NPTEL ONLINE CERTIFICATION COURSES

Blockchain and its applications

Prof. Shamik Sural
Department of Computer Science & Engineering
Indian Institute of Technology Kharagpur

Lecture 06: Basic Cryptographic Primitives - IV

CONCEPTS COVERED

- Basic Concepts of Cryptography
- Public Key Cryptography
- Encryption and Decryption using Public Key Cryptography
- Digital Signature

KEYWORDS

- Public Key Cryptography
- RSA

What we have learnt so far

- **Cryptographically Secure Hash Function**
  - Collision Free
  - Information Hiding
  - Puzzle Friendly

- **Hash Pointers and Data Structures**
  - Hashchain
  - Hash Tree – Merkle Tree

Basic Concepts of Cryptography

- **Symmetric Key Cryptography**
  - Same key used for encryption and decryption
  - How to share the key securely
  - Cannot address certain requirements

- **Public Key Cryptography**
  - One key for encryption, one for decryption
  - Handles several requirements like those in blockchain

Digital Signature

- A **digital code**, which can be included with an electronically transmitted document to verify
  - The content of the document is authenticated
  - The identity of the sender
  - Prevent *non-repudiation* – sender will not be able to deny about the origin of the document

Purpose of Digital Signature

- Only the **signing authority** can sign a document, but everyone can verify the signature

- Signature is **associated with** the particular document
    - Signature of one document cannot be transferred to another document

Public Key Cryptography

- Also known as **asymmetrical cryptography** or **asymmetric key cryptography**

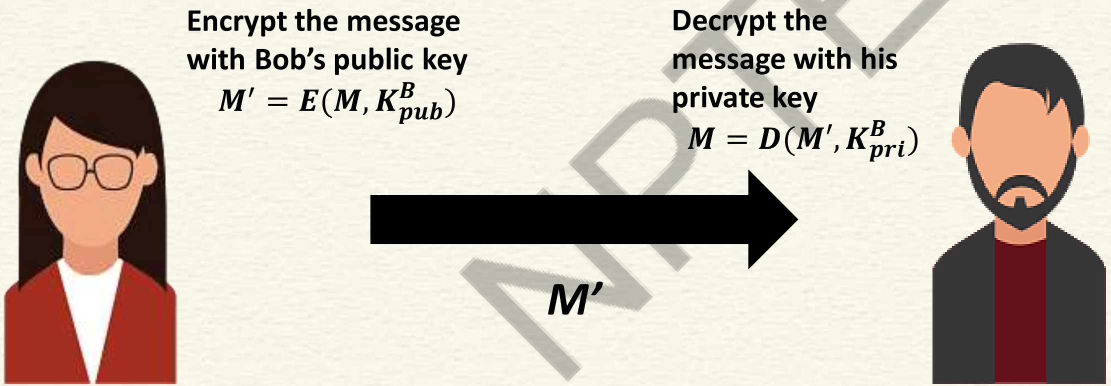
- **Key:** A parameter that determines the functional output of a cryptography algorithm
  - **Encryption:** The key is used to convert a plain-text to a cypher-text; $M' = E(M, k)$
  - **Decryption:** The key is used to convert the cypher-text to the original plain text; $M = D(M', k)$

Public Key Cryptography

- Properties of a cryptographic key (you need to prevent it from being guessed)
    - Generate the key truly randomly so that the attacker cannot guess it
    - The key should be of sufficient length – increasing the length makes the key difficult to guess
    - The key should contain sufficient entropy, all the bits in the key should be equally random

Public Key Cryptography

- Two keys are used
    - **Private key**: Only Alice has her private key
    - **Public key:** “Public” to everyone – everyone knows Alice’s public key



Public Key Encryption - RSA

- Named over (Ron) Rivest – (Adi) Shamir – (Leonard) Adleman – inventors of the public key cryptosystem

- The encryption key is public and decryption key is kept secret (private key)
    - Anyone can encrypt the data
    - Only the intended receiver can decrypt the data

RSA Algorithm

- Four phases
  - Key generation
  - Key distribution
  - Encryption
  - Decryption

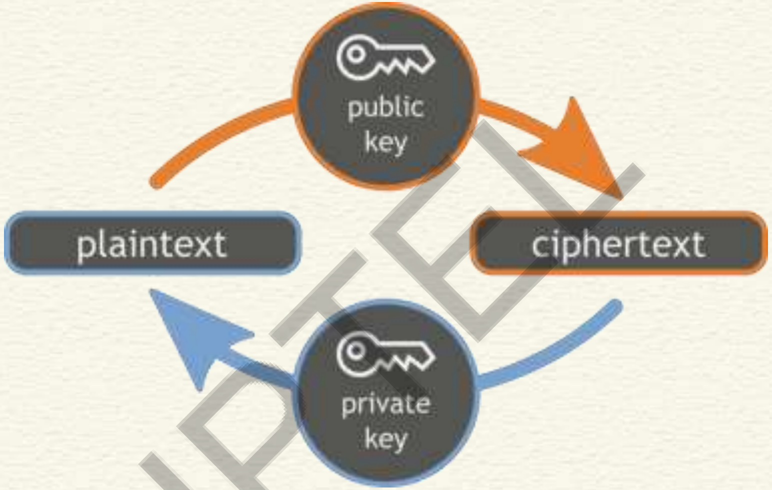


Image source: https://commons.wikimedia.org/

Public and Private Keys in RSA

- It is feasible to find **three very large positive integers** e, d and n; such that *modular exponentiation* for integers m $(0 \leq m < n)$:

$$(m^e)^d \equiv m \ (mod\ n)$$

- Even if you know e, n and m; it is extremely difficult to find d
- Note that

$$(m^e)^d \equiv m \ (mod\ n) = \left(m^d\right)^e \equiv m \ (mod\ n)$$

- (e, n) is used as the public key and (d, n) is used as the private key. m is the message that needs to be encrypted.

RSA Key Generation and Distribution

- Chose two distinct prime integers p and q
  - p and q should be chosen at random to ensure tight security
- Compute $n = pq$; n is used as the modulus, the length of n is called the key length
- Compute $\phi(n) = (p-1)(q-1)$ (*Euler totient function*)
- Choose an integer e such that $1 < e < \phi(n)$ and $\gcd\left(e, \phi(n)\right) = 1$; e and $\phi(n)$ are co-prime
- Determine $d \equiv e^{-1}(mod\ \phi(n))$: d is the *modular multiplicative inverse* of $e(mod\ \phi(n))$

  [Note $d.e \equiv 1(mod\ \phi(n))$]

CONCLUSIONS

- We have discussed the basic concepts of public key cryptography
- How to generate keys in RSA

REFERENCES

- Cryptography and Network Security – Principles and Practice by William Stallings, Pearson (2017)

Thank you

NPTEL ONLINE CERTIFICATION COURSES

Blockchain and its applications

Prof. Shamik Sural
Department of Computer Science & Engineering
Indian Institute of Technology Kharagpur

Lecture 07: Basic Cryptographic Primitives - V

CONCEPTS COVERED

- RSA Encryption and Decryption
- Digital Signature
- Hashing and Digital Signature

KEYWORDS

- RSA
- Digital Signature

RSA Encryption and Decryption

- Let m be the integer representation of a message M.

- **Encryption with public key $(\boldsymbol{e}, \boldsymbol{n})$**
  - $c \equiv m^e \ (mod\ n)$

- **Decryption with private key $(\boldsymbol{d}, \boldsymbol{n})$**
  - $m \equiv c^d \ (mod\ n) \equiv (m^e)^d (mod\ n)$

RSA Encryption and Decryption - Example

Key Selection

- Select 2 prime numbers: $p=17$, $q=11$
- Calculate $n=pq=17\times11=187$
- Calculate $\phi(n)=(p-1)(q-1)=16\times10=160$
- Select e such that e is relatively prime to $\phi(n)=160$ and less than $\phi(n)$; Let $e=7$
- Determine d such that $d.e \equiv 1 \bmod 160$ and $d<160$; Can determine $d = 23$ since $23\times7 = 161 = 1\times160+1$

RSA Encryption and Decryption - Example

Encryption of Plaintext M = 88

- $C=88^7 \bmod 187$
- $= [(88^4 \bmod 187)\times(88^2 \bmod 187)\times(88^1 \bmod 187)] \bmod 187 = (88\times77\times132) \bmod 187 = 11$

Decryption of Ciphertext C = 11

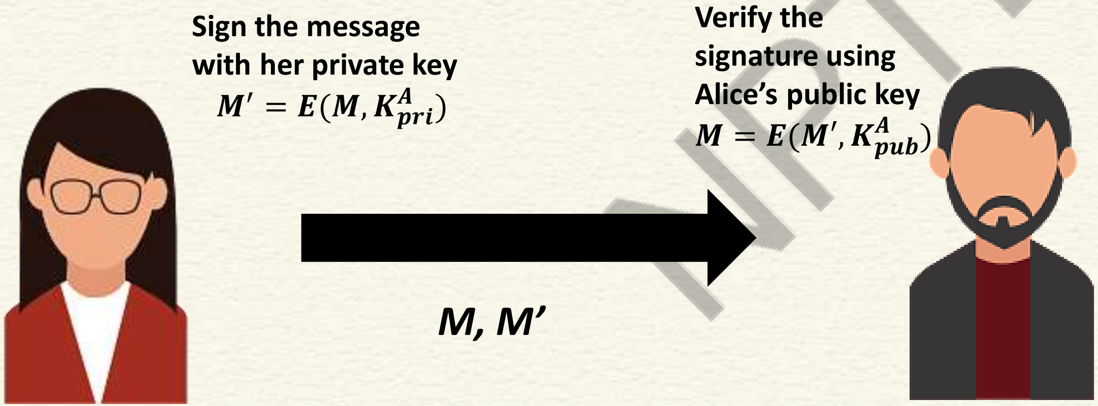
- $M=11^{23} \bmod 187$
- $=[(11^1 \bmod 187)\times(11^2 \bmod 187)\times(11^4 \bmod 187)\times(11^8 \bmod 187)\times(11^8 \bmod 187)] \bmod 187$
- $=(11\times121\times55\times33\times33) \bmod 187 = (79720245) \bmod 187 = 88$

RSA Encryption and Decryption - Illustration

https://www.devglan.com/online-tools/rsa-encryption-decryption

Digital Signature using Public Key Cryptography

- **Sign the message using the Private key**
    - Only Alice can know her private key
- **Verify the signature using the Public key**
    - Everyone has Alice's public key and they can verify the signature



