorithm:
 - new transactions are broadcast
 - each node collects new transaction into a block
 - each node works on solving the puzzle or its block
 - **intensive computational resources**
 - when a node finds a solution, it broadcasts the block
 - nodes accept the block only if
 - all transactions are valid (check digital signature)
 - coins not already spent (check public ledger)
 - nodes accept and work on creating the next block
 - if multiple valid blocks exist, **choose the longest chain**

Conflicting Blocks



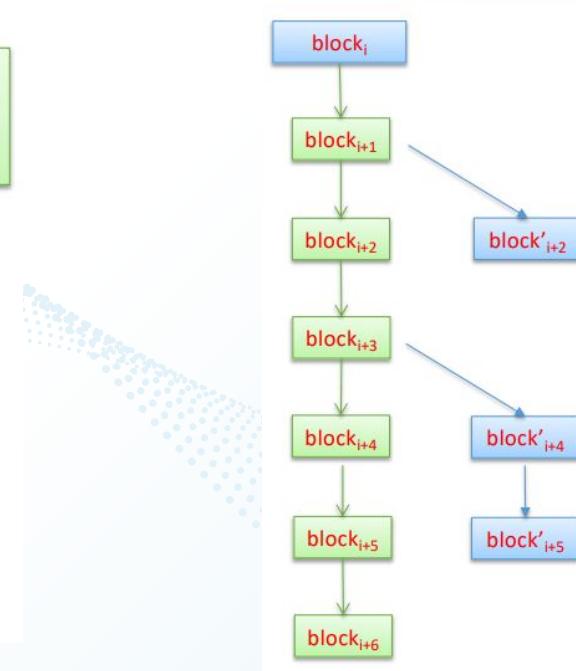
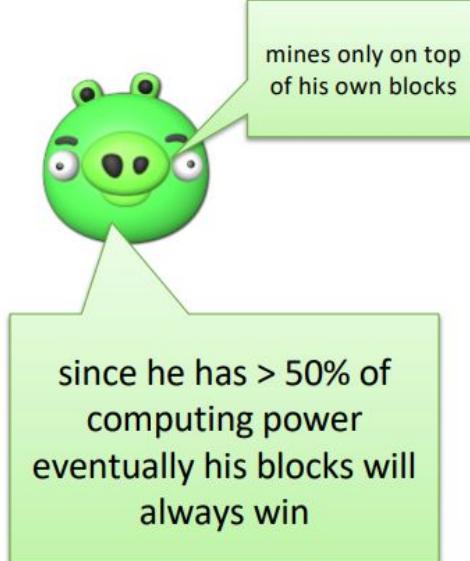
Proof of Stake

- Like shareholders of a company
- Participants must have a stake Stake:
 - usually by owning some cryptocurrencies
 - to have a chance of selecting, verifying and validating transactions
- Factors of having the chance
 - the amount of stake
 - the duration of the stake
- No mining involved
- No need for the entire network to be involved in validation process

Person who holds the most coins wants to secure the chain the most

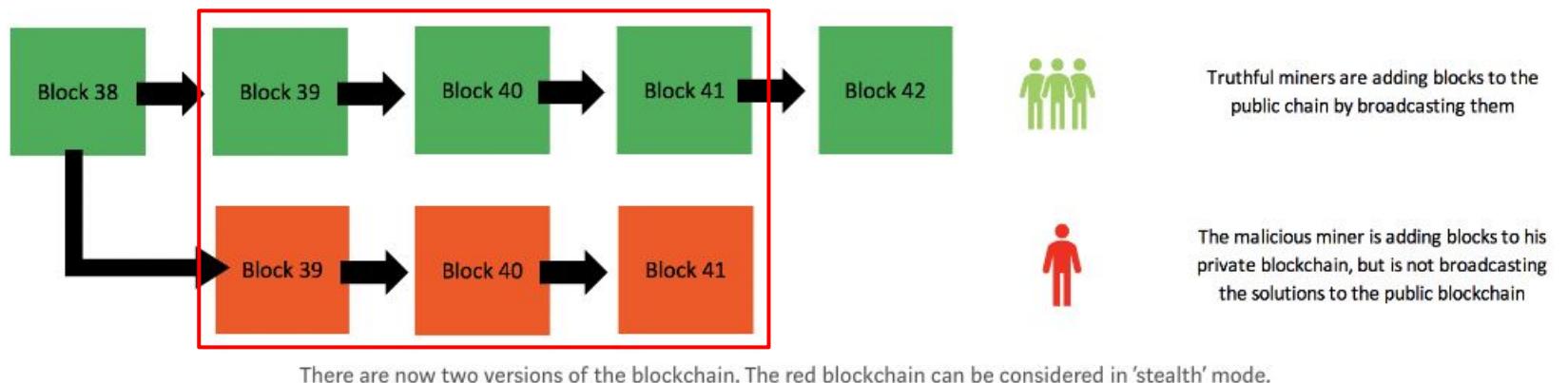
The 51% Attack

- Anyone with a 51% computation power can get a full control over the blockchain



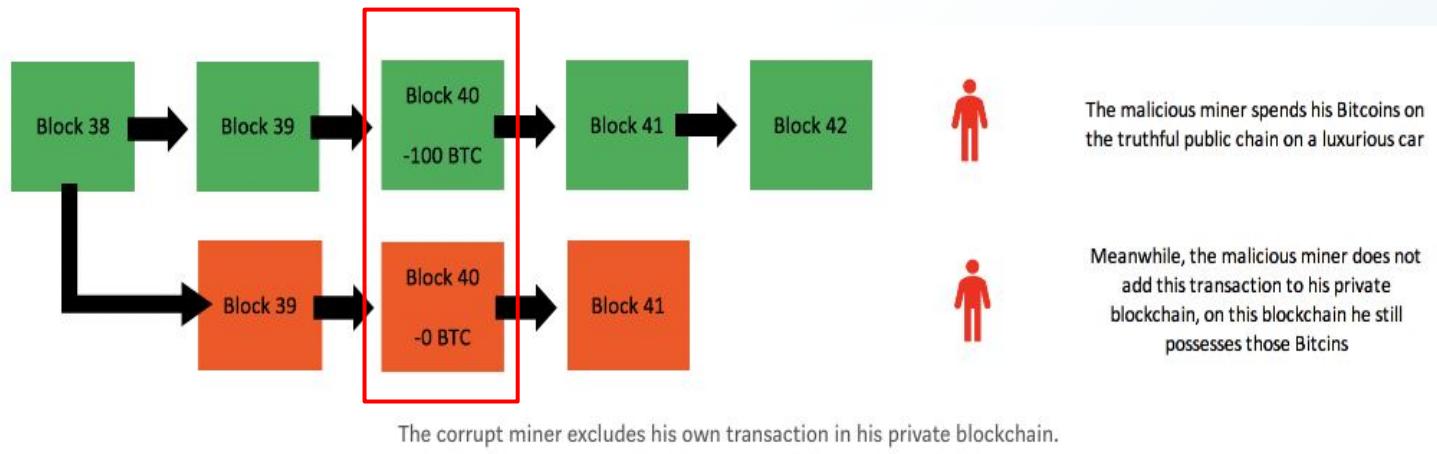
The 51% Attack: Double-spending

- A malicious miner creates an offspring of the blockchain by not broadcasting the solutions



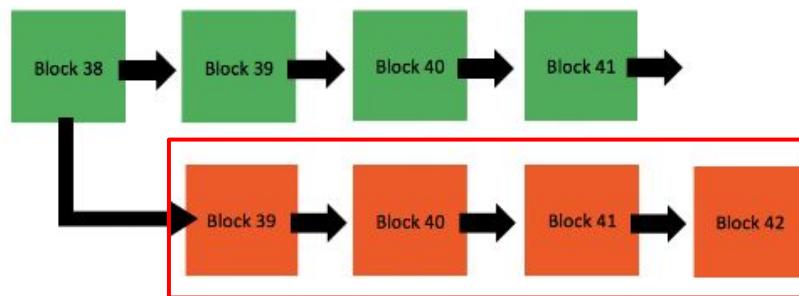
The 51% Attack: Double-spending

- The malicious miner then spends cryptocurrency on public chain.
- This transaction, however, is not shown in his private chain.



