



Introduction to Blockchain Block

What is Blockchain?

Ledger Book Analogy -> Digital Ledger Book

- **Blocks as Pages:** Each block in the blockchain is like a page in a ledger book, containing a list of transactions.
- **Linking Pages:** Each block refers to the previous one using a cryptographic hash, forming a secure, linked chain.
- **Decentralization:** The blockchain is maintained by a network of computers, called nodes. Each node has an identical copy of the entire blockchain. My ETCMC example.
- **Consensus Mechanism:** To add a new transaction (block), all nodes must agree on its validity through a process called consensus.

Mining

- **Definition:** Mining is the process by which new blocks are added to the blockchain.
- **How It Works:** Miners use powerful computers to solve complex mathematical problems. The first miner to solve the problem gets to add the new block to the blockchain and is rewarded, usually with cryptocurrency.
- **Methods:** Proof of work, Proof of stake, proof of authority,

Use Cases:

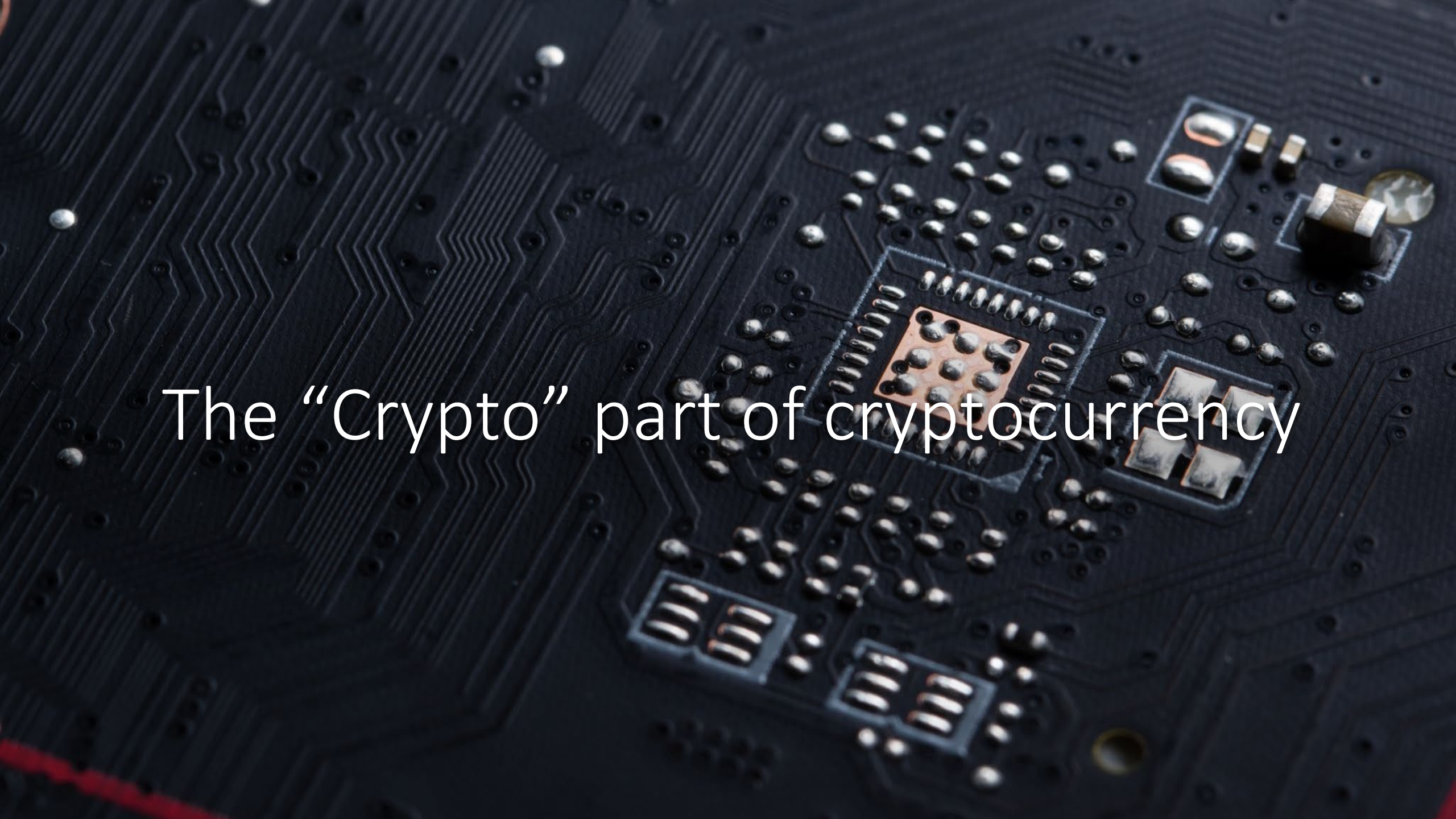
- **Cryptocurrency:** Enables secure, peer-to-peer digital transactions (e.g., Bitcoin, Ethereum).
- **Shipping/Supply Chain Management:** Enhances transparency and traceability from origin to consumer (e.g., Walmart, IBM Food Trust).
- **Smart Contracts:** Automates and enforces agreements without intermediaries (e.g., Ethereum).
- **Artificial Intelligence (AI):** Ensures data integrity and enables decentralized AI networks (e.g., Ocean Protocol, SingularityNET).