Reduce the Signature Size

- Use the message digest to sign, instead of the original message

Sign the message with her private key

$$S = E(H(M), K^{A}_{pri})$$

Verify the signature using Alice's public key

$$H(M) = E(S, K^{A}_{pub})$$

Digital Signature - Illustration

https://www.devglan.com/online-tools/rsa-encryption-decryption

http://www.blockchain-basics.com/HashFunctions.html

Digital Signature in Blockchain

- Used to validate the origin of a transaction
  - Prevent non-repudiation
    - **Alice cannot deny her own transactions**
    - **No one else can claim Alice's transaction as his/her own transaction**

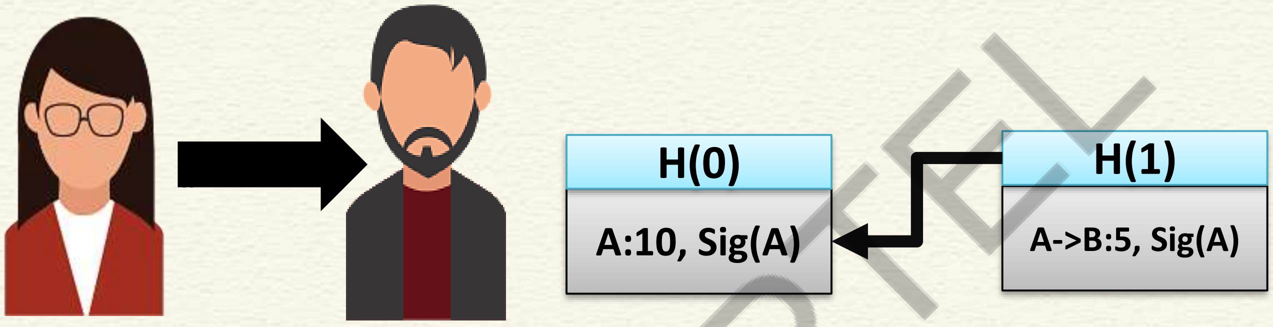
- Bitcoin uses *Elliptic Curve Digital Signature Algorithm* ***(ECDSA)***
  - Based on elliptic curve cryptography
  - Supports good randomness in key generation

A Cryptocurrency using Hashchain and Digital Signatures

A:10, Sig(A)

- Alice generates 10 coins
- Sign the transaction A:10 using Alice's private key and put that in the blockchain

A Cryptocurrency using Hashchain and Digital Signatures



- Alice transfers 5 coins to Bob
- Sign the transaction A-B:5 using Alice's private key and put that in the blockchain

CONCLUSIONS

- We have shown how to encrypt and decrypt using public key cryptography
- Application in digital signature
- Use of digital signature in blockchain

REFERENCES

- Cryptography and Network Security – Principles and Practice by William Stallings, Pearson (2017)
- Blockchain Basics: A Non-Technical Introduction in 25 Steps by Daniel Drescher, Apress (2017)

Thank you

Blockchain and its applications
Prof. Sandip Chakraborty
Department of Computer Science & Engineering

Lecture 08: Distributed Systems for Decentralization – The Beginning

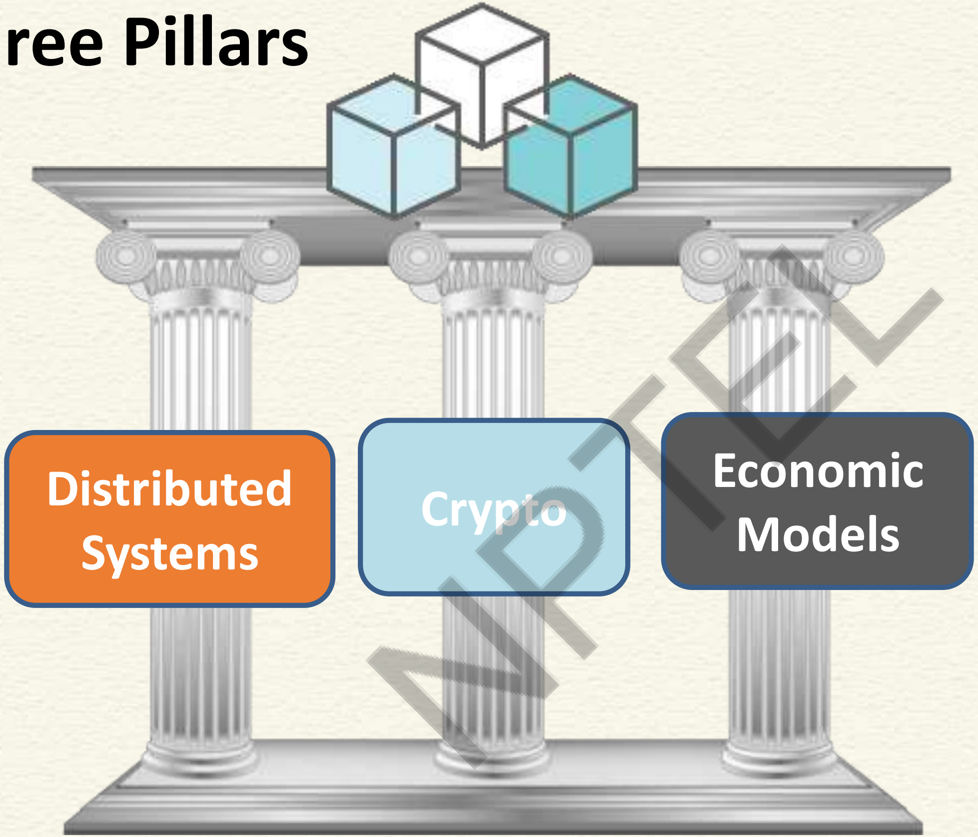
CONCEPTS COVERED

- Distributed Systems
- Blockchain as a Distributed System
- Distributed Consensus – A History

