Cryptocurrency Technology

CSIE 7190 Cryptography and Network Security, Spring 2018

https://ceiba.ntu.edu.tw/1062csie_cns

cns@csie.ntu.edu.tw

Hsu-Chun Hsiao

National Taiwan University

Housekeeping

6/10: HW3 due

6/12: Reading critique #7 due

6/12: 2nd midterm exam

6/19, 6/26: final presentation

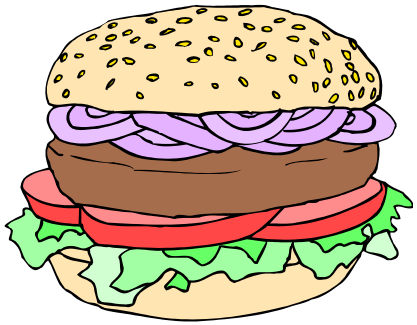
- 報告 15mins + Q&A 3mins

7/03: final report

Check **w5_project_info** for more info

What your talk is for

Your paper = **The beef**



Your talk = **The beef advertisement**

Do not confuse the two

Reference

Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. 2016. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, Princeton, NJ, USA.

- http://bitcoinbook.cs.princeton.edu/
- Many figures in today's slides are from this book

Special thanks to TA Tzu-Wei Chao

Agenda

Recent security incidents

Introduction to cryptocurrencies

- What is cryptocurrency? Why do we need it?
- How does it work?

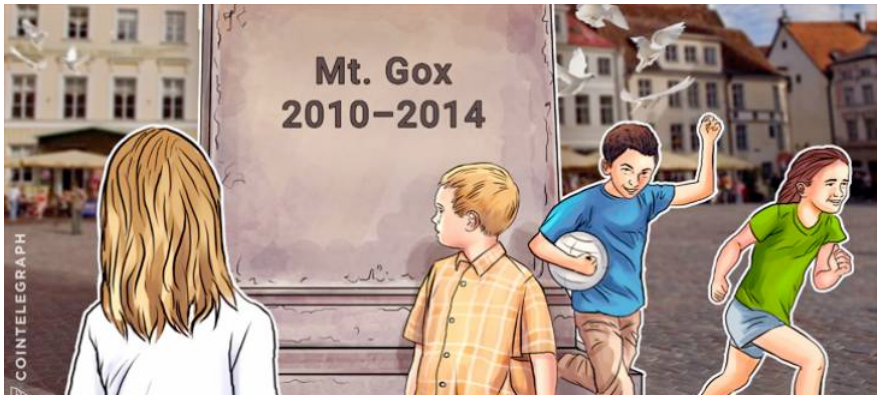
Consensus algorithms

- What is consensus?
- How to reach consensus?

Smart contract

Security issues in cryptocurrencies

Mt. Gox Hack



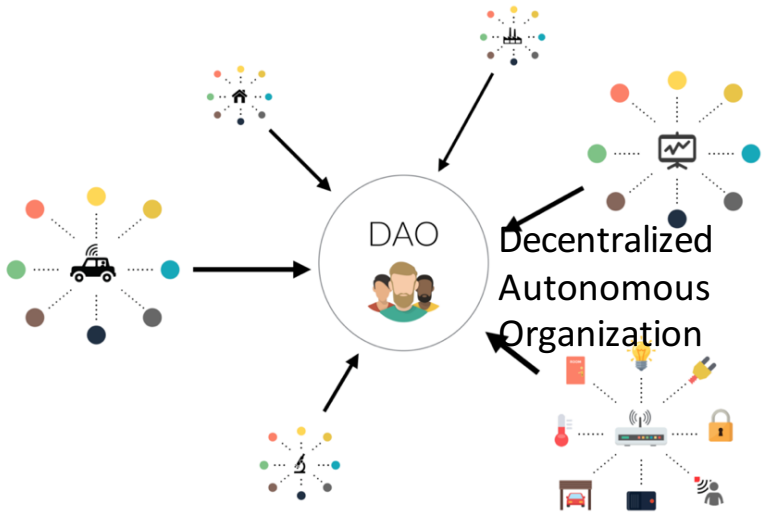
Mt. Gox was the biggest Bitcoin exchange based in Tokyo, Japan. In February 2014, Mt. Gox closed its website and service, and they claimed it was related to security issues: 850,000 BTC disappeared.

DAO Hacked

The DAO was crowdfunded via a token sale in May 2016. It set the record for the largest crowdfunding campaign in history.

It was hacked on June 2016, losing 3.6 million ether (60 million USD).

- Ethereum community created a controversial hard fork to save it.



Reentrancy Vulnerability

The most well-known vulnerability due to the DAO attack.

```
1 contract SendBalance {
2  mapping (address => uint) userBalances;
3  bool withdrawn = false;
4  function getBalance(address u) constant returns(uint){
5   return userBalances[u];
6  }
7  function addToBalance() {
8   userBalances[msg.sender] += msg.value;
9  }
10 function withdrawBalance(){
11  if (!(msg.sender.call.value(
12    userBalances[msg.sender])())) { throw; }
13  userBalances[msg.sender] = 0;
14  }}
```

Figure 7: An example of the reentrancy bug. The contract implements a simple bank account.

51% Attacks in Practice

小幣、獎勵減半後的熱門幣，有被double spent的風險

【Verge｜Bitcoin Gold｜Monacoin】加密貨幣「51%攻擊」頻傳，兩週內已三起，問題均未獲解決

by Steve Jr Lin — 2018-05-26 in 其他幣別 602

Charlie Lee [LTC ⚡]
@SatoshiLite

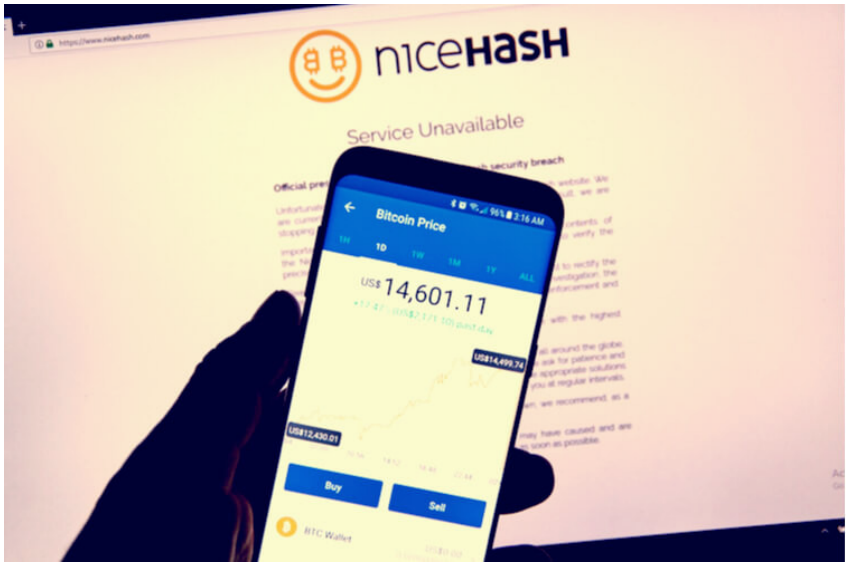
Follow

crypto51.app shows how easy it is to 51% attack some of the smaller PoW coins. For some coins, 100% of the hashrate can be rented from NiceHash, which removes the capital costs of the attack!

For example, Bytecoin (marketcap ~$1B) can be 51% attacked for $557! 😲

Learn More

Name	Symbol	Market Cap	Algorithm	Hash Rate	1h Attack Cost	NiceHash-able
Bitcoin	BTC	$121.55 B	SHA-256	32,798 PH/s	*$553,982*	2%
Ethereum	ETH	$51.96 B	Ethash	213 TH/s	*$360,114*	3%
Bitcoin Cash	BCH	$15.27 B	SHA-256	4,268 PH/s	*$72,093*	12%
Litecoin	LTC	$6.35 B	Scrypt	313 TH/s	*$64,954*	6%
Monero	XMR	$2.39 B	CryptoNightV7	402 MH/s	*$21,151*	13%
Dash	DASH	$2.35 B	X11	2 PH/s	*$15,439*	27%
Ethereum Classic	ETC	$1.47 B	Ethash	6 TH/s	*$10,643*	89%
Bytecoin	BCN	$944.33 M	CryptoNight	158 MH/s	$557	225%
Zcash	ZEC	$928.92 M	Equihash	576 MH/s	*$65,984*	19%
Bitcoin Gold	BTG	$694.32 M	Equihash	34 MH/s	$3,858	329%
Bitcoin Private	BTCP	$446.90 M	Equihash	7 MH/s	$778	1,632%
Dogecoin	DOGE	$366.81 M	Scrypt	197 TH/s	*$40,908*	10%
MonaCoin	MONA	$199.41 M	Lyra2REv2	2 TH/s	$2,889	464%
Electroneum	ETN	$151.89 M	CryptoNight	2 GH/s	*$7,383*	17%
ZenCash	ZEN	$111.52 M	Equihash	86 MH/s	$9,880	128%
Vertcoin	VTC	$71.30 M	Lyra2REv2	982 GH/s	$1,326	1,012%



Cryptocurrencies

What is currency?

To understand *cryptocurrency*, we need to understand *currency* first.

- A currency refers to money in any form when in actual use or circulation as a medium of exchange, such as banknotes or coins.
- As a medium of exchange, is our currency good enough?

Source: https://www.emaze.com/@ACCRTLZR/Money-project

Currency Evolution

1. Exchange things (past): not convenient
2. Gold (past): not convenient
3. Fiat money (now): not convenient, and you need to trust the issuers (i.e., government / bank)
4. Cryptocurrency: convenient and decentralized

cf.
Digital Currency: Don't need to bring cash / bill / banknote
Cryptocurrency: Decentralized, not own by any third-party / bank / government

What is cryptocurrency?

All currencies need security measures to prevent cheating—such as double spending, tampering, inconsistent state, etc.

- Fiat currencies are protected by central authorities.
- Cryptocurrencies are protected technologically, with the help of cryptography, without relying on a central authority.

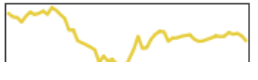
Why do we need to use cryptocurrency?

As a medium of exchange, cryptocurrency is different from (better than?) traditional currency in the following aspects:

- Decentralization: Not issued and controlled by any third-party, including governments or banks.
- Convenient: Real value can transfer over the Internet, fast with low fee (some with zero fee).
- Security: protected by cryptography and consensus algorithm. Security of your money is reduced to the security of your private key.