The 51% Attack: Double-spending

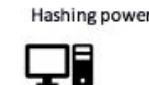
- The malicious miner now tries to add more blocks in his private chain than the public chain.



Truthful miners are adding blocks to the public chain, but in a considerably slower pace than the malicious miner is adding blocks to his private and stealth blockchain



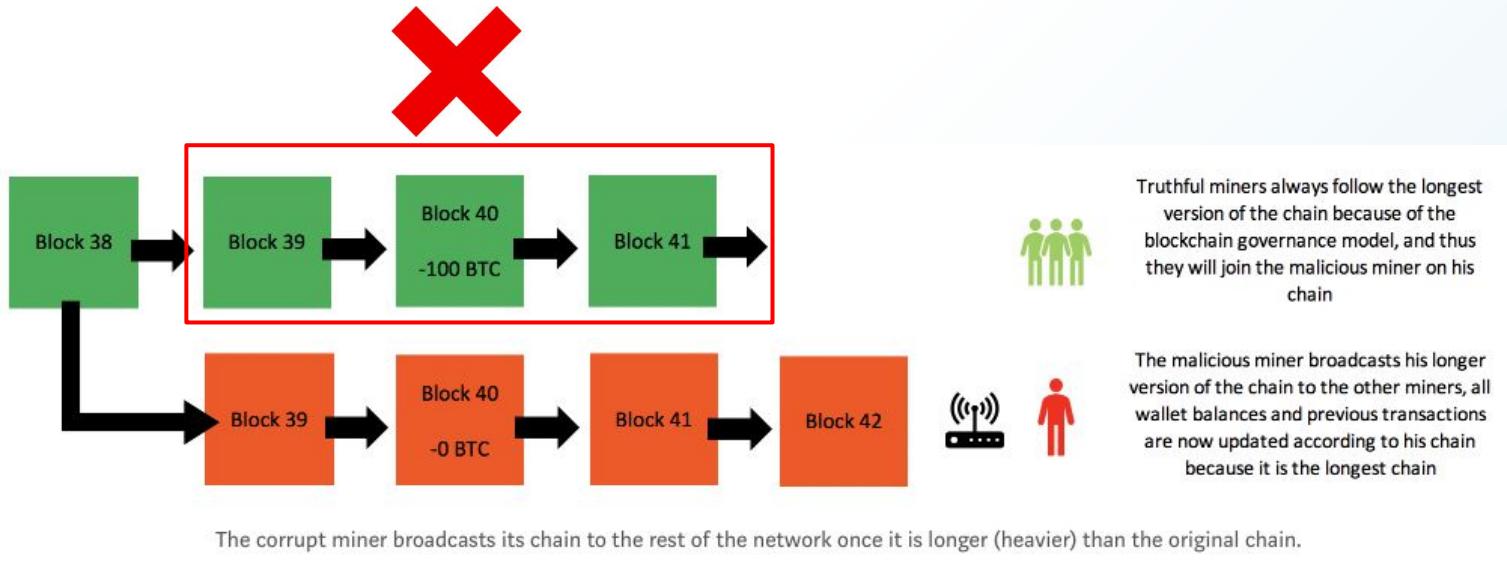
The malicious miner is adding blocks to his private blockchain faster, trying to catch up with the private blockchain



The corrupt miner is now adding blocks to his private chain faster because he has more hashing power.

The 51% Attack: Double-spending

- Eventually, malicious miner broadcasts his longer version of the chain, rendering his previous transaction reversed.



Majority is not Enough: Bitcoin Mining is Vulnerable

Ittay Eyal and Emin Gun Sirer

FC 2013

Selfish Mining Attack

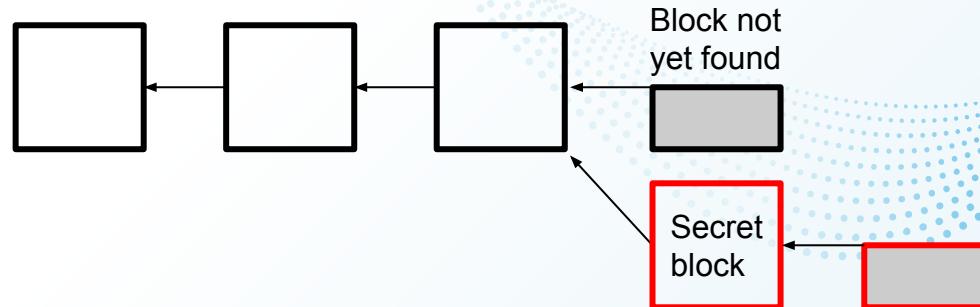
- Blockchain protocol works if majority (51%) of miners follow the protocol.



- Two-thirds must be honest
 - best theoretical case
- Practical solution
 - three-quarters of miners are honest

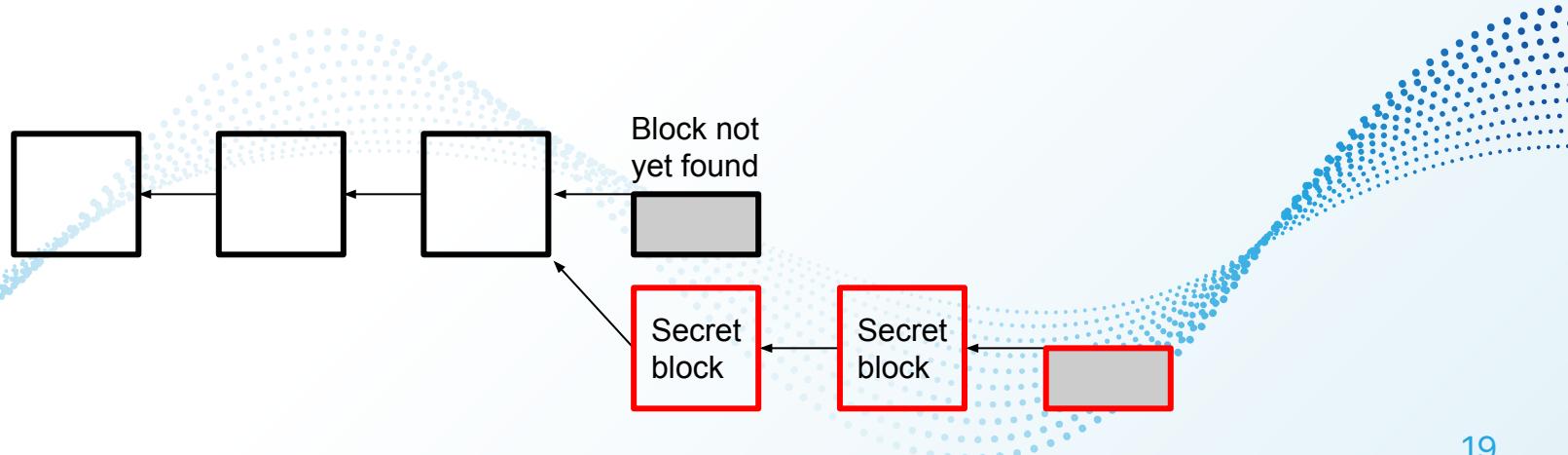
Selfish Mining Attack

- Suppose an attacker just found a block
 - normal behavior:
 - broadcasting the block
 - receive reward
 - Instead,
 - keep the block secret
 - try to find two blocks in a row before the network finds the next one