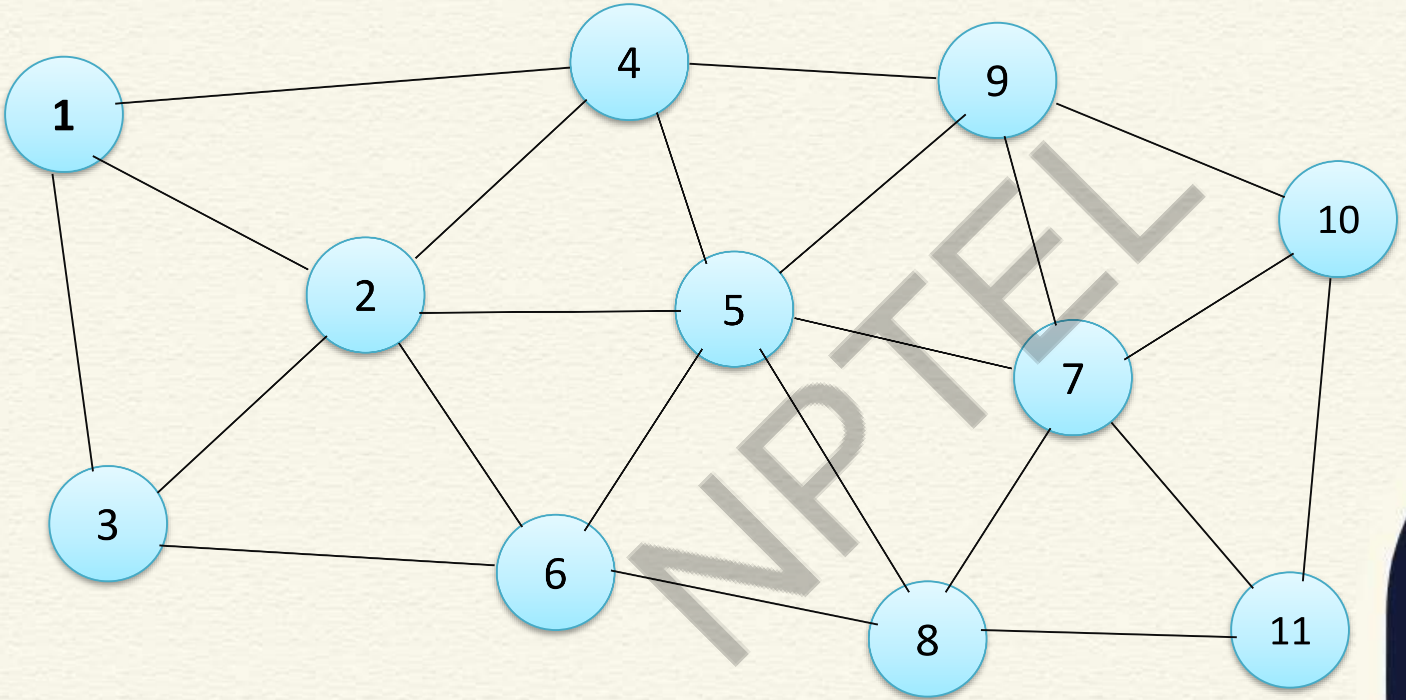
KEYWORDS

- **Distributed System**
- **Consensus**

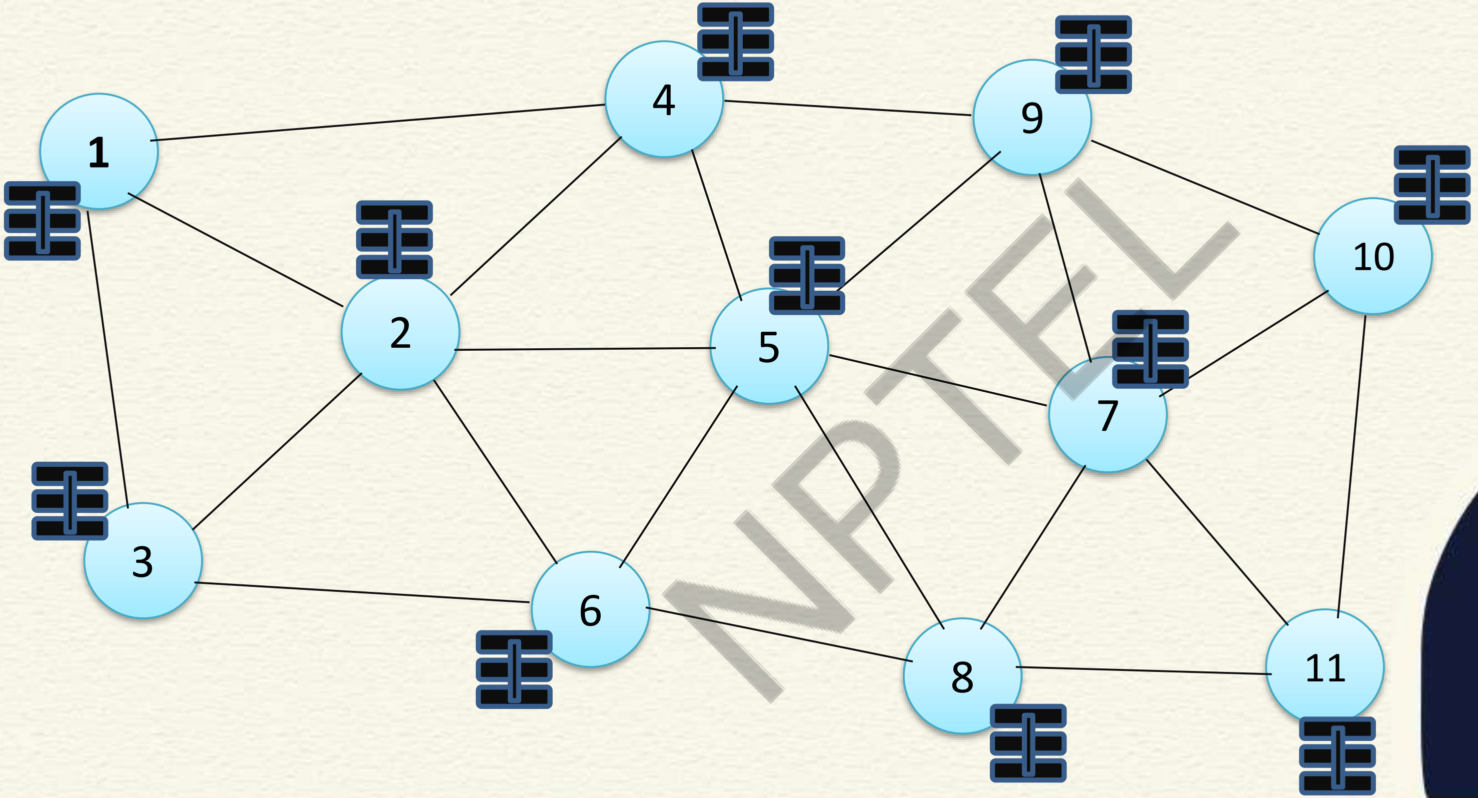
The Three Pillars



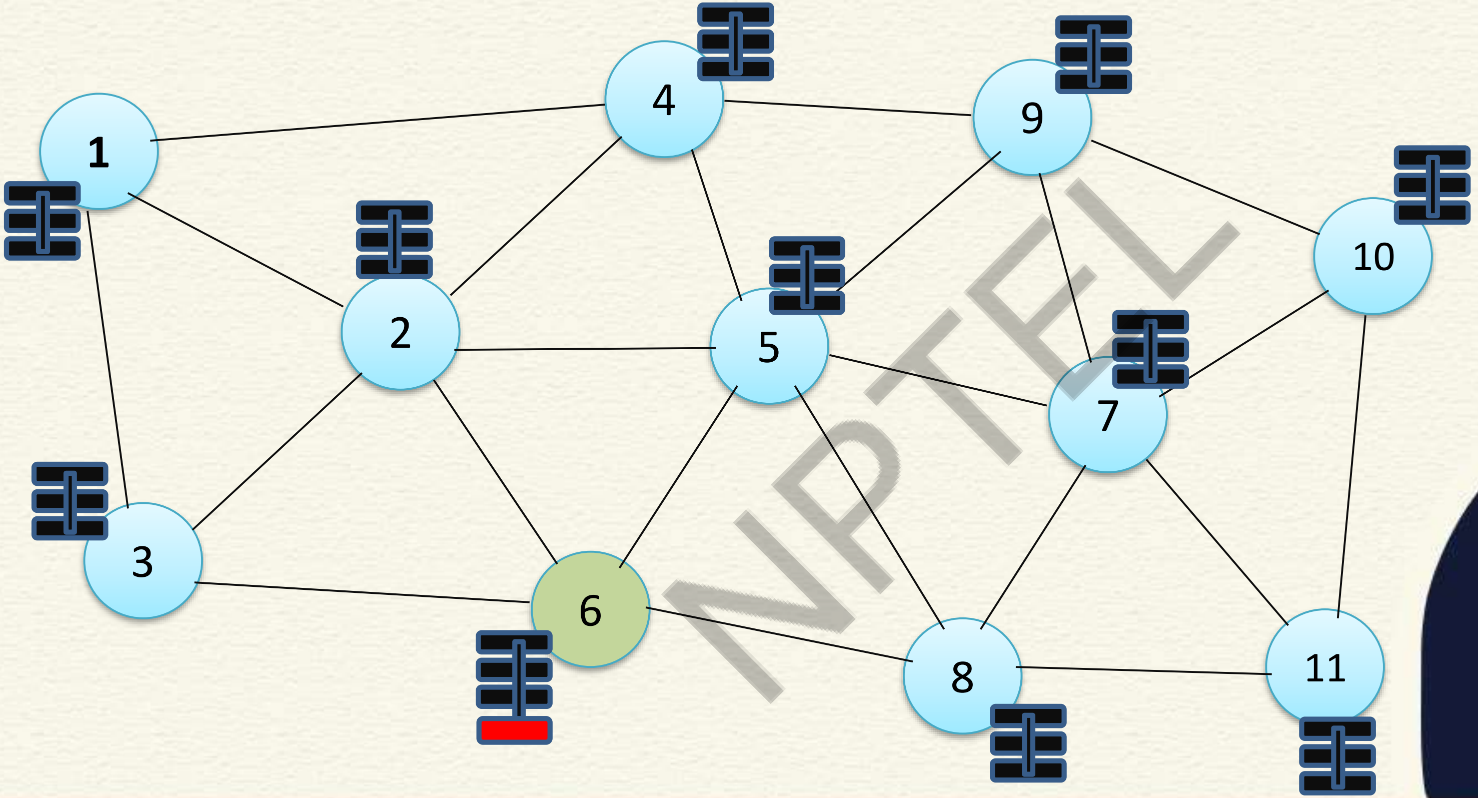
Our Core Problem



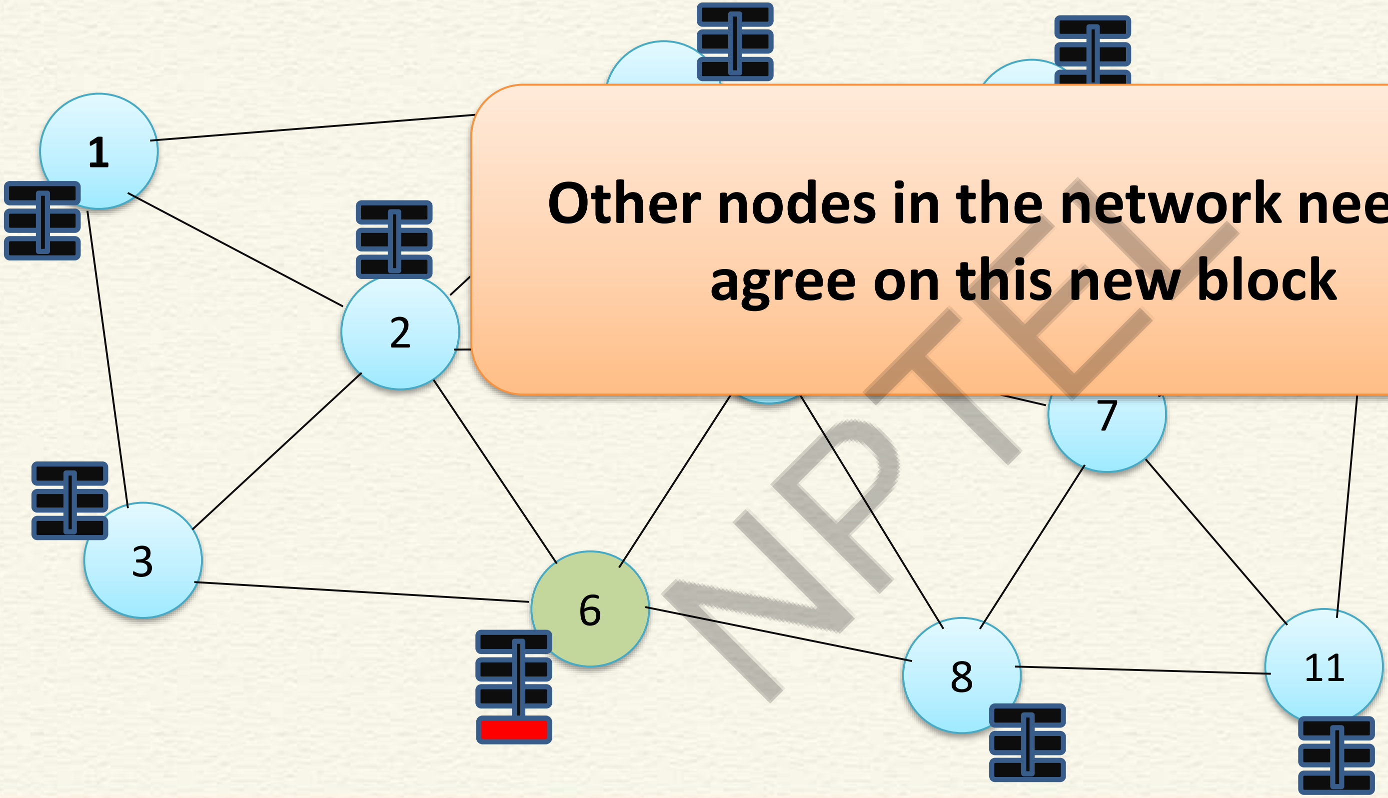
Our Core Problem



Our Core Problem

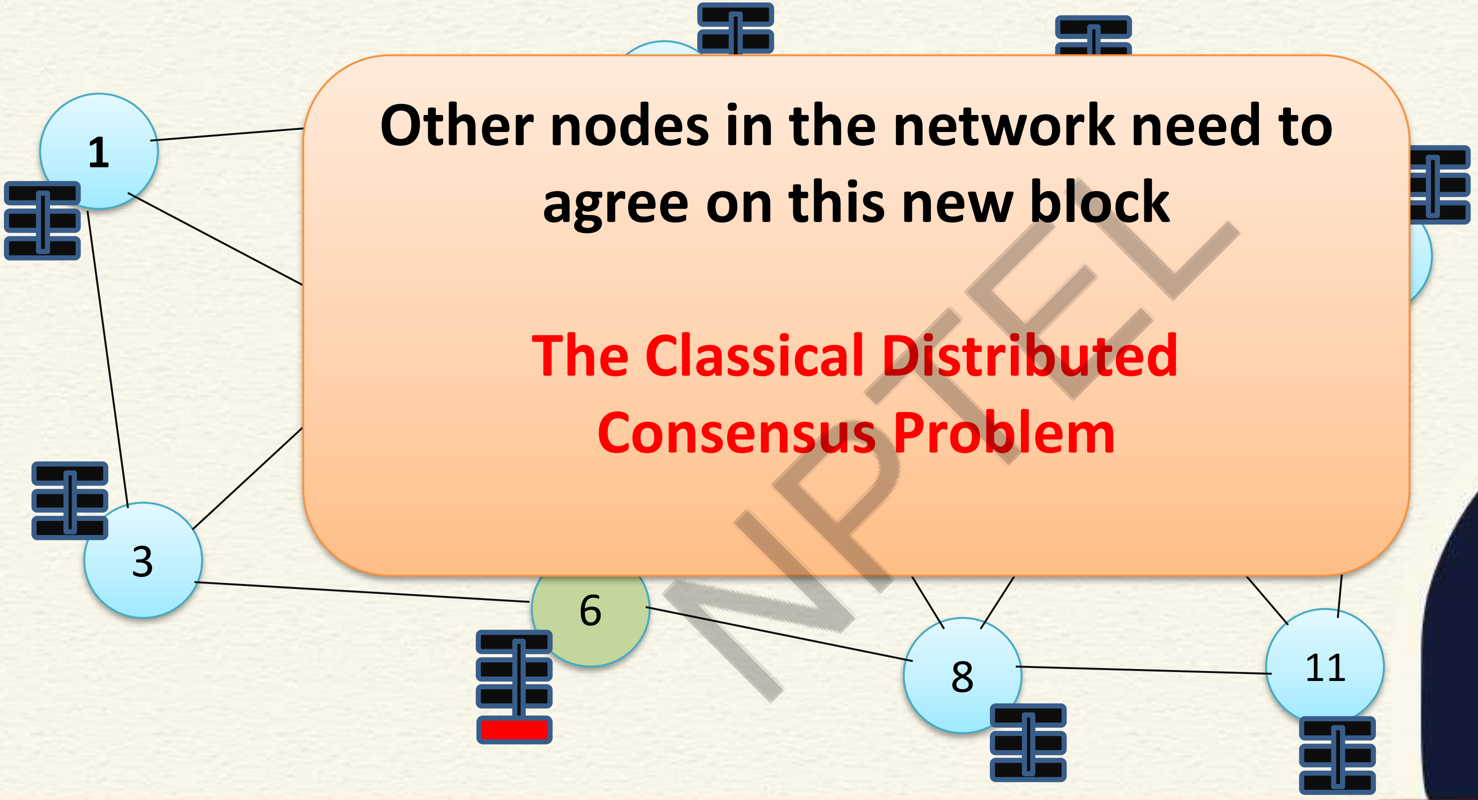


Our Core Problem



Other nodes in the network need to agree on this new block

