Week 9

Blockchains and Cryptocurrencies

THE UNIVERSITY OF SYDNEY

Currencies

- Suppose we are designing a currency. What properties does it need to have?
  - Divisibility: A unit of currency should be able to be subdivided into units with equal buying power. Gold is divisible, an iPhone is not.
  - Fungibility: Individual units of currency can be exchanged for each other. Every $10 note is the same, while diamonds are different.
  - Scarcity: There should be a reasonable restriction on the projected rate of availability of the currency.
  - Recognisability: It should not be difficult to verify that a piece of currency is genuine.

Limitations of Centralised Digital Currency

- Example: Alice has $100 in her PayPal account and wants to buy some item from Bob for $25.
  - Alice asks the PayPal server to transfer $25 to Bob. Alice tells Bob that the transaction has been processed.
  - Bob checks with the PayPal server and confirms he has the $25 and sends Alice the item.
- Advantages:
  - Transactions can be reversed, in the case of fraudulent transactions.
  - Transactions are secure, and double spending or cheating cannot occur.
- Disadvantages:
  - PayPal servers are a single point of failure. Alice and Bob can't perform a transaction without the PayPal server.
  - PayPal can shift or move money at their discretion.

Bitcoin

- Bitcoin was the first successful "Crypto Currency", and the first widespread implementation of peer-to-peer distributed blockchain technology.
  - White Paper published in 2008 by Satoshi Nakamoto Bitcoin: A Peer-to-Peer Electronic Cash System

- Bitcoin uses public key encryption to secure transactions.
  - The public key is like a bank account number.
  - The private key is like a PIN / password.
- A blockchain takes the place of a central server.
  - Transactions are announced to the peer-to-peer network.
  - All transactions are visible to nodes in the network.
  - Bitcoin miners are rewarded for operating the blockchain.
    - The creator of the next block is awarded some newly minted Bitcoins.
    - The creator of the next block is awarded the transaction fees for transactions processed during the last time period.

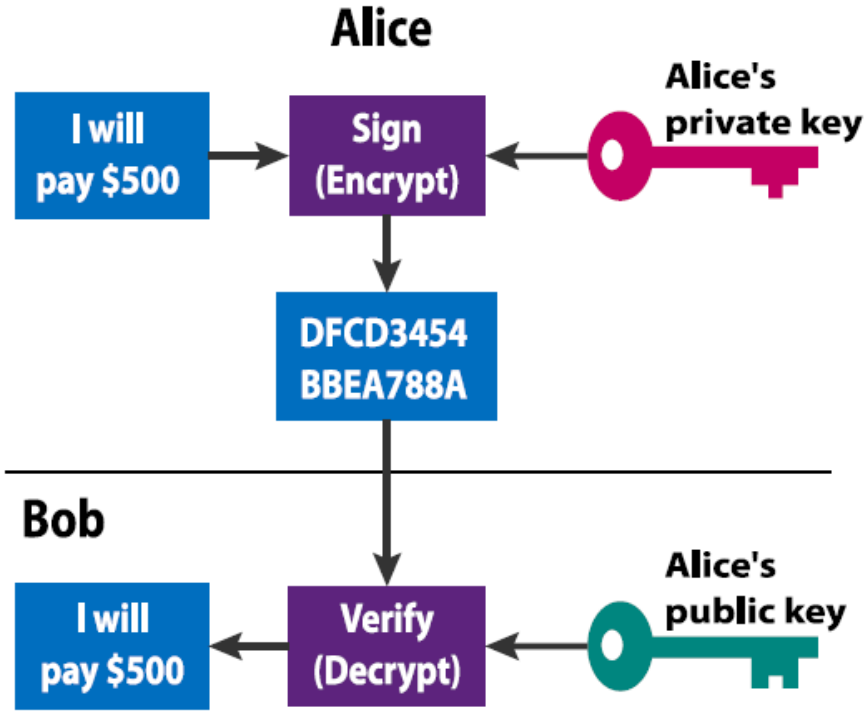
Bitcoin Wallet

- Create a public/private keypair.
  - Public: The public key is similar to a bank account number. Giving someone this number allows them to transfer Bitcoin to you. This is used to create a Bitcoin address.
  - Private: The private key allows you to transfer money away from the corresponding address.