The background is a dark teal gradient with an abstract digital cityscape. It features isometric buildings made of glowing blue and green dots, resembling a data matrix. Several bright green and red light streaks, like digital rain or data paths, crisscross the scene. There are also some soft, out-of-focus circular light spots in shades of green and red.

What we will cover

- Basic Cryptography and Networks (Precursor to understanding blockchain)
- Basic blockchain architecture (Start with Bitcoin)
- Consensus Mechanisms
- Case Studies of Various Blockchains
- Smart Contracts and ERC Token Standards



The “Crypto” part of cryptocurrency

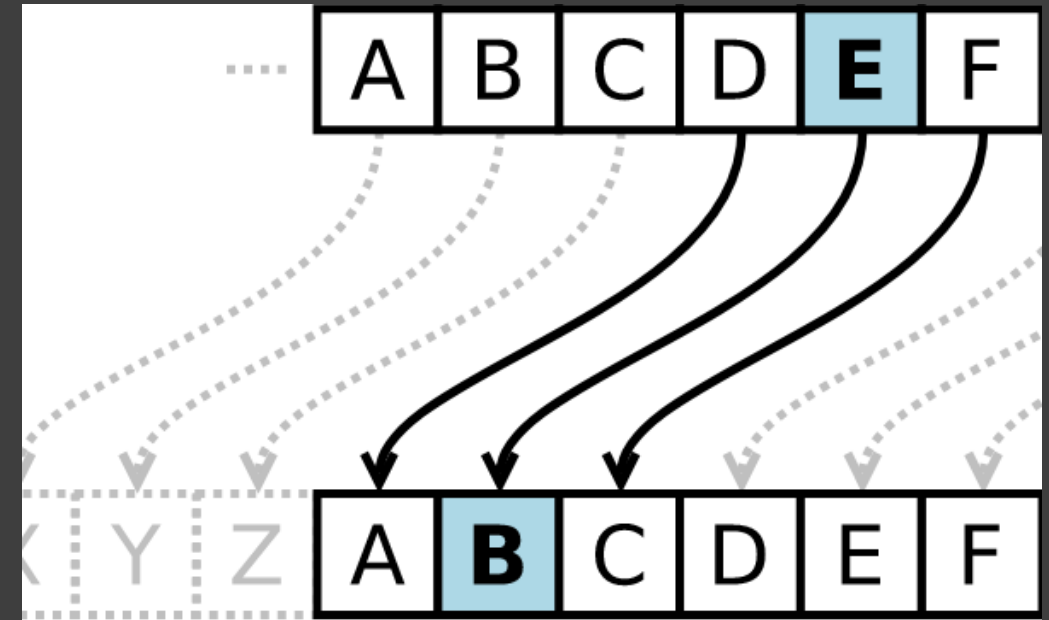


Cryptography

- The art of writing or solving code
- How do we keep our data secure and communicate with each other safely?
- Encryption: scrambling data from “plaintext” to “ciphertext”
- Decryption: unscrambling data from “ciphertext” to “plaintext”
- Want to encrypt our data such that only the people we desire to decrypt can decrypt it

Examples of basic encryption

- Caesar Shift
- Row and column permutation
- Using a key to encrypt
- Data Encryption Standard
- And far far more