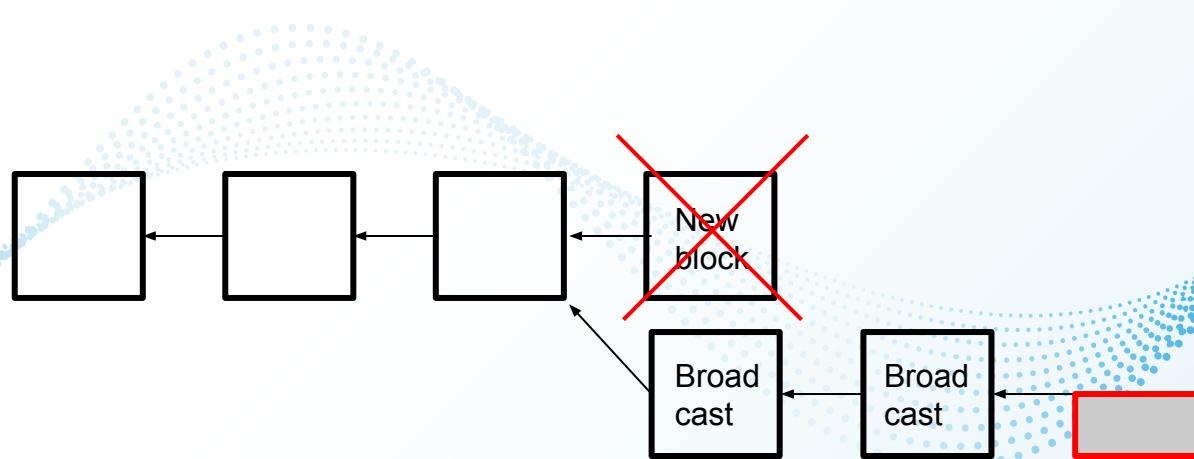
Selfish Mining Attack

- If the attacker succeeds in finding a second block
 - network still believes it is mining on the longest chain
 - attacker can continue to mine on his own chain
 - keep doing this, until the network is only 1 block behind



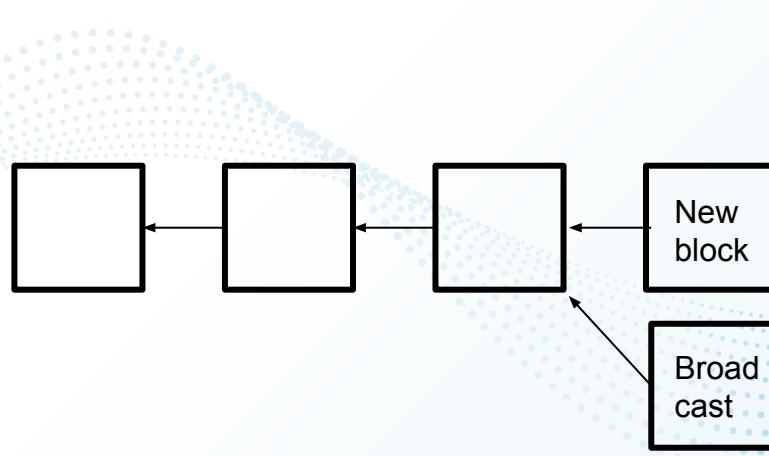
Selfish Mining Attack

- If the network finds a block, the attacker broadcasts the two secret blocks
- The network block becomes invalid
 - other miners waste work building off of an invalid block



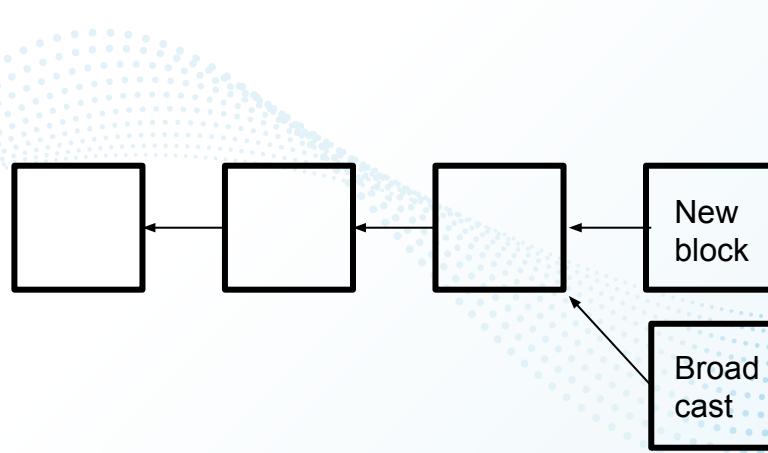
Selfish Mining Attack

- What if the network found their new block before the attacker could find a second one?
- **Race to propagate!**



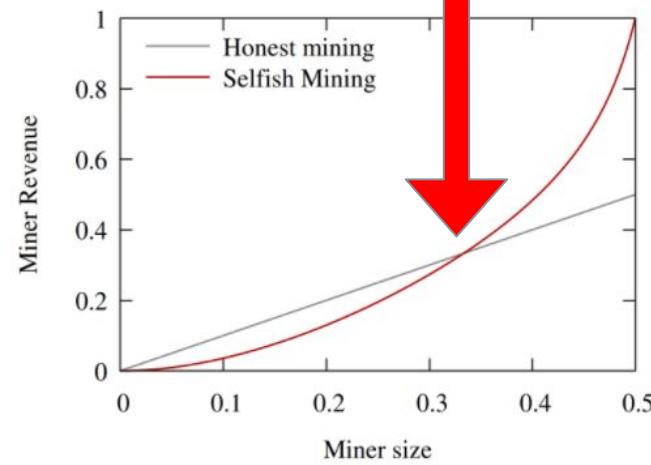
Selfish Mining Attack

- Sybil nodes can be used to increase the chance
- When sybil nodes receive new blocks:
 - share only the selfishly mined block
 - selfish miners will more likely win with enough sybils



Selfish Mining Attack

- Even without sybil nodes
 - a selfish miner larger than $1/3$ of the mining power would increase her revenue by deviating from the prescribed protocol and performing Selfish Mining



