UNIVERSITY *of* NICOSIA

Session:

Bitcoin and Digital Money

We are here

Session	Session Name	Professor
1	Introduction to Blockchain and Web3	George Giaglis
2	Bitcoin and Digital Money	Garrick Hileman
3	Ethereum and Programmable Blockchains	Apostolos Kourtis
4	Digital Wallet Security and Privacy	Charles Guillemet **(Ledger)**
5	Decentralized Finance (DeFi)	Lambis Dionysopoulos
6	Prediction Markets	Apostolos Kourtis & Lambis Dionysopoulos
7	The Geopolitics of Cryptocurrency	Garrick Hileman
8	Tokenization and Stablecoins	Lambis Dionysopoulos & Lauren Berta **(Ripple)**
Advanced Track & MSc Students		
9	NFTs and Digital Ownership	Punk6529 & Mohsen El-Sayed **(Ledger)**
10	Regulation and Policy of Digital Assets	Jeff Bandman
11	Advanced Topics in Web3	Apostolos Kourtis
12	The Future: Convergence of Blockchain, AI, and IoT	George Giaglis

Session Objectives

- Understand the **fundamentals of Bitcoin** and how it operates as a decentralized digital currency.
- Learn key cryptographic concepts such as **hash functions** and **public-private key cryptography**.
- Explore how Bitcoin transactions are **verified and recorded** on the blockchain.
- Analyze the significance of **Proof-of-Work** and the mining process in securing the Bitcoin network.
- Investigate the **Byzantine Generals' Problem** and how Bitcoin solves this challenge in a decentralized system.
- Examine advanced topics such as **Bitcoin address generation** and the mechanics of maintaining **blockchain synchronization**.

Agenda

1. Basics of Bitcoin cryptography (Algorithms, Transactions, and Addresses)
2. The Byzantine Generals' Problem
3. Proof-of-Work and Mining in Bitcoin
4. Unspent Transaction Output (UTXO) Model
5. Conclusions
6. Further Reading

Session: Bitcoin and Digital Money

1. Basics of Bitcoin Cryptography

First, Some Useful Definitions

- **bitcoin**: Without capitalization, is used to describe the currency and unit of account for the Bitcoin network.
- **Bitcoin**: With capitalization, is used to describe the concept, network, development project, and enthusiast community.
- **Bitcoin address**: A string of letters and numbers where bitcoin can be sent, similar to how one sends email to an email address.
- **transaction:** A record informing the network of a transfer of bitcoin from one bitcoin address to another.
- **blockchain***: The complete transaction ledger of the Bitcoin network, showing how bitcoin have been transferred from one address to another over time. The blockchain is a public record of all bitcoin transactions in chronological order.

Think of a Bitcoin transaction as a single value transfer recorded in a ledger, a block as a page of transactions from the last ten minutes, and a blockchain as the whole ledger book.

Bitcoin: Foundations and Cryptographic Technologies

Bitcoin is a collection of concepts and technologies that form the basis of a digital money ecosystem, including:

- A decentralized peer-to-peer network (enabled by the Bitcoin protocol)
- A public transaction ledger (the blockchain)
- A decentralized mathematical and deterministic currency issuance mechanism (distributed mining and the “Proof-of-Work” consensus algorithm)
- A decentralized transaction verification system (transaction script) (From “Mastering Bitcoin” Andreas Antonopoulos, 2014)

To achieve this, Bitcoin relies heavily on cryptographic technologies, such as:

- Hash functions (i.e. SHA-256 and RIPEMD-160)
- Public Key Cryptography (i.e. ECDSA – the Elliptic Curve Digital Signature Algorithm)

Bitcoin and Cryptography

A transaction is a record that informs the network of **a transfer of funds or value** from one owner to another.

- Think of a transaction as a **single line** on a page in a notebook.
- Think of a block as the **page** in that notebook.
- And think of the blockchain as being equivalent to the entire **notebook**.
- All the users are able to **read, write, and get updated** on what is written in this notebook.