Encryption

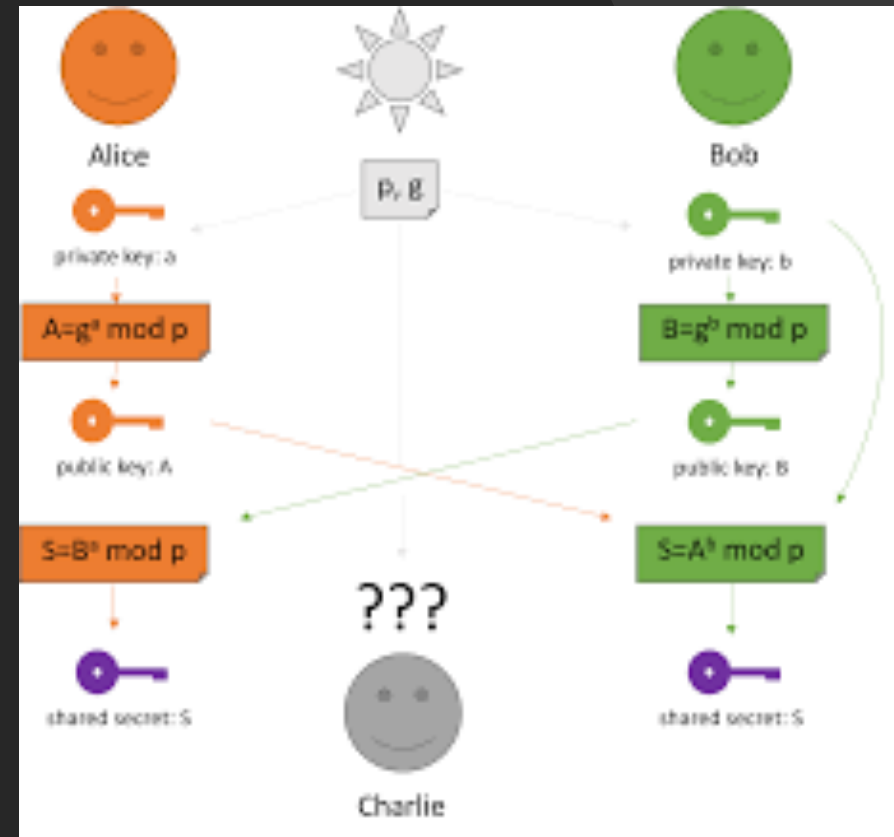
Given text = Geeks for Geeks
Keyword = HACK **Length of Keyword** = 4 (no of rows) **Order of Alphabets in**

H	A	C	K
3	1	2	4
G	e	e	k
s	-	f	o
r	-	G	e
e	k	s	-

Print Characters of column 1,2,3,4
Encrypted Text = e kefGsGsrekoe_

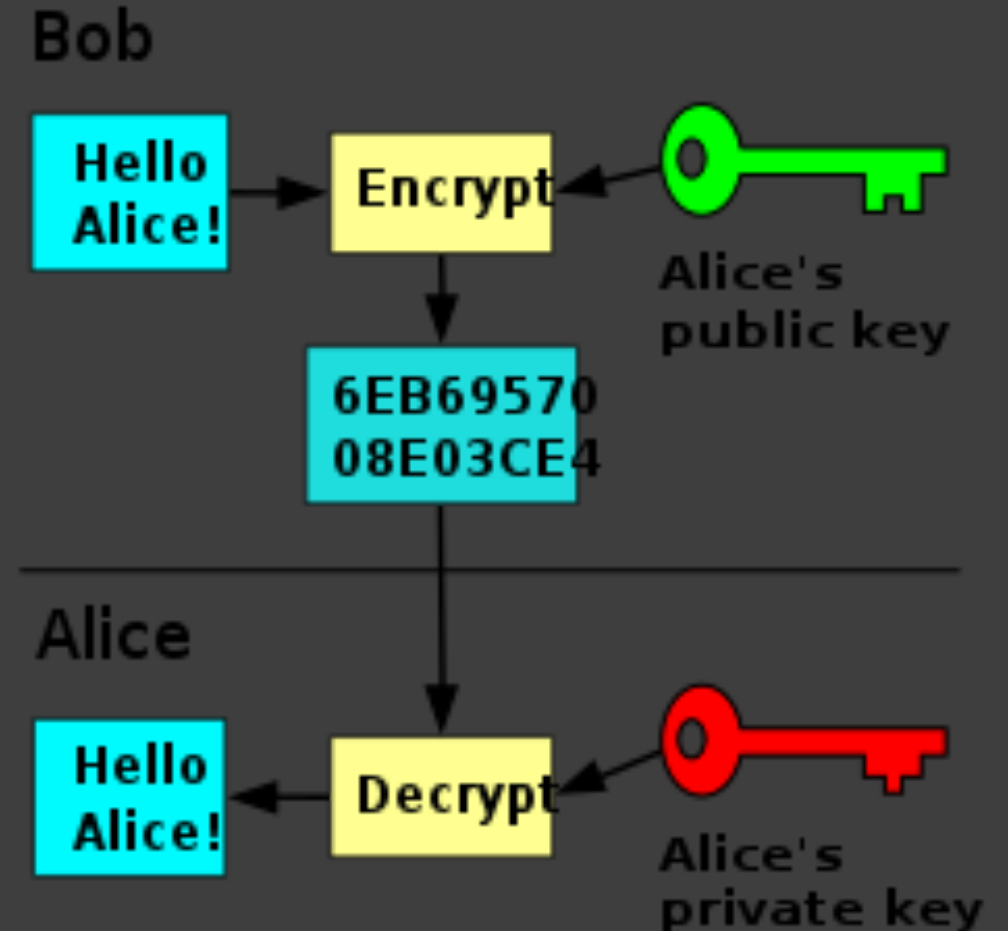
Symmetric Keys

- First, Symmetric Keys: we use these for both encrypting and decrypting a message
 - One would establish a symmetric key with another user for a communication channel (Diffie-Hellman Key Exchange)
 - This is out of scope of class, but takes advantage of repeated squares with mod of a large prime number and a “generator”