Eclipse Attacks on Bitcoin's Peer-to-Peer Network

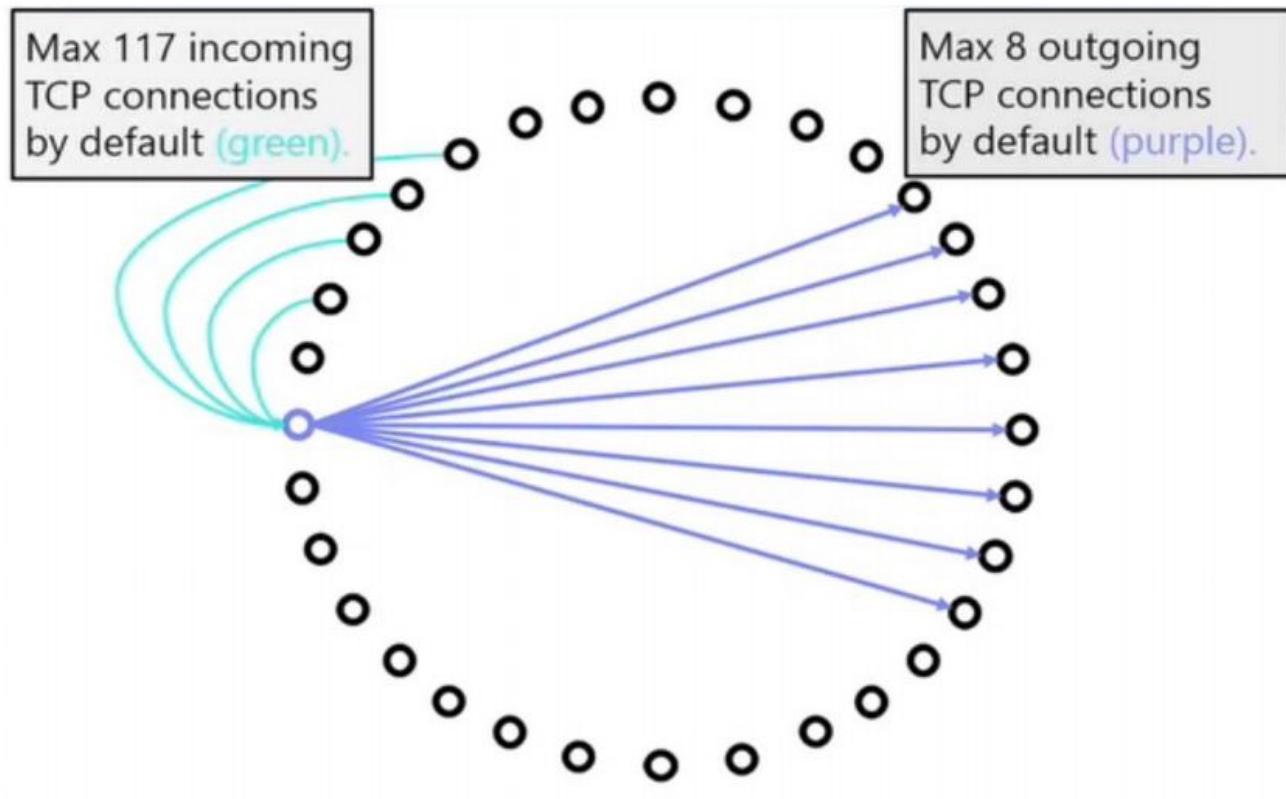
Ethan Heilman, Alison Kendler Aviv Zohar, Sharon Goldberg

Usenix Security 2015

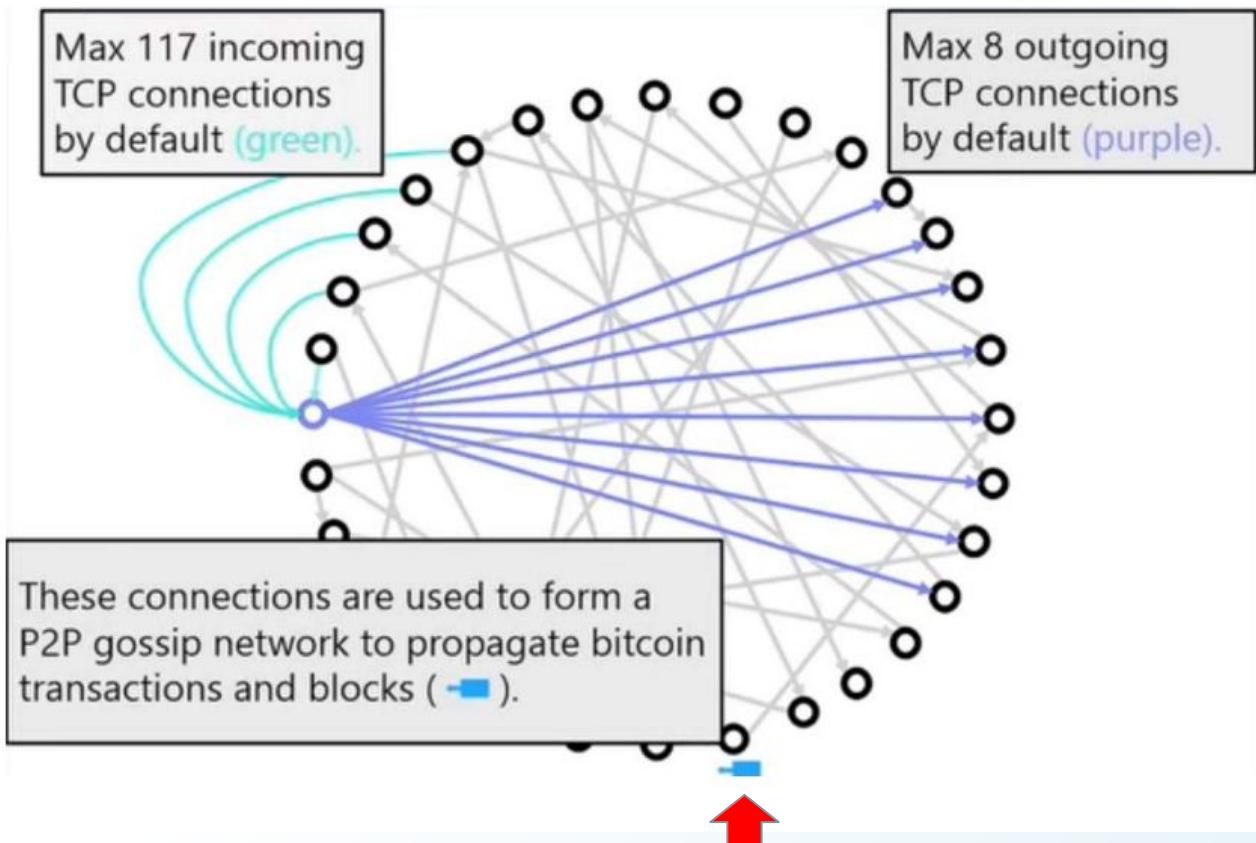
P2P Network

- Bitcoin is thought to be secure if **51%** of the mining power is honest
- Assumption:
 - all miners see all blocks/transactions
- Bitcoin relies on its P2P network to deliver this information
- Controlling the network → Controlling the blockchain