Our Core Problem



Other nodes in the network need to agree on this new block

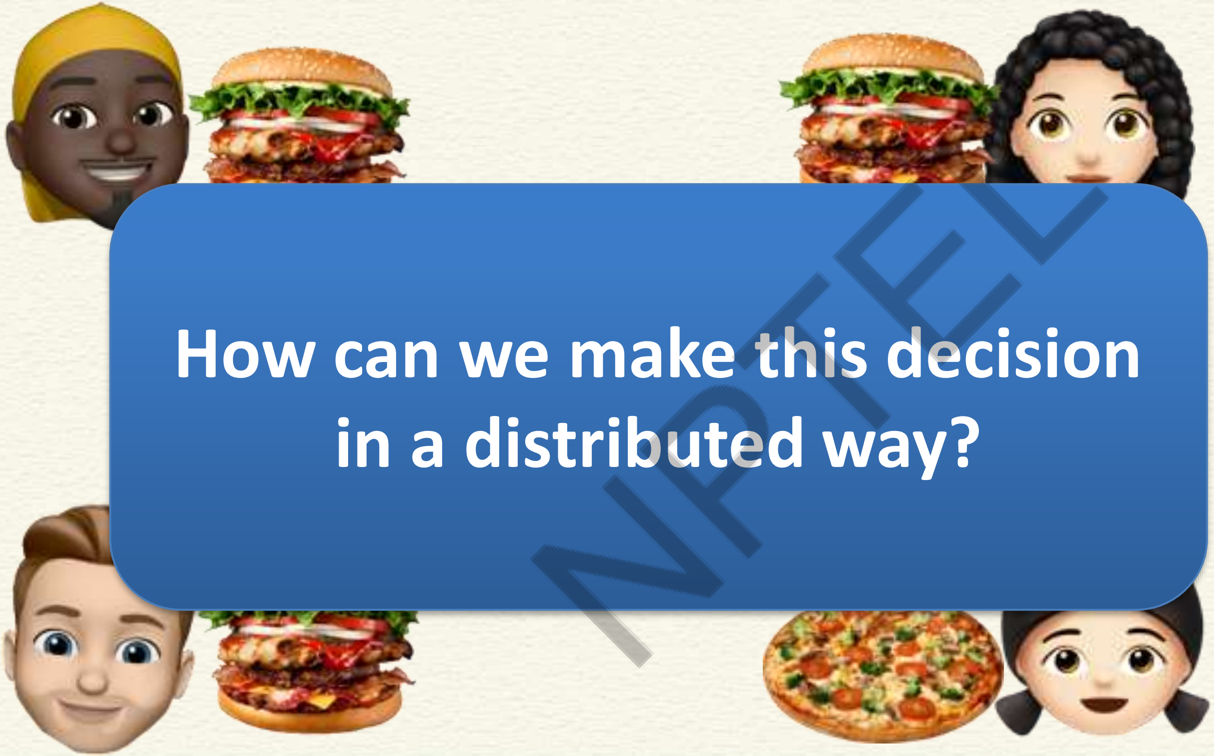
The Classical Distributed Consensus Problem

Distributed Consensus

Distributed Consensus

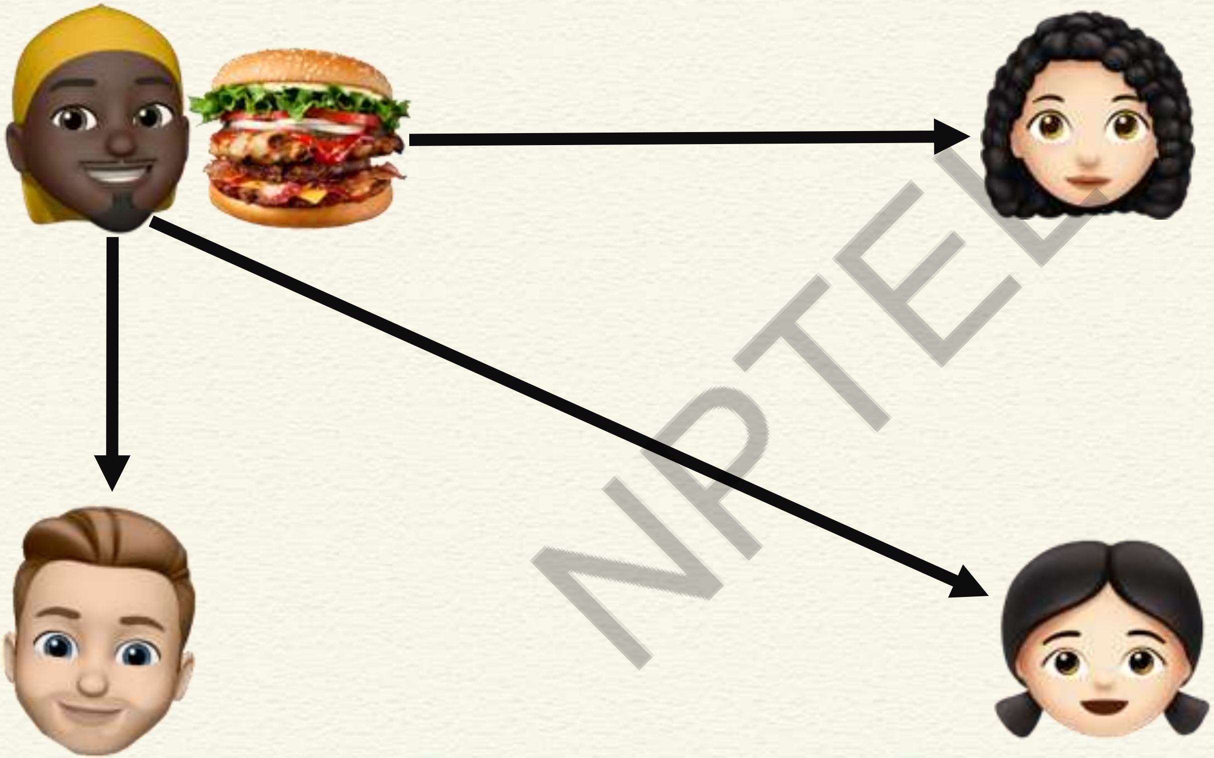
Distributed Consensus

Distributed Consensus

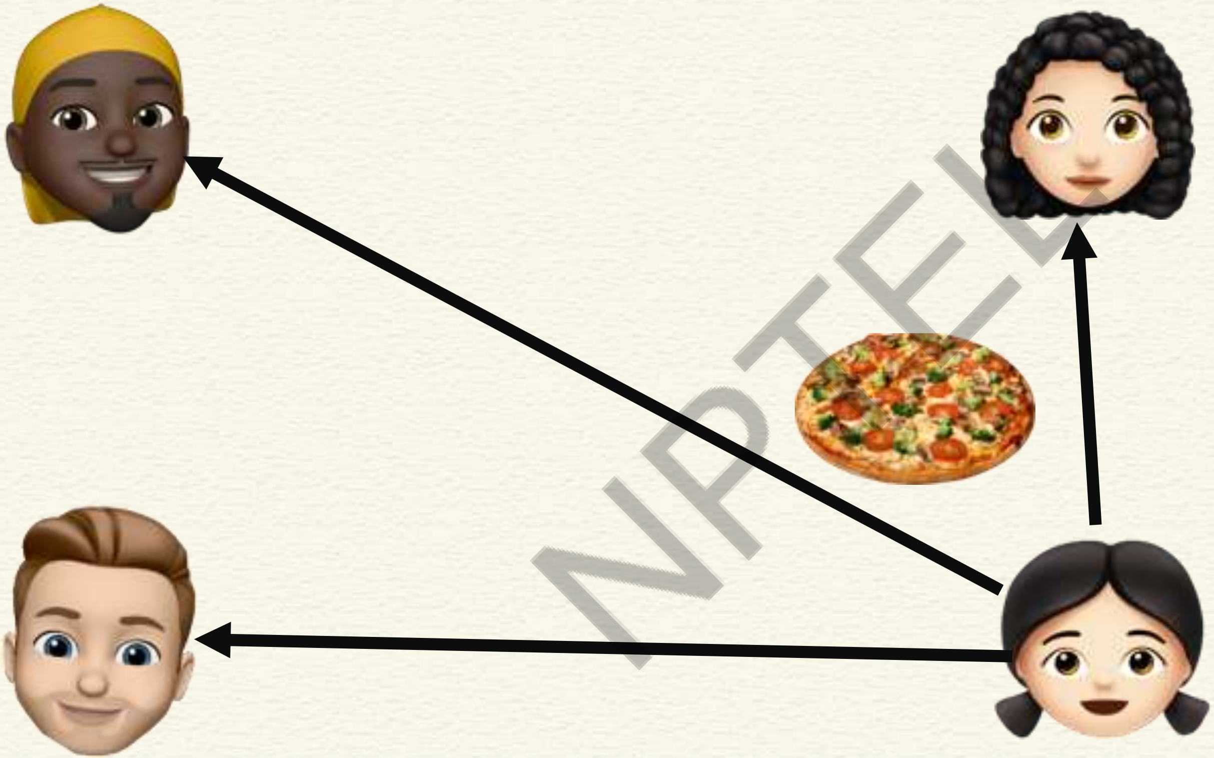


How can we make this decision in a distributed way?

