Blockchain

DONG Chengzu东承祖

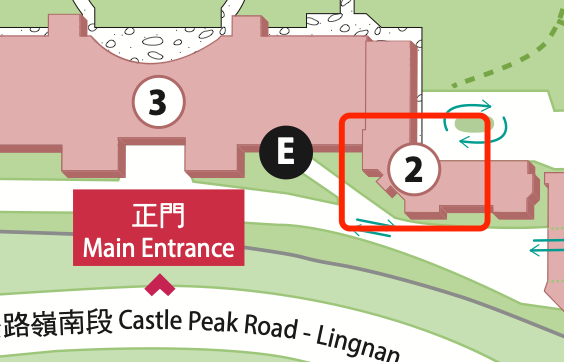
Email: [chengzudong@ln.edu.hk](mailto:chengzudong@ln.edu.hk)

Phone: 26168135

# General

meet him at office time at Wong Administration Building

Office Hour: all day long



## Assessment

attendance and participation 10%

Assignments 20%

Group Project 30%

Presentation 10%

Examination 30%

GAI is NOT permitted in assignment, case study and examination in this course. Files will be examined by Turnitin System to detect plagiarism.

## Required/Essential Readings

<https://github.com/inoutcode/bitcoin_book_2nd>

<https://github.com/inoutcode/ethereum_book>

## metamask

qwertyuiop

0x7cD57590fC0767F6F541F48c64C8237cD652341F

broom lounge front orbit sustain retire problem inject chaos find try nephew

**目录**

[第1章 General - 1 -](#_Toc254)

[1.1 Assessment - 1 -](#_Toc10411)

[1.2 Required/Essential Readings - 1 -](#_Toc28778)

[1.3 metamask - 1 -](#_Toc19912)

[第2章 Week 1 Lecture 1 - 5 -](#_Toc1296)

[2.1 what is blockchain? - 5 -](#_Toc27006)

[2.2 main features of blockchain - 5 -](#_Toc29715)

[2.3 who established bitcoin - 5 -](#_Toc26792)

[2.4 bitcoin history - 5 -](#_Toc23067)

[2.5 cryptocurrency transaction speed - 5 -](#_Toc13218)

[2.6 foundation of bitcoin - 5 -](#_Toc2845)

[2.7 technical challenges - 6 -](#_Toc20330)

[2.8 Solidity - 6 -](#_Toc32123)

[2.9 Token Standards - 7 -](#_Toc12493)

[2.10 altcoin - 7 -](#_Toc4853)

[2.11 Cryptographic Hash Function - 7 -](#_Toc6671)

[2.12 Hash Pointer - 7 -](#_Toc22079)

[2.13 blockchain - 7 -](#_Toc24229)

[2.14 block - 7 -](#_Toc14911)

[2.15 Merkle Tree - 8 -](#_Toc1536)

[2.16 Merkle Proof - 8 -](#_Toc4155)

[2.17 Digital Signature - 9 -](#_Toc19840)

[2.18 POW Prof of Work - 9 -](#_Toc21635)

[2.19 Longest chain protocol - 10 -](#_Toc19190)

[2.20 DAO Attack(Decentralized Autonomous Organization) - 10 -](#_Toc11609)

[2.21 Bitcoin difficulty - 10 -](#_Toc15470)

[2.22 Bitcoin difficulty formula - 10 -](#_Toc5667)

[2.23 Reward Halving - 10 -](#_Toc14103)

[第3章 Week 2 Lecture 2 - 11 -](#_Toc16688)

[3.1 Networking Requirements - 11 -](#_Toc13945)

[3.2 Types of Network Architecture - 11 -](#_Toc19601)

[3.3 Data Broadcast - 11 -](#_Toc22114)

[3.4 Dandelion protocol - 11 -](#_Toc2776)

[3.5 Probability of Detection - 12 -](#_Toc5419)

[3.6 Protocol level attacks - 12 -](#_Toc26302)

[第4章 Week 3 Lecture 3 - 13 -](#_Toc10572)

[4.1 Chain quality - 13 -](#_Toc24828)

[4.2 Fruit chain - 13 -](#_Toc3153)

[4.3 GHOST(Greedy Heaviest Observed Sub Tree) - 13 -](#_Toc25046)

[4.4 Bitcoin-NG(Next Generation) - 14 -](#_Toc11947)

[4.5 Blockchain Trilemma - 14 -](#_Toc9264)

[4.6 Scaling N2S(Node to shard allocation) - 14 -](#_Toc8959)

[4.7 Uni-consensus - 15 -](#_Toc28202)

[4.8 Multi-consensus - 15 -](#_Toc17620)

[第5章 Week 4 Lecture 4 - 16 -](#_Toc11141)

[5.1 Proof of Stake - 16 -](#_Toc15828)

[5.2 NaS(Nothing at Stake Attack) - 16 -](#_Toc24686)

[5.3 Main concepts in Blockchain - 16 -](#_Toc9728)

[5.4 Data Availability Attack - 16 -](#_Toc24411)

[5.5 Incorrect-coding Attack - 16 -](#_Toc7807)

[5.6 Reentrancy Attack - 16 -](#_Toc17188)

[5.7 Timestamp Dependence Attack - 17 -](#_Toc1013)

[5.8 Front running Attack - 17 -](#_Toc30327)

[5.9 DDoS(Distributed Denial of Service Attack) - 17 -](#_Toc31874)

[5.10 Logic Error Attack - 17 -](#_Toc7256)

[5.11 Insecure Randomness Attack - 17 -](#_Toc1762)

[第6章 Week 5 Lecture 5 - 18 -](#_Toc2390)

[6.1 Write Smart Contract Code together - 18 -](#_Toc6966)

[第7章 Week 6 Lecture 6 - 19 -](#_Toc15246)

[7.1 Sidechain - 19 -](#_Toc13194)

[7.2 Payment channel - 19 -](#_Toc28608)

[7.3 Multisig - 19 -](#_Toc272)

[7.4 Lightening network - 19 -](#_Toc9209)

[7.5 Fraud proof - 19 -](#_Toc31908)

[7.6 Rollup - 19 -](#_Toc7328)