Address (Public key)	Bitcoin Balance
1CkH8epnCee2jSnoYKVf2no8564LygpZcr	1.5027
1Lm9AuUUcazH54qFFW1Rt3V35mNvUCVFb1	17.3723
1D8L2KPG2U8mUqu6seE1GrYCRw2tkCxBHR	0.2
1NbLhL5xGS1YF8LEcXo588EPWSswgsvizb	23.5643

Public Key Crypto and Signatures

- An encryption & decryption key pair is generated.
- Blockchain address is public key.
  - Anyone can verify your signature with the public key
- Blockchain wallet is private key
  - Only you can sign a transaction with your private key



Bitcoin Transactions

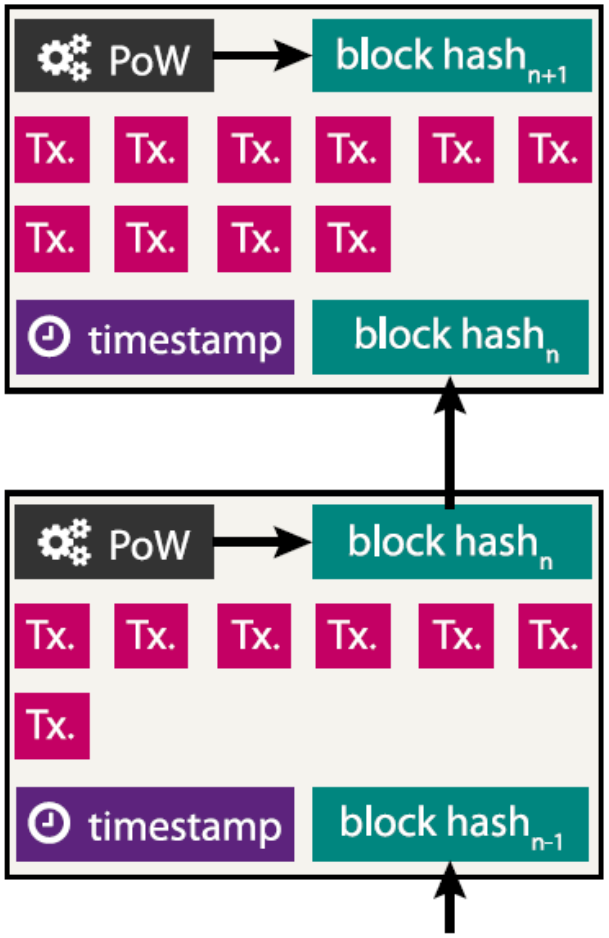
- In Bitcoin, the sender submits a transaction to the blockchain network (a distributed authority).
  - If valid, the network will agree that the receiver is the new owner.

- The payee of a transaction must be able to "prove" that the previous owners of the bitcoin did not double spend it at any point. In bitcoin, this is achieved by making everyone in the network aware of all previous transactions.

- In order to accomplish this without a trusted third party,
  - 1. Transactions must be publicly announced.
  - 2. All participants must agree on a single history for the order of transactions.

Blockchain Network

- The concept of a blockchain incorporates a series of interdependent blocks, which store a consistent history of information.
- Each block contains:
  - Tx.: A set of transactions.
  - Block hash: block identifiers linking blocks together.
  - Timestamp: time the block was "completed".
  - Proof of Work (PoW): Hashing solution that is calculated by the miners.
- Data is appended only – with a new block.



Blockchain Network

- The blockchain must definitively record all transactions
  - It must show the order in which these transactions occurred; hence it must implement some form of timestamping that is agreed upon by all observers.
- The blockchain must be difficult to modify
  - Specifically, it should be difficult to modify an existing blockchain such that past transactions may be modified, added, or removed.

- The blockchain achieves both these goals by segregating work into chunks, called blocks, which are processed on average once every 10 minutes. This time limit, and the difficulty of modifying previous blocks, is accomplished via a proof-of-work function.

Blockchain Timestamp

- The blockchain can be thought of as implementing a **timestamp server,** which takes a group of items from timestep n, and combines them into a block B_n.