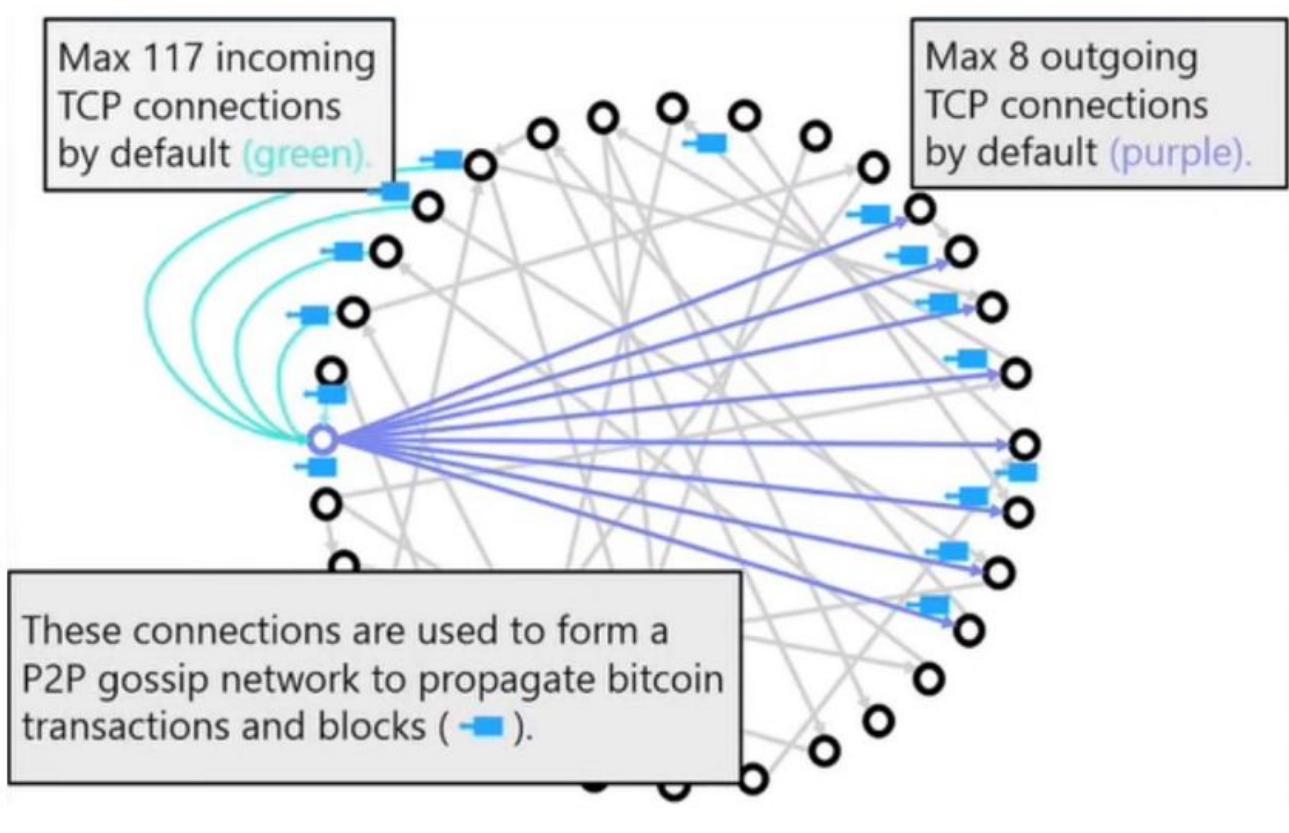
Eclipse Attack



Eclipse Attack



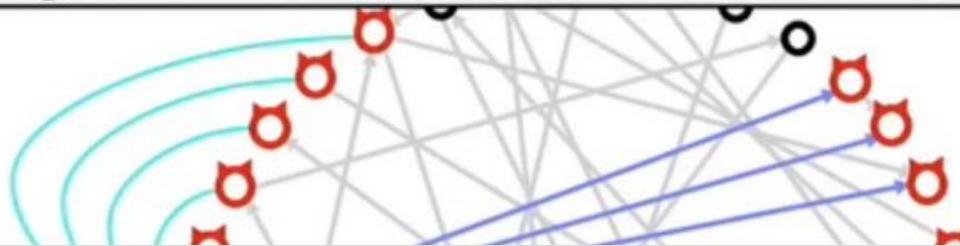
Eclipse Attack



Eclipse Attack

Information Eclipse Attack (def):

Gaining control over a node's access to information in a P2P network.



By manipulation the P2P net, the attacker eclipses the node



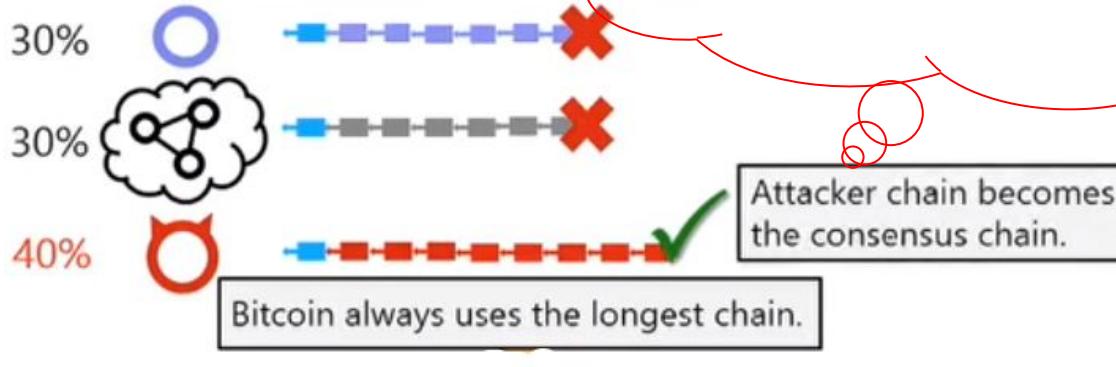
Example 1: 51% attack



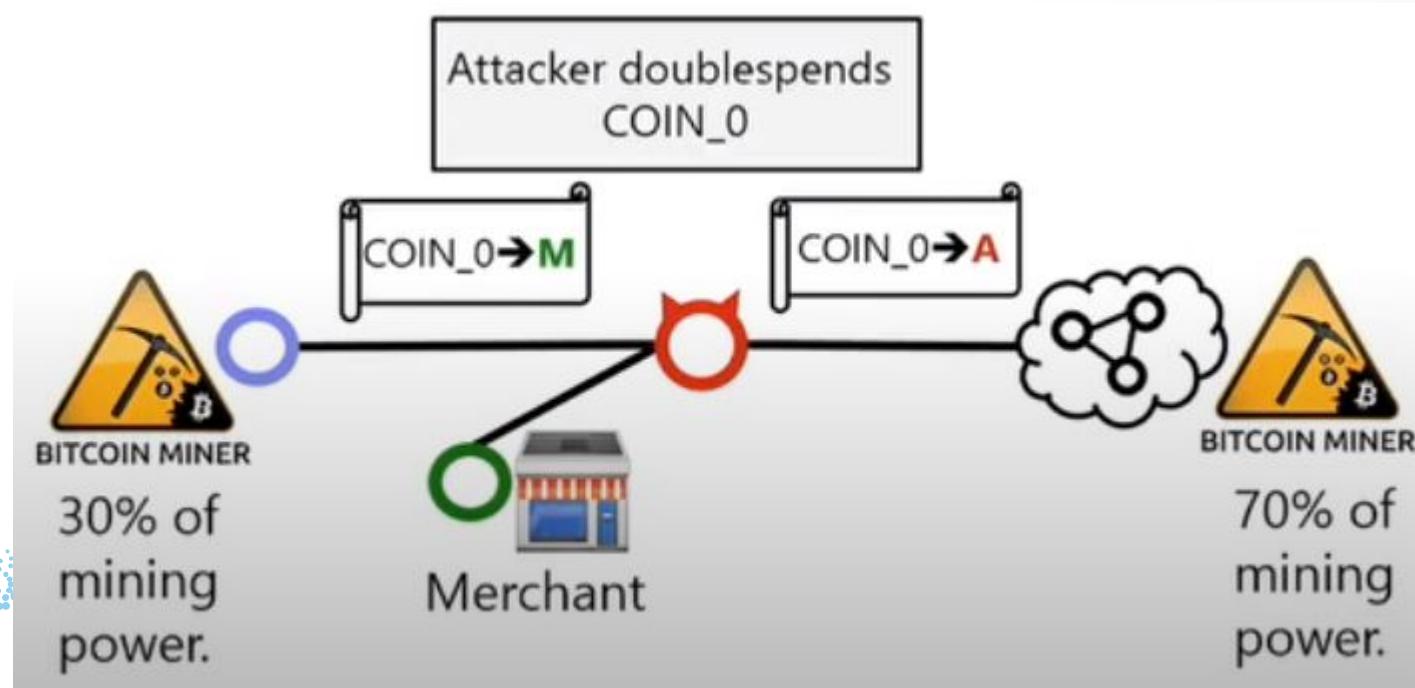
Attacker partitions miners so they cannot build on each others block