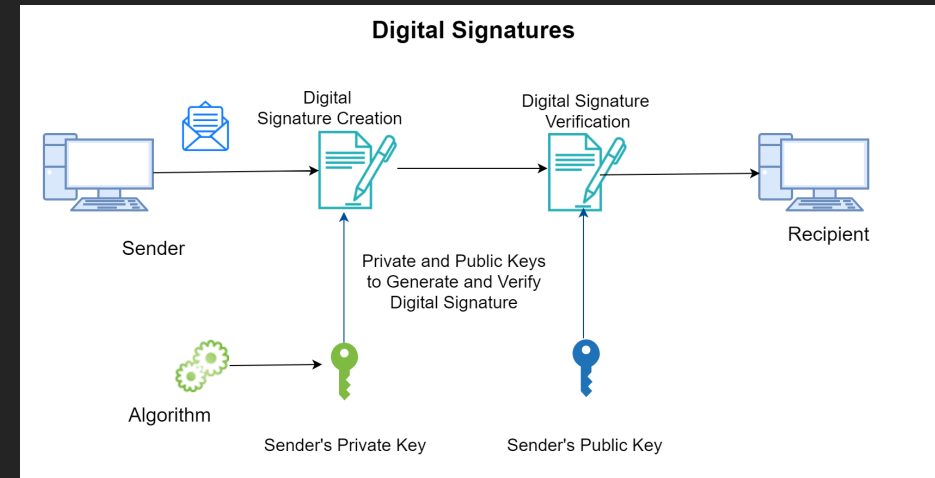
Public and Private Keys

- Unlike symmetric keys, we have two keys here
 - Now, we can keep one private such that it cannot be intercepted by a malicious user
- Public key: The key accessible to everybody will be public
- Private key: kept to yourself, nobody else will have it
- We can use this for anyone to send a message to the user with private key, or use the private key to sign things



Signing with Private Key

- We can verify that a person sent a specific message through a “digital signature”
- The digital signature is basically just your private key
- Everyone can verify this with public key