- The hash of current block, H_n, is a function of both the current block's contents and the previous hash H_{n-1}.

$$H_n = \text{Hash}(H_{n-1} \parallel B_n)$$

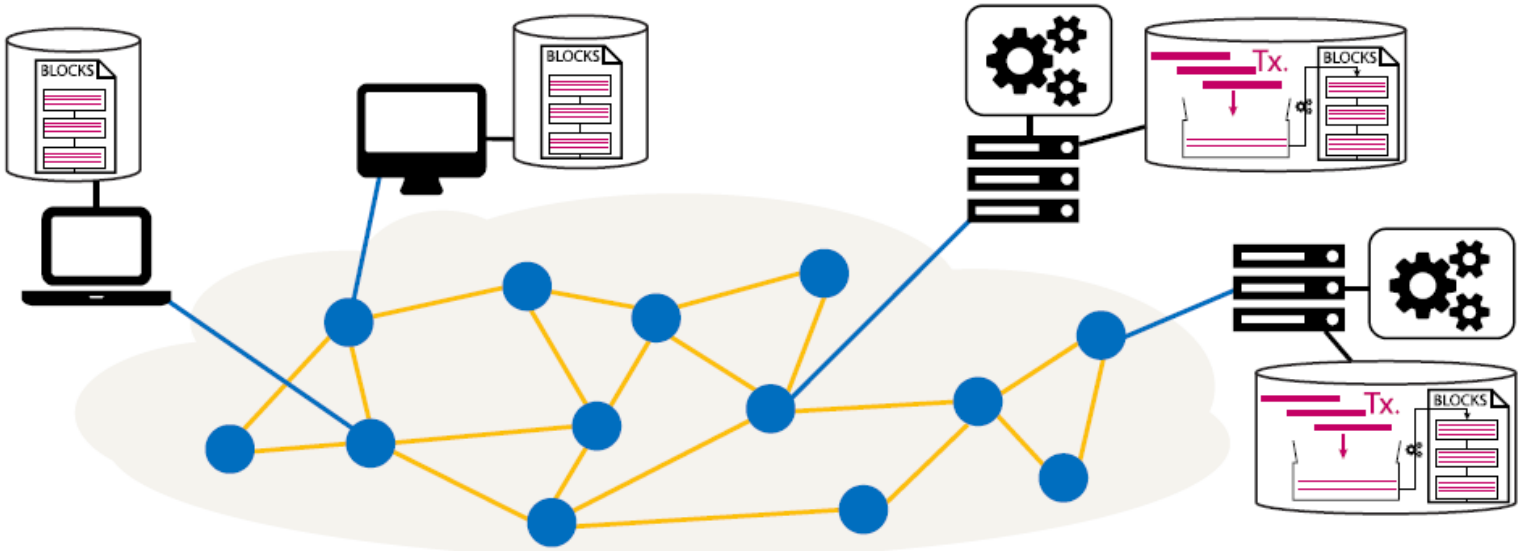
- To modify a previous item, all hashes following the modification must be recalculated.

Blockchain Proof of Work

- Pure hashing is fast (in fact, most hash algorithms are designed to be fast).
- The proof-of-work involves making the hashing more time consuming, by including a nonce in the block B_n.
- The nonce has to be changed until Hash($H_{n-1} \parallel B_n$) starts with a set number of zero bits.