How do cryptocurrencies work? Bitcoin as an example

All ▾ Coins ▾ Tokens ▾ USD ▾ Next 100 → View All

#	Name	Market Cap	Price	Volume (24h)	Circulating Supply	Change (24h)	Price Graph (7d)
1	Bitcoin	$205,976,429,707	$12,248.70	$10,766,300,000	16,816,187 BTC	-1.95%	
2	Ethereum	$106,820,255,263	$1,099.96	$3,518,220,000	97,112,854 ETH	-0.48%	
3	Ripple	$56,610,284,173	$1.46	$2,092,210,000	38,739,142,811 XRP *	-6.29%	
4	Bitcoin Cash	$31,850,599,972	$1,882.07	$1,048,240,000	16,923,175 BCH	0.60%	
5	Cardano	$17,038,778,143	$0.657181	$682,458,000	25,927,070,538 ADA *	-3.56%	
6	Litecoin	$10,886,653,891	$198.48	$486,994,000	54,850,683 LTC	-2.07%	
7	NEM	$10,011,869,999	$1.11	$105,784,000	8,999,999,999 XEM *	-1.24%	
8	NEO	$9,160,385,000	$140.93	$443,595,000	65,000,000 NEO *	-1.69%	
9	Stellar	$8,980,675,035	$0.501924	$169,705,000	17,892,499,731 XLM *	-3.46%	

https://coinmarketcap.com/

Bitcoin (BTC) - Pioneer of Cryptocurrency

Proposed in Jan. 2009 by Satoshi Nakamoto

35% marketcap

Gateway to other cryptocurrencies

Bitcoin (BTC) - Pioneer of Cryptocurrency

Bitcoin relies on several well-known cryptographic constructions:

- Cryptographic hash functions (Merkle tree, hash chain)
- Digital signatures
- Public key as identities
- Proof-of-work

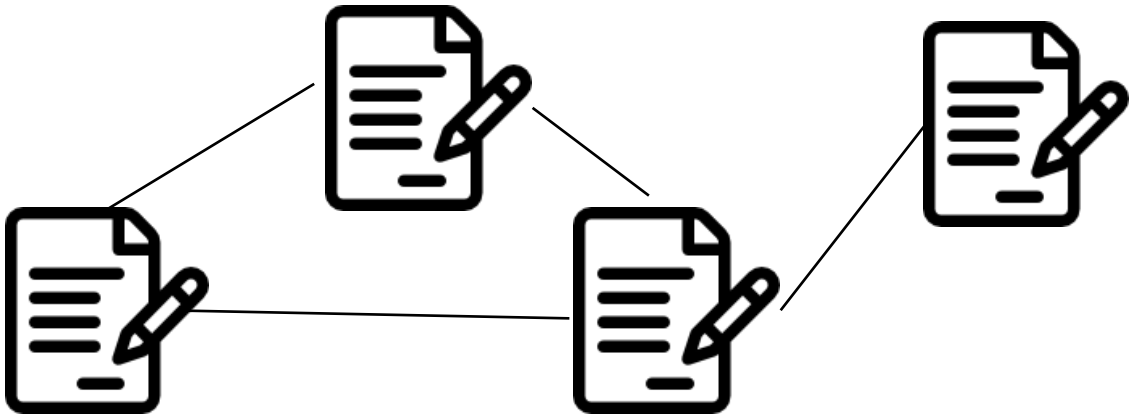
It's success is largely due to the combination of technical methods and incentive engineering

The Double-spending Problem

With a cash / banknote in real world, no one can double-spend his money.

How about a digital cash?

- A *centralized trusted* entity can maintain an append-only ledger to detect double spending
- Challenge: How to achieve *decentralization*, such that a set of nodes can agree on a consistent view without trusting each other?



Bitcoin Blockchain: Append-only Ledger

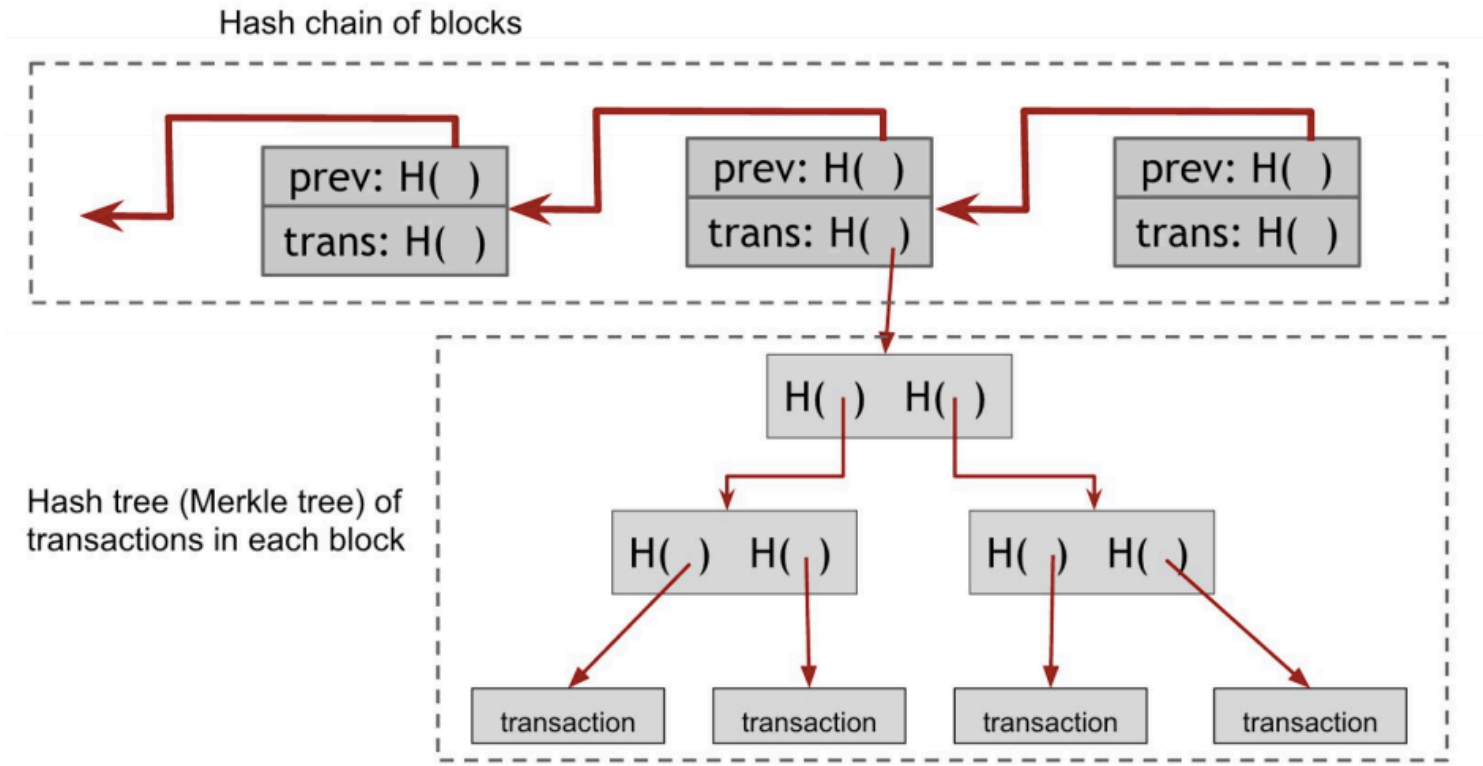


Figure 3.8. The Bitcoin block chain contains two different hash structures. The first is a hash chain of blocks that links the different blocks to one another. The second is internal to each block and is a Merkle Tree of transactions within the blocks.

Hash Chain of Blocks

A useful data structure to create tamper-evident, append-only log

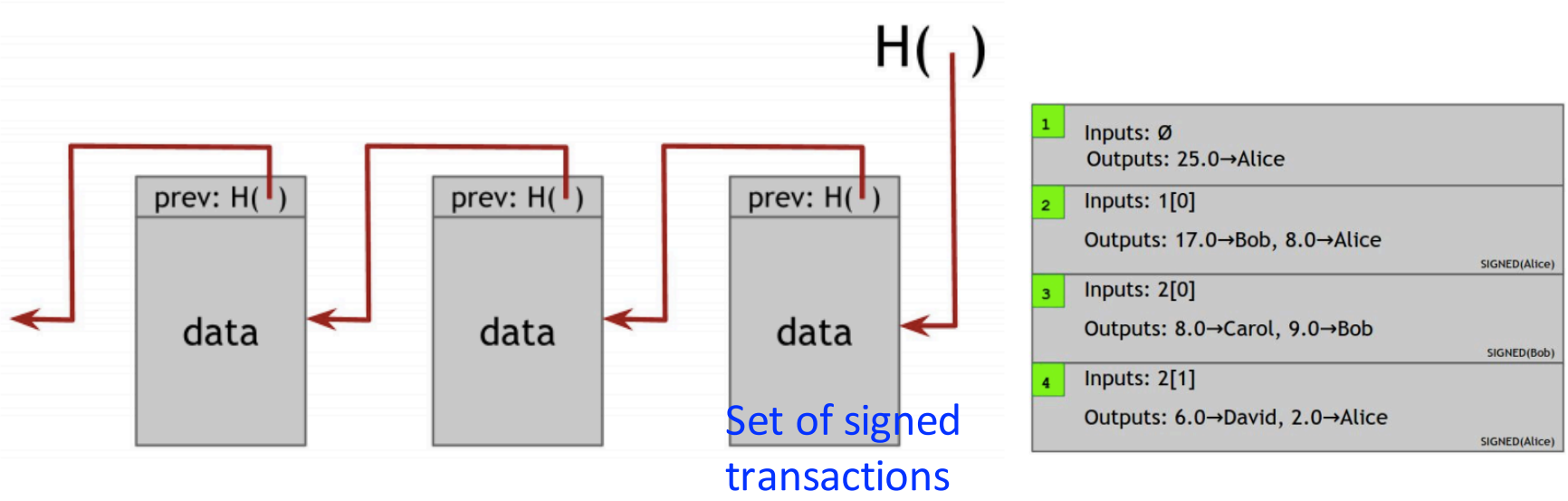


Figure 1.5 Block chain. A block chain is a linked list that is built with hash pointers instead of pointers.

Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. 2016. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, Princeton, NJ, USA.

Merkle Hash Tree

A useful data structure to supports concise proof-of-membership

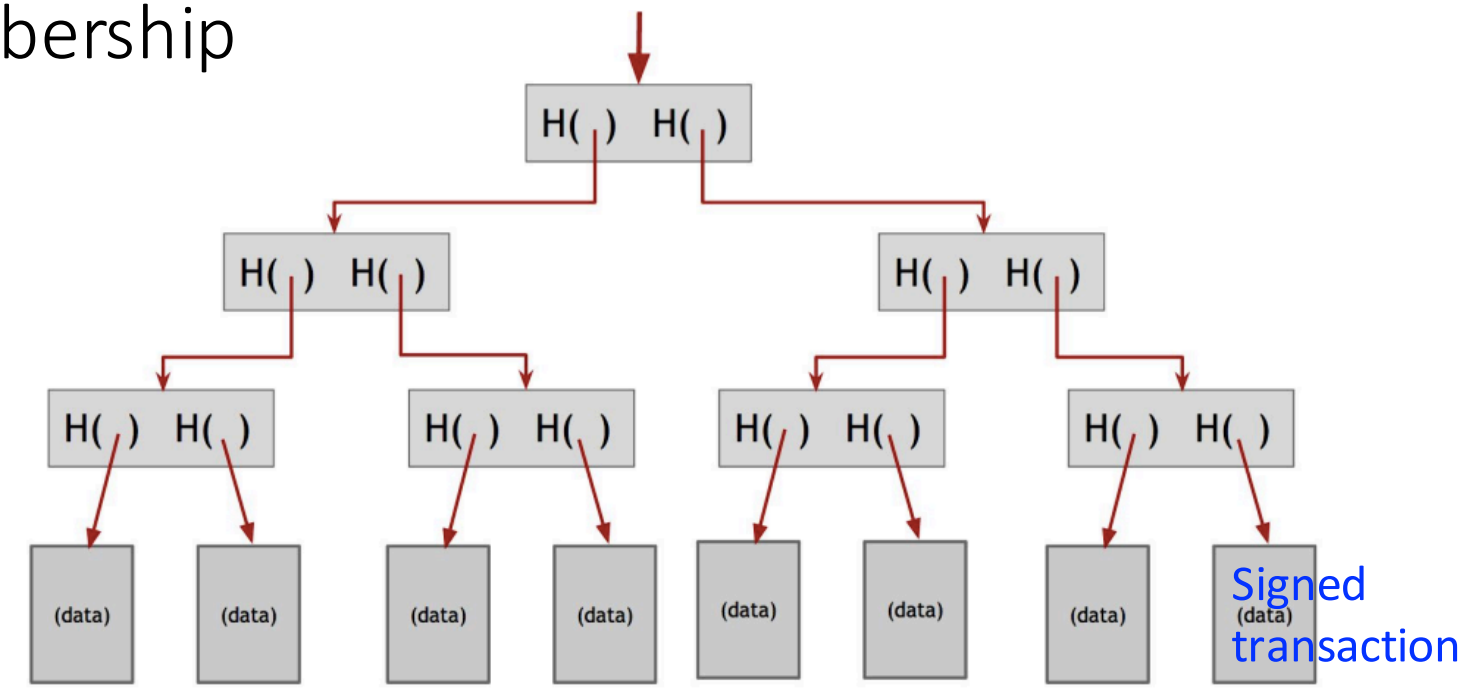


Figure 1.7 Merkle tree. In a Merkle tree, data blocks are grouped in pairs and the hash of each of these blocks is stored in a parent node. The parent nodes are in turn grouped in pairs and their hashes stored one level up the tree. This continues all the way up the tree until we reach the root node.