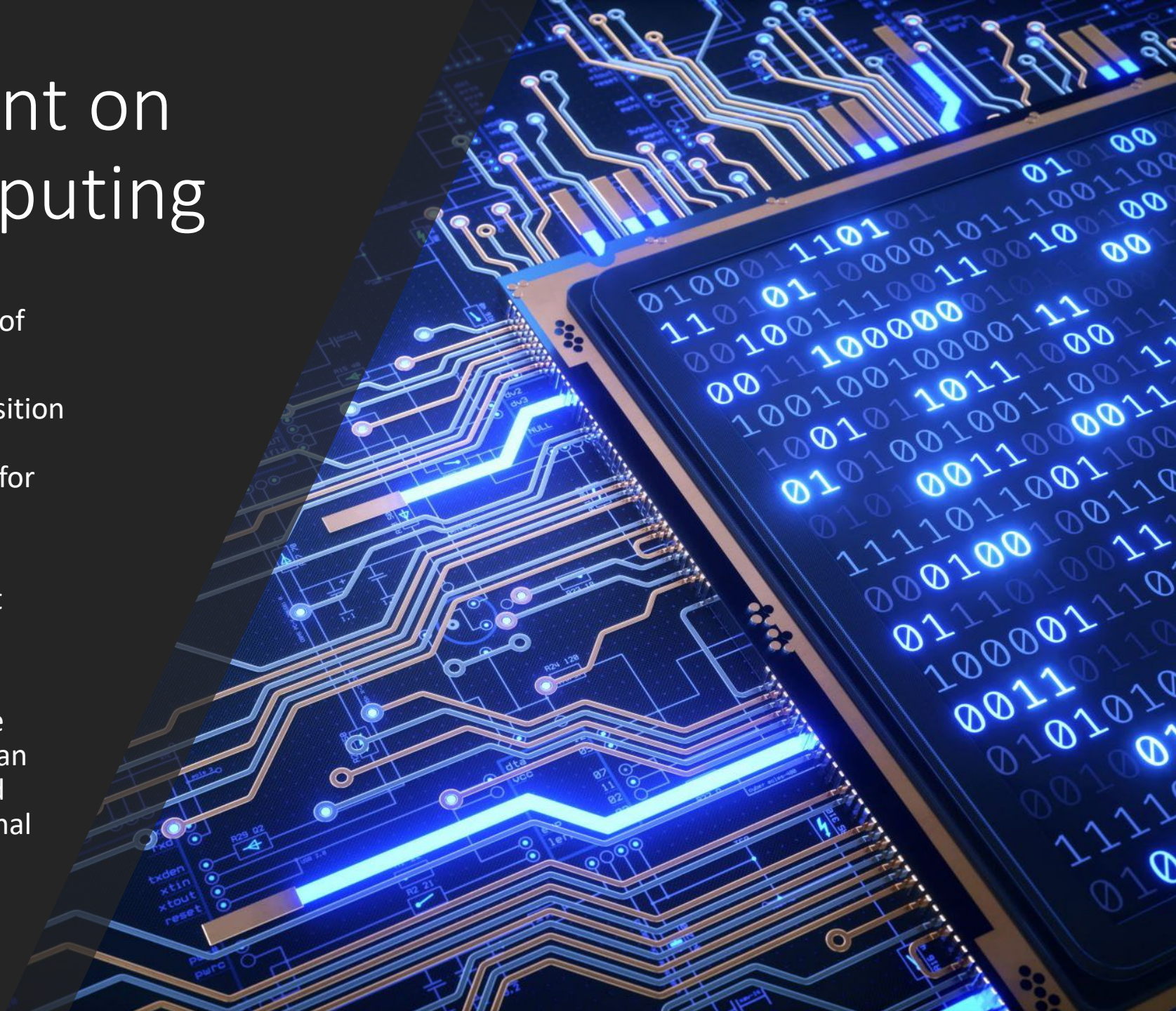


Public and Private Key Pairs

- We will use the example of RSA to explain this but many forms of key generation exist
- Select two large primes p and q to get $n = p * q$
- Choose e s.t. e and $(p-1)*(q-1)$ are relatively prime
- Public Key pair of (e,n) to get Ciphertext $C = M^e \bmod n$
- Find d s.t. $(d*e) \bmod ((p-1)*(q-1)) = 1$
- Private Key is (d,n) use $M = C^d \bmod n$

Random Tangent on Quantum Computing

- Quantum is capable of breaking all of these
- Qubits can be in a state of superposition between 0 and 1 (represented as vectors for computer science spins for hardware)
- Discrete Log or Simon's Problem, quantum computing is very good at finding that exponent
- Due to superposition, quantum in a form (assuming you can bring these qubit states out of superposition) can query multiple options at once, and rotate all states if they are orthogonal



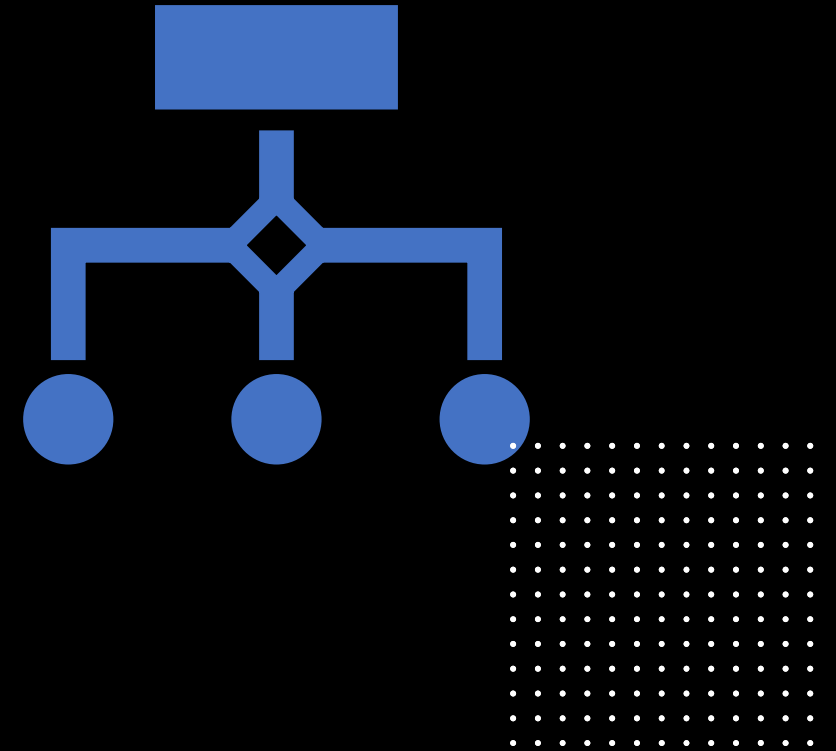
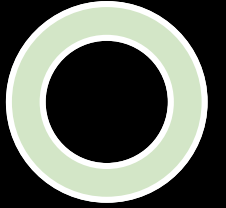
Hashing