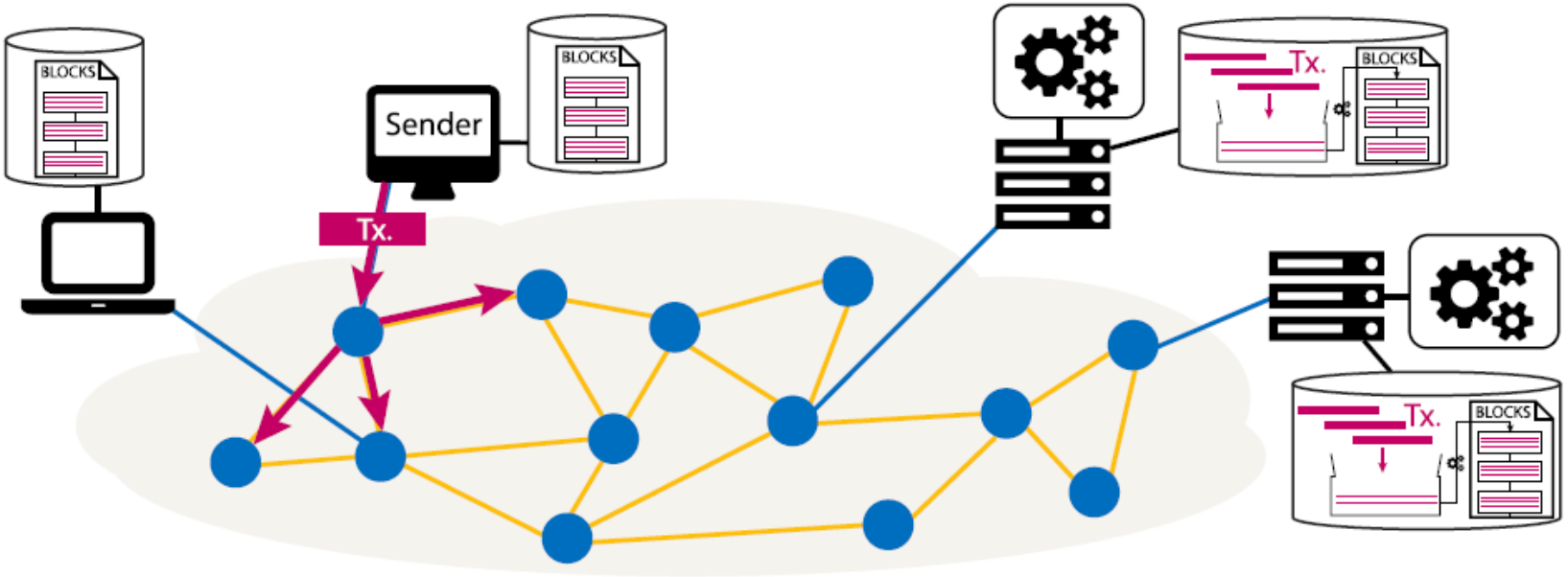
Hash Format	XXXXXXXXXXXXX
Trivial	00XXXXXXXXXXX
Easy	0000XXXXXXXXX
Harder	000000XXXXXXX

Peer-to-Peer Blockchain Network



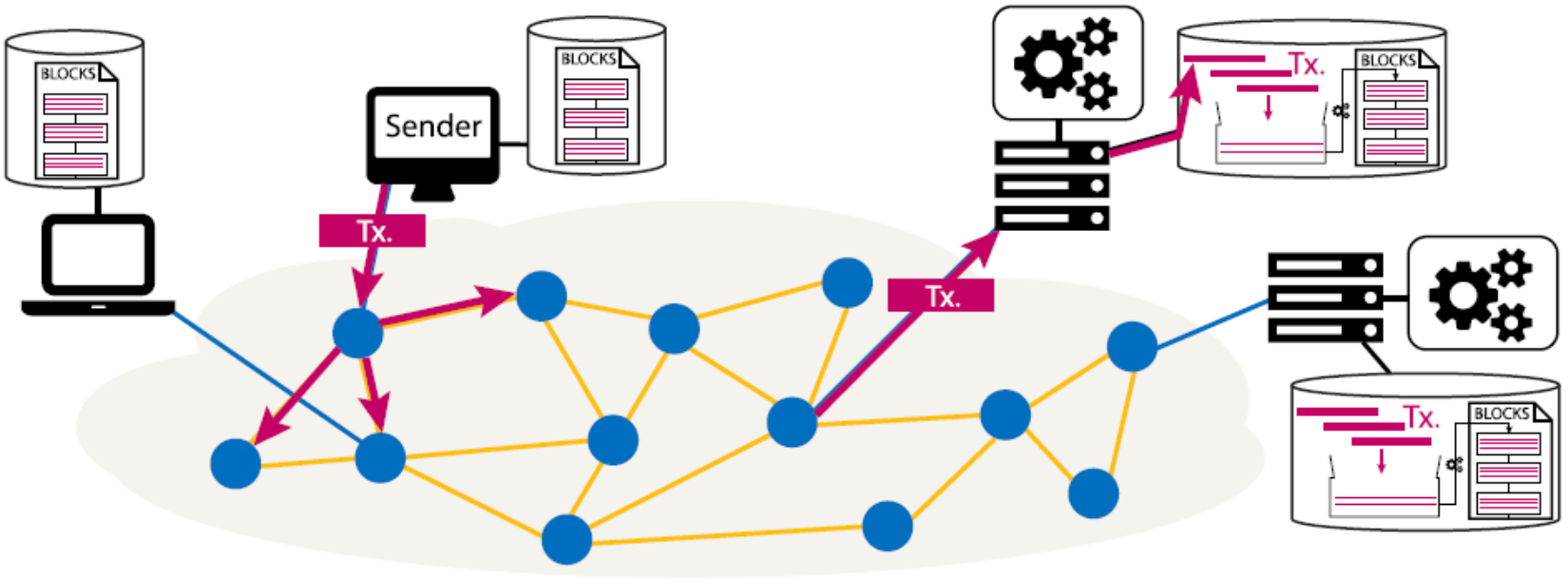
- All Bitcoin clients are sharing the latest block information via a P2P network.
  - Some of those clients are also miners that are responsible for mining new blocks and including transactions in the blocks.
  - New transactions are announced on the network by clients and included in new blocks.