Ownership of bitcoin is established through **the relationship** between public keys and the digital signatures produced from the corresponding private keys.

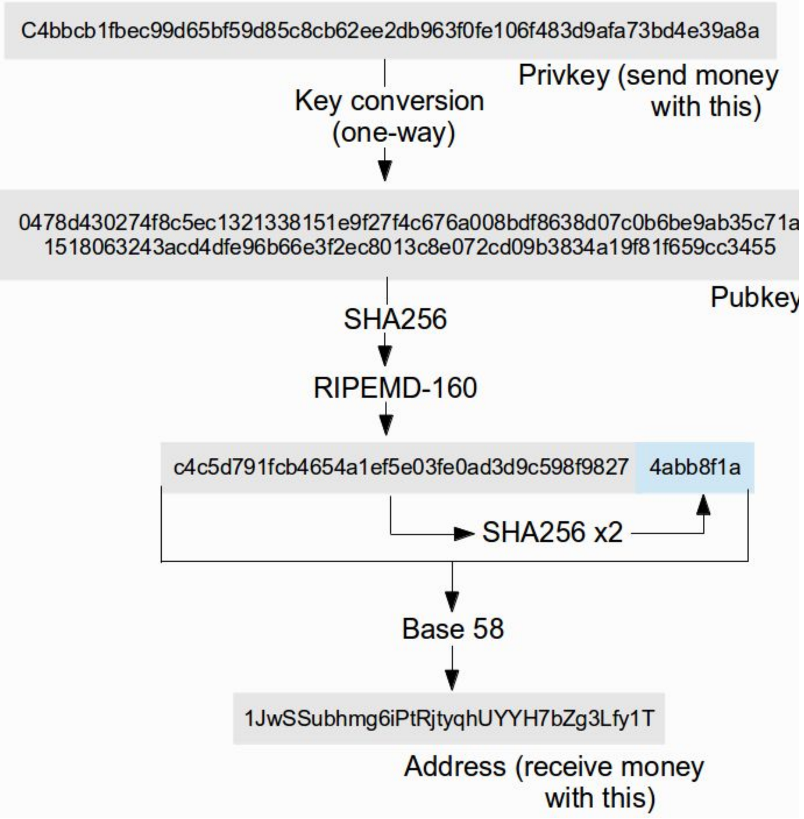
Bitcoin and Cryptography

Digital Keys: A mathematically-related public-private key pair, created using the Elliptic Curve Digital Signature Algorithm (ECDSA).

1. **Private key (Privkey)**: is essentially a randomly generated number that should be kept secret. It is used to generate digital signatures and confirm ownership, authorizing the spending of bitcoin.

1. **Public key (Pubkey):** Is generated from the private key using the elliptic curve multiplication process. When spending and receiving, the public key is represented by a bitcoin address.

"Think of the public key as similar to a bank account number and the private key as similar to the secret PIN, or signature on a check, that provides control over the account."