- Hashing is the concept of a one way function that can scramble an input and give you some output
- Deterministic scrambling
- One way: this basically means once you hash it, it is very very difficult to “unhash it”
- Two similar inputs should look very different
- abc:
ba7816bf8f01cfea414140de5dae2223b00361a396177a9cb410ff61f20015ad
- Abc:
06d90109c8cce34ec0c776950465421e176f08b831a938b3c6e76cb7bee8790b



More Hashing

- If one bit changes in input, we ideally want 50% of bits to flip on average for the output
- Obviously we want this to be as close to one-to-one as possible
- We have way too much possible inputs for this to happen but can have a computationally high amount of options for outputs it will take too computationally long for computers to solve
- We want it as close to only input x generates output y , or at least very difficult to get the same output y with a different input






Networking





Client-Server Networks

- Client-Server networks are what are used in a lot of web service and general networking right now
 - The client requests some information, and the server serves it
 - One central entity processing all requests and serving information as necessary
 - Obviously some will get more servers and spread them out for service, but generally central entity controlling communication and processing of requests
 - Could have an issue with a lot of requests at once
 - Denial of Service attacks and general server downtime becomes an issue
 - Also this entity needs to be trusted to keep your data safe
- 



Peer-to-Peer Networks

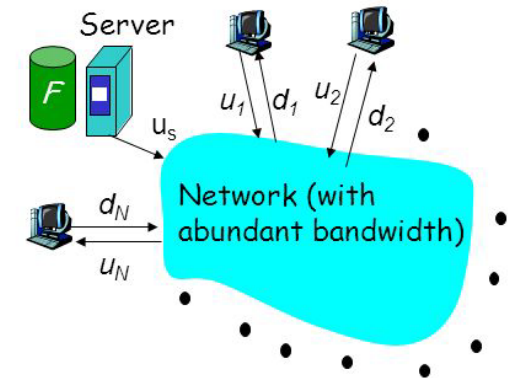
- Can be qualified as P2P with something as simple as connecting two PCs via USB to communicate
- Generally directly communicate with peers rather than through a middle server
- All users in a network cooperate to deal with requests and transactions etc.
- Can also lead to faster download speeds in the case of something like BitTorrent



Client Server Distribution Time (Credit CMPEN362)

File distribution time: server-client

- server sequentially sends N copies:
 - ❖ NF/u_s time
- client i takes F/d_i time to download



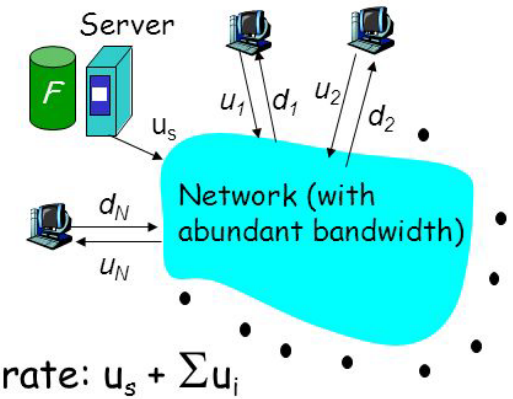
Time to distribute F
to N clients using
client/server approach $= d_{cs} = \max \{ NF/u_s, F/\min_i(d_i) \}$

increases linearly w.r.t. N (for large N)

P2P (Credit CMPEN 362)

File distribution time: P2P

- ❑ server must send one copy: F/u_s time
- ❑ client i takes F/d_i time to download
- ❑ NF bits must be downloaded (aggregate)
 - ❑ fastest possible upload rate: $u_s + \sum u_i$



$$d_{p2p} = \max \{ F/u_s, F/\min_i(d_i), NF/(u_s + \sum u_i) \}$$

Comparison



Decentralization

The Buzzword to Rule Them All



Distributed Ledgers

- Instead of keeping a ledger of transactions on a centralized server, why not distribute it to many users
- All users can witness transactions and we can make sure the majority agree on validity rather than a central entity
- Controlled by all not one



Bitcoin

What is it?

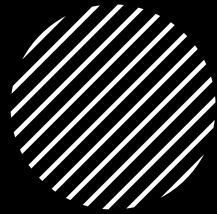
Bitcoin

- First Generation of Blockchain
- Created in 2009 by “Satoshi Nakamoto”
- Genesis block created Jan. 3, 2009 with a headline from The Times
- First transaction was 10000 BTC for two Papa John’s pizzas on May 22, 2010
- Obviously created at a time of general distrust in the economy





Basic Structure



- A Public List of transactions shared across a P2P network
- Nodes would agree on history of transactions
- Mine this new block with a connection to the previous blocks in the chain
- Hence, blockchain, a chain of mined blocks which contain transactions
- Mining takes lot of computational work, allowing for trust in decentralization rather than a central entity.