Step 1: Clients send transactions to the network



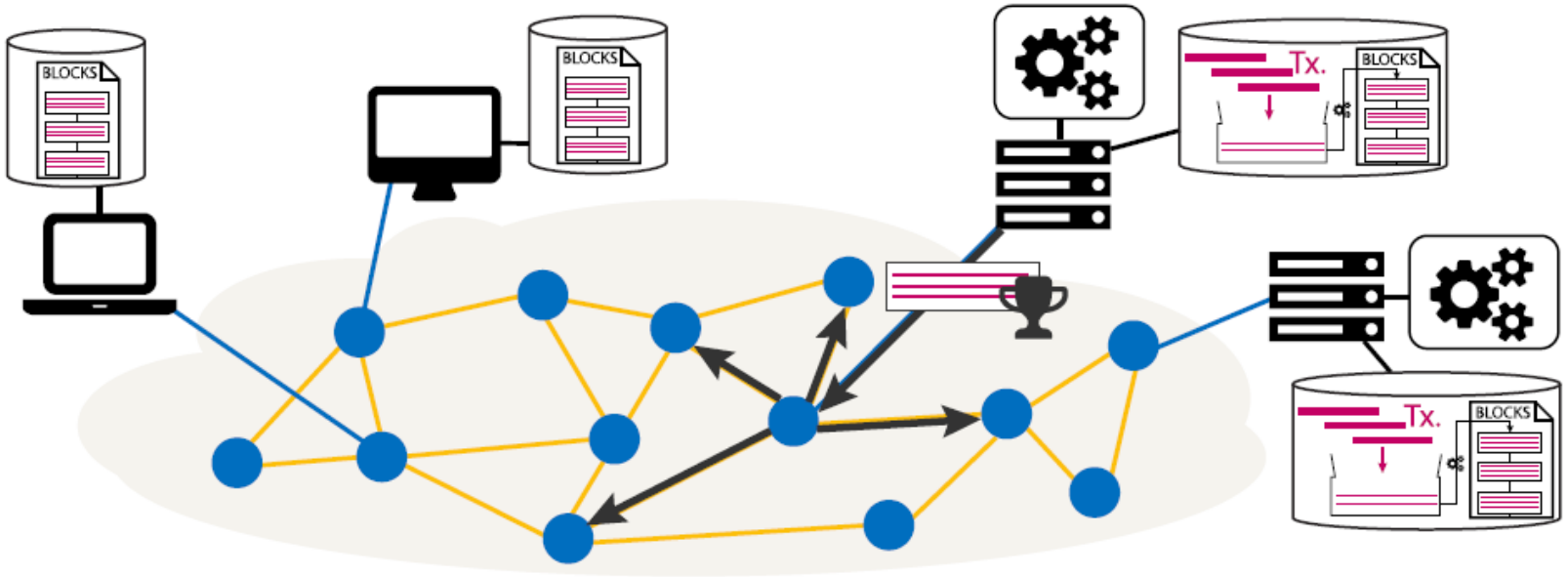
A sender authors a transaction Tx. and broadcasts it to the network.