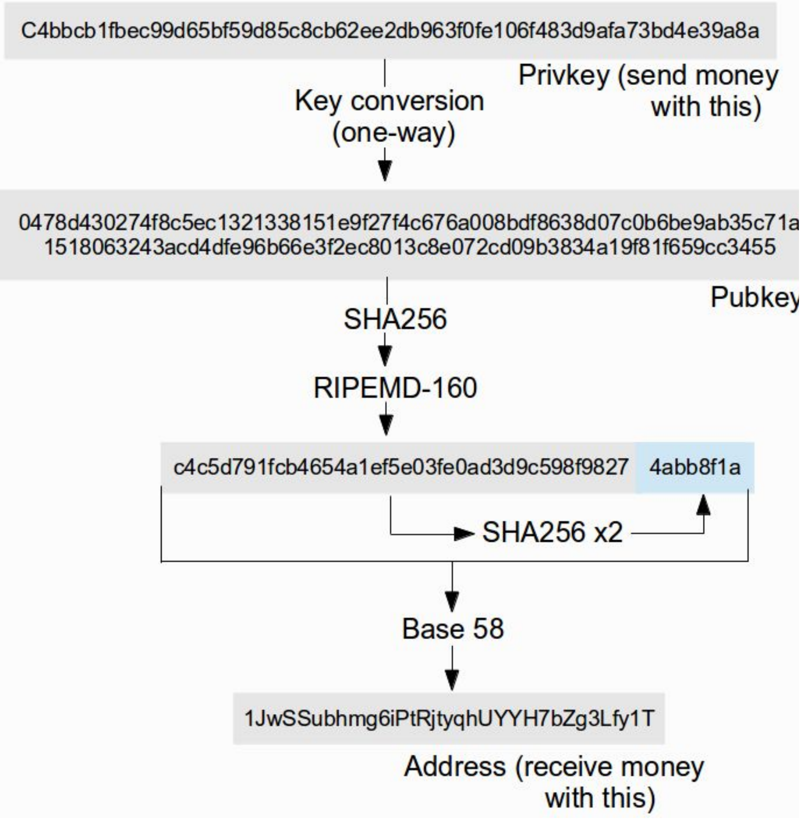

C4bbcb1fbec99d65bf59d85c8cb62ee2db963f0fe106f483d9afa73bd4e39a8a
Privkey (send money with this)
Key conversion (one-way)
0478d430274f8c5ec1321338151e9f27f4c676a008bdf8638d07c0b6be9ab35c71a
1518063243acd4dfe96b66e3f2ec8013c8e072cd09b3834a19f81f659cc3455
Pubkey
SHA256
RIPEMD-160
c4c5d791fcb4654a1ef5e03fe0ad3d9c598f9827
4abb8f1a
SHA256 x2
Base 58
1JwSSubhmg6iPtRjtyqhUYYH7bZg3Lfy1T
Address (receive money with this)

Generating a Bitcoin Address

Bitcoin and Cryptography

3. **Bitcoin Address:** An address is a unique identifier for the destination of a bitcoin payment, generated from and corresponding to a public key or script.
   - It is usually generated by applying the SHA-256 and RIPEMD-160 cryptographic hash functions (explained on slide 36), in series, on the public key.
   - These addresses are encoded using Base58 encoding, which represents an address in a human-readable form of 58 alphanumeric characters.

Fun fact: There are 52 characters in the alphabet, if we include all upper and lower-case letters. There are also 10 numbers (0 through 9). To avoid confusion and copying errors, Satoshi removed 4 commonly mistaken characters from the address generation process: uppercase letter 'O' and number '0,' uppercase letter 'I' and lowercase letter 'l.'


C4bbcb1fbec99d65bf59d85c8cb62ee2db963f0fe106f483d9afa73bd4e39a8a
Privkey (send money with this)
Key conversion (one-way)
0478d430274f8c5ec1321338151e9f27f4c676a008bdf8638d07c0b6be9ab35c71a
1518063243acd4dfe96b66e3f2ec8013c8e072cd09b3834a19f81f659cc3455
Pubkey
SHA256
RIPEMD-160
c4c5d791fcb4654a1ef5e03fe0ad3d9c598f9827
4abb8f1a
SHA256 x2
Base 58
1JwSSubhmg6iPtRjtyqhUYYH7bZg3Lfy1T
Address (receive money with this)

Generating a Bitcoin Address

Hash Functions

- A cryptographic hash function **transforms data** into a fixed-size digest, verifying data integrity. Any change in the input data results in a **completely different hash output**.
- Cryptographic hash functions are crucial in Bitcoin for **verifying transaction integrity, securing bitcoin addresses**, and in the Proof-of-Work (PoW) mining process.
- Bitcoin uses the **SHA-256** hash function, generating a 256-bit (32-byte) output.
- Hash functions ensure block integrity and establish the chronological order of the blockchain by referencing the hash of the previous block.
- Key properties of hash functions:
  - **Same input** always produces the **same hash**.
  - **Slight changes** in input generate a **completely different hash**.
  - Hash collisions (different inputs generating the same hash) should not occur.
  - Hashes are **difficult to reverse-engineer,** enhancing security for commitment schemes.