[7.7 Optimistic Rollups: pros and cons - 19 -](#_Toc13346)

[7.8 ZK Rollup - 20 -](#_Toc26349)

[7.9 ZK Rollups: pros and cons - 20 -](#_Toc898)

[7.10 Data compression of an individual transaction - 20 -](#_Toc27666)

[7.11 Comparison - 21 -](#_Toc8165)

[第8章 Week 7 Lecture 7 - 22 -](#_Toc663)

[8.1 Longest Chain Protocol Pros and Cons - 22 -](#_Toc19442)

[8.2 BFT Byzantine Fault Tolerant Protocols - 22 -](#_Toc6078)

[8.3 Streamlet Confirmation Rule - 22 -](#_Toc2101)

[8.4 From Streamlet to Hotstuff - 22 -](#_Toc6380)

[8.5 Clock Synchronization - 22 -](#_Toc26466)

[8.6 Comparison - 22 -](#_Toc17765)

[8.7 Hybrid Consensus - 23 -](#_Toc11677)

[8.8 Finality and Availability - 23 -](#_Toc4831)

[8.9 CAP - 23 -](#_Toc23367)

[8.10 Checkpoints - 24 -](#_Toc25529)

[第9章 Week 8 Lecture 8 - 25 -](#_Toc1749)

[9.1 Legislative BFT Protocols - 25 -](#_Toc983)

[9.2 Forensic Support - 25 -](#_Toc18111)

[9.3 HotStuff - 25 -](#_Toc18802)

[9.4 Algorand BA - 25 -](#_Toc13193)

[9.5 UTXO Model - 25 -](#_Toc24928)

[9.6 Trusted Third-party Mixer - 25 -](#_Toc2932)

[9.7 Zerocoin - 25 -](#_Toc19102)

[9.8 Security Requirements - 26 -](#_Toc3164)

[9.9 Privacy Bridge - 26 -](#_Toc25376)

[第10章 Week 9 Lecture 9 - 27 -](#_Toc32196)

[10.1 Interoperability in blockchains - 27 -](#_Toc14392)

[10.2 Wrapped Bitcoin - 27 -](#_Toc29741)

[10.3 More Decentralized Solutions - 27 -](#_Toc7950)

[10.4 Hash TimeLock Contract (HTLC) - 27 -](#_Toc24608)

[10.5 Atomic Swap - 27 -](#_Toc32253)

[10.6 Trustless and Trusted Bridges - 28 -](#_Toc20121)

[10.7 Validator-based Bridges - 28 -](#_Toc13790)

[10.8 Bitcoin Light Client Bridge - 28 -](#_Toc11698)

[10.9 What is DeFi - 28 -](#_Toc21115)

[10.10 DeFi elements are smart contracts - 28 -](#_Toc16321)

[10.11 Nine elements of DeFi - 29 -](#_Toc9151)

[第11章 Week 10 Lecture 10 - 30 -](#_Toc30787)