Step 2: Miners include transactions in the next block



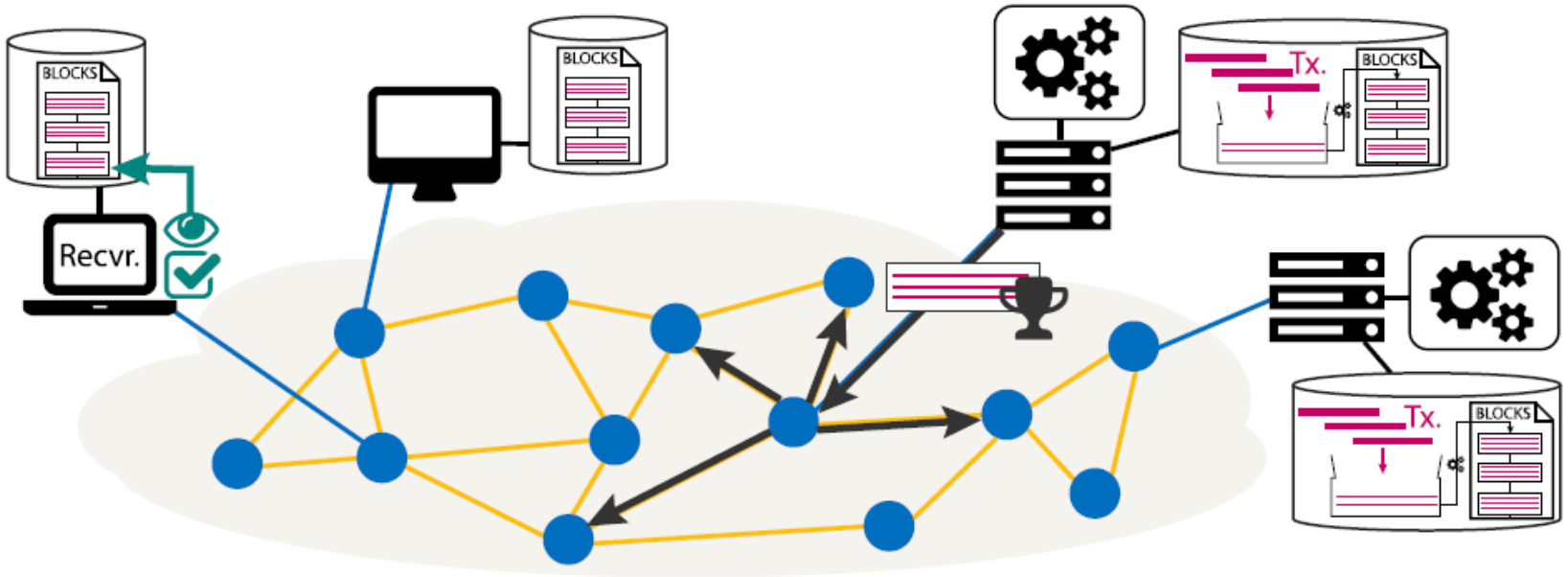
The miners include **Tx.** in the next block they work on.

Step 3: A miner generates new block by solving POW first



One miner **solves** 🏆 a block that includes Tx.
It announces the new block to the network.

Step 4: New block and transactions are confirmed by network



The receiver watches the blocks to see that Tx. is confirmed.
After a few blocks go by, the receiver accepts that the funds have cleared.

Transaction Confirmation

- In proof-of-work, sometimes ties can occur between competing miners.
  - The conflicts are resolved when one of the competing miners solves the next block. The longest chain at any time is the most authoritative.

- Transaction Confirmation
  - Wait a certain number of blocks before being 'sure' that your transaction will be permanent and irrefutable.
  - Six confirmed blocks (≈ one hour) is generally considered very safe for Bitcoin.

Incentives for Mining

- If you solve the proof-of-work and mine the next block, you receive two significant rewards:
  - The mining reward is essentially "free money in a block" rewarded to the successful miner.
    - The only way to create new bitcoin.
    - Started at 50 BTC per block, halves approximately every four years. Eventually will hit zero. (Currently 3.125 BTC per block)
  - Transaction fees incentivise miners to include your transaction.
    - A successful miner takes the transaction fees associated with each transaction in the block.
    - These fees will drive mining in the long run, once the "mining reward" runs dry.

Proof of Work Difficulty

- The SHA256 algorithm outputs 256 bits which should be "approximately a random string of bits", meaning each bit has an equal probability of being a 0 or 1.
  - Probability of any hash beginning with 10 zero bits is $2^{-10} \approx 0.1$ %.
  - Probability of any hash beginning with 50 zero bits is $2^{-50} \approx 10^{-13}$ %.

