Goals:
• Describe how distributed ledgers and blockchains work
• Review the basics of Bitcoin (with emphasis on distributed algorithms)

Reading:
• Bitcoin: A peer-to-peer electronic cash system, Satoshi Nakamoto. 
Distributed Databases versus Distributed Ledger

Centralized relational databases
- (Multiple) clients talk to central server
- Server manages competing requests, assigns priority, prevents conflicts
- Examples: web server

Distributed databases
- Distribute storage of information across multiple locations
- Distributed consensus algorithm is used to ensure consistency
- Examples: 2PC, 3PC, Paxos

Distributed ledgers
- Adds additional features
  - Complete (unalterable) history of transactions
  - Protection against (Byzantine) attack
Example: Bitcoin

Bitcoin = mechanism for maintaining a distributed ledger of payment transactions

Operation of the bitcoin network

- Transactions requests are collected by “miners” (acceptors) and bundled into blocks
- Blocks are added to the ledger in an append-only chain (blockchain)
- Miners run a consensus protocol to agree on the blockchain
- Miners are paid for the effort required to maintain the blockchain
- Digital signatures are used to insure validity of all transactions and blocks
- “Proof of work” is used to implement consensus and break “ties” (including “double spend” attacks)

Signed by Alice

Pay to $pk_{Bob} : H( )$

Signed by Alice

Pay to $pk_{Bob} : H( )$

Signed by Alice

Pay to $pk_{Bob} : H( )$

Signed by Alice

Pay to $pk_{Bob} : H( )$

Signed by Alice

Pay to $pk_{Bob} : H( )$

Signed by Alice

Pay to $pk_{Bob} : H( )$

Signed by Alice

Pay to $pk_{Bob} : H( )$

Signed by Alice

Pay to $pk_{Bob} : H( )$

Signed by Alice

Pay to $pk_{Bob} : H( )$

Signed by Alice

Pay to $pk_{Bob} : H( )$

Signed by Alice

Pay to $pk_{Bob} : H( )$

Signed by Alice

Pay to $pk_{Bob} : H( )$
Encryption Technologies

Cryptographic hash functions (SHA256)
- Collision resistant: $H(x) \neq H(y)$ if $x \neq y$
- Hiding: given $H(\text{nonce} \, || \, x)$, can’t find $x$
- Puzzle friendly: finding nonce such that $H(\text{nonce} \, || \, x) < Y$ is hard

Hash pointers and data structures
- Hash pointer = pointer to data + hash of data
- Block chain: sequence of blocks whose integrity is maintained via hash pointers
- Optimization: Merkle trees
  - $O(\log n)$ check of data validity
  - Sorted Merkle: $O(\log n)$ check for invalid data block

Digital signatures (PKI)
- Means of signing a transaction and proving ownership
- Implement via standard public key infrastructure (PKI), ala PGP
- Note: eliminates possibility of forging a message in Byzantine agreement problem
**Preliminary Examples: GoofyCoin and ScroogeCoin**

**Goofy coin: simple block chain implementation**
- Build chain of transactions, protected by hash pointers
- Start with set of coins and allow owners to spend
- Payee can verify payment by checking transfers
- Problem: coin owner can spent the same coin twice

**ScroogeCoin:**
- Mod #1: Implement append only ledger via transaction chain
  - Require banker to sign => avoid double-spend
- Mod #2: implement create and spend transactions
  - CreateCoins: create coin with unique ID
  - PayCoins: transfer coins between owners (with for ownership, validity, double-spend, etc)
- Problem: centralized solution (but we know what to do…)

---

**Diagrams and Flowcharts:**
- Chain of transactions with hash pointers
- Transactions signed by public and private keys
- Table of transactions with ID, type, and signatures
Distributed Consensus for Distributed Ledger

Properties for distributed ledger

- Safety: stable (accepted transactions)
- Progress: valid transaction $\sim$ accepted

Distributed ledger using blockchain

- Implement standard distributed consensus (Paxos): agree on transactions (in blocks)
- No forged messages, but could see failed nodes, replay attacks, DDOS, etc
- Avoid coordinated attacks by implementing random choice of acceptors (= miners)

Bitcoin consensus algorithm (simplified)

- New transactions are broadcast to all nodes
- Each node collects new transactions into a block
- Each round, random node gets to broadcast its block
- Other nodes accept the block only if all its transactions are valid (signed, unspent, …)
- Nodes express acceptance of the block by including its hash in next block they create

signed by Alice
Pay to $pk_{Bob}$ : $H(\_)$
Bitcoin Incentives and Proof of Work

Bitcoin uses incentives to implement consensus

- Basic idea: trust nodes that spend lots of time doing useful work => traitors have to work (very) hard
- Reward #1: Block rewards = 12.5 BTC/block created (today)
- Reward #2: Transactions fees = leave some change on the table