# Week 1 Lecture 1

## what is blockchain?

Blockchains are decentralized digital trust platforms

## main features of blockchain

* + immutable
  + distributed
  + decentralized

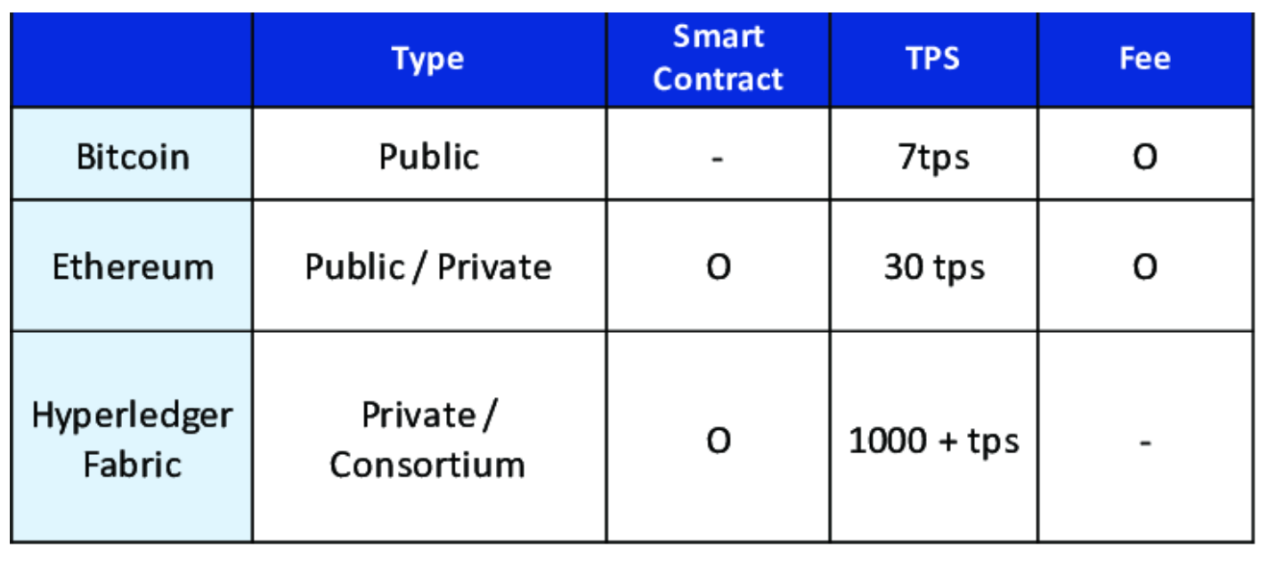
## who established bitcoin

Satoshi Nakamoto 中本聪

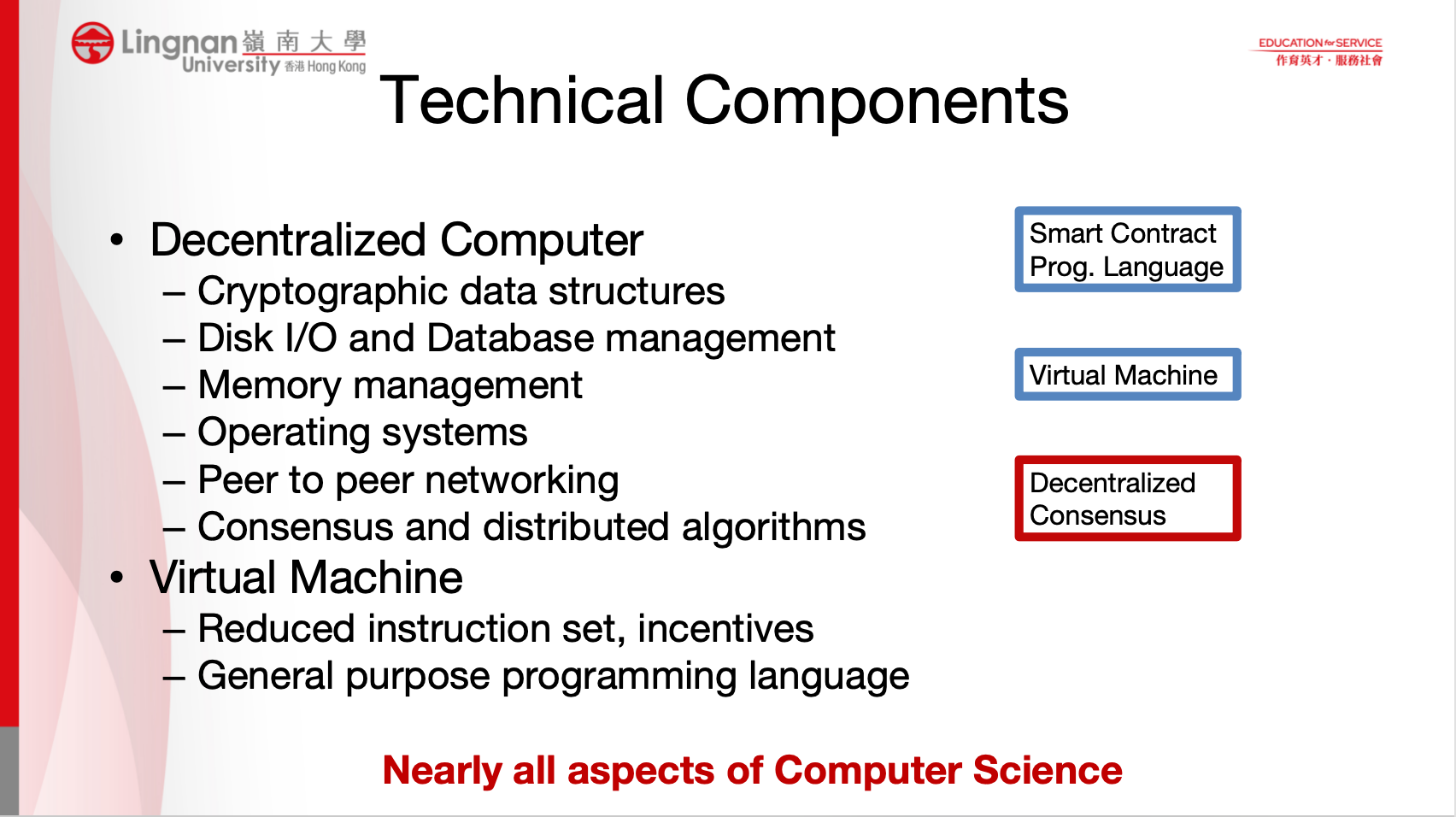
## bitcoin history

Bitcoin is a cryptocurrency used to exchange and store of value which was originated during 2008 Financial Crisis