The Double-spending Problem in Decentralized Setting

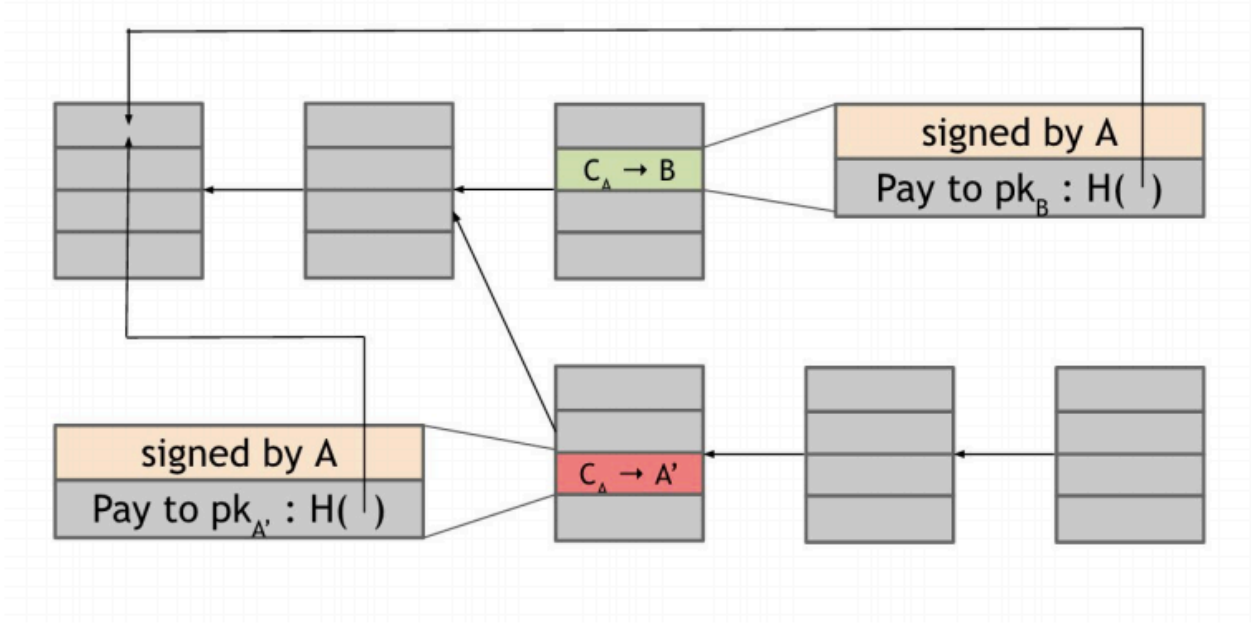


Figure 2.2 A double spend attempt. Alice creates two transactions: one in which she sends Bob Bitcoins, and a second in which she double spends those Bitcoins by sending them to a different address that she controls. As they spend the same Bitcoins, only one of these transactions can be included in the block chain. The arrows are pointers from one block to the previous block that it extends including a hash of that previous block within its own contents. C_A is used to denote a coin owned by Alice.

How to prevent double spending in decentralized setting?

Cryptography (digital signatures and hashes) can protect against invalid transactions and modify old transactions

However, cryptography along cannot protect against double spending or inclusion of invalid transactions!

A consensus algorithm is needed to ensure

- Invalid transactions are not included in the block chain (when the majority of nodes are honest)
- Protection against double spending

How to prevent double spending in decentralized setting?

Distributed Ledger Technology (DLT)

- A public, distributed database that can be easily verified but cannot be modified arbitrarily.
- Guarded with cryptography and consensus algorithms

Ownership of bitcoins: other nodes agreeing that a given party owns those bitcoins

How to achieve consensus?

Bitcoin randomly elects one node to create the next block using proof-of-work.

- A node with more computing power has a higher probability to be elected.

Bitcoin incentivizes nodes to help create blocks and validate transactions through *block reward* and *transaction fee.*

- Reward those that created the blocks that did end up on the long-term consensus chain.

Incentives and Proof-of-Work

Try different nonces such that ethe hash is less than a target value.

H(**nonce** || prev_hash || merkle_root) < target

This target can be adjusted to control the difficulty of calculation and the speed of *block mining*.

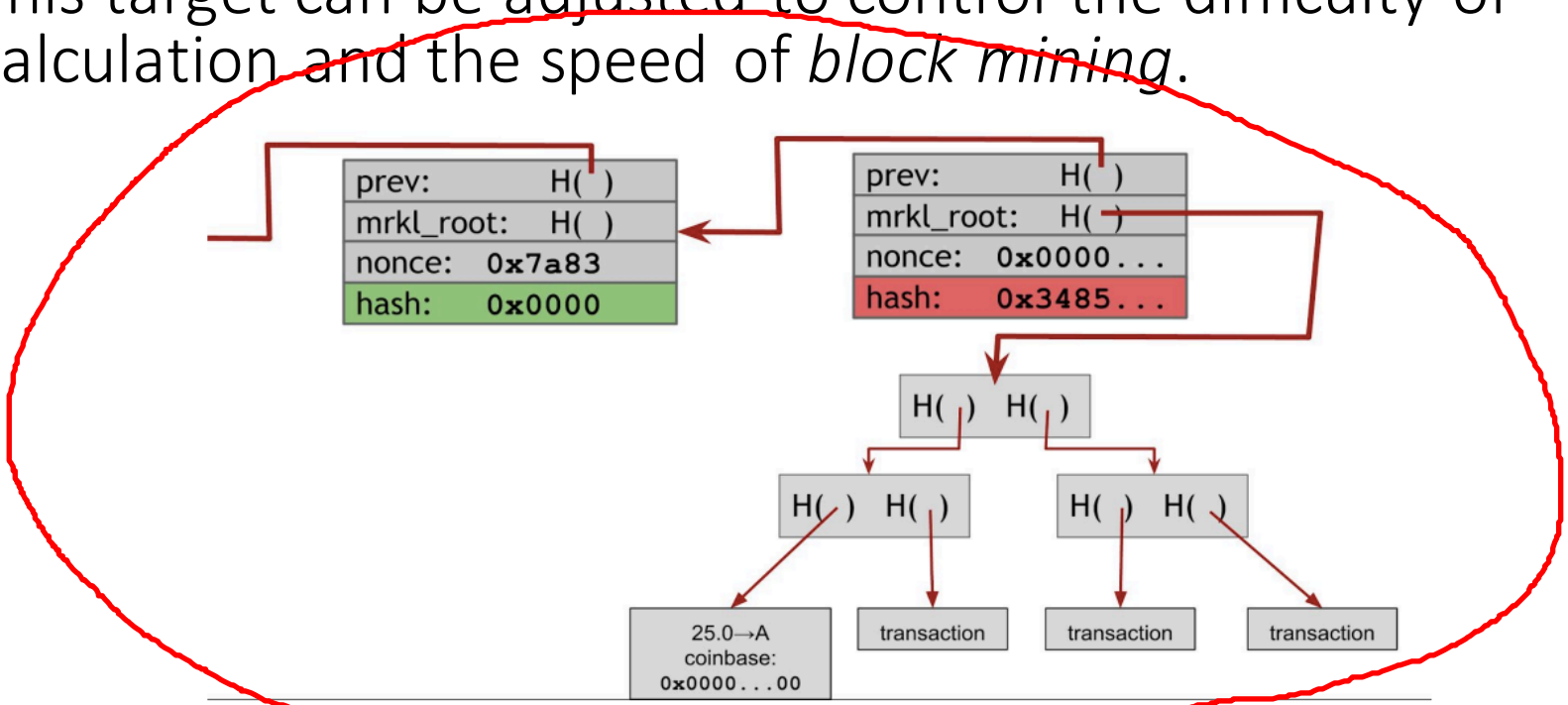


Figure 5.1: Finding a valid block. In this example, the miner tries a nonce of all 0s. It does not produce a valid hash output, so the miner would then proceed to try a different nonce.

Miners and Mining

Mining: finding a nonce generating a legitimate hash

Miner: people who mine

Mining Pool: a collaboration of miners to get advantages on mining

Reward: transaction fee of the transactions put in the block + newly created Bitcoin

The Task of a Miner

1. Listen for transactions
2. Maintain a blockchain and listen for new blocks
3. Assemble a candidate block
4. Fine a nonce that makes your block valid
5. Broadcast your block and hope it is accepted
6. Profit
    - Block reward was 25 Bitcoins in 2015
    - Transaction fee, about 1% of block reward

Validating transactions and blocks are the fundamental steps to Bitcoin. Race to find blocks and profit are ways to incentivize miners to perform the fundamental steps!