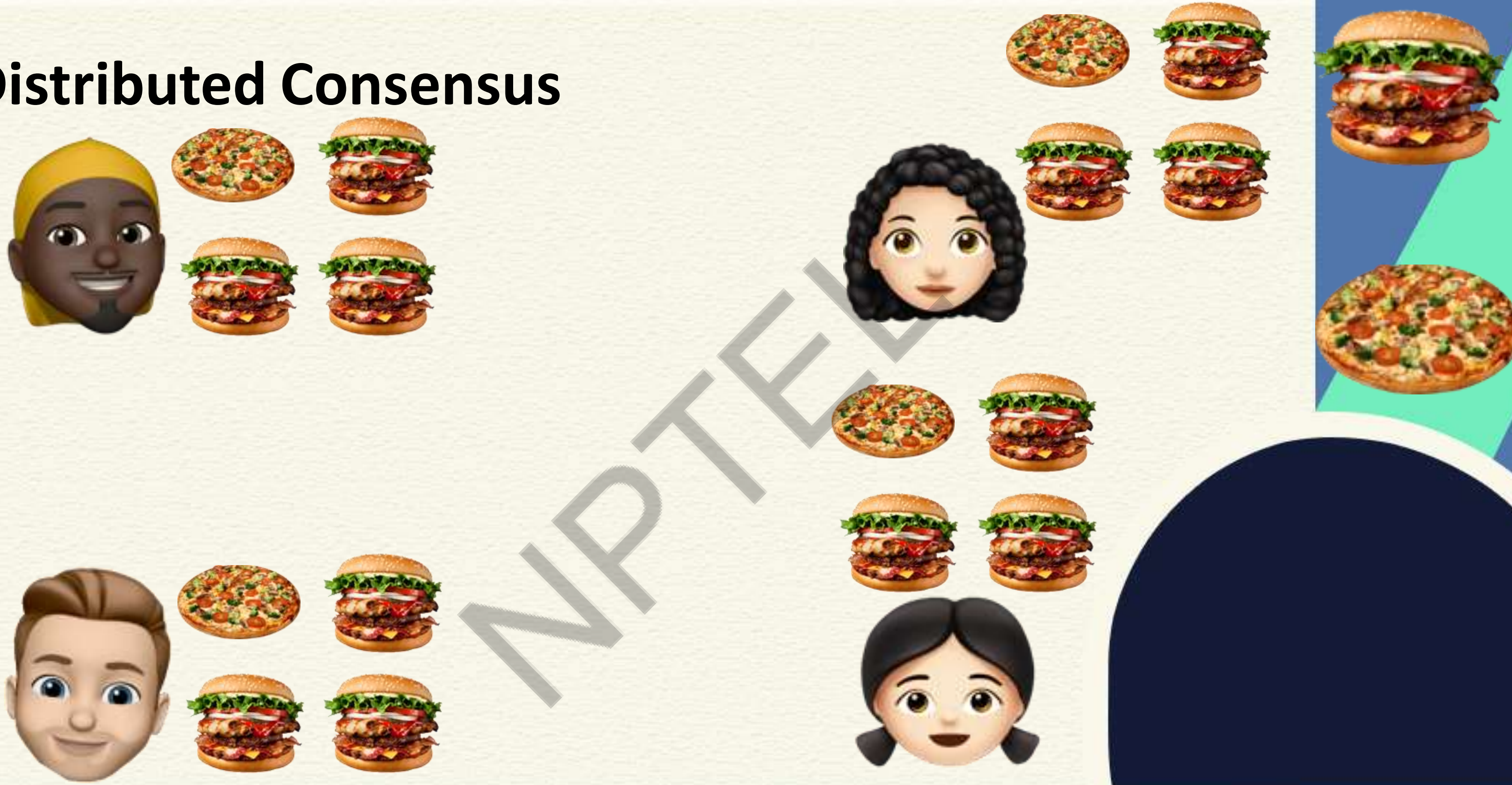
Distributed Consensus



Distributed Consensus



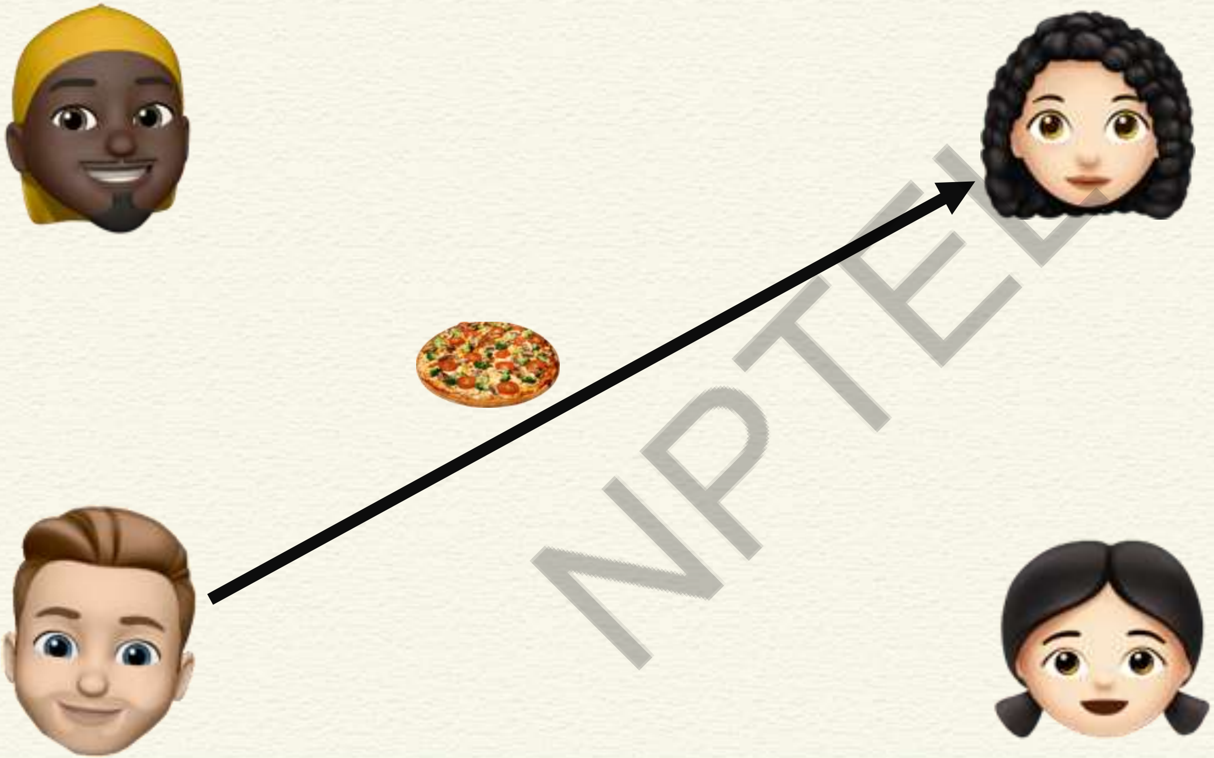
Distributed Consensus



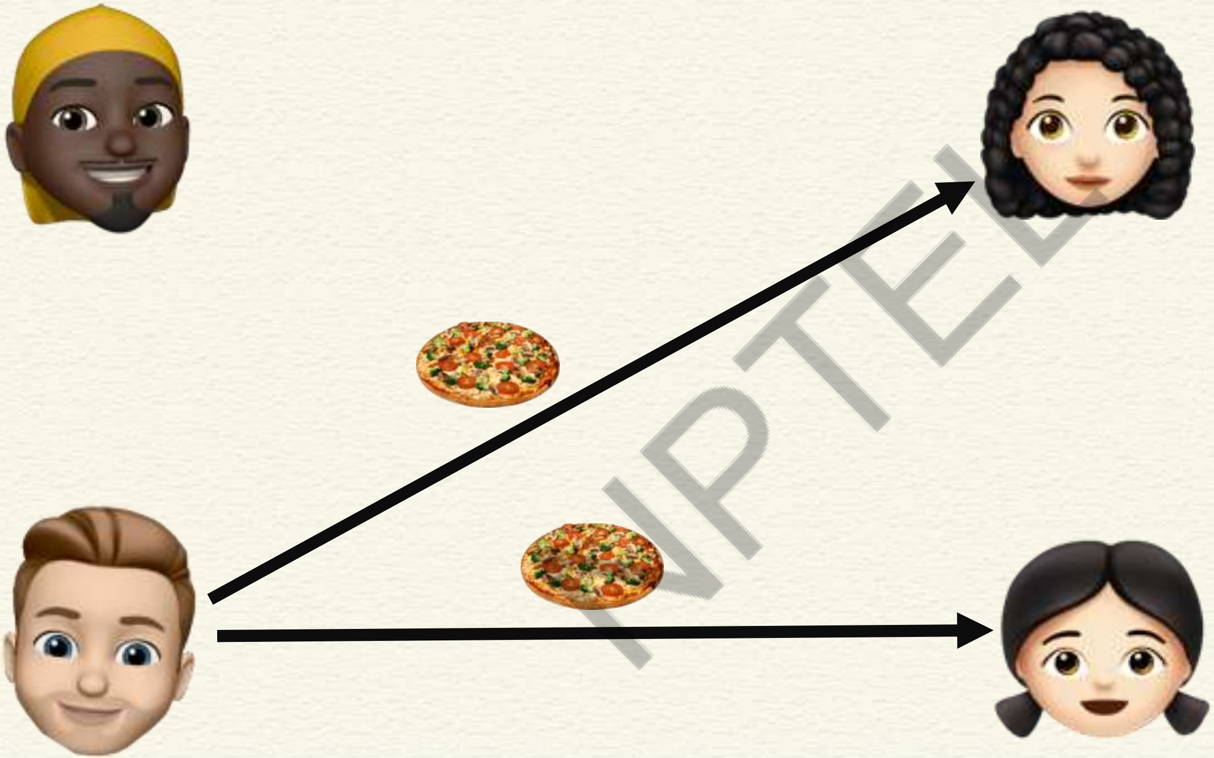
Distributed Consensus

Take a majority voting and decide

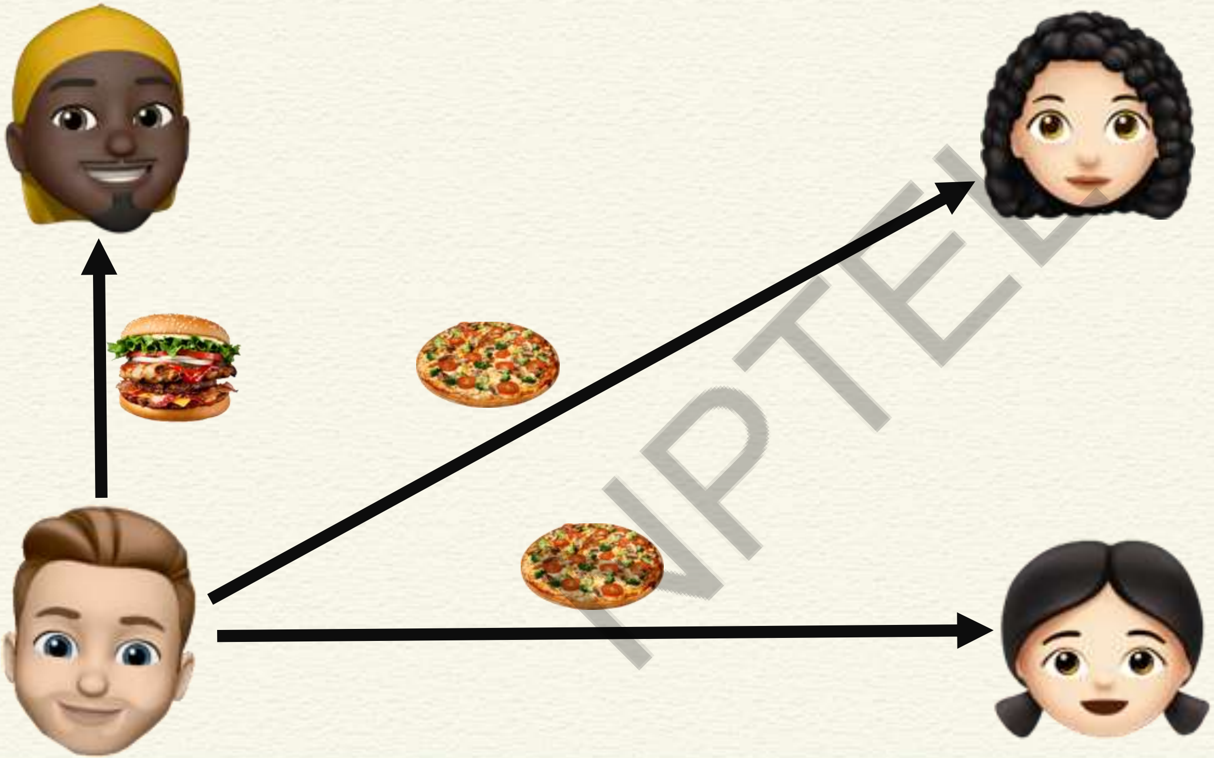
Distributed Consensus



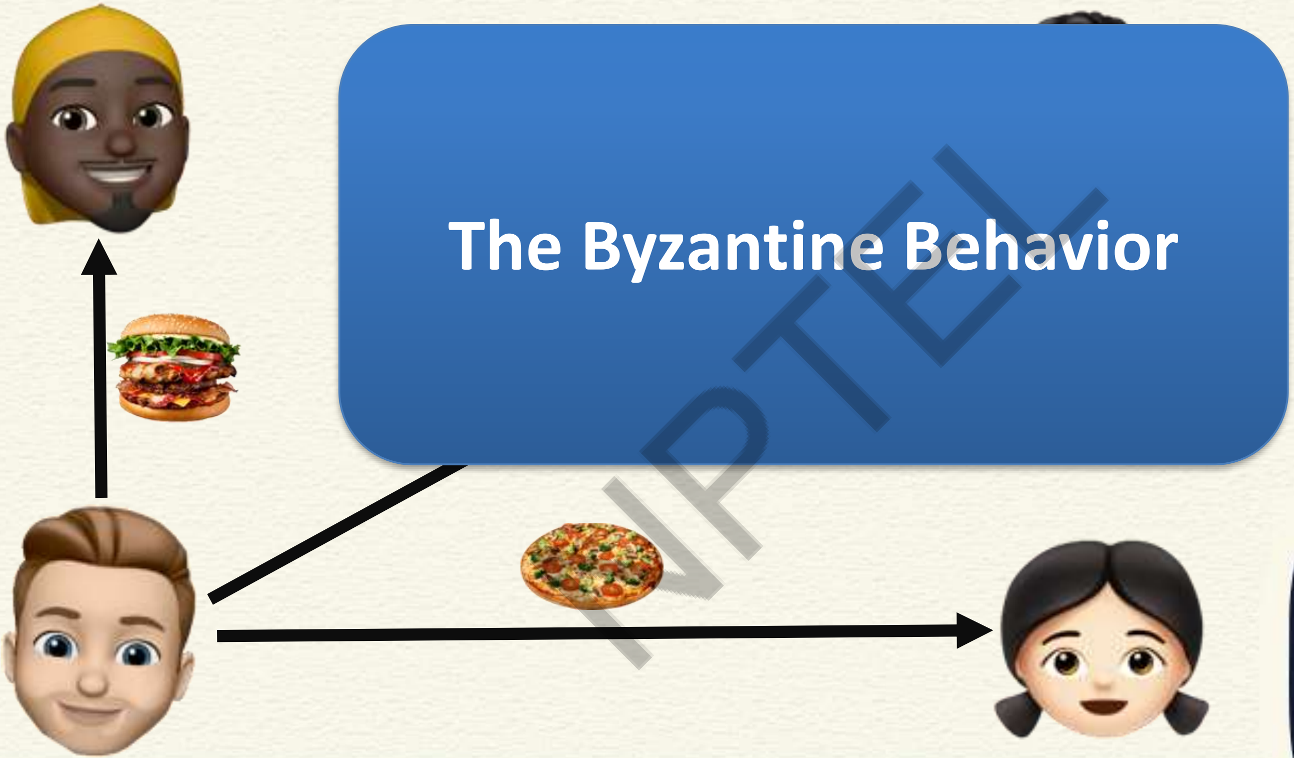
Distributed Consensus



Distributed Consensus



Distributed Consensus



Distributed Consensus – The Literature

- 1985: **FLP Impossibility Theorem** – Fischer, Lynch, Paterson
  - Consensus is impossible in a fully asynchronous system even with a single crash fault

Distributed Consensus – The Literature

- 1985: **FLP Impossibility Theorem** – Fischer, Lynch, Paterson
  - Consensus is impossible in a fully asynchronous system even with a single crash fault
  - Cannot ensure "**<u>Safety</u>**" and "**<u>Liveness</u>**" together

Distributed Consensus – The Literature

- 1985: **FLP Impossibility Theorem** – Fischer, Lynch, Paterson
  - Consensus is impossible in a fully asynchronous system even with a single crash fault
  - Cannot ensure "**<u>Safety</u>**" and "**<u>Liveness</u>**" together

Correct processes will yield the correct output

The output will be produced within a finite amount of time (eventual termination)

Distributed Consensus – The Literature

- 1985: **FLP Impossibility Theorem** – Fischer, Lynch, Paterson
  - Consensus is impossible in a fully asynchronous system even with a single crash fault
  - Cannot ensure "**Safety**" and "**Liveness**" together

- 1989: Lamport started talking about "Paxos"
  - Supports safety but not the liveness

Distributed Consensus – The Literature

- 1985: **FLP Impossibility Theorem** – Fischer, Lynch, Paterson
  - Consensus is impossible in a fully asynchronous system even with a single crash fault
  - Cannot ensure "**Safety**" and "**Liveness**" together
- 1989: Lamport started talking about "Paxos"
  - Supports safety but not the liveness
- 1990's: Everyone were confused about the correctness of Paxos