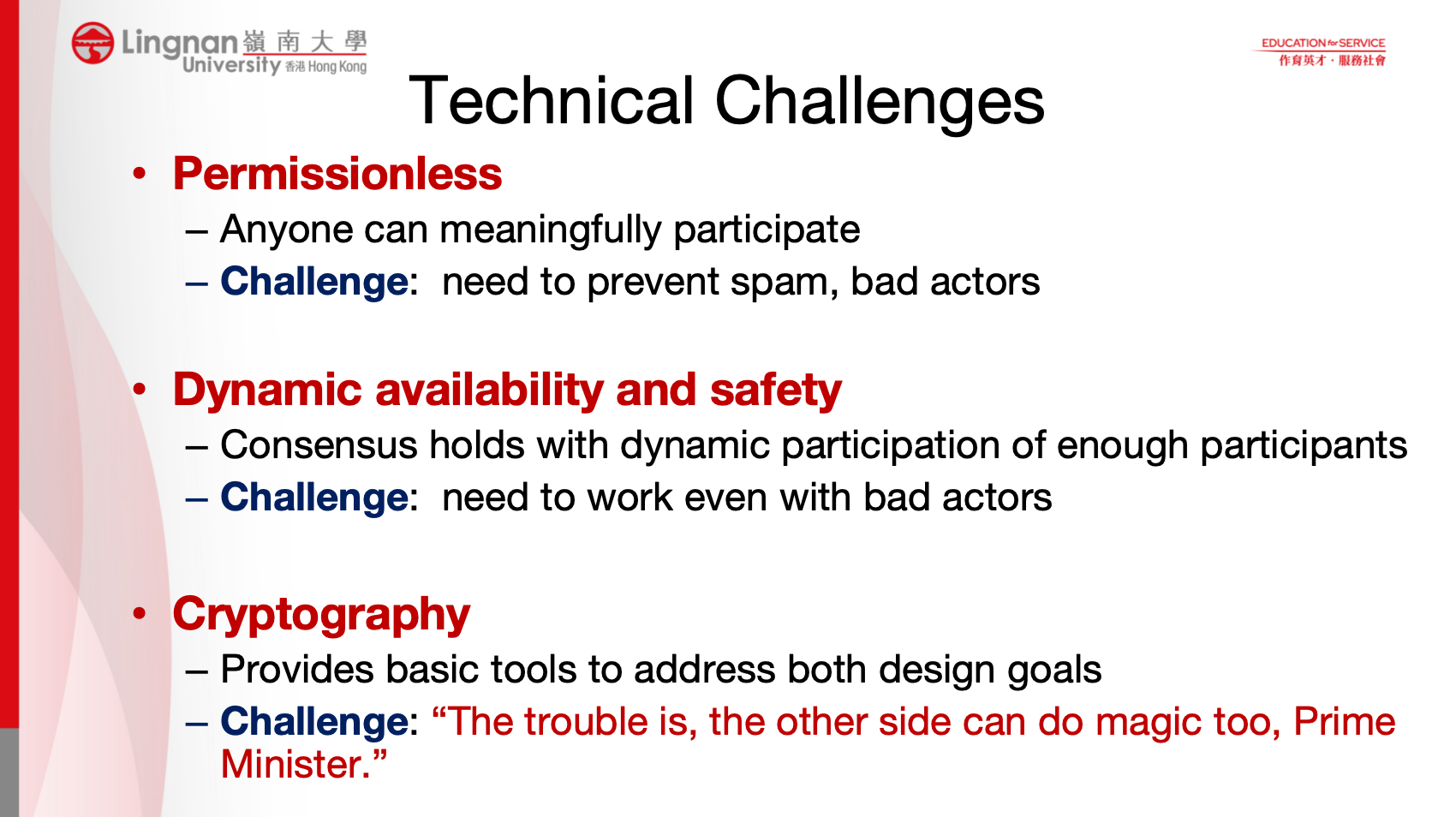
## cryptocurrency transaction speed



## foundation of bitcoin



## technical challenges



## Solidity

Solidity is a high-level programming language designed for implementing smart contracts on the Ethereum blockchain

<https://remix.ethereum.org/>

<https://ethereum.org/en/developers/>

<https://docs.soliditylang.org/en/latest/introduction-to-smart-contracts.html>

## [Token Standards](https://ethereum.org/en/developers/docs/standards/tokens/" \l ":~:text=Many Ethereum development standards focus on token" \t "https://cn.bing.com/_blank)

### ERC-20

each token is the same and has the same value

used in cryptocurrency

### ERC-721

each coin is distinct and cannot be compared

used in NFT(non fungible token)

## altcoin

altcoin is a collection of all alternatives to bitcoin

|  |  |  |
| --- | --- | --- |
| Ethereum (ETH) | Tether (USDT) | BNB (BNB) |
| Solana (SOL) | Ripple (XRP) | USDC (USDC) |
| Cardano (ADA) | Dogecoin (DOGE) |  |