- Given the speed of hashing across the network (calculated by how fast blocks are mined), bitcoin will adjust how many zeros are required each 2016 blocks (≈ two weeks), to try to make the next 2016 blocks take two weeks.

Hash Format	XXXXXXXXXXXXX
Trivial	00XXXXXXXXXXX
Easy	0000XXXXXXXXX
Harder	000000XXXXXXX

Mining Hardware

- Bitcoin mining (similar to password cracking) can be done much faster on dedicated hardware, rather than general-purpose CPUs.
- Ordered by increasing specialisation, mining hardware is categorised as CPU, GPU, FPGA, or ASIC.
  - An ASIC is entirely custom hardware and is very expensive to start producing. Both ASICs and FPGAs consume far less power than CPUs or GPUs.

Type of hardware	Millions of Hashes / sec
Multi-Core CPU	35
Average GPU	200
Multi-Card GPU	2000
FPGA	400
$1,000 ASIC	10 000
$30,000 ASIC	14 000 000

Bitcoin Security

- Bitcoin's security resides in three places:
  - The strength of public-key authentication (ECDSA) to protect accounts.
  - The strength of SHA256 in the proof-of-work function.
  - General secure open-source programming that is designed to be upgradeable.
- Potential Attacks
  - Improper verification: Verification wasn't properly done on transactions before they entered the blockchain.
  - Blockchain Forks: The blockchain temporarily forked into two independent chains due to a major software bug. The new Bitcoin client produced a transaction that wasn't accepted by the older client, splitting the blockchain and producing the first real world examples of "double spending".
  - Control of Network: If an attacker had more than 50% of the computing power of the network, the attacker could perform double spending and also reject other people's transactions from receiving confirmations.

Blockchain Security

THE UNIVERSITY OF SYDNEY

Hardware Key Security

- Physically storing keys in hardware is more secure because breaking into hardware is difficult.
- Some challenges of hardware security:
  - Hardware still depends on software: Need more sophisticated hardware to counteract. Case study: Trezor Hardware Wallet (Official) | Bitcoin & Crypto Security
  - Can give users a false sense of security, e.g. falling victim to card skimming
  - Carrying hardware is less convenient, e.g., storing cards in wallet or RSA tokens on your keychain. Solution: standardisation & cooperation. Case study: Discover YubiKeys | Strong Two-Factor Authentication for Secure Login (yubico.com)
  - Developing hardware is expensive. Need to consolidate solutions while allowing flexibility and upgradability.

Distributed Key Solutions

- Since we can't trust people to look after their own keys, we need ways to spread the risk to multiple people. It's harder to hack all 5 individuals, or have 5 individuals all lose their keys, than one.
- Copying the key to 5 people reduces likelihood of losing key, but increases likelihood of it being stolen or misused. How do we distribute the responsibility of a key?
  - Need to hack 2 or 3 people.
  - If one person loses their key, the others can "reset" or "reissue" their key
- Multi-signature is a technique to enforce complex key requirements. In a multi-sig scenario, a group of n people have their own independent keys which they self-manage.
- Shamir's Secret Sharing could be used for key splitting to divide a secret into m pieces, where only n of them are required to reconstruct the original secret.

Cryptocurrency Exchanges

- Many people store their cryptocurrency on exchanges to balance security vs usability.
  - Authenticated with:
    - Password
    - MFA
    - E-mail address
- Common target for hackers and crypto exchange is responsible for ensuring security. Case study: Mt. Gox https://www.investopedia.com/terms/m/mt-gox.asp

Trust

- Blockchain systems are often said to be “trustless” or that they “remove the need to trust a 3rd party”. In fact, blockchains do not remove trust, they move trust.
- Instead of trusting a central entity (e.g. a bank) to perform some action with integrity, the users are trusting a whole network of computers to perform some action with integrity. A network is a lot more complicated than a central authority and has many points of failure.
- Many attacks and security considerations in decentralised systems are solved using game theoretic solutions. By choosing incentive structures carefully, Denial-of-Service attacks can be avoided. Case study: The Ethereum network is currently undergoing a DoS attack | Ethereum Foundation Blog

Decentralisation

- A major selling point of public blockchains is their decentralisation.
- There are a number of different ways to measure the amount of centralisation when it comes to blockchain networks.
- We will discuss the following metrics:
  - Control of Mining Power
  - Location of Network Nodes
  - Diversity of Node Software

Decentralisation: Mining

- Mining is a game of efficiency; the two extremes are:
  - Mining Farm
    - If you rent space to a hydro-electric power plant to house a warehouse full of miners, your cost of operations are low. You can probably also afford to buy ASICs.
  - Individuals
    - If you live in an apartment in Sydney, power is very expensive, and the economies of scale are not competitive. There is less incentive to mine.
- Miners control the network, and the economics encourage them to consolidate and centralise.