Example 1: 51% attack

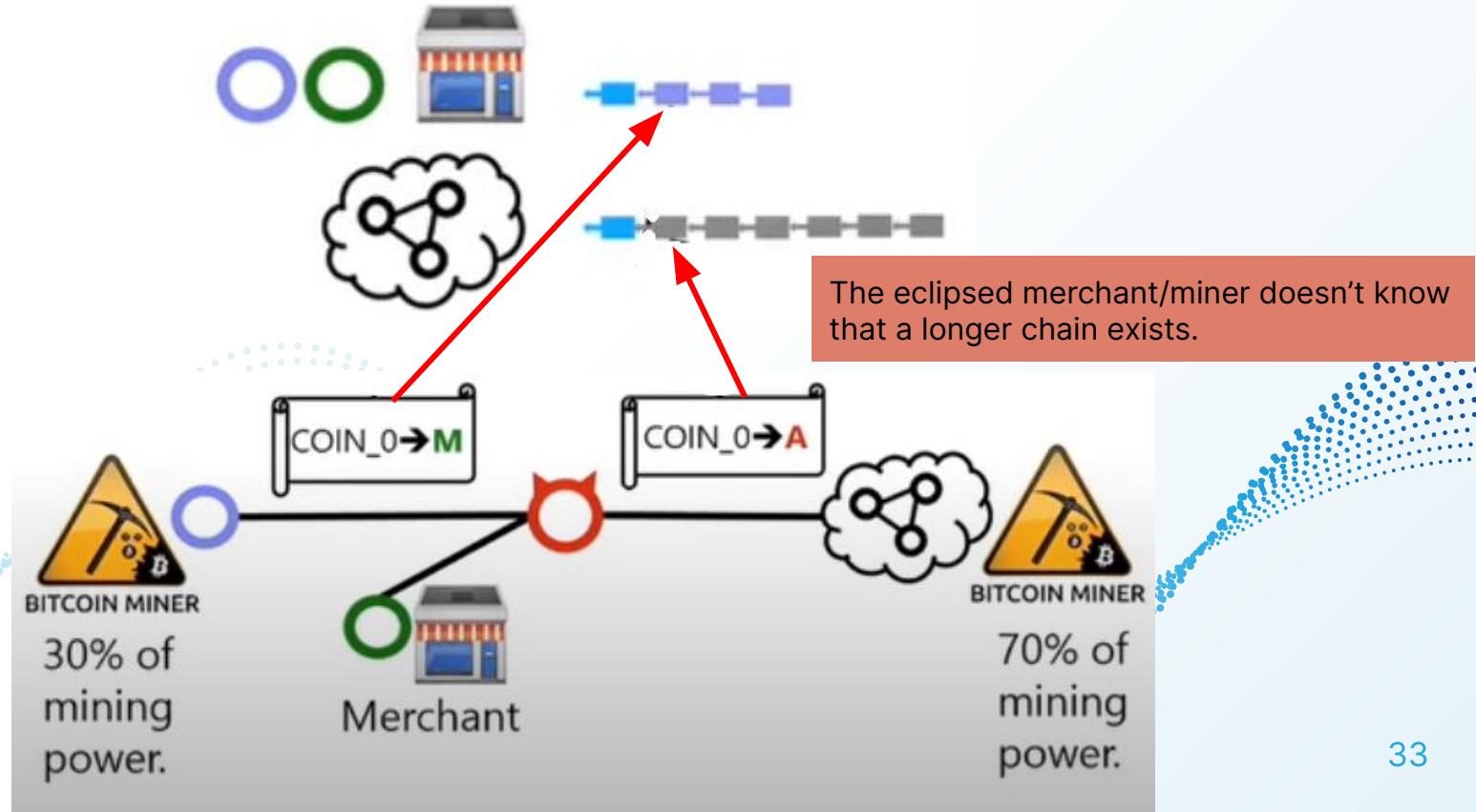
The attacker then outcompetes each partitioned miner.



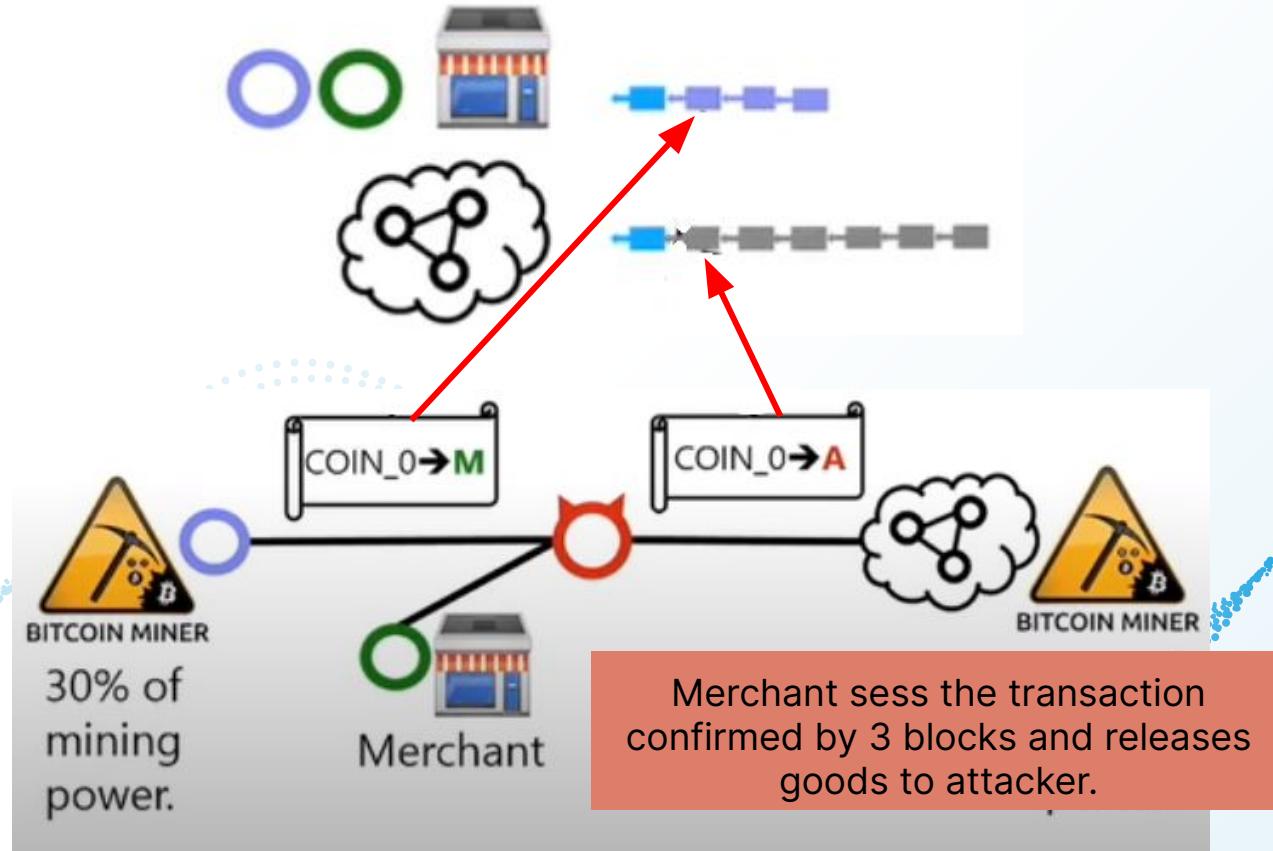
Example 2: N-Confirmation Double Spending