## Cryptographic Hash Function

SHA-256 is used by both bitcoin and altcoin

## Hash Pointer

a special chain data structure used in cryptographic

## blockchain

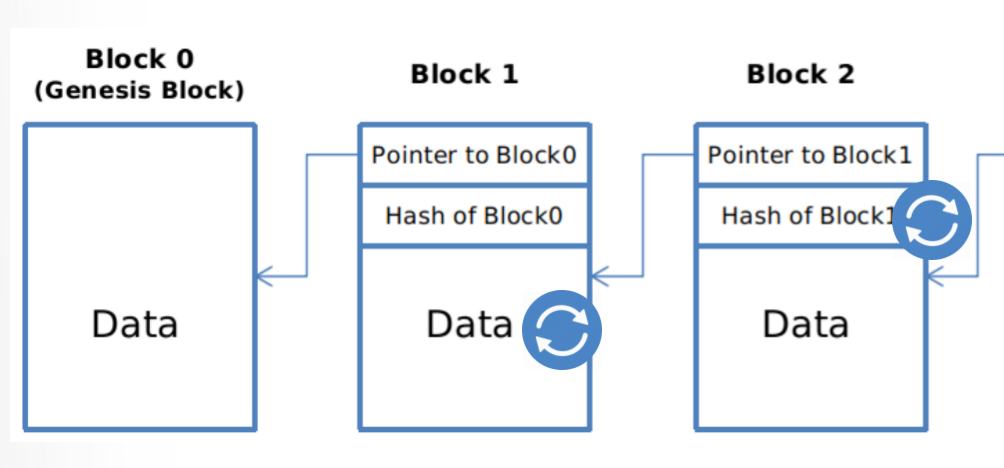
a linked list via hash points

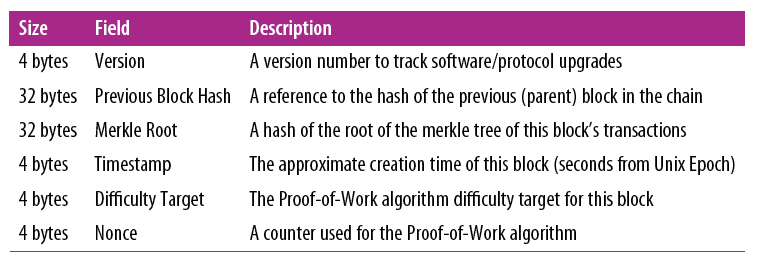
## block

one block consist of Header and Data

Header: Pointer to previous block + hash of the previous block

Data: information specific to the block



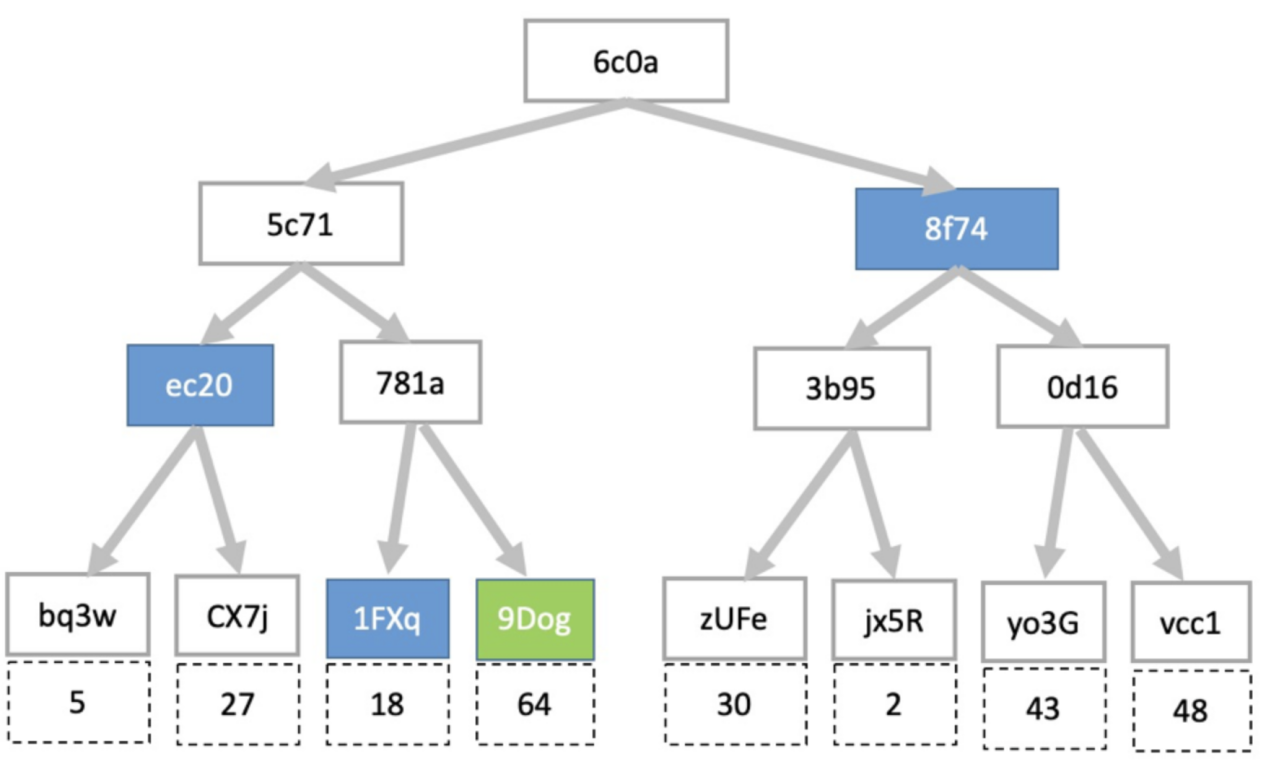


## Merkle Tree

Merkle Tree is constructed from leaf nodes to root and adapting to decentralized network.

## Merkle Proof

We only need hash value of sibling nodes without downloading their data to know whether the data is correct or which part go wrong.



## Digital Signature

<https://ruanyifeng.com/blog/2011/08/what_is_a_digital_signature.html>

Key Generation : get public key and private key

Signature : generate hash code of content and private key

Verification : use public key from certificate to assure that public key is correct and then decode the content

DIY Signature process

## POW Prof of Work

### Mining

cryptographic hash function creates computational puzzle

Hash(nonce, block-hash) < Threshold

nonce is the proof of work

include nonce inside the block

### Threshold

chosen such that a block is mined successfully on average once in 10 minutes

a successfully mined block will be broadcast to all nodes in the network

## Longest chain protocol

Always consider the longest chain as the main branch

## DAO Attack(Decentralized Autonomous Organization)

call withdraw() function multiple times to steal money from ETH market

## Bitcoin difficulty

a measure of how many hashes (statistically) must be generated to find a valid solution to solve the next Bitcoin block and earn the mining reward.

## Bitcoin difficulty formula

difficulty = difficulty\_1\_target / current\_target

next\_difficulty = (previous\_difficulty \* 2016 \* 10 minutes) / (time to mine last 2016 blocks)

¼ < next\_difficulty/previous\_difficulty < 4

the difficulty changes every 2 weeks(2016\*10min=14day) to assure that mining one block consume at about 10 minutes and it changed mildly

Mining difficulty of target Hash number 0x1b 0404cb is :

0x00000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF /0x00000000000404CB000000000000000000000000000000000000000000000000 = 16307.669773817162 (pdiff)

## Reward Halving

Bitcoin halving mechanism reduces the block reward by 50% every four years(210,000 blocks) and has a significant ramification. It introduces built-in scarcity by gradually lowering the pace at which new Bitcoin enters circulation.

Number of halving = Current block height / Reward halving interval

Effective mining reward = Initial mining reward per block (50) / 2^(Number of halving)

# Week 2 Lecture 2

## Networking Requirements

### Key Primitive

Broadcast blocks and transactions to all nodes

### Robustness

some nodes go offline then new nodes join

## Types of Network Architecture

|  |  |
| --- | --- |
| 1 | 2 |

## Data Broadcast

Data the so called transactions are broadcast using Gossip-flooding or epidemic protocol