Similarity to Gold Mining

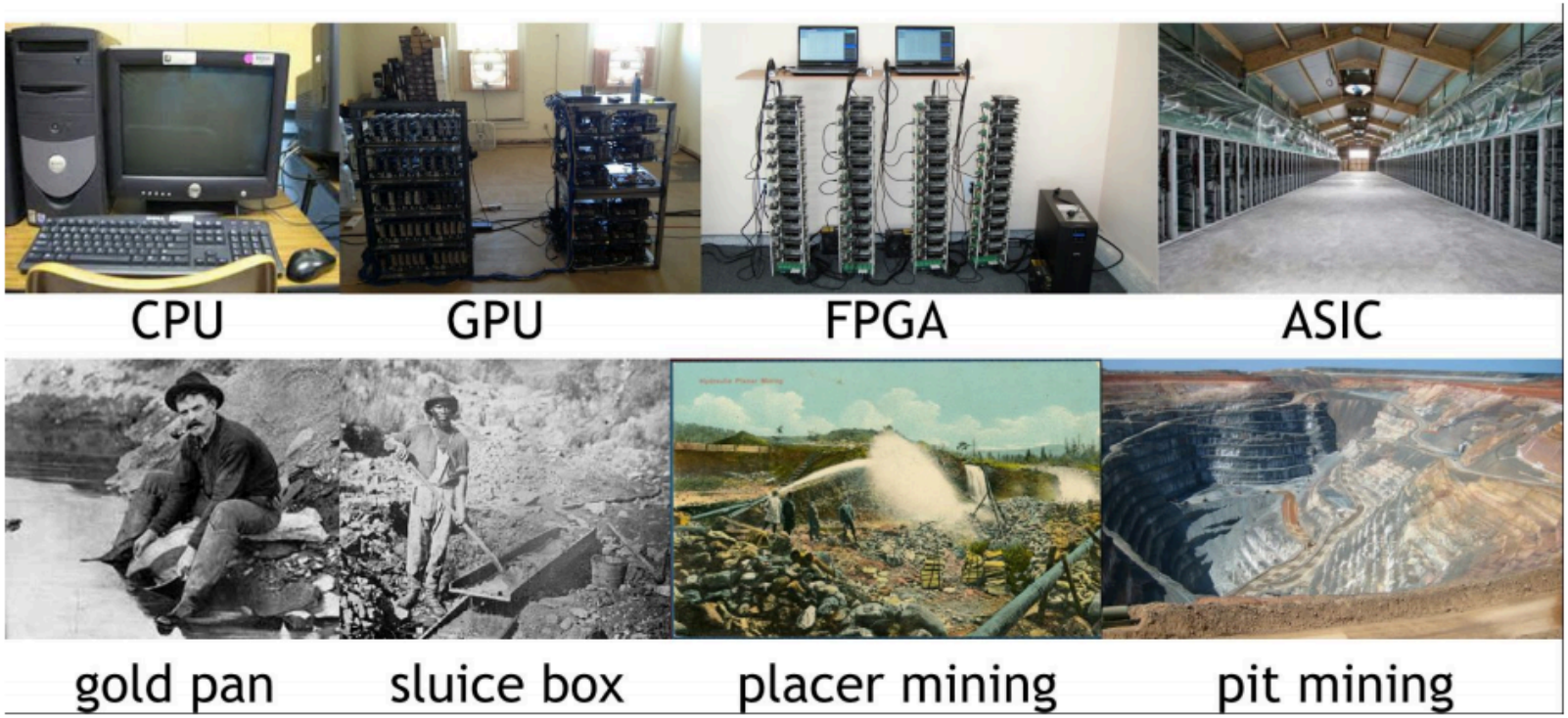


Figure 5.10: Evolution of mining. We can see a clear parallel between the evolution of Bitcoin mining and the evolution of gold mining. Both were initially friendly to individuals and over time became massive operations controlled by large companies.

How Does It Work - Value Transfer

1. A payer, Bob, signs a transaction with his private key, generating a digital signature that can be verified by his public key.
2. The payer broadcasts his transaction to the Bitcoin P2P network.
3. When a miner, Alice, mined a new block, she will (possibly) put the transaction into the block.
4. After several confirmations (6 in Bitcoin), the payment is considered to be received and completed.

Challenges of Cryptocurrencies

Trilemma

- Scalability
- Security
- Decentralization

Other issues

- Speed / High transaction fee
- Waste of power
- Anonymity (or lack of anonymity)
- Attack via non-default strategies
- Hard to update
- …

Built-in limitations to the Bitcoin protocol

Many constraints are hardcoded into bitcoin protocol

- Total numbers of bitcoins and structure of mining rewards
- Size of block
  - Limits to about 7 transactions per second
  - Visa: 2000 transactions per second on average, 10000 at peak
  - Paypal: 100 per second
- Fixed cryptographic algorithms
  - Signature: ECDSA over secp256k1, …

Changing the protocol?

Hard fork (unacceptable)

- A change that makes the blockchain split permanently
- The new feature was previously considered invalid
- Nodes running different versions of software will extend different branches

Soft fork

- The newly added feature makes validation rules stricter
- Requiring enough nodes to switch to the new version
- Will have small, temporary forks

Does Bitcoin provide anonymity?

Depending on the interpretations of anonymity:

1. No real name (but might have a pseudonym)
2. No name at all

Bitcoin provides psudonymity but not unlinkability

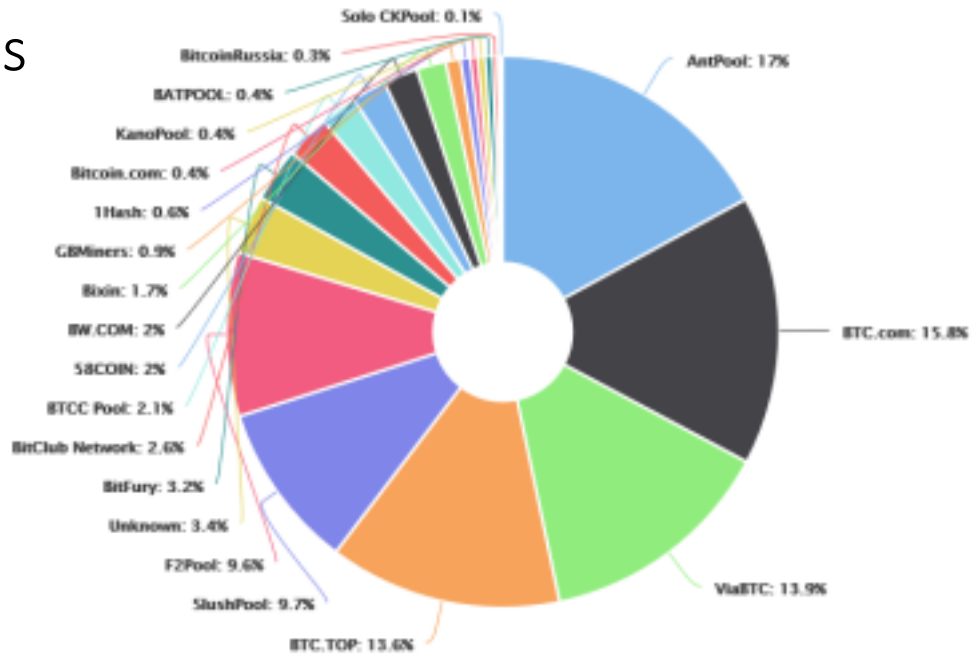
Cryptocurrencies that improve unlinkablity:

- CoinJoin (mixing)
- Zerocoin, Zerocash (zero-knowledge proof)

Decentralized vs. Centralized

While bitcoin is designed to be decentralized, its use in practice causes centralization to some extent:

- Mining pools
- Internet routing infrastructure
- Cryptocurrency exchanges
- Online wallets



Consensus

Distributed Consensus

There are *n* nodes that each have an input value. Some of these nodes are faulty or malicious.

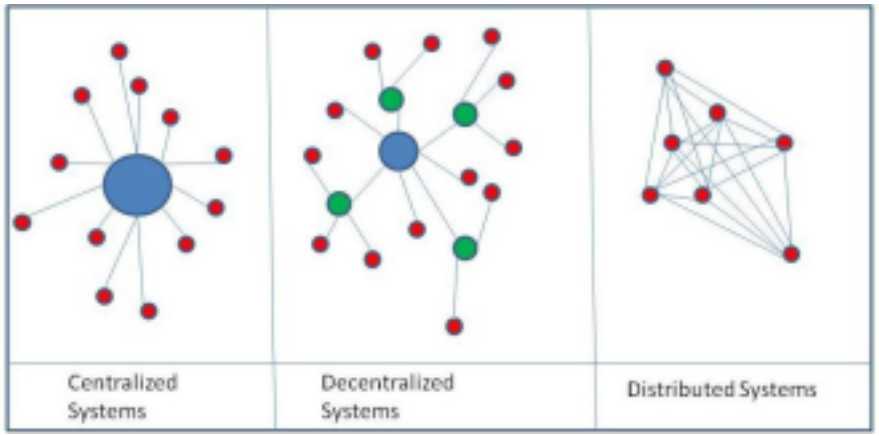
A distributed consensus protocol has the following two properties:

1. It must terminate with all honest nodes in agreement on the value
2. The value must have been generated by an honest node

Distributed Consensus

Advantages

- Trustless / Permissionless
- Fair / Uncensored
- Stable and DDoS-resistant



Disadvantages

- Slow
- Waste of energy
- Waste of network traffic and computational power

Why is it so hard to reach consensus?

Distributed Consensus – the Difficulties

Reaching a consensus is difficult because

1. Imperfections in the network (e.g., latency and nodes crash)
2. Dishonest nodes deliberately subvert the process

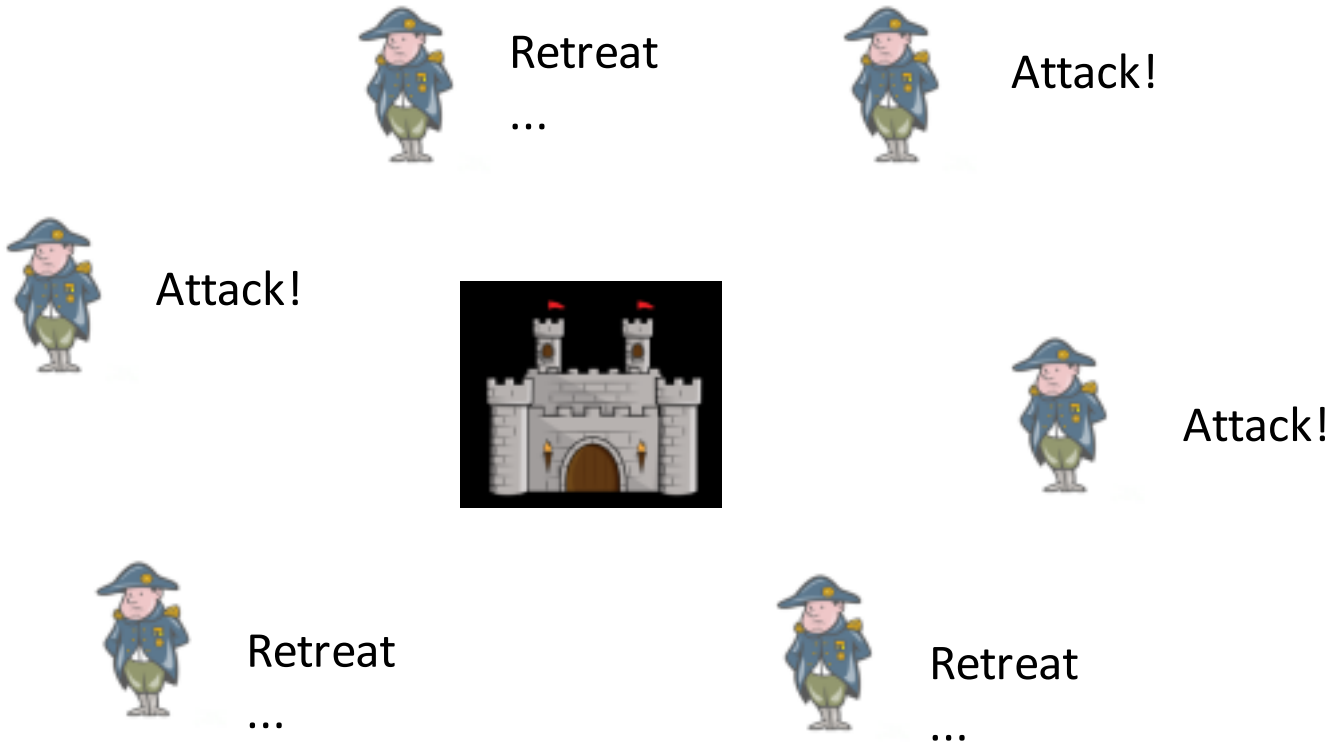
Dishonest node can:

- Never send any message / pretend to be a crashed honest node
- Send faulty messages
- Does not follow any steps of execution or algorithm

Dishonest node cannot:

- Send virus / kill other honest nodes (failure detector)
- Modify messages from honest node (protected by DSA & hash)

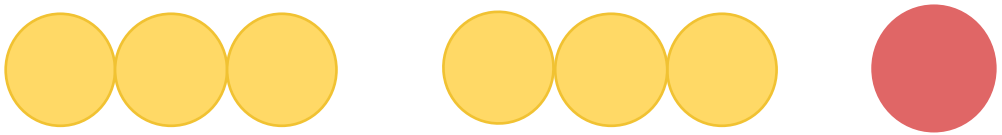
Impossibility result: Byzantine Generals' Problem