Transactions

- Transactions will take place when you are sending money between wallets
- Sender must sign (using the private key we discussed)
- Transaction leads to a fee to “mine” the block
 - With bitcoin this is abstracted away by a mining reward
- This transaction is broadcasted and spread across the network to synchronize the distributed ledger

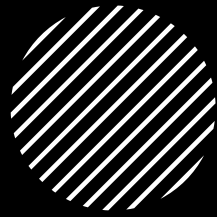


“Mining” A Block

- Miners are these nodes that collect the transactions and have a copy of the distributed ledger
- Once a timestamp has passed for the ledger of new transactions to be added as a block, the miners will all try and “solve” their block
- First solution will be broadcasted, and checked, and if it is validated by all other mining nodes, it will be added
- If not valid, keep trying until one is agreed upon by the majority of nodes



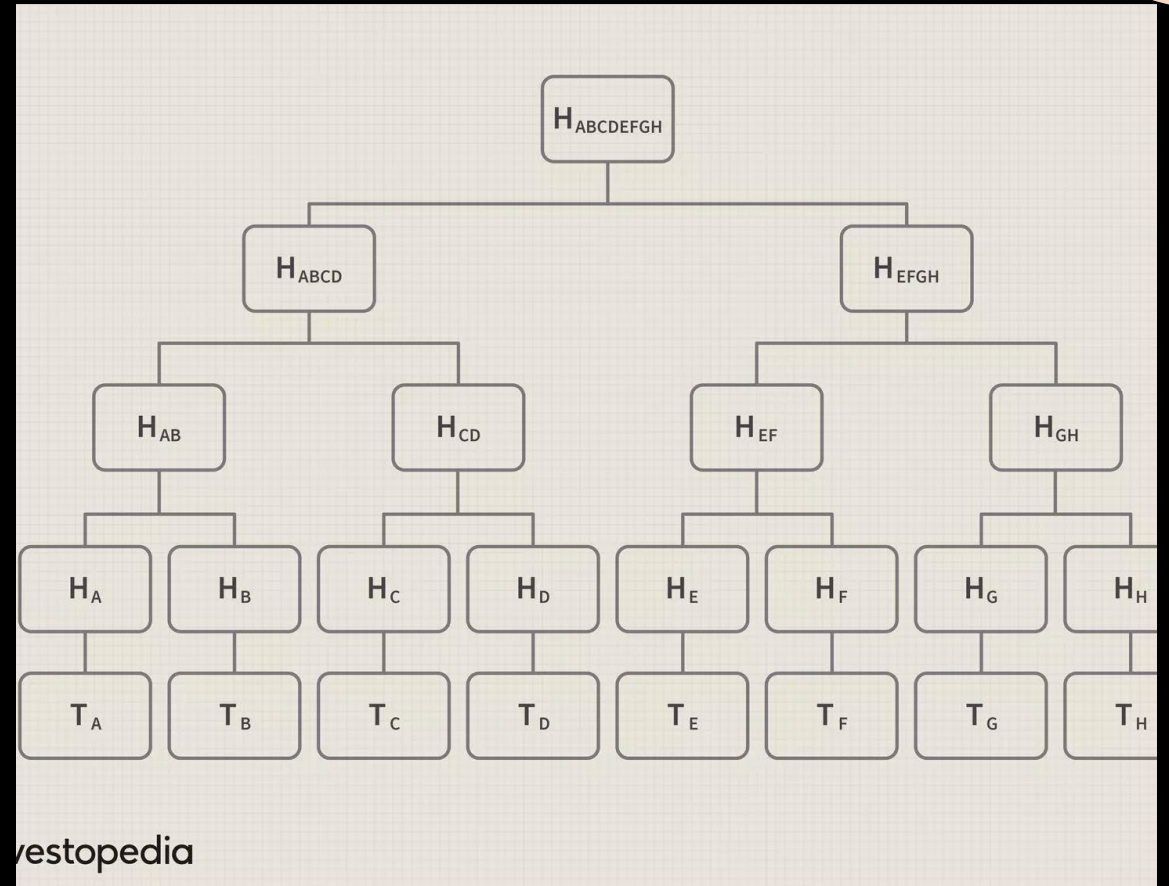
Merkle Tree



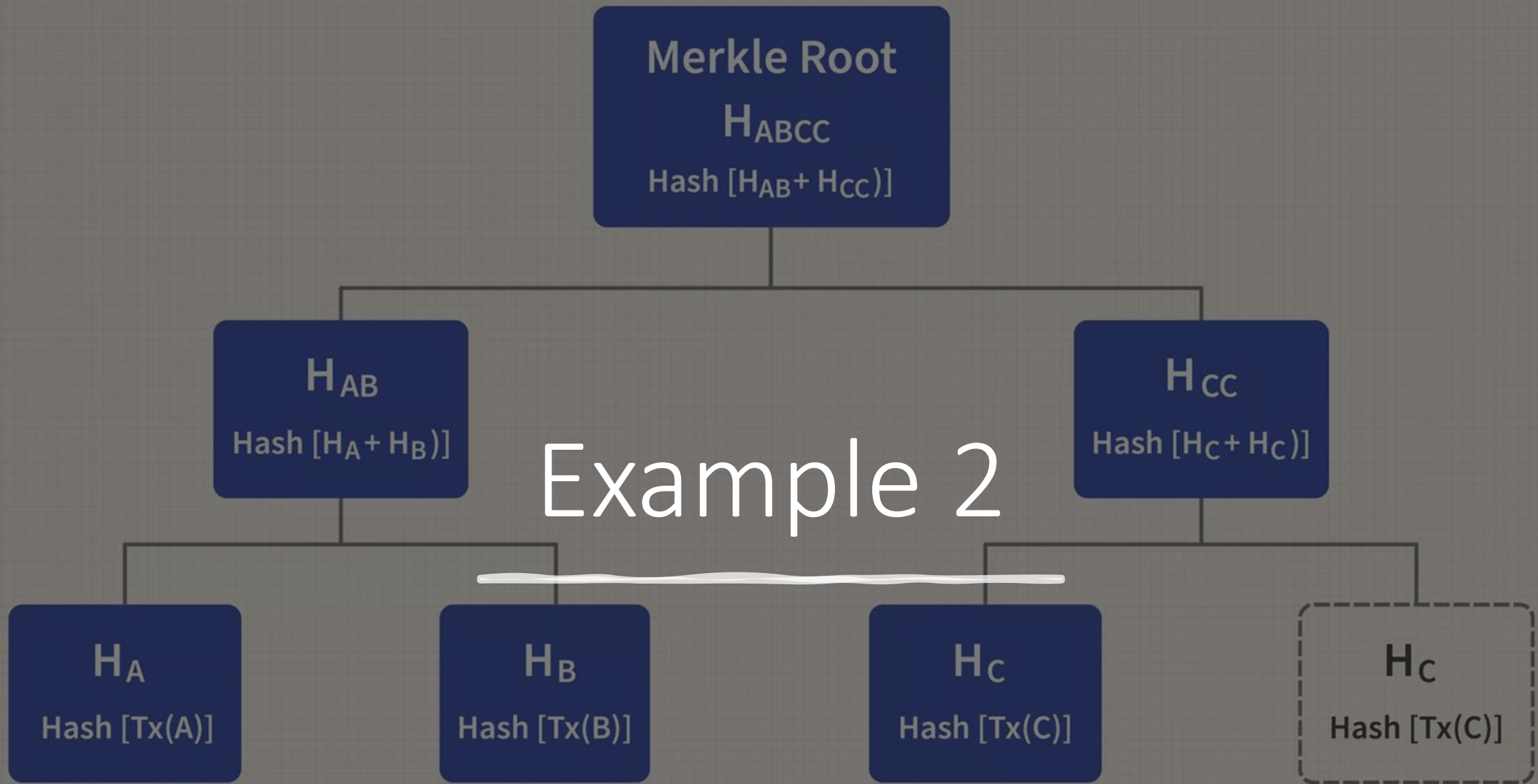
- Transactions are hashed together
- We concatenate each hashed transaction with its neighbor and hash again
- Keep repeating until we get a singular “root” hash
- If we have odd transactions just concatenate with itself



Example 1



Example 2



What a Block Looks Like

- Contains the software version
- Hash of the Previous Block
- Merkle Root of the Transactions
- Timestamp
- Difficulty Target
- Nonce
- The difficulty target determines how hard it is to find the right nonce

Version
Previous Block Hash
Merkle Root
Timestamp
Difficulty Target
Nonce

So What is Mining?

- Mining is trying nonces to hit the difficulty target
- GPUs are particularly fast at calculating the output from the hash, so can try more attempts faster with a good GPU
- The miner that hits that right nonce can broadcast it out
- Everyone else can check the nonce with ease as it is just one attempt of the hash to validate a nonce
- If majority agrees, we have a valid block (will get into consensus mechanisms deeper later but this is Bitcoin)

