# Overview

1. Part I: Cryptocurrency and blockchain systems

Bitcoin and how it works, scalability issues, and smart contracts 比特币工作原理、可扩展性问题和智能合约

1. Part II: Confidential computing technologies

Focus on ZKP, MPC, homomorphic encryptions 研究 ZKP，MPC，同态加密

1. Part III: Web tracking and privacy in the wild

Targeting in ad and social networks, safe browsing广告和社交网络，安全浏览

1. Part IV: Machine learning and data privacy

Security and privacy in ML systems, differential privacy机器学习系统的安全和隐私差分隐私

# Part I: Cryptocurrency and blockchain systems

## Lecture 1 (Bitcoin, PoW)

### Bitcoin dark sides

Money laundering洗钱

Theft of Bitcoin wallets盗窃

Illicit marketplaces (SilkRoad) 非法市场

Rogue mining流氓挖矿

Ransomware勒索软件

### How Bitcoin works?

Background: hash function：Map data of arbitrary size to data of a fixed size

Security properties:

• Collision-resistant：Difficult to find x and y such that x != y and H(x)=H(y)

• Hiding：Given H(x), it is infeasible to find x

• Puzzle-friendly

For every possible output value y, k is chosen uniformly at random,

then it is infeasible to find x such that H(k | x) = y.

### blockchain

key idea: build data structures with hash pointers

Background: Digital Signatures

Useful trick: public key == an identity有用的技巧：公共密钥==身份

**double-spending attack**

the main design challenge in digital currency

Scrooge publishes a history of all transactions (a block chain, signed by Scrooge)

optimization: put multiple transactions in the same block 优化: 将多个事务放在同一块中

**Bitcoin: A Peer to Peer Electronic Cash System**

property #1 pseudonymous假名的

**What a bank will do to prevent double spending?**

Maintain a ledger to record every transaction! 维护一个帐本记录每一笔交易！

Bitcoin entities emulate a public trusted bulletin-board (ledger) 公共信任的公告栏(帐本)

The public ledger prevents double spending公共账目可以防止双重支出

property #2 decentralized

Has a ledger, but with no Bank. The ledger is called blockchain in Bitcoin.

**Key challenge in Bitcoin**

Distributed consensus:

All “correct” nodes decide on the same value. This value must have been proposed by some correct node.

### How consensus could work in Bitcoin?

All nodes have a sequence of blocks of transactions they’ve reached consensus on. 所有节点都有一系列已经达成共识的事务块

But it’s hard to ensure one vote per machine in a P2P system.

“Sybil” attacks”: one user creates multiple identities.

So “voting” (cleverly) in Bitcoin takes the form of hash power.

one vote per CPU (roughly speaking)

**“Mining” in Bitcoin**

All miners execute communal, computationally–intensive process called mining.

Together, mining community defines blockchain采矿社区共同定义区块链

All miners collectively search for hard-to-compute “signature” on new block (solve a puzzle).

Attacker with little computing power unlikely to mine new valid block faster than honest ones.

Security: assume less than 50% malicious

比特币挖矿是比特币网络系统的一个重要组成部分，用于达成对账本当前状态的共识。矿工是向账本提出更新的人。

## Lecture 2 (BGP, New consensus protocols)

**Recall: Nakamoto Consensus**

Securely reaching distributed consensus on the same ledger across the globe.

mining process (i.e., PoW)

挖矿就是：有用户发出事务块，然后矿工去验证这些事务是否有效，然后向账本提出更新，从而赚取交易费。

### Byzantine Generals Problem (BGP)

Impossibility Results

One traitor makes it impossible with three generals (to solve the BGP problem)

一个叛徒使得三个将军不可能解决 BGP 问题

when 𝑓 nodes can behave arbitrarily (Byzantine), 2𝑓 + 1 nodes are not enough to tolerate it.

**Asynchronous Model异步模型**

Message delays are finite, but unbounded or unknown. 消息延迟是有限的，但是无限的或未知的

More realistic/general than synchronous model.

Strictly harder/weaker than synchronous model.

• Consensus is not always possible

**Case Study: Byzantine Fault Tolerance (BFT) Question**

Given 𝑓, how many nodes do we need to tolerate 𝑓 Byzantine failures?

Let’s assume we have 𝑛 servers, and maximum 𝑓 Byzantine failures

What is the minimum # of replies that you are always guaranteed to get? 𝑛 - 𝑓

This means that if a client receives 𝑛 – 𝑓 replies, the client needs to determine what the correct answer is at that time.

However, upon receiving 𝑛 – 𝑓 replies, how many replies can come from malicious servers (i.e., lies)? Still 𝑓. Since some servers can just be really slow!

What can be the minimum 𝑛 to determine the correct answer?

if 𝑛 == 2𝑓 + 1?