Impossible to reach consensus if >=1/3 of the generals are traitors

Byzantine Generals' Problem - Naive Voting

7 generals with 1 betrayer

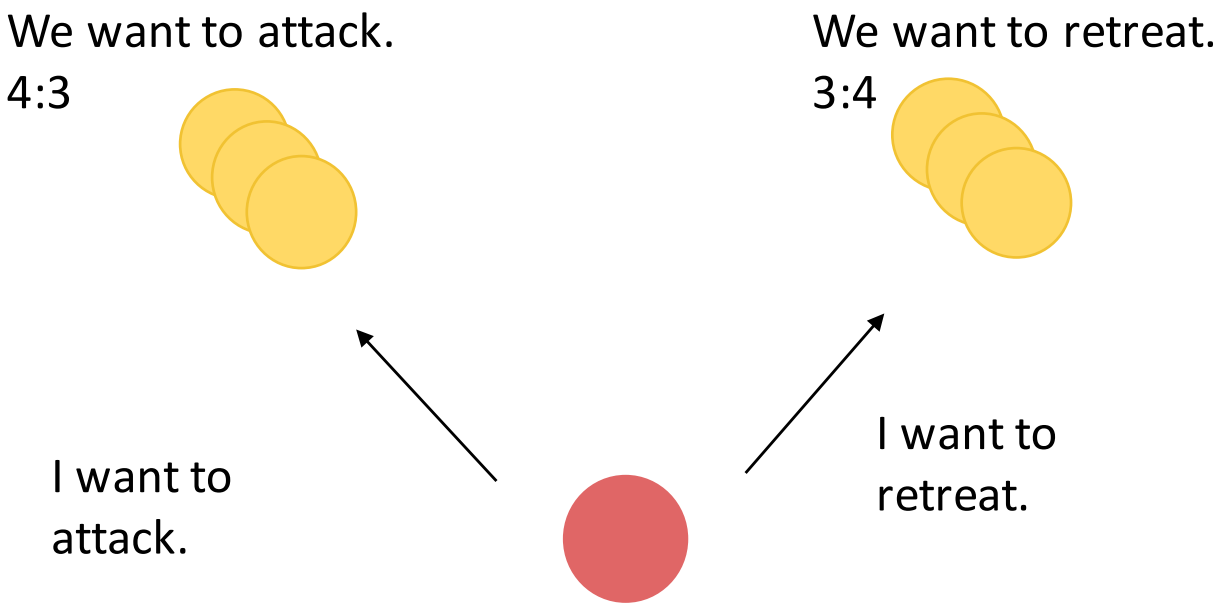


Byzantine Generals' Problem - Naive Voting

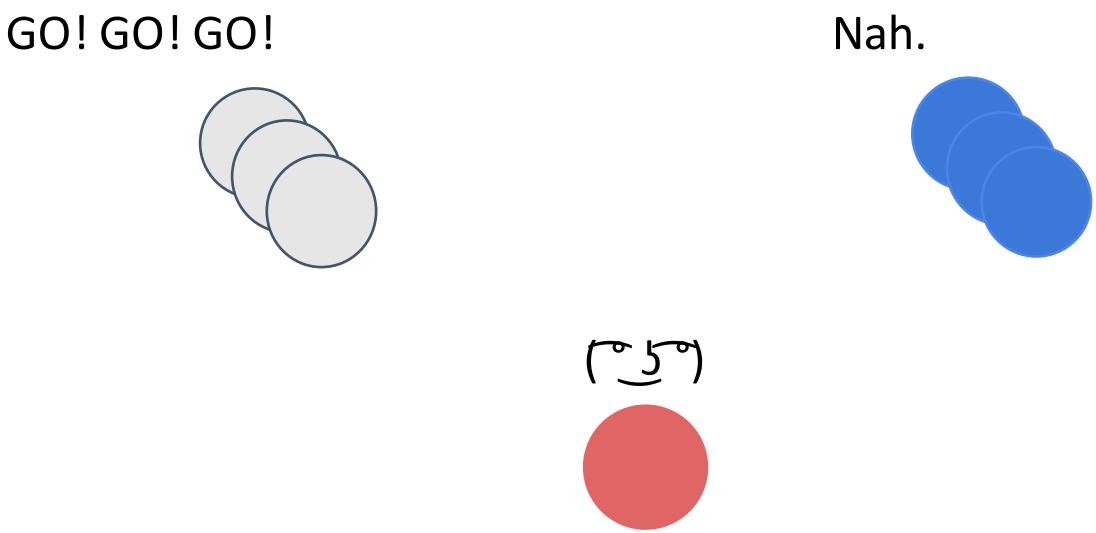
We want to attack.
3:3

We want to retreat.
3:3

Byzantine Generals' Problem - Naive Voting



Byzantine Generals' Problem - Naive Voting



Impossible to reach consensus if >=1/3 of the generals are traitors

Breaking Traditional Assumptions

Such impossibility results were proven in the context of distributed databases.

Bitcoin consensus works better in practice than theory!

Bitcoin violates traditional assumptions:

- Introduces the idea of incentives (possible in the context of currency systems, but not distributed databases)
- Embrace the notion of randomness (such that consensus takes place over a long time with high probability)

How to reach consensus?

Many different types of consensus algorithms

- PoW / PoS in cryptocurrency
- Atomic broadcast / total ordering in distributed database

Some basic blocks of consensus algorithms:

- Reliable Broadcast (RB)
- Byzantine Agreement (BA)
- Asynchronous Common Subset (ACS)

The Evolution of Consensus System

With the rise of cryptocurrency and other applications, the design of consensus algorithm has been discuss a lot lately.

Many promising projects have built new consensus algorithms to meet their own need.

First Generation: Proof-of-Work (PoW)

Mining / miner

Waste of energy and computational power

Slow

Example: Bitcoin, Ethereum...

Puzzles Requirements (for replacing PoW in Bitcoin)

Core requirements

1. Fast to verify
2. Adjustable difficulty
3. Progress free (chance of getting reward is independent of previous progress)

Additional requirements considered in altcoins

- ASIC-resistant puzzles
- Proof of *useful* work
- Non-outsourceable puzzles
- Virtual mining

Second Generation: Proof-of-Stake (PoS)

Validation / validator

No energy and computational power waste

(Should be) Fast

Cardano, EOS

Second Generation: Proof-of-Stake (PoS)

Mining is basically an election. We randomly pick a leader among miners to create the next block, proportionally to hashing power.

How about we just use a random process?

Proportionally to stake, because:

- It is recorded on blockchain itself.
- One can always buy hashing power with a lot of stake.
- To attack the system, you need to own 51% of the stake, which is difficult.
- One with a lot of stake cares more about the system.

PoS: elect a validator to mint a block

Fast and energy-efficient

Third Generation: Two Phase Commit

Use 2PC (RB, BA, ACS...) to reach consensus

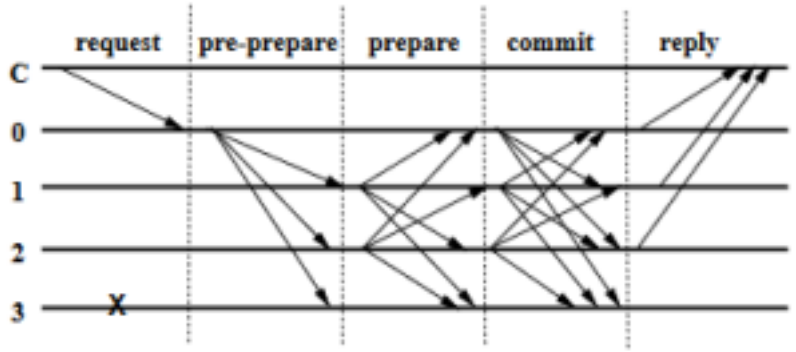
No energy and computational power waste

Should be the fastest

Example: PBFT, Honey Badger, NEO, HashGraph

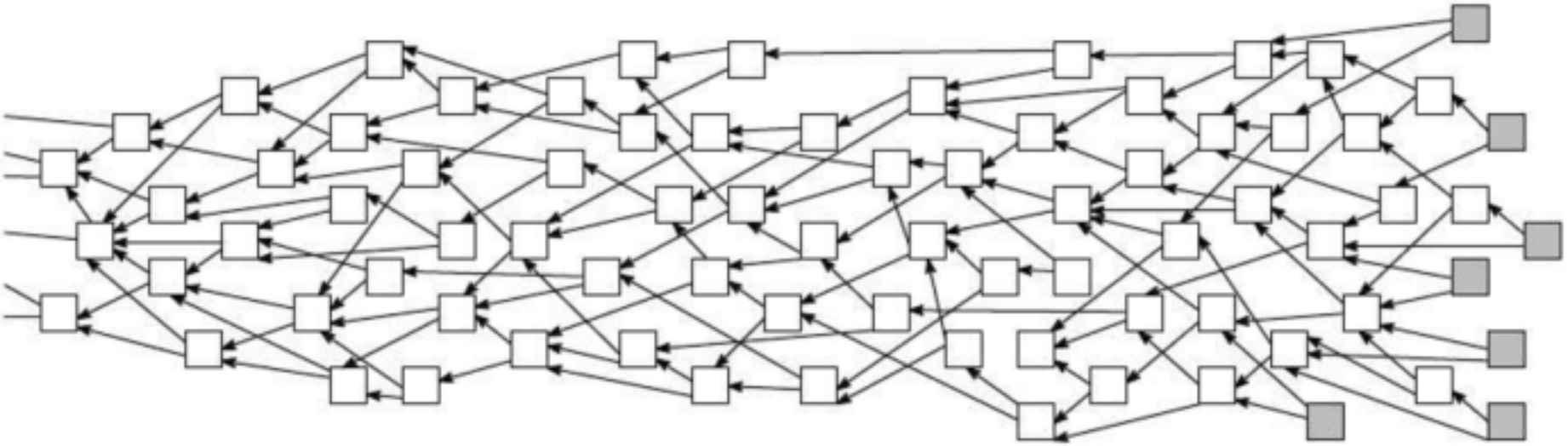
PBFT

- Each node takes turn to become leader in a round.
- Use RB to make sure the leader node won't fork (propose two different value).
- Leader makes a block, and the blockchain grows.
- A fast, beautiful solution with 2PC.



Trend of Next Generation Blockchain?

- DAG
- 2PC + PoS



Smart Contract and DApp

What is a Smart Contract?

A computer protocol to verify and enforce the negotiation or performance of a *contract*.

- A program running on blockchain that can execute transfer when certain events happen.
- Basically, a wallet + program.

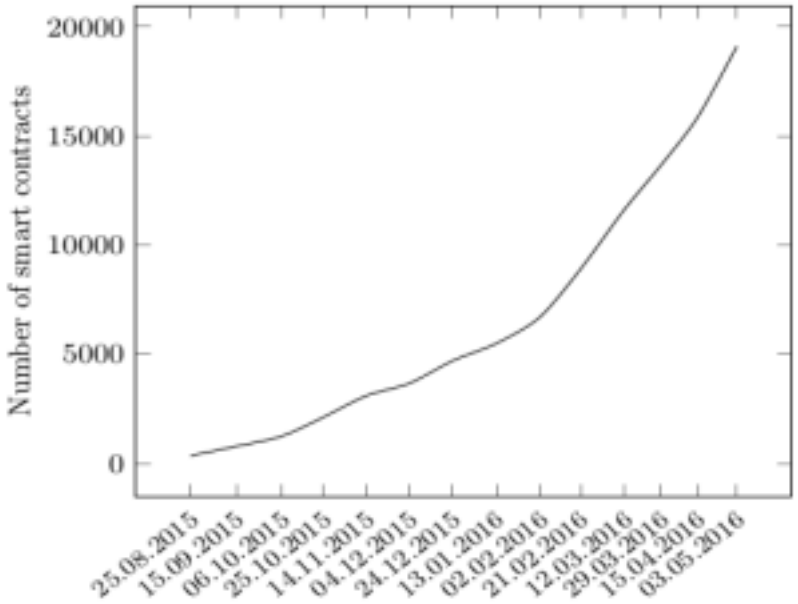
Used mostly with cryptocurrency (Ethereum). Executed by miners (rewarded with “gas”) with EVM when a transaction happens.