Decentralisation: Network

- The nodes which form the underlying peer-to-peer network are often hosted in similar scenarios, such as:
  - Hosted on Amazon AWS
  - Located in a particular locality with economic advantage
- Large-scale outages can have severe impacts on network availability. Imagine an Amazon AWS outage and all the mining power on AWS suddenly drops off the network. The result:
  - A poorly connected P2P network
  - Very long block times
  - Losing hash power = slower blocks…until the network adjusts.

Decentralisation: Software

- A software monoculture can be a dangerous thing.
- Blockchain networks operate according to well defined protocols, but software implementations differ.
- Ethereum has clients written in:
  - C++: Aleth (cpp-ethereum)
  - Go: go-ethereum (geth)
  - Rust: parity
  - Python: pyethapp
  - Java: ethereumj

Beyond Bitcoin

THE UNIVERSITY OF SYDNEY

Blockchain = Timestamp Server

- At its heart, a blockchain is just a decentralised, append-only timestamp server.
- It does not:
  - Remove the need for trust
  - Process transactions at scale
  - Provide efficient computation or information transfer
- It can:
  - Move trust
  - Improve data integrity
  - Strengthen non-repudiation
  - Provide a verifiable order of events

Public vs Private Blockchains

- Public: Moves trust from a centralised organisation to a decentralised group of (humans running) computers.
  - Publicly accessible (Not necessarily publicly readable)
    - Bitcoin
    - Ethereum
    - Monero
- Private: Improves efficiency and data integrity across large organisations or groups of organisations.
  - Only verified parties can participate in the network.
    - Eris Industries
    - Rubix

Public Blockchains

- Properties:
  - Anyone can participate.
  - Sometimes, anyone can read.
  - Secured by many participants, the more the better.
- Use Cases:
  - Censorship avoidance
  - Prevent corruption
  - Move trust away from organisations
- Developing Economies
  - In developed countries, organisations are more trusted → less need for cryptocurrency.
  - In developing countries, organisations are not so reliable → trading can be performed more securely using cryptocurrency.

Private Blockchains

- Properties:
  - Only authorised parties can participate or read.
  - a.k.a. “permissioned” blockchain.
  - Secured by authorisation lists.
  - Not very decentralised
- Use Cases:
  - More rigorous audit trails
  - Prevent corruption/improve integrity
  - Improve cooperation & transparency
  - More efficient, append-only database

Ethereum

- Ethereum extends the concept of a distributed payment network, by creating a distributed supercomputer.
- Executes “smart contracts”; independent apps that are governed by their source code, written in Solidity language.
- Allows deploying distributed apps on a blockchain without having to create a whole new blockchain infrastructure.
- Deploying and executing smart contracts are done using transactions which accompany a transaction fee, calculated in “gas”.
- The internal currency of Ethereum is “ether” and nodes processing transactions are paid the corresponding “gas” amount in “ether”.

- A transaction is like a road trip. Based on the type of trip (uphill, downhill), it will require some amount of gas. A given trip will cost you differently in $$ based on when you go on the trip, as gas prices fluctuate.

Smart Contracts

- A smart contract is a program on the blockchain. Like a regular program, a smart contract has:
  - # Code
  - # Inputs
  - # Outputs
  - # Data storage
- Unlike a regular program, a smart contract is executed by all nodes in the network. Anyone can send a message (input) to a smart contract, causing all nodes in the network to agree on the state change caused as a result of the message and code.

Smart Contracts

- A smart contract:
  - is fully defined by its initial code. (code is public)
  - cannot be modified. (unless its code explicitly permits)
  - can receive input from anyone on the network.
- Since a smart contract could fully encapsulate simple business logic, they can be used to create decentralized autonomous organisations (DAOs).
- Further reading: Ethereum roadmap | ethereum.org

THE UNIVERSITY OF SYDNEY