## Dandelion protocol

anonymize the IP address of the original transaction to protect privacy of all

## Probability of Detection

d-regular trees

Anonymity Metric

Estimators

## Protocol level attacks

Longest Chain Protocol Attack

Double Spend Attack

Private attack

Balance attack

Liveliness attack

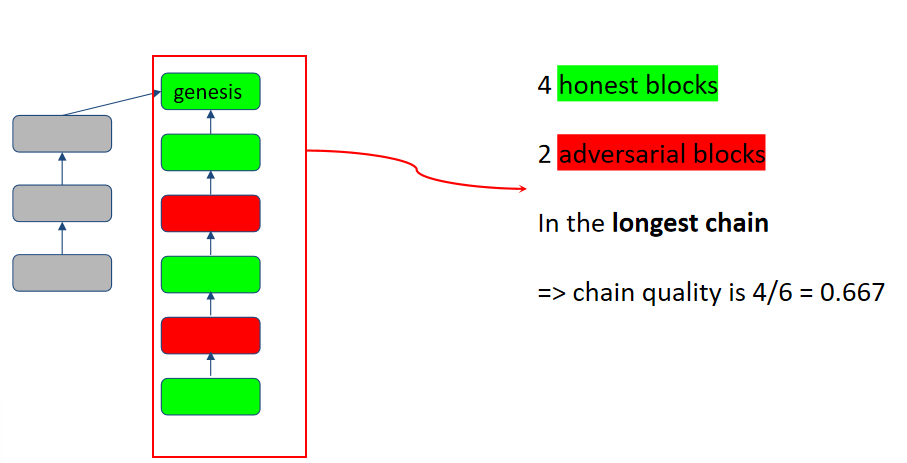
Selfish mining attack

Block withholding attack

Bribing attack

# Week 3 Lecture 3

## Chain quality

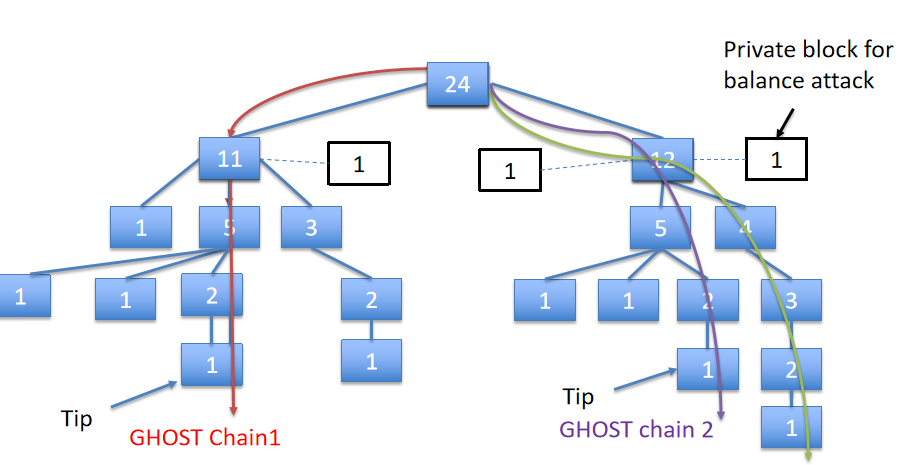


## Fruit chain

Separate transactions & their rewards from blocks in the longest chain

## GHOST(Greedy Heaviest Observed Sub Tree)

The GHOST chain is harder to displace by a private attack but easier infected by balance attack



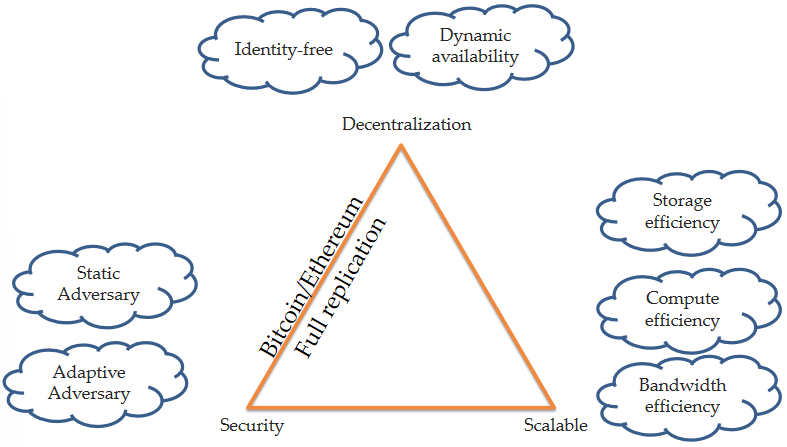
## Bitcoin-NG(Next Generation)

Positive: Throughput is high because transaction blocks are many in number and only limited by network capacity

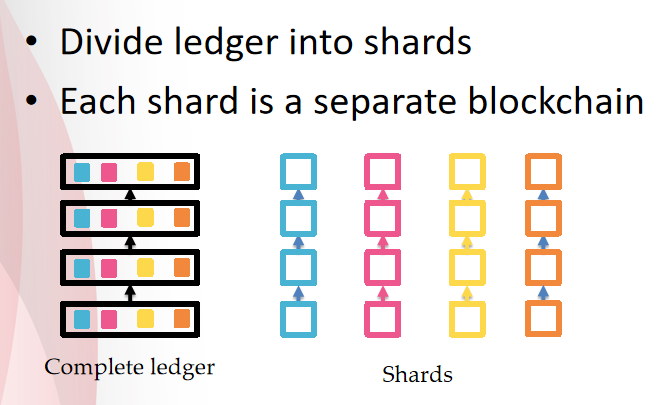
Negative: Bitcoin-NG is permission-less but does not have the full security of longest chain protocol

Bribing attack on Bitcoin-NG

## Blockchain Trilemma



## Scaling N2S(Node to shard allocation)



## Uni-consensus

### Proposer chain

Every node maintains a proposer chain therefore adversarial majority in a shard does not violate safety

## Multi-consensus

Assumption: Each shard has honest majority

### Interactive fraud-proof

Interactive fraud proof mechanism for detecting incorrect state commitments

# Week 4 Lecture 4

## Proof of Stake

Allow meaningful participation based on stake block proposers own coins

## NaS(Nothing at Stake Attack)

Adversary deviates can grow on all blocks even Genesis

## Main concepts in Blockchain

### Layer 1

Layer 1 is the base blockchain. Ethereum and Bitcoin are both layer 1 blockchains because they are the underlying foundation that various layer 2 networks build on top of

### Simple Payment Verification

Download chain of block headers from main node

### Layer 2 Scaling

Layer 2 projects will post their transaction data onto Ethereum, relying on Ethereum for data availability to increase transaction throughput

## Data Availability Attack

A malicious block producer publishes a block header so that it is impossible for honest full nodes to validate the block and generate the fraud proof

## Incorrect-coding Attack

Attack the computer system or cryptocurrency codes other then attack blockchain directly

## Reentrancy Attack

A reentrancy attack exploits the vulnerability in smart contracts when a function makes an external call to another contract before updating its own state

## Timestamp Dependence Attack

Smart contracts often use block.timestamp for time-sensitive functions.

However, miners can slightly adjust this timestamp, creating a vulnerability where they can manipulate the timing to gain an unfair advantage.