Distributed Consensus – The Literature

- 1998: Paxos got published in ACM Transactions on Computer Systems
- 2001: FLP Impossibility paper wins Dijkstra Prize
  - People starts talking about Distributed Systems
- 2009: Zookeeper released
  - Service for managing distributed applications

Distributed Consensus – The Literature

- 2010's onward: Different types of consensus algorithms released
  - Multi-Paxos
  - Raft
  - Byzantine Fault Tolerance
  - PBFT
  - ...

Conclusion

- Blockchain needs consensus at its back
- There is a vast literature on distributed consensus
- Can we use them for blockchain?


Thank you

NPTEL ONLINE CERTIFICATION COURSES

Blockchain and its applications

Prof. Sandip Chakraborty
Department of Computer Science & Engineering

Lecture 09: The Evolution of Cryptocurrencies

CONCEPTS COVERED

- Cryptocurrencies – Requirements
- The evolution of cryptocurrencies
- Design Goals for Cryptocurrency Development

KEYWORDS

- **Cryptocurrency**
- **eCash, b-money, bit gold**

Issues with Physical Currencies

Issues with Physical Currencies

Cryptocurrency

- An automated payment system having the properties

  - **Inability** of the third parties to determine payee, time, or the amount of payments made by individuals

  - **Ability to show** the proof of payment

  - **Ability to stop** the use of payment media reported stolen

Digital Money: The Evolution of Cryptocurrencies

- 1983: **eCash** by David Chaum
  - Money is stored in the computer – digitally signed by the bank
  - Use a concept "blind signature" to make the payment anonymous – the content of a message is "blinded" (disguised) before it is signed

Blind Signature


Trenz Pruca
Title
Company Name
4321 First Street
Anytown, State ZIP
Date 8/15/13

Dear Trenz,

Sincerely yours,

Kenneth Bears

Blind Signature

Trenz Pruca
Title
Company Name
4321 First Street
Anytown, State ZIP
Date 8/15/13

Dear Trenz,

Sincerely yours,

Kenneth Beare

- Wants to get your credentials verified
- But do not want to reveal the text of the letter to the person who is verifying the credentials