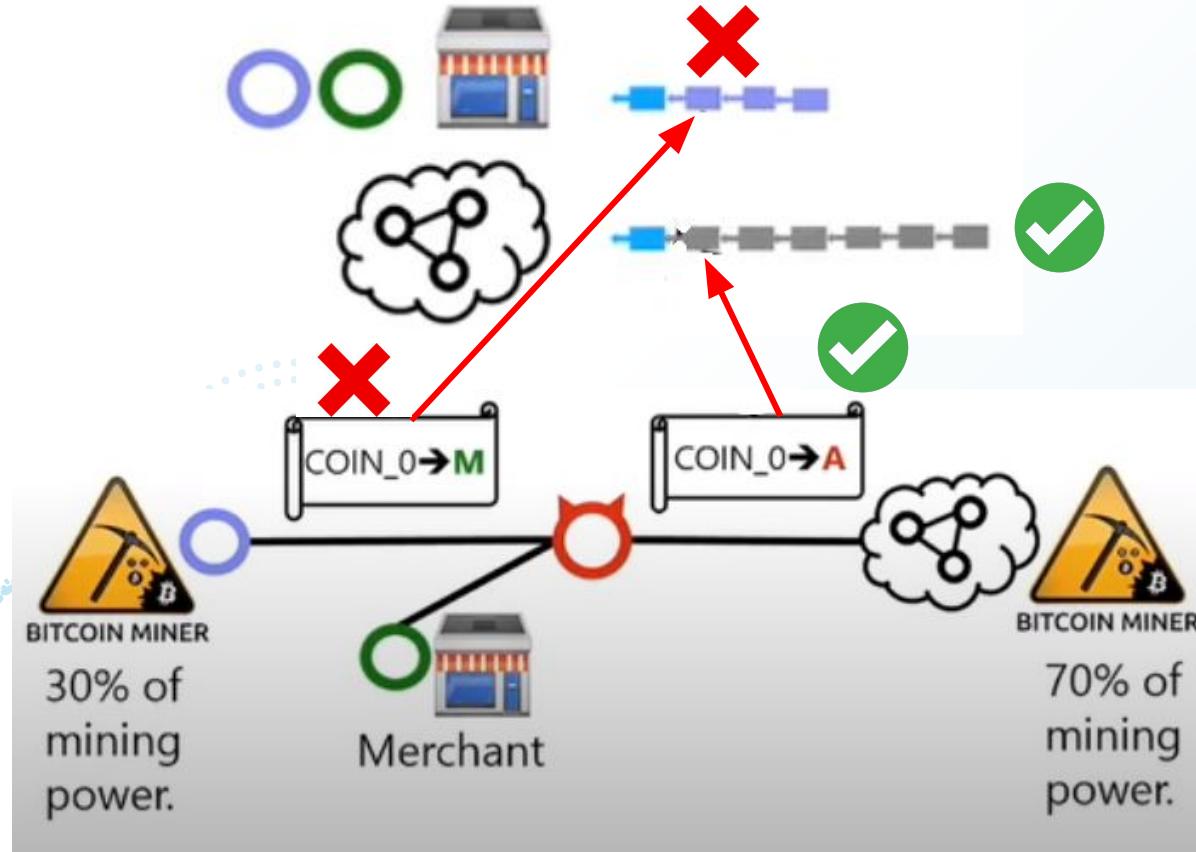
Example 2: N-Confirmation Double Spending



Example 2: N-Confirmation Double Spending

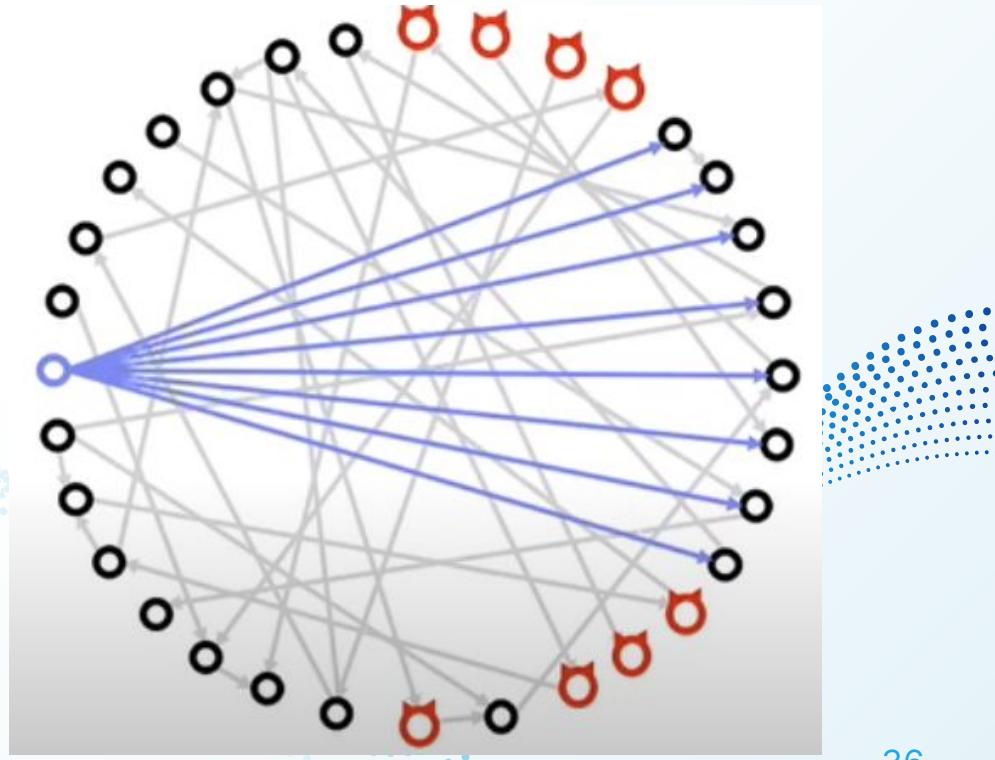


Example 2: N-Confirmation Double Spending



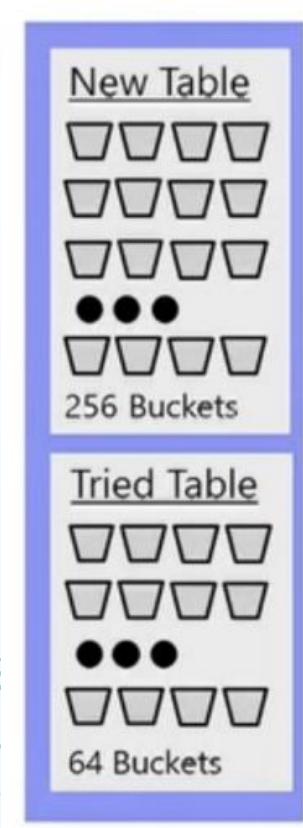
How to Launch the Attack

- Manipulate the node so all its outgoing connects are to attacker IPs.
- Fill node's **peer table** with attacker IPs
 - The node **restarts** and loses its current outgoing connections.
 - Node makes new connections to only attacker IPs.