## Front running Attack

Attackers observe the memory pool and place their own transactions with higher gas fees to be processed before the target transaction, leading to potential financial losses and disruption of smart contract functionality

## DDoS(Distributed Denial of Service Attack)

exhaust critical resources such as gas, CPU cycles, or storage

## Logic Error Attack

Logic errors, or business logic vulnerabilities, are subtle flaws found in smart contracts

## Insecure Randomness Attack

Predictability or influence over a supposedly random number can allow attackers to exploit contracts for their benefit

# Week 5 Lecture 5

## Write Smart Contract Code together

Practice lecture

# Week 6 Lecture 6

## Sidechain

maintain a blockchain pegged to Ethereum

## Payment channel

Make payment off chain

* + One-way payment channel
  + Two-way payment channel
  + Commitment transaction
  + Opening transaction broadcast

## Multisig

Locking transaction has to be signed by k out of n pub keys

## Lightening network

Base fee - 1 Satoshi per forward

Fee rate – 0.5 Satoshi per million – 0.00005%

Total fees = Amount\*(Fee rate) + Base fee

## Fraud proof

The fraud proof is sufficient to execute the batch and compute the post-state root

## Rollup

Execute transactions off-chain Report compressed data on-chain

## Optimistic Rollups: pros and cons

|  |  |
| --- | --- |
| Pros | Cons |
| Offers massive improvements in scalability without sacrificing security | Security model relies on at least one honest node executing rollup transactions and submitting fraud proofs |
| Permissionless (anyone can force the chain to advance by executing transactions and posting assertions) | Users must wait for the one-week challenge period to expire before withdrawing funds back to Ethereum |
| Compatibility with EVM and Solidity allows developers to port Ethereum-native smart contracts to rollups | Rollups must post all transaction data on-chain, which can increase costs. |

## ZK Rollup

A ZK rollup coordinator generates a SNARK proof π that proves it knows the private transactions such that the post-state is correctly updated from the prestate

## ZK Rollups: pros and cons

|  |  |
| --- | --- |
| Pros | Cons |
| Short withdrawal delays | Building EVM-compatible ZK-rollups is difficult due to complexity of ZK technology |
| Relies only on trustless cryptographic mechanisms for security | High cost on computing and verifying validity proofs |
| Only TX summaries on chain | Some proving systems (e.g., ZK-SNARK) require a trusted setup |

## Data compression of an individual transaction

|  |  |  |
| --- | --- | --- |
| Parameter | Ethereum | Rollup |
| Nonce | ~3 | 0 |
| Gasprice | ~8 | 0-0.5 |
| Gas | 3 | 0-0.5 |
| To | 21 | 4 |
| Value | ~9 | ~3 |
| Signature | ~68 (2 + 33 + 33) | ~0.5 |
| From | 0 (recovered from sig) | 4 |
| Total | ~112 bytes | ~12 bytes |

## Comparison

|  |  |  |
| --- | --- | --- |
| Property | Optimistic rollups | ZK rollups |
| Fixed gas cost per batch | ~40,000 | ~500,000 |
| Withdrawal period | ~1 week | Very fast |
| Complexity of technology | Low | High |
| Generalizability | Easier | Harder |
| Per-transaction on-chain gas costs | Higher | Lower |
| Off-chain computation costs | Lower | Higher |

# Week 7 Lecture 7

## Longest Chain Protocol Pros and Cons

### Liveness

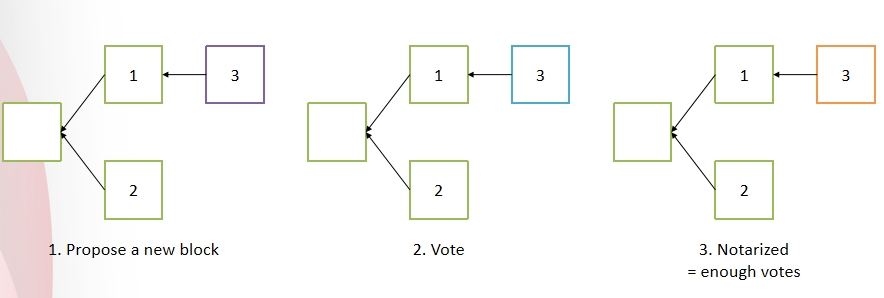
Even a single honest miner with a small hash power can extend the longest chain

### Safety

Guaranteed when hash power of honest nodes is more than 50% But with 2 caveats (Probabilistic guarantee Network/must be synchronous)

## BFT Byzantine Fault Tolerant Protocols

Deterministic safety even under asynchronous network



## Streamlet Confirmation Rule

confirming a block as soon as it is notarized is not safe

## From Streamlet to Hotstuff

Every node keeps the QC with the highest round number it knows of

## Clock Synchronization

HotStuff does not require round synchronization for safety

## Comparison



## Hybrid Consensus

Bring HotStuff to PoW for fast confirmation

## Finality and Availability

### Availability:

a protocol that remains live and safe, despite variable participation

PoW longest chain has this property

### Finality:

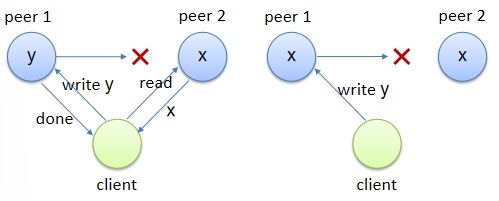
a protocol that remains safe, despite asynchrony

BFT protocol has this property

## CAP

Consistency/Availability/Partition Tolerance

Blockchain CAP Theorem says NO



## Checkpoints

entire chain leading up to prospective checkpoint block

# Week 8 Lecture 8

## Legislative BFT Protocols

When f≤t , non-faulty parties eventually agree on the same sequence of values.

## Forensic Support

Provide irrefutable evidence of bad behavior

## HotStuff

Partially synchronous protocol, tolerates 1/3 Byzantine faults (n = 3t + 1)

