# 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 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。

**Longest Chain Rule**

## Lecture 2

**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

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.