• 𝑛 − 𝑓 = 3 – 1 = 2, Obviously, it doesn’t work

If 𝑛 – 𝑓 replies always contain more replies from honest nodes than Byzantine nodes, we’re safe! 在保证能收到的消息中，真节点比假节点多，才会安全

We set 𝑛 == 3𝑓 + 1

We can always obtain 𝑛 – 𝑓, i.e., 2𝑓 + 1 votes. Then we have at least 𝑓 + 1 votes from honest nodes. One more than the number of potential faulty nodes.

### Consensus in the Era of Cryptocurrency Systems时代的共识

**Nakamoto Consensus中本聪**

Maintain a chain of blocks where each block is filled with transactions. New block is added via conducting PoW minings. If conflicts, choose the longest chain.

|  |  |
| --- | --- |
| **Bitcoin scalability: How to scale Bitcoin？**Increase the block size？mining rate? new algorithms？ | |
| **Attempt 1: Increase the block size** | Pro: more transactions per sec  Con: Require sufficient consensus to do a hard fork |
| **Attempt 2: Increase the mining rate**  Decrease the mining difficulty. | Potential problems:  1. More forks in a short time period, which makes the longest chain hard to find!  2. Requires increased bandwidth.  3. A greater portion of the raw hash power will be wasted. |

**New consensus protocols are required**

1. Incompatible with the original Bitcoin design. 不符合原始比特币的设计

Deployment needs hard fork.

1. New consensus strategies

• Bitcoin-NG • GHOST (adopted by Ethereum) • Proof-of-Stake

1. New structure for parallel blockchain processing. Sharding.

Bitcoin-NG

The GHOST Protocol (Greedy Heaviest Observed Sub-Tree)

An alternative chain selection rule. Instead of “longest chain”.

At every split, pick the heaviest sub-tree.

Both Bitcoin-NG and GHOST still rely on Proof-of-Work (PoW) mining

## Lecture 3 (REM, PoS)

Challenge: Replace PoW with Alternate Resource Lottery

* Other physical resources, with different properties?

Disk space, Useful computation/storage, Etc ...

* What about the coin itself?

“Virtual resource mining” --> Proof of Stake (PoS)

First question: can we recycle the waste of Bitcoin and do something useful?

**Greening Bitcoin** (Proof-of-useful-works)

Repurposing Bitcoin Work重新利用比特币

Permacoin硬币

Permacoin: Features

Storage: a large public dataset

Recoverability: even after catastrophic failure. (Erasure Coding)

Diversity: geographical, as well as administrative多样性: 地域以及行政管理

### REM (Resource Efficient Mining for Blockchains)

**SGX-backed Blockchain**: A New Security Model

Permission-less：Anyone with SGX equipped can join

Partially decentralized：Intel correctly manages the group signature

**Proof of Useful Work (PoUW)**

Replace the hash calculation in PoW with “useful” mining work.

**Useful Work Metering**有用的工作计量

|  |  |  |
| --- | --- | --- |
|  | **Key question 1**: how to meter the effort a miner has conducted? 如何计量一个矿工所做的努力？ | **Key question 2**: how to determine whether an effort is successful, i.e., resulting in a new block? 如何确定一项努力是否成功，即产生一个新的区块？ |
| Proof of Work | perform two SHA256 hash operations. 工作证明 | check if the computed hash value is smaller than a target value.  Overall mining effort is measured in terms of the number of executed hashes. 执行哈希的数量 |
| REM | perform a CPU instruction that is defined in an PoUW task. CPU 指令 | each performed CPU instruction wins a chance to conduct a Bernoulli trial 伯努利试验  Overall mining effort is measured in terms of the number of executed useful-work instructions. 执行的有用工作指令的数量. |

### PoS: Stake-based Lottery

Blockchain tracks ownership of coins among parties区块链追踪各方之间的硬币所有权

Idea: participants elected proportionally to stake参与者按比例选举

No need for physical resources

Different ideologies意识形态

• Proof-Of-Work (POW), Vote ∝ compute power

• Proof-Of-Stake (POS), Vote ∝ stake

Ideal PoS Consensus

PoS: The Challenge

|  |  |
| --- | --- |
| PoS Designs “in the wild” | • NXT  • Peercoin  • DPoS (BitShares, Steem, EOS)  • Casper (Ethereum)  • Etc.. |
| PoS Designs with Rigorous Guarantees | Eventual (Nakamoto-style) Consensus:  • Ouroboros  • Ouroboros Praos  • Ouroboros Genesis  • Snow White  Block-wise Byzantine Agreement:  • Algorand |

Nakamoto-style VS Byzantine

|  |  |  |
| --- | --- | --- |
|  | Nakamoto-style | Byzantine |
| Scalability | Hard (>1000 TPS) | Easy (~10 TPS) |
| Safety | >50% | 33% |
| Finality | Probabilistic | Near-instant |

### PoS designs: Ouroboros

**Ouroboros**: A Provably Secure Proof-of-Stake Blockchain Protocol

**Algorand**: Scaling Byzantine Agreements for Cryptocurrencies **(not to be included in final exam)**

## Lecture 4 (smartcontract)