Blind Signature

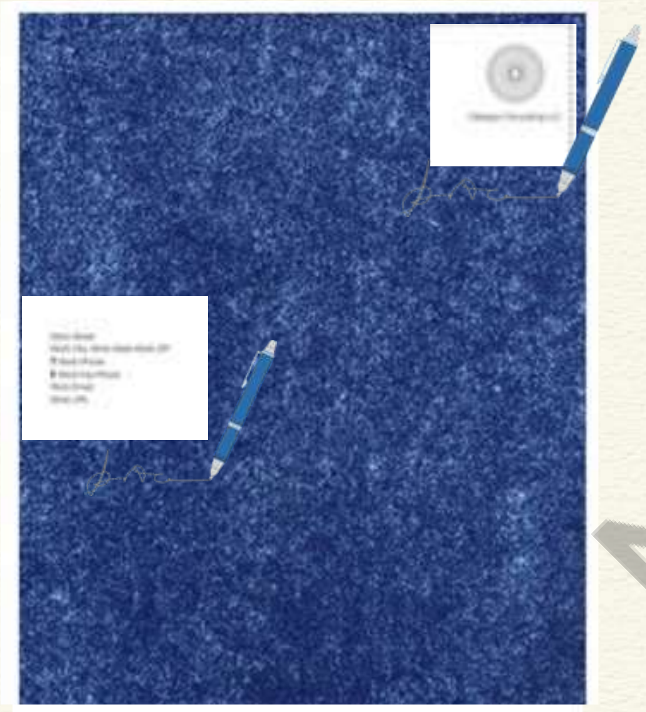
- Wants to get your credentials verified

- But do not want to reveal the text of the letter to the person who is verifying the credentials

Blind Signature

- Wants to get your credentials verified
- But do not want to reveal the text of the letter to the person who is verifying the credentials

Blind Signature



- Wants to get your credentials verified

- But do not want to reveal the text of the letter to the person who is verifying the credentials

Blind Signature

- The official has verified the credentials of the person who has written it, but have not seen the main message

- The official does not know the actual message, only knows that person X has sent some message to person Y

eCash to DigiCash

- 1989: DigiCash Inc. founded by David Chaum
  - ECash could not provide much additional benefit
  - Not very popular among people – currency management overhead is more than bank notes
  - 1998: The company got bankrupted

Morphing the Definition

- An automated payment system having the properties
  - Inability of the third parties to determine payee, time, or the amount of payments made by individuals – **Even the banks will not be able to track it**
  - Ability to show the proof of payment
  - Ability to stop the use of payment media reported stolen

Morphing the Definition

A complete distributed platform for cryptocurrency exchange

e, or

the

- Ability to stop the use of payment media reported stolen

Moving Further ...

- 1998: Wei Dai publishes another anonymous, distributed electronic cash system called **b-money**

- Nick Szabo describes "bit gold"
  - Participants solve a cryptographic puzzle that depends on the previous puzzle
  - Some central control still needs to verify that the puzzle has been solved correctly

Moving Further ...

- 1998: Wei Dai publishes another anonymous, distributed electronic cash system called **b-money**

- Nick Szabo describes "bit gold"
  - Participants solve a cryptographic puzzle that depends on the previous puzzle
  - Some central control still needs to verify that the puzzle has been solved correctly

The Open Question

Can we verify the proof of the puzzle solving in a distributed way?

The Open Question

Can we verify the proof of the puzzle solving in a distributed way?

Distributed Consensus

Majority agrees that the puzzle has been solved correctly


Thank you

NPTEL ONLINE CERTIFICATION COURSES

Blockchain and its applications

Prof. Sandip Chakraborty
Department of Computer Science & Engineering

Lecture 10: Open Consensus and Bitcoin

CONCEPTS COVERED

- Consensus over an Open Network
- Bitcoin – Open Blockchain Network
- The success of Bitcoin as a cryptocurrency

KEYWORDS

- Bitcoin
- Open Consensus
- PoW

The Open Question

Can we verify the proof of the puzzle solving in a distributed way?

Distributed Consensus

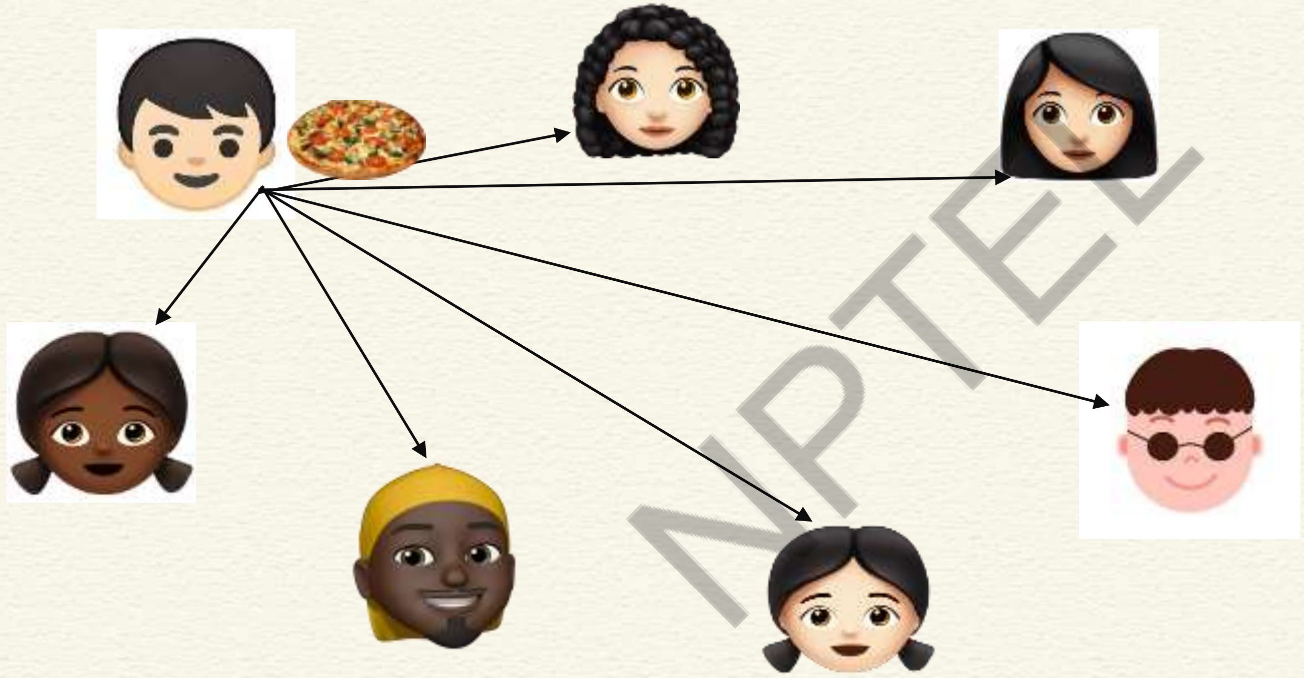
Majority agrees that the puzzle has been solved correctly

The Open Question

Can we verify the proof of the puzzle solving in a distributed way?