Linear communication complexity and responsiveness

Consensus engine for multiple blockchains

## Algorand BA

Synchronous protocol, tolerating 1/3 faults

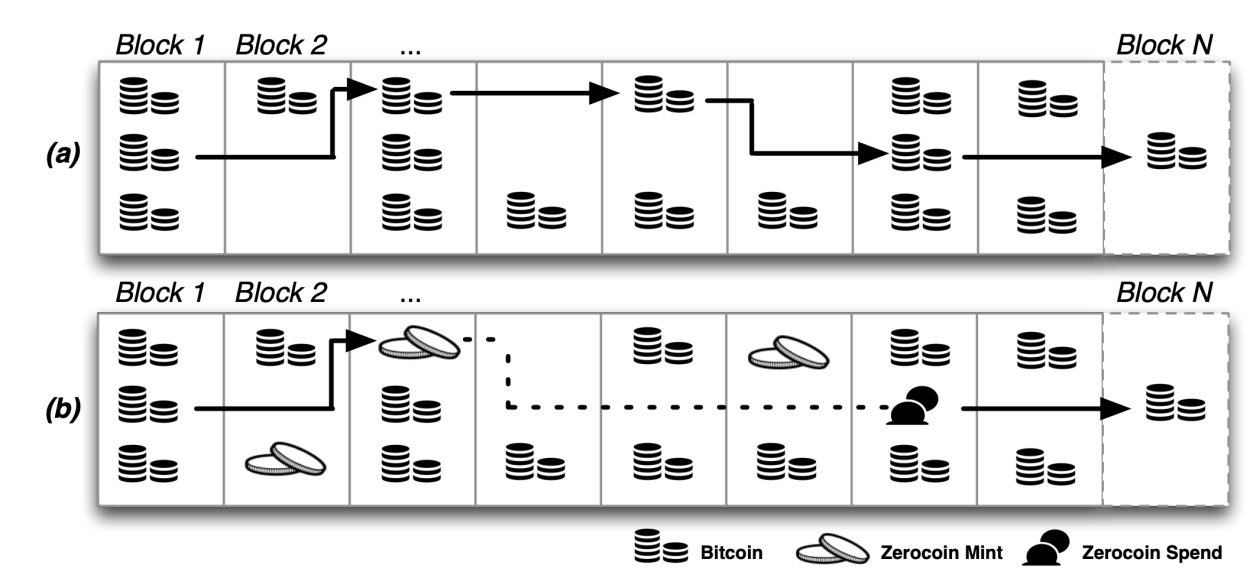
## UTXO Model

One can create many public keys to keep privacy

## Trusted Third-party Mixer

Laundry service Exchanges the coins

## Zerocoin



## Security Requirements

### Completeness

An honest prover with a valid witness can always convince an honest verifier

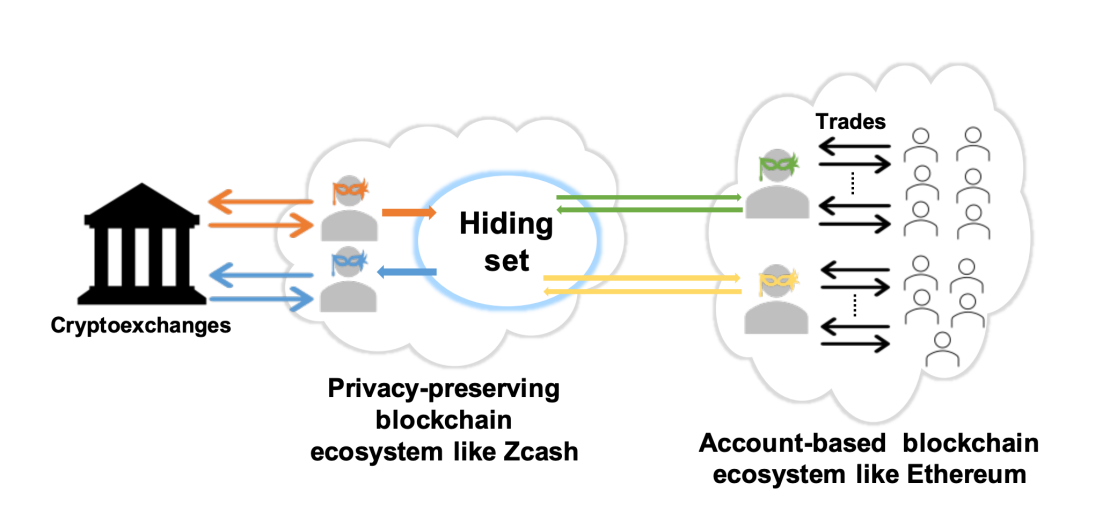
### Soundness

An honest verifier cannot accept a proof if x∉L

### Zero-knowledge

The proof does not reveal any information about w

## Privacy Bridge



1. Week 9 Lecture 9

## Interoperability in blockchains

move assets from one blockchain to another

Example: move BTC from Bitcoin blockchain to ETH in Ethereum

## Wrapped Bitcoin

Lock BTC on Bitcoin and get equivalent amount of WBTC on Ethereum

WBTC is a new token in the Ethereum blockchain (tracks value of BTC)

## More Decentralized Solutions

### External layer of trust

Committee of validators

### Atomic swaps

HTLC

## Hash TimeLock Contract (HTLC)

### Hashlock key

Restrict spending an output until some specific data is provided by the recipient.

The recipient can spend the output by providing that data and a valid signature in a predetermined time limit.

### Timelock key

If the proof is not submitted within a time limit, the coins are returned to the original owner.

In other words, the owner can spend the output by providing a valid signature after the time limit is passed.

## Atomic Swap

Swap simultaneously

Need a counter party for every swap; hard to create a market

## Trustless and Trusted Bridges

### Trusted: validator-based bridges

Validators move the data from one blockchain to another

Off-chain verification module

Trust to committee of validators for the verification module

### Trustless: light client and zero-knowledge bridges

Relayer nodes move the data from one blockchain to another

On-chain verification module

Hard to scale for many blockchains

## Validator-based Bridges

Use Multisig or Threshold Signature Scheme (TSS)

Not safe in practice - many attacks

## Bitcoin Light Client Bridge

A Bitcoin light client implemented on another blockchain (e.g. Ethereum) can verify any data that gets included on the Bitcoin blockchain.

Destination blockchain needs to be programmable.

## What is DeFi

DeFi is tokenized finance on decentralized platforms

DeFi is Non-custodial that Users control ownership of their assets

DeFi is Permissionless that Anyone can participate and interact with contracts

DeFi is Composable that Interoperability across financial instruments

## DeFi elements are smart contracts

Each element implemented via smart contracts

Smart contracts “manage” the input/output of the tokens

Smart contracts “regulate” the logic of the DeFi element

The underlying blockchain ledger maintains the time sequence ordered contract operations

## Nine elements of DeFi

Token transfers: native blockchain transactions

Market making via smart contracts

Oracles: importing external data

Borrow/Lending: banking functionality

Cross border finance: bridges, wrapped tokens

Stable coins: tying tokens to fiat

Synthetics and Perpetuals: self-adapting financial instruments

NFT: digital collectibles

DAO: tokenized governance

1. Week 10 Lecture 10