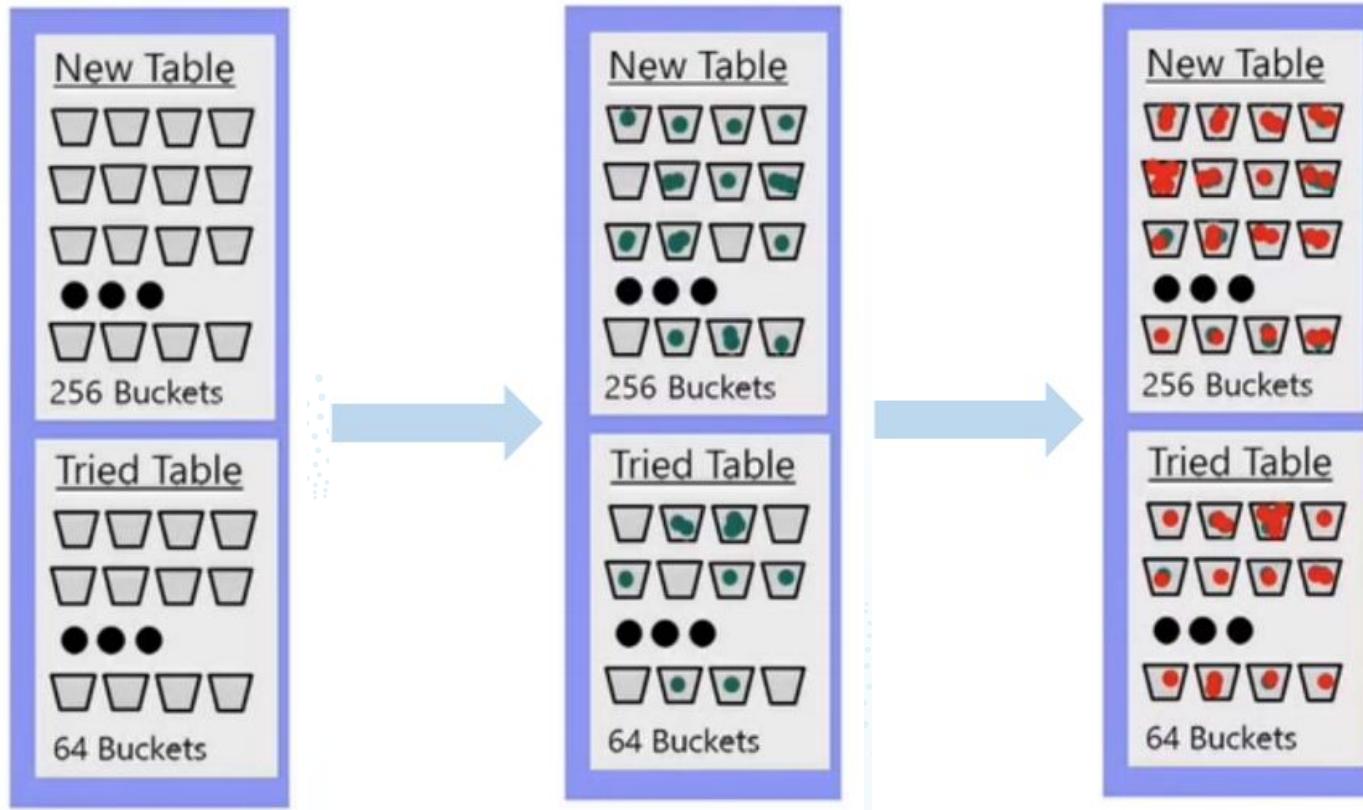


How to Launch the Attack

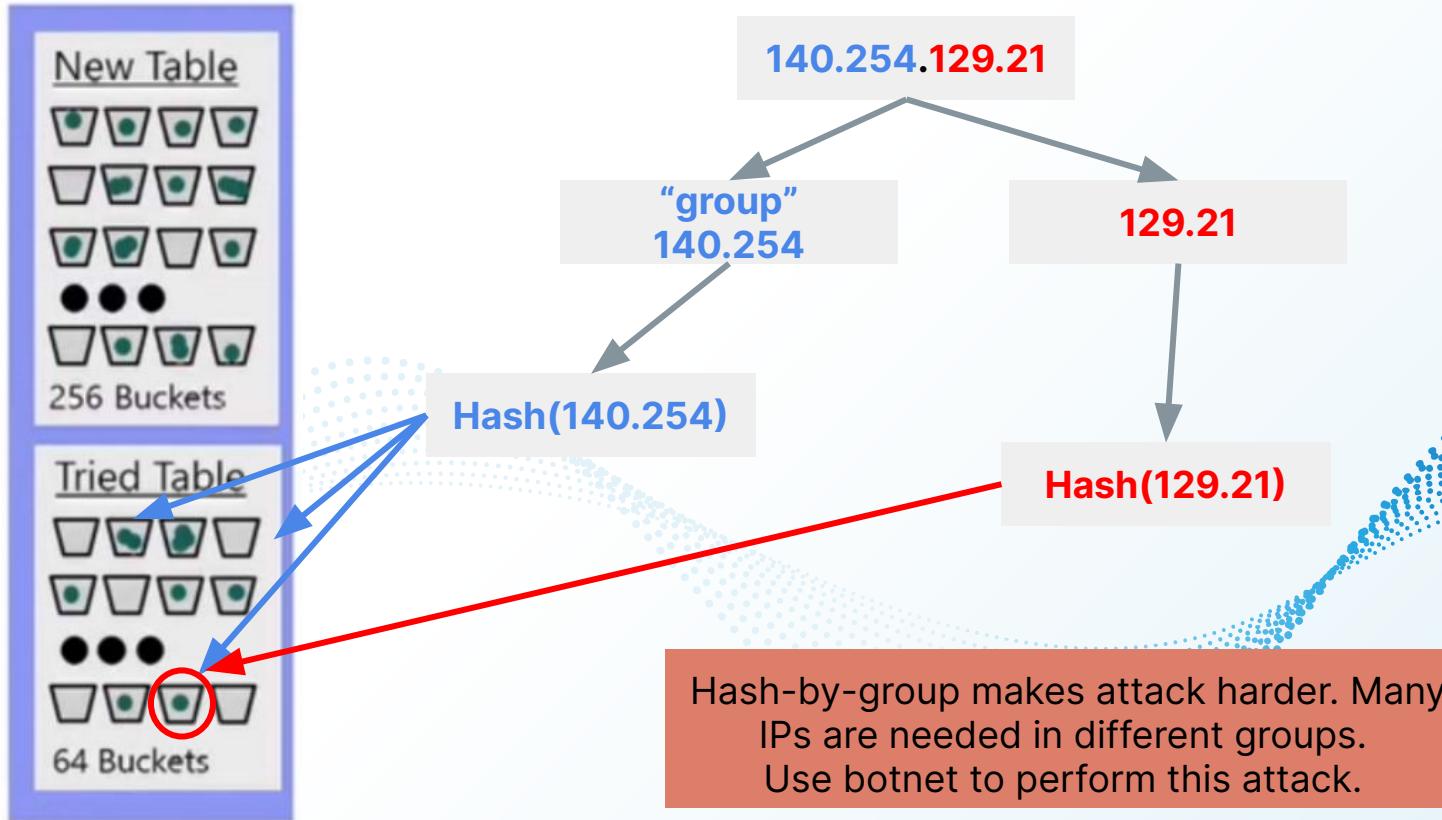
- Each node selects its peers from IP addresses stored in two tables
 - New table : IPs the node has heard about
 - Tried table: IPs the node peered with at some point
- The tables also store a timestamp for each IP
- To find an IP to make an outgoing connection:
 - choose new or tried table
 - select an IP biased towards '**newer**' timestamps
 - attempt an outgoing connection to that IP



How to Launch the Attack



How to Launch the Attack



Thank you!

Questions?