The network is open

Distributed Consensus: The Limitation



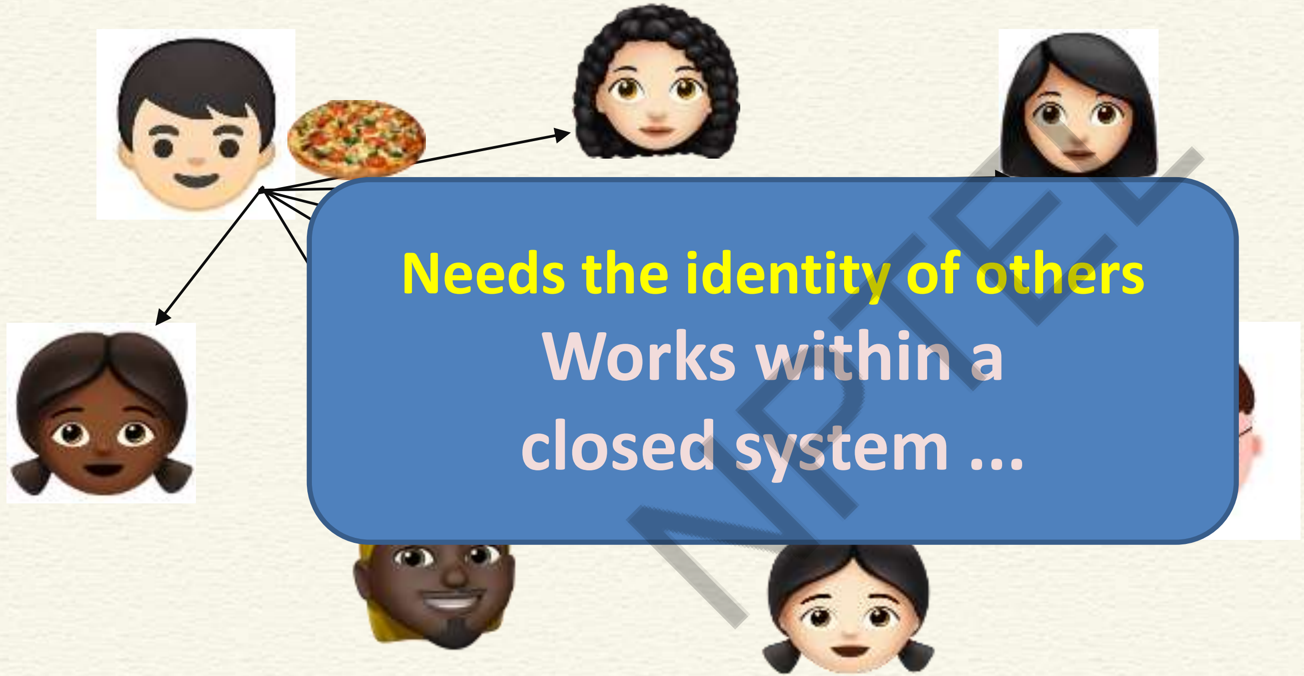
Distributed Consensus: The Limitation

Distributed Consensus: The Limitation

Message Passing

Needs the identity of others

Distributed Consensus: The Limitation



Needs the identity of others

Works within a closed system …

Bitcoin Proof of Work: An Open Consensus

- 2008: A whitepaper got floated on the Internet

Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

Consensus in an Open Network: Puzzle Solving

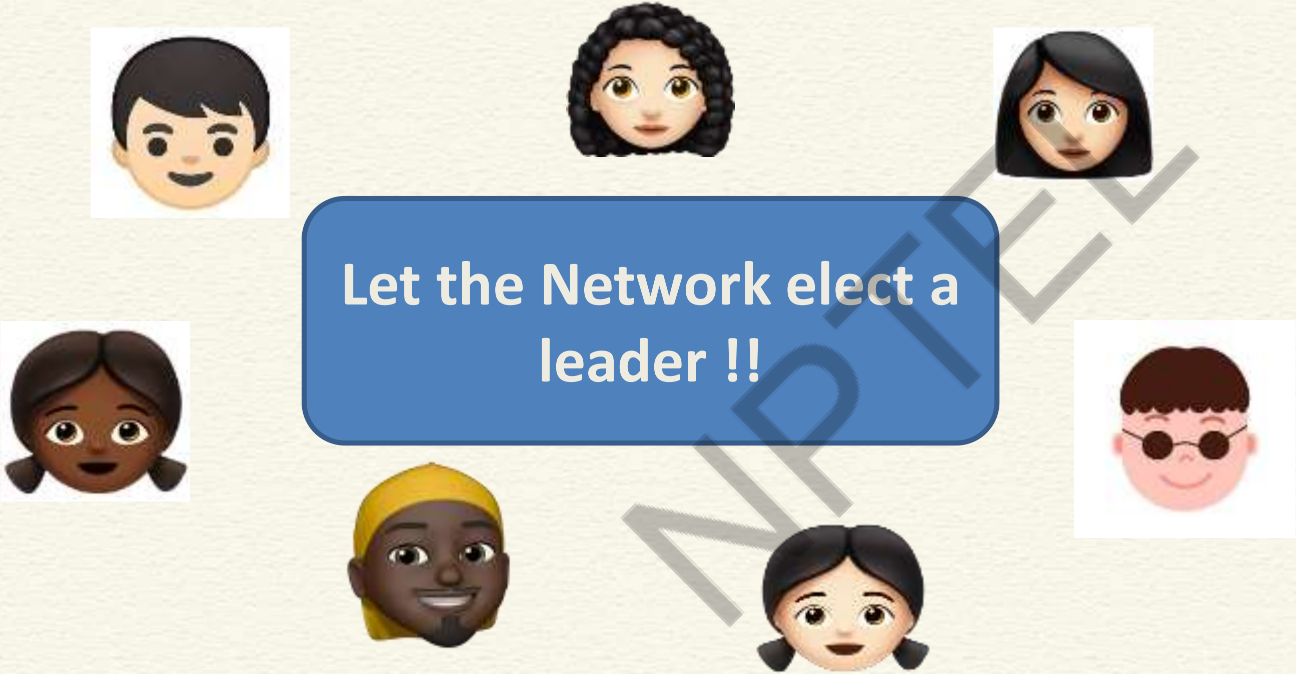


We need a leader

But nobody knows each other!

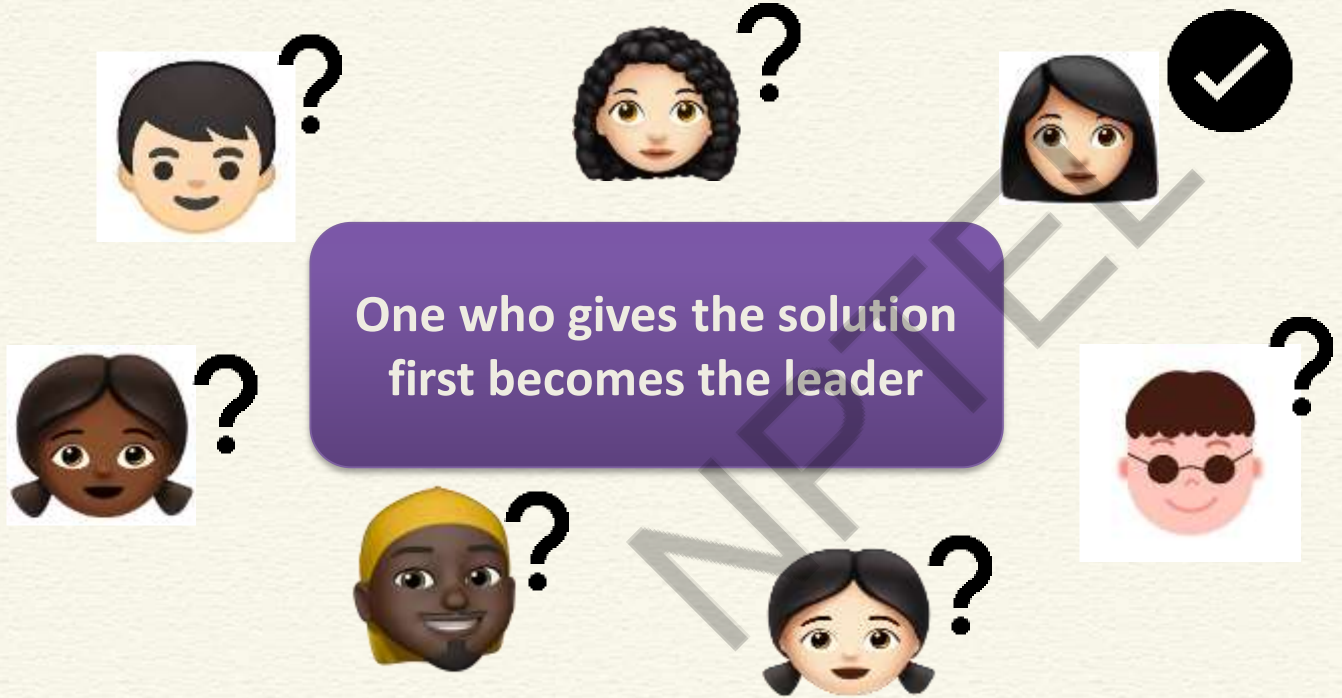



Consensus in an Open Network: Puzzle Solving

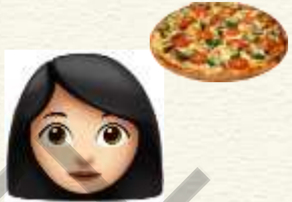


Consensus in an Open Network: Puzzle Solving

Consensus in an Open Network: Puzzle Solving



Consensus in an Open Network: Puzzle Solving



Whatever the leader says, everyone agrees to that

Consensus in an Open Network: Puzzle Solving

Consensus in an Open Network: Puzzle Solving

- Need a good puzzle
  - Difficult to solve
  - Easy to verify

Consensus in an Open Network: Puzzle Solving

- Need a good puzzle
  - Difficult to solve
  - Easy to verify

- **Y = H (X || N)**, Given X and Y, find out N

Bitcoin Proof of Work: An Open Consensus

- 2008: A whitepaper got floated on the Internet
  - Hash Chain + **Puzzle Solving as a Proof** (from Bit Gold) + Coin Mining in an open P2P setup

Bitcoin Proof of Work: An Open Consensus

- 2008: A whitepaper got floated on the Internet
  - Hash Chain + Puzzle Solving as a Proof (from Bit Gold) + Coin Mining in an open P2P setup
  - **Proof of Work** (PoW) -- Nakamoto Consensus

Bitcoin Proof of Work: An Open Consensus

- 2008: A whitepaper got floated on the Internet
  - Hash Chain + Puzzle Solving as a Proof (from Bit Gold) + Coin Mining in an open P2P setup
  - **Proof of Work** (PoW) -- Nakamoto Consensus

The Key to Success:

Give more emphasis on "Liveness" rather than "Safety"

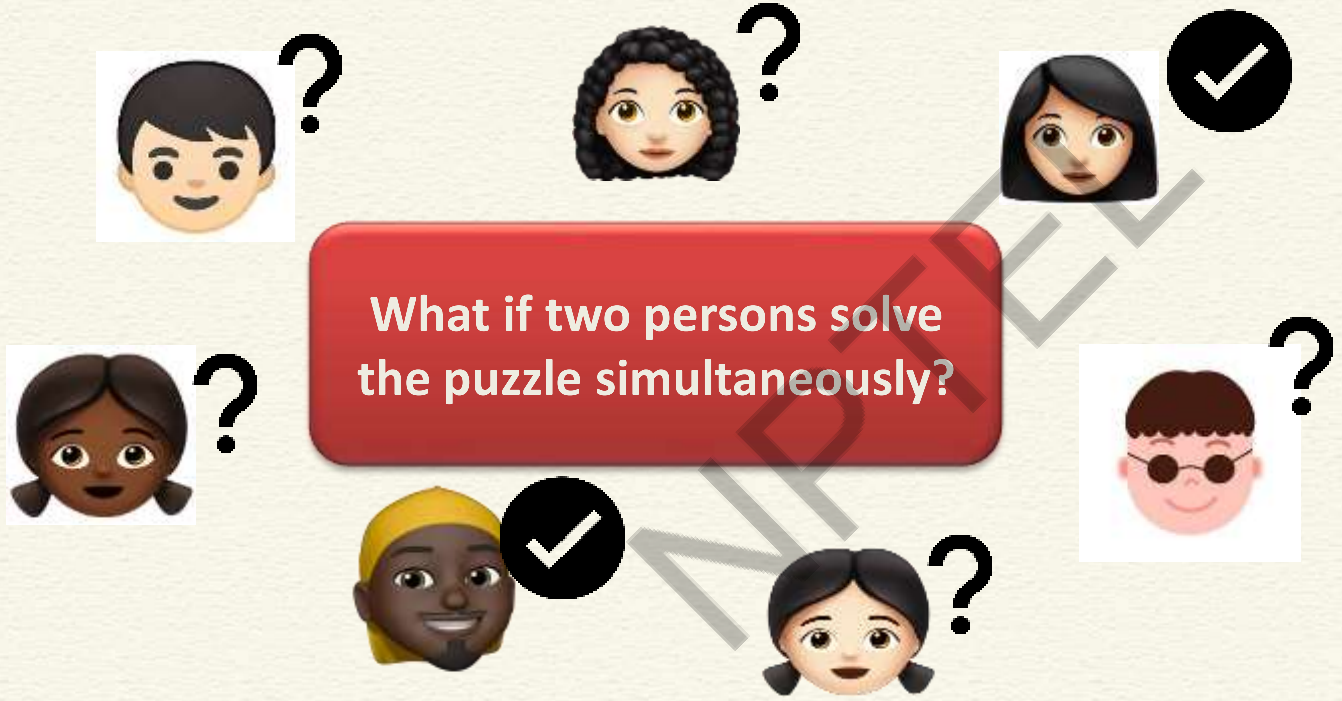
Bitcoin Proof of Work: An Open Consensus

- 2008: A whitepaper got floated on the Internet
  - Hash Chain + Puzzle Solving as a Proof (from Bit Gold) + Coin Mining in an open P2P setup
  - **Proof of Work** (PoW) -- Nakamoto Consensus

Give more emphasis on "Liveness" rather than "Safety"

Participants may agree on a transaction that is not the final one in the chain

Consensus Finality over an Open Network



Consensus Finality over an Open Network

Bitcoin Proof of Work: An Open Consensus

- 2008: A whitepaper got floated on the Internet
  - Hash Chain + Puzzle Solving as a Proof (from Bit Gold) + Coin Mining in an open P2P setup
  - **Proof of Work** (PoW) -- Nakamoto Consensus
  - **Have not coined the term "Blockchain" in the paper !!**

From Cryptocurrency to Blockchain

- 2011: Litecoin got introduced

- 2015: Ethereum network went live

- Sometime around 2016: Term "Blockchain" got popular

Conclusion

- Classical distributed consensus can't be applied on the blockchain for cryptocurrencies
    - Open network, can't support message passing
- Use puzzle solving to reach open consensus – used on Bitcoin
- But, why should someone solve the puzzle?
    - The **puzzle is hard to solve**, needs computing power


Thank you