Also known as DApp (decentralized application).

Turing complete.

Why do we need smart contract?

- Programmable money
- Fair and transparent: no one can modify the program once it's deployed.
- Decentralized & Trustless: not controlled/owned/executed by any centralized party.



What is a Smart Contract - Example

- CryptoKitties
- The DAO (Decentralized Autonomous Organization) - a capital venture in Ethereum
- Slock.it - https://slock.it/
- ICO
- Voting

Security Issues in Smart Contract

A smart contract operates on an open network that arbitrary participants can join.

Once a smart contract is deployed, it can not be upgraded and patched.

Security issues in smart contract should be taken seriously since it is related to large amount of assets. Hackers are definitely well-incentivized to exploit flaws in smart contract to gain profit.

Security Issues in Smart Contract

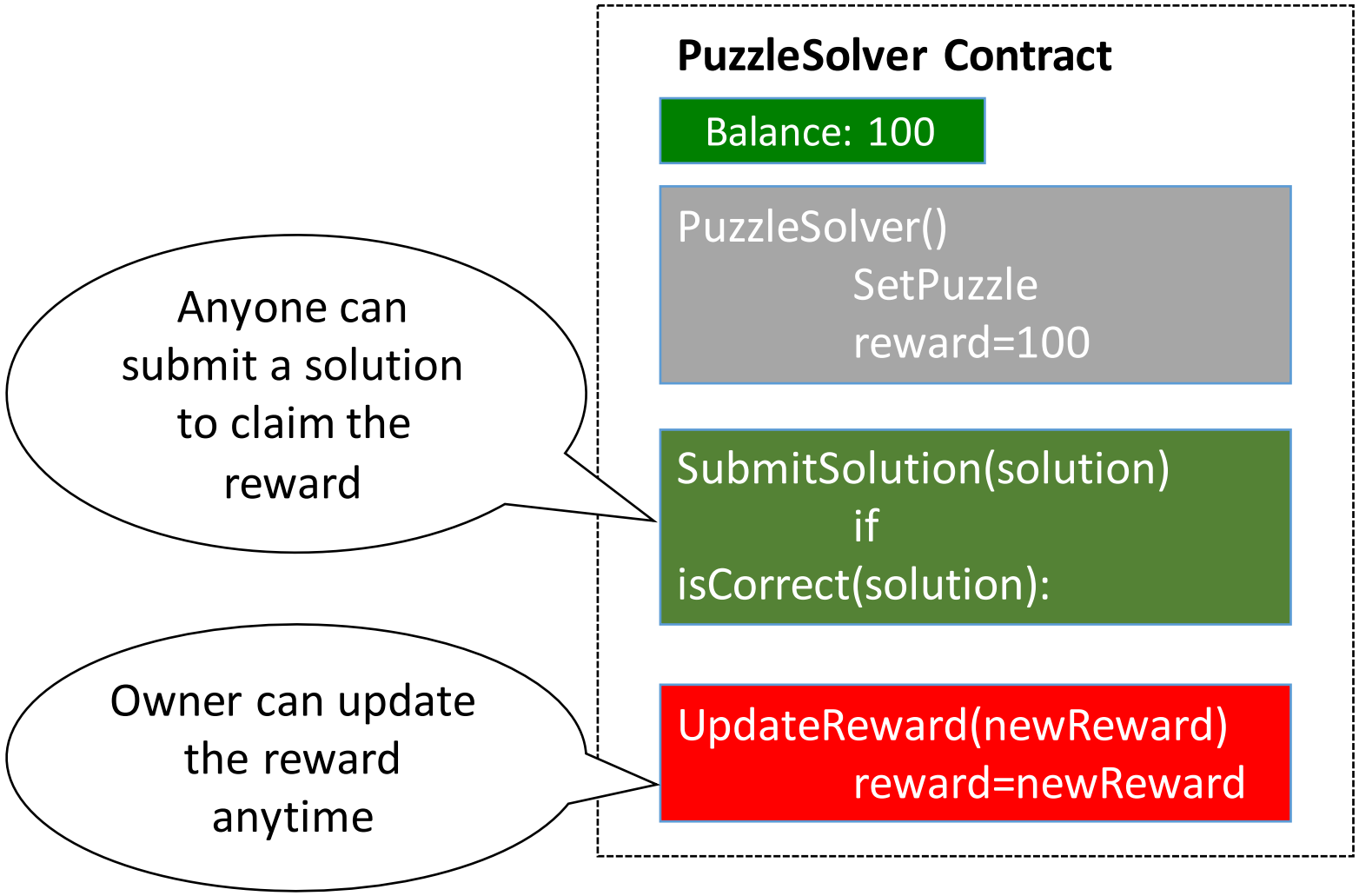
- Lack of oracle: timestamps and RNG can be faked.
- Front-run attack: miner can decide the order of execution.
- Human factors: codes with bugs, such as integer overflow, condition check failed, logic flaw…
- Bugs in EVM

Transaction-ordering Dependence

```
1 contract Puzzle{
2   address public owner;
3   bool public locked;
4   uint public reward;
5   bytes32 public diff;
6   bytes public solution;
7
8   function Puzzle() //constructor{
9     owner = msg.sender;
10    reward = msg.value;
11    locked = false;
12    diff = bytes32(11111); //pre-defined difficulty
13  }
14
15  function(){ //main code, runs at every invocation
16    if (msg.sender == owner){ //update reward
17      if (locked)
18        throw;
19      owner.send(reward);
20      reward = msg.value;
21    }
22    else
23      if (msg.data.length > 0){ //submit a solution
24        if (locked) throw;
25        if (sha256(msg.data) < diff){
26          msg.sender.send(reward); //send reward
27          solution = msg.data;
28          locked = true;
29        }}}}
```

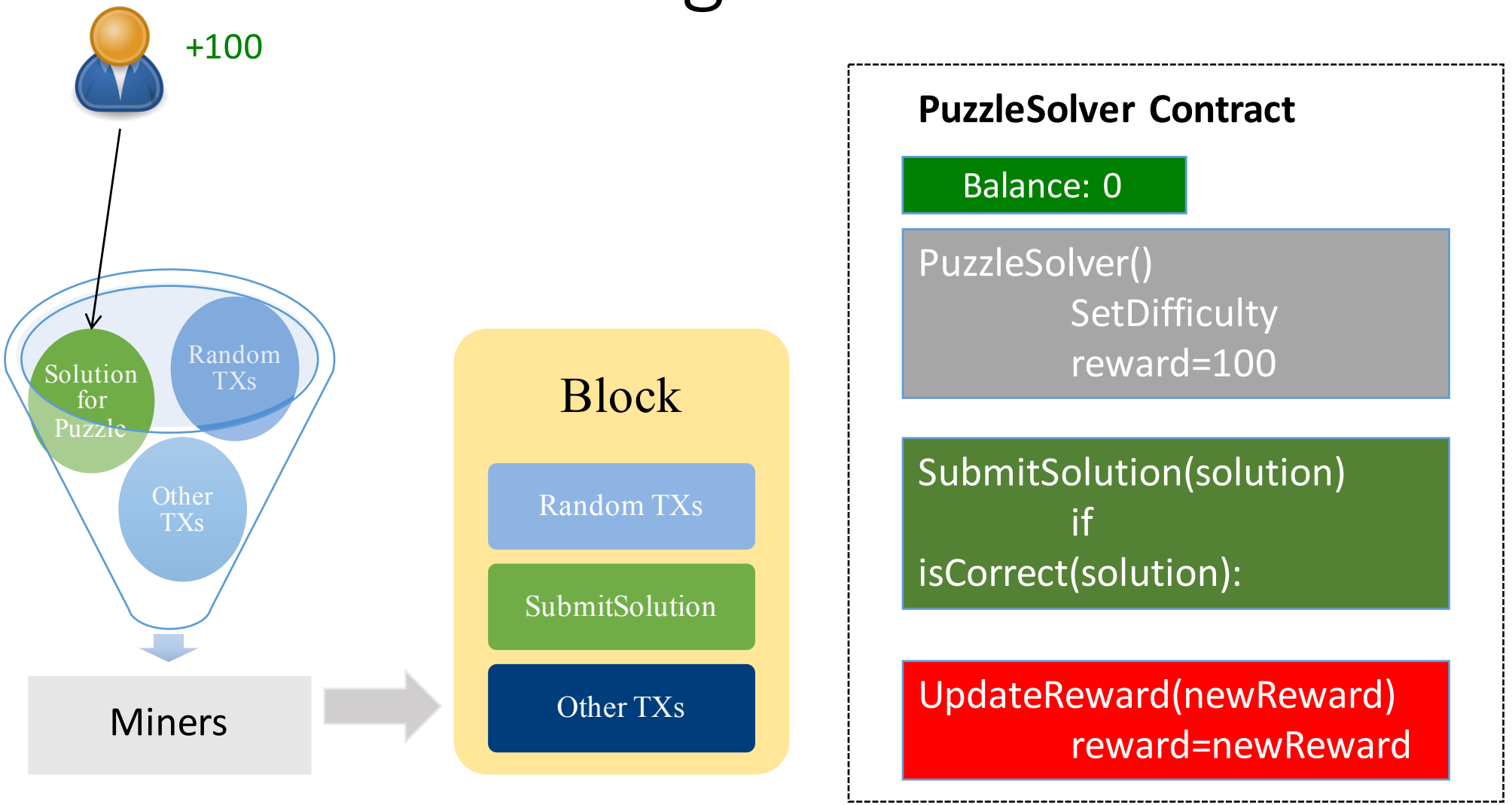
Figure 3: A contract that rewards users who solve a computational puzzle.