Mining and proof of work

- Miners must solve hash puzzle: create a nonce such that
  \[ H(\text{nonce} \ || \ \text{prevH} \ || \ \text{tx} \ || \ ... \ || \ \text{tx}) < \text{target} \]
- Properties of hash function => this takes a long time on average (~10 min)
- Can propose a block of transactions whenever you solve the puzzle
- Use consensus algorithm to agree on blocks in chain (by building on them)

Bitcoin Attacks

Impersonation attack
- Not possible as long as crypto is strong enough

Change history attack
- Not possible as long as crypto is strong enough

Double spend attack
- What happens if you try to spend same coin twice?
- A: either one could be valid
- A: consensus will be achieved around the *longest* block => eventually one of the transactions will be selected by most miners
- Can show probability of the second block “winning” goes down exponentially in the number of blocks that have been built on the chain
- From a practical point of view, need to wait ~1 hour before providing goods…
No formal analysis of the safety of the Bitcoin protocol is yet available

- Safety properties are available, but not guarantee of progress
- Most proofs focus on showing the prefix of honest peers is stable

**Problem specification (following Garay, Kiayias, Leonardos)**

- Safety: All honest peers will have the same prefix for some depth $k$
- Progress: A conflict-free transaction will eventually be deeply confirmed in the blockchain of an honest peer

**Formal definitions and analysis:**

- Let $C$ be a blockchain and let $C^{\lceil k\rceil}$ be the chain with the last $k$ blocks removed
- Let $n =$ number of players, $t =$ number of traitors, $\mu = t/(n-t)$
- Common-Prefix Property: For any two honest players $P_1$ and $P_2$ adopting chains $C_1$, $C_2$ and round $r_1 = r_2$, it holds that $C_1^{\lceil k\rceil} = C_2^{\lceil k\rceil}$
- Chain Quality Property: For any host party $P$ with chain $C$, it holds that for any $\ell$ consecutive blocks of of $C$, the ratio of adversarial blocks is at most $\mu$
- Chain Growth Property: For any honest party $P$ with chain $C$ it holds that for any $s$ rounds there are at least $\tau s$ blocks added to the chain
Formal Analysis of Distributed Ledger (2 of 2)

**Bitcoin backbone protocol**
- Read instruction [M1]: return content of chain
- Insert instruction [M2]: extend chain, solve proof-of-work, and broadcast extended chain to all
- Validate instruction [M3]: receive newly extended chain and adopt if better than local chain
- Note: miners agree on prefix to chain, but not on latest transactions

**Properties of the protocol**
- Likelihood that common prefix not present drops exponentially in length of chain
- Exponentially unlikely that adversary contributed to chain as the chain gets longer

**Note:** all properties are in terms of probabilities… (more on Wed)
Bitcoin Implementation

Surprising facts about Bitcoin
- Transaction data are unencrypted
- Transactions are coded using Forth-like language (“Script”)
  - 1 byte opcode + arguments

Optimizations to make this work
- Transactions are kept small (~250 bytes/transaction)
- Blocks are stored as hashes of Merkle trees

Some current Bitcoin statistics
- Source: https://blockchain.info
- Number of miners: ~300K
- New blocks mined every 10 min
- Market price: $7,600/BTC
- Miner revenue: $15M/day
- Hash “difficulty”: $13 \times 10^{12}$
- Size of blockchain: ~240GB
Summary: Distributed Ledger and Bitcoin

Transactions
1. A user generates a request to transfer a bitcoin value from their account to another using a mobile device or computer.

2. The request floats on the bitcoin network until users in the network called “miners” pick it up for processing.

Mining
3. During the mining process, transactions are packed into data blocks and are randomly assigned with a header.

4. Miners compete to match the block’s header with a nonce, an arbitrary number used only once, to get a short alphanumeric code called hash, which must have a value below a certain difficulty target.

5. Each hash accepted by the bitcoin network is rewarded with bitcoins, currently at 25, but this will exponentially decrease as more miners join the network.

6. The hash values are then added to the next block’s header, creating a block chain which serves as the public ledger of all transactions ever made in the bitcoin network.

Sources: Bitcoin.org; Bitcoin Ladder
C. Inton, staff, 09/12/2015

Rest of the week:
- Wed: bitcoin probabilities
- Fri: final exam review