Example: Puzzle Solver



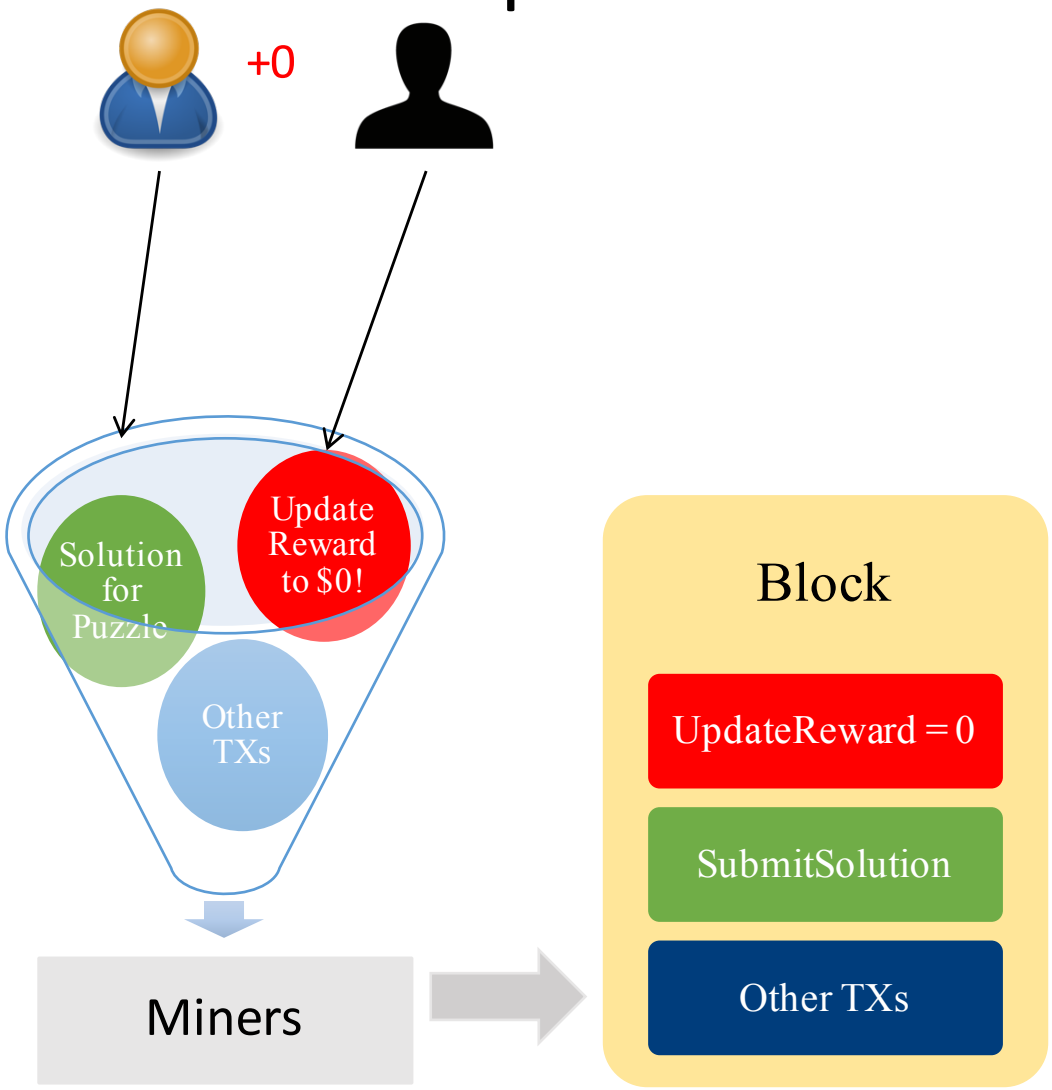
Loi Luu, Duc-Hiep Chu, Hrishi Olickel, Prateek Saxena, and Aquinas Hobor, "Making Smart Contracts Smarter," In ACM CCS 2016.

Scenario 1: SubmitSolution is trigerred



Loi Luu, Duc-Hiep Chu, Hrishi Olickel, Prateek Saxena, and Aquinas Hobor, "Making Smart Contracts Smarter," In ACM CCS 2016.

Scenario 2: Both SubmitSolution and UpdateReward are triggered



PuzzleSolver Contract

Balance: 0

```
PuzzleSolver()
        SetDifficulty
        reward=100
```

```
SubmitSolution(solution)
        if
isCorrect(solution):
```

```
UpdateReward(newReward)
        reward=newReward
```


Loi Luu, Duc-Hiep Chu, Hrishi Olickel, Prateek Saxena, and Aquinas Hobor, "Making Smart Contracts Smarter," In ACM CCS 2016.

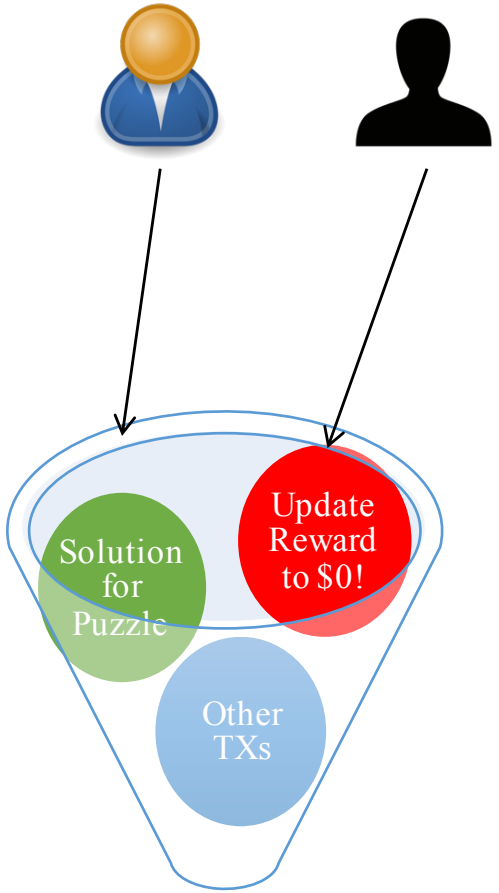
Transaction Ordering Dependence

Observed state != execution state

- The expectation of the state of the contract may not be true during execution.
- Miners decide the order of TXs

Can be coincidence

- Two transactions happen at the same time



Loi Luu, Duc-Hiep Chu, Hrishi Olickel, Prateek Saxena, and Aquinas Hobor, "Making Smart Contracts Smarter," In ACM CCS 2016.

Transaction Ordering Dependence

Observed state != execution state

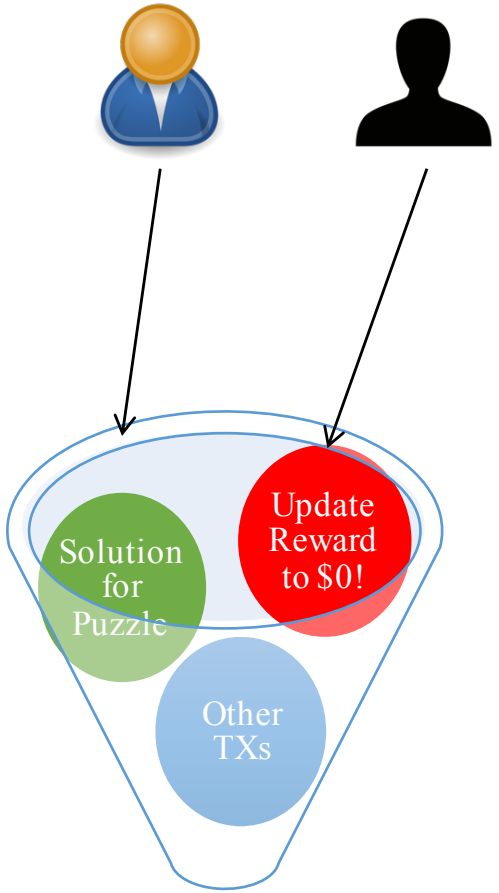
- The expectation of the state of the contract may not be true during execution.
- Miners decide the order of TXs

Can be coincidence

- Two transactions happen at the same time

Can be malicious

- Saw the targeted TX from the victim
- Submit the second TX to update the reward
- Both TXs enter the race



Loi Luu, Duc-Hiep Chu, Hrishi Olickel, Prateek Saxena, and Aquinas Hobor, "Making Smart Contracts Smarter," In ACM CCS 2016.

Reentrancy Vulnerability

The most well-known vulnerability due to the DAO attack.

```
1 contract SendBalance {
2  mapping (address => uint) userBalances;
3  bool withdrawn = false;
4  function getBalance(address u) constant returns(uint){
5   return userBalances[u];
6  }
7  function addToBalance() {
8   userBalances[msg.sender] += msg.value;
9  }
10 function withdrawBalance(){
11  if (!(msg.sender.call.value(
12    userBalances[msg.sender])())) { throw; }
13  userBalances[msg.sender] = 0;
14  }}
```

Figure 7: An example of the reentrancy bug. The contract implements a simple bank account.

Security Issues on Cryptocurrencies

Making Others Mine for You

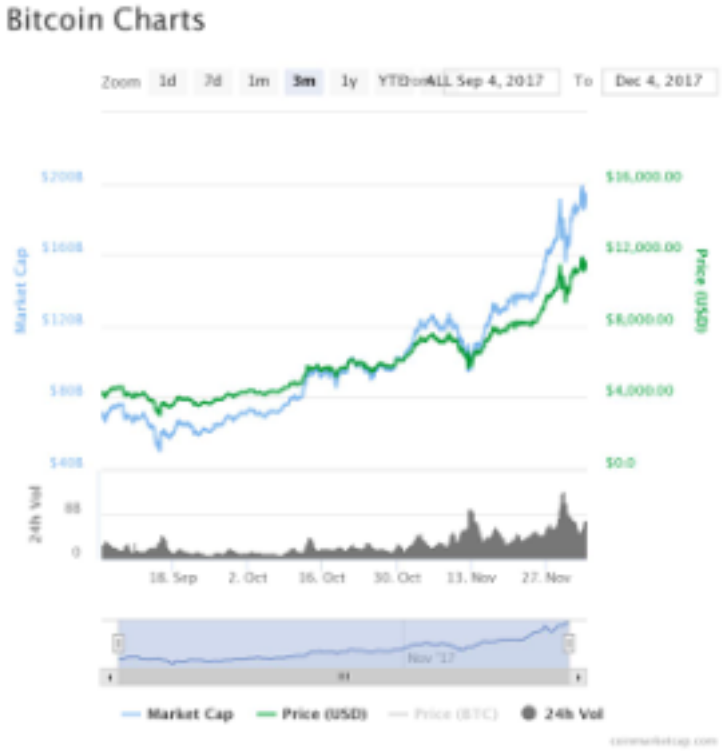
Deceiving

Attacking on Protocol / Networking

Introduction

Given the amount and price of Bitcoin, it obviously becomes an attractive attack target.

Blockchain itself is ideally secure, but other aspects like mining, wallet implementation, trading and networking might have security issues.



Making Others Mine for You - Trojans

Old-school mining trojans

After Bitcoin can only be efficiently mined by ASIC, those trojan has switch to other cryptocurrency such as Ethereum or Monero.

Today, mining can be done by client-side javascript with CPU.

Making Others Mine for You - Browser

Pirate Bay caught running browser based cryptocurrency miner

Coinhive

Evaluation: 0.011 XMR (2.75USD) in 24h with 1k sessions

Works with XSS

Prevention:

- Notice your abnormal CPU usage
- Use MinerBlock or No coin
- Use NoScript or ScriptSafe to block JavaScript

Deceiving

We can only get small profit from trojan mining.

How about directly make victims send cryptocurrency to us?

Deceiving - Clipboard Hijacking

- CryptoShuffler discovered by Kaspersky Lab
- Stole over 23 bitcoins already
- Also steal Ethereum, Monero, Litecoin, Zcash…
- Double-check before you send a transaction!

```
reg_create(&bitcoin_regexp, "(^)[13][a-km-zA-HJ-NP-Z0-9]{26,33}($| )", v7);
v76 = 0;
reg_create(&dogecoin_regexp, "(^| )D[a-km-zA-HJ-NP-Z0-9]{33}", v8);
LOBYTE(v76) = 1;
reg_create(&litecoin, "(^| )L[a-km-zA-HJ-NP-Z0-9]{33}", v9);
LOBYTE(v76) = 2;
reg_create(&wallet1_regexp, "(^| )X[a-km-zA-HJ-NP-Z0-9]{33}", v10);
LOBYTE(v76) = 3;
reg_create(&ether_regexp, "(^| )0x[a-fA-F0-9]{40}", v11);
LOBYTE(v76) = 4;
reg_create(&nonero_wallet, "(^| )4[a-km-zA-HJ-NP-Z0-9]{33}", v12);
LOBYTE(v76) = 5;
reg_create(&wallet2_regexp, "(^| )(M|M)[a-km-zA-HJ-NP-Z0-9]{33}($| )", v13);
LOBYTE(v76) = 6;
reg_create(&wallet3_regexp, "(^| )t[a-km-zA-HJ-NP-Z0-9]{34}", v14);
LOBYTE(v76) = 7;
reg_create(&wallet4_regexp, "(^| )P[a-km-zA-HJ-NP-Z0-9]{33}($| )", v15);
```

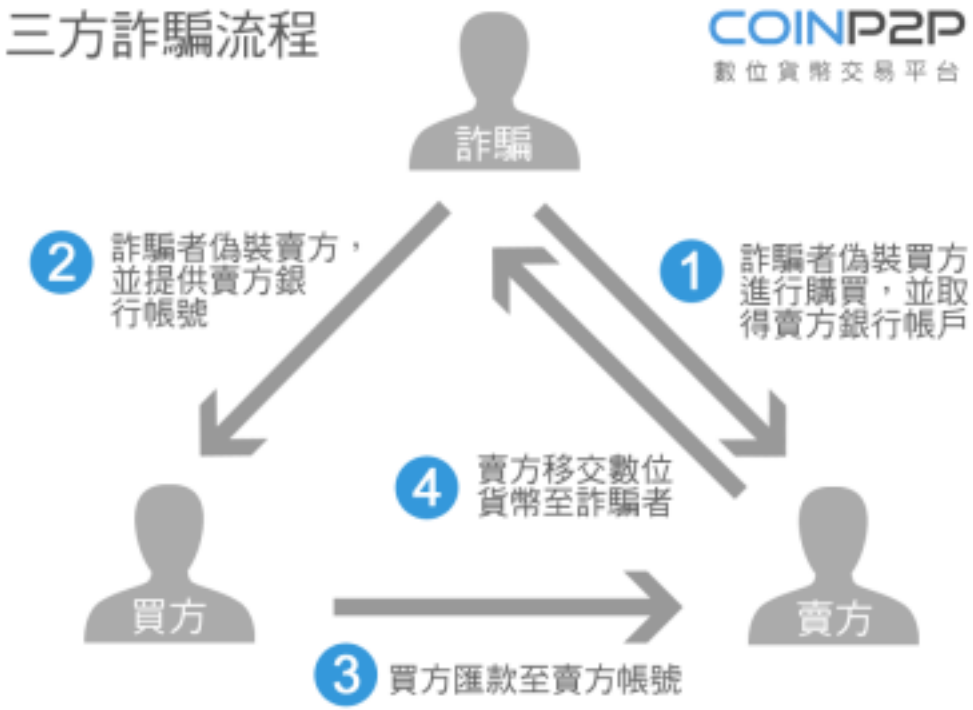
Deceiving - MITM

Buyer will not get Bitcoin.

Seller's bank account will be frozen.

Prevention:

- Verified by transferring a nonce?
- Verified by video chat?
- Use bank transfer as a secure channel?



Deceiving - Fake Wallet and Weak Wallet

Bitcoin Gold fake online wallet

Bitcoin Gold official wallet is compromised

Weak recovery phrases / weak password may be used to store private key.

Attack on Protocol / Networking

Selfish mining and routing attack.

These two attacks can hugely affect a blockchain system (51% attack).

Double-spending may be possible.

Miners / mining pool may lose a lot of profit.

Transaction confirmation will be even slower, causing a DoS.

Selfish Mining: a 25% attack against bitcoin network

Majority is not enough: Bitcoin mining is vulnerable

Hijacking Bitcoin: routing attacks on cryptocurrencies

Selfish Mining

Maintain a private chain and publish the blocks you mined selectively.

Try to invalidate other honest miners' work.

Now those honest miners have incentive to join you.

=> 51 % attack

Routing Attack

After all, Bitcoin is a peer-to-peer network…

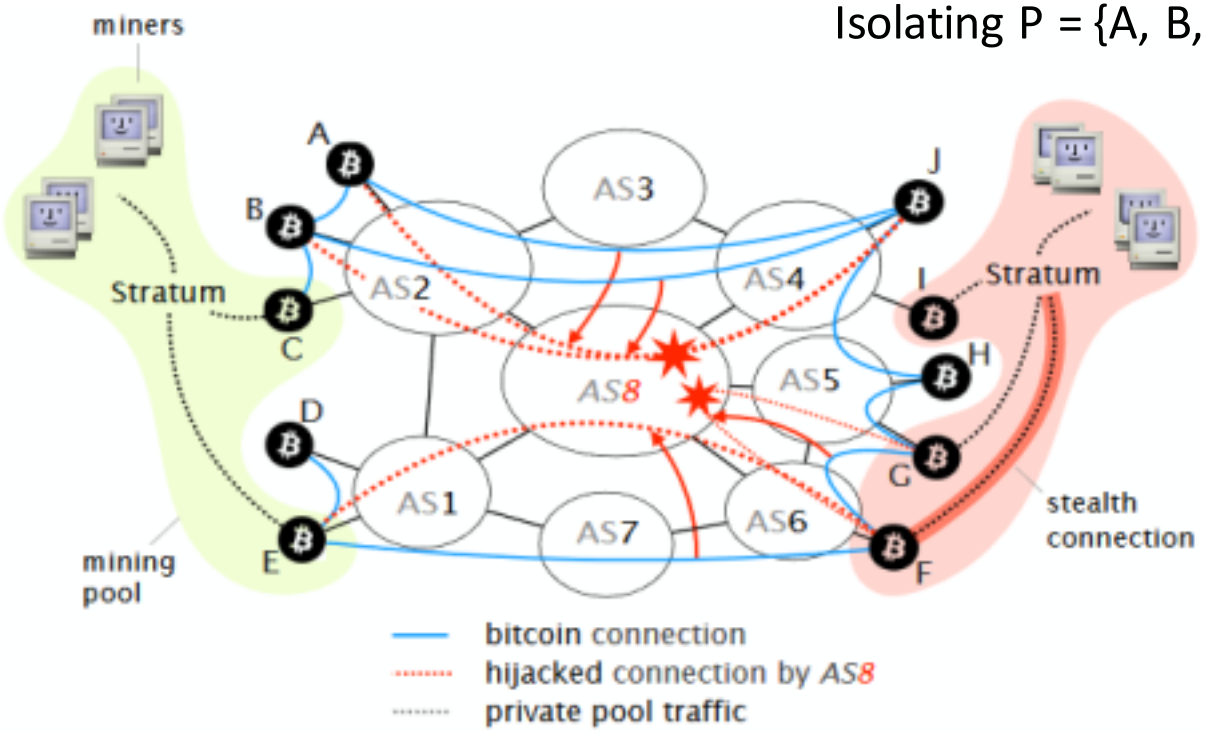
How about attack on the Internet routing itself?

Routing Attack Example - Partition Attack

The goal of this attack is to completely disconnect a set of nodes from the network.

BGP hijacking by AS-level attacker.

Isolating P = {A, B, C, D, E, F}



BGP hijacking

Ways to hijack:

- An AS announces a IP that it does not have.
- An AS announces a more specific prefix.
- An AS announces it has shorter path.

Pakistan blackholed traffic to Youtube

Resource public key infrastructure (RPKI)

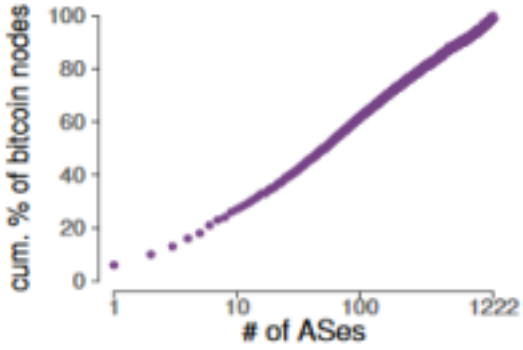
- BGP version of CA
- AS can verify a received BGP information, including IP prefix and its max length.

Routing Attack

Routing attack can only succeed because:

- Bitcoin connections are routed in plaintext without integrity check.
- Bitcoin nodes are centralized from routing perspective.

Works on other P2P cryptocurrency such as Ethereum, Litecoin and Zcash.



(a) Only 13 ASes host 30% of the entire network, while 50 ASes host 50% of the Bitcoin network.

Other Cryptocurrencies

Ethereum (ETH) - Program Runs on BlockChain

2015 July by Vitalik Buterin

Faster transaction with lower fee, a lot closer to currency

Turing-complete script and smart contract, a decentralized application (Dapp)

Same blockchain problems: scalability, security and decentralization...

Cardano (ADA) - PoS version Ethereum

Created by ETH co-founder, Charles Hoskinson

Proof of Stake

Random generator and follow the Satoshi

Nash equilibrium

Ouroboros: The first provably secure PoS

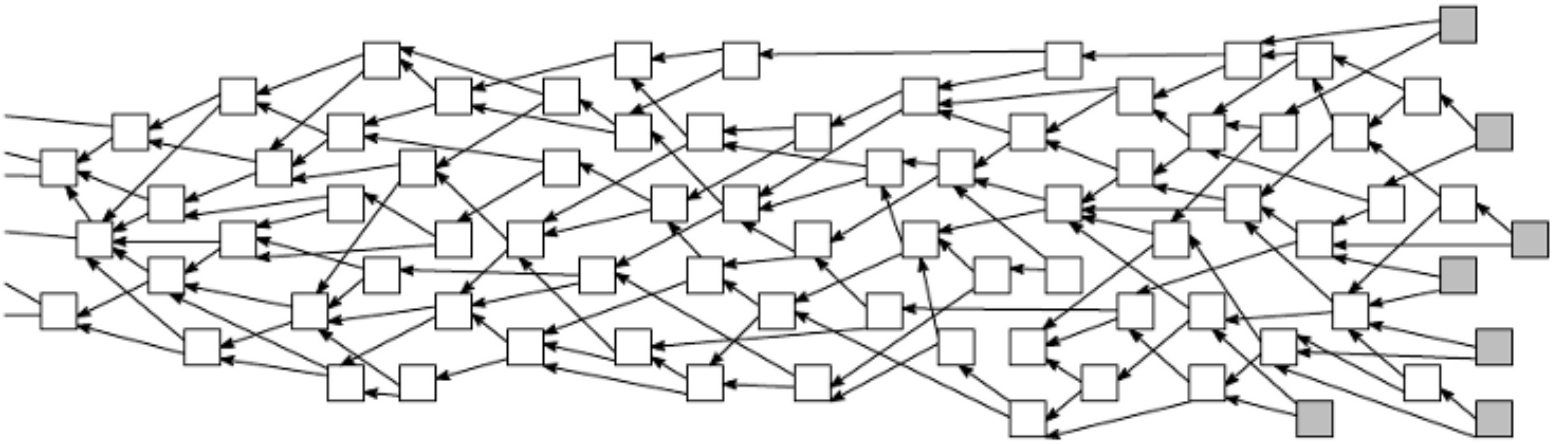
IOTA (IOTA)
- Blockchain to block graph

Designed for IOT devices to perform M2M transactions and data exchange.

Tangle: a DAG-based hash chain

Fast, unlimited scalable, no fee

Other DAG-based cryptocurrency: hashgraph, raiblock, dagcoin, byteball...



Problems of IOTA

Asynchronous, advantage or disadvantage?

Different types of attack

Consensus on DAG can be more easily to diverge

Incentive to run full node?

Coordinator

Monero (XMR) - Privacy-focused Cryptocurrency

Since Bitcoin/Ethereum is transparent, anyone can find out how much you spend and where your payment goes

These kinds of currencies is a better choice for Dark Web payment and money laundering

Often have low TPS

Usually use cryptography techniques like Ring Signature or Zero-knowledge proof

Other privacy-focused cryptocurrency: Zcash, Dash, Verge, PIVX...