Input	Hashing algorithm: SHA-1
Fox	dfcd3454bbea788a751a696c24d97009ca992d17
fox	ff0f0a8b656f0b44c26933acd2e367b6c1211290
fox1	fcb9f413aa14b3fbec3c29d53dcf880994282874

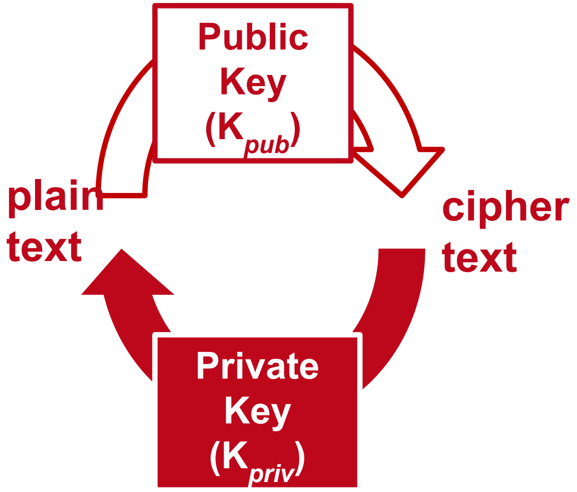
```
# sha256sum
Bitcoin
b4056df6691f8dc72e56302ddad345d65
fead3ead9299609a826e2344eb63aa4
```

Hash Functions

- Public key cryptography, also known as asymmetric cryptography, is used in Bitcoin primarily for digital signatures, not for encrypting transaction data (since transaction details are publicly visible on the blockchain).
- Public key cryptography operates as a two key system:
  - The **public key** is used to **encrypt** a message.
  - The **private key** is used **decrypt** the message.
- **Asymmetric encryption** means the **public key** can be easily derived from the **private key**, but the private key is nearly impossible to derive from the public key.
- In Bitcoin, the **Elliptic Curve Digital Signature Algorithm (ECDSA)** is used to generate digital signatures that authenticate transactions, ensuring that only the holder of the private key can authorize the spending of Bitcoin.
  - The private key **signs the transaction**, authorizing the spending of Bitcoin.
  - The public key (revealed after spending) **verifies** the transaction's authenticity.

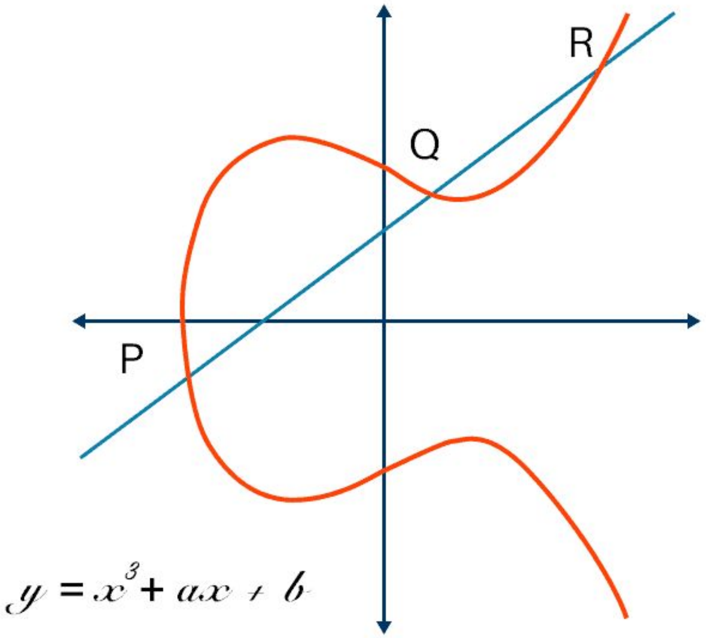
$\mathbf{C}$ = encrypt(M, $\mathbf{K}_{pub}$)

$\mathbf{M}$ = decrypt(C, $\mathbf{K}_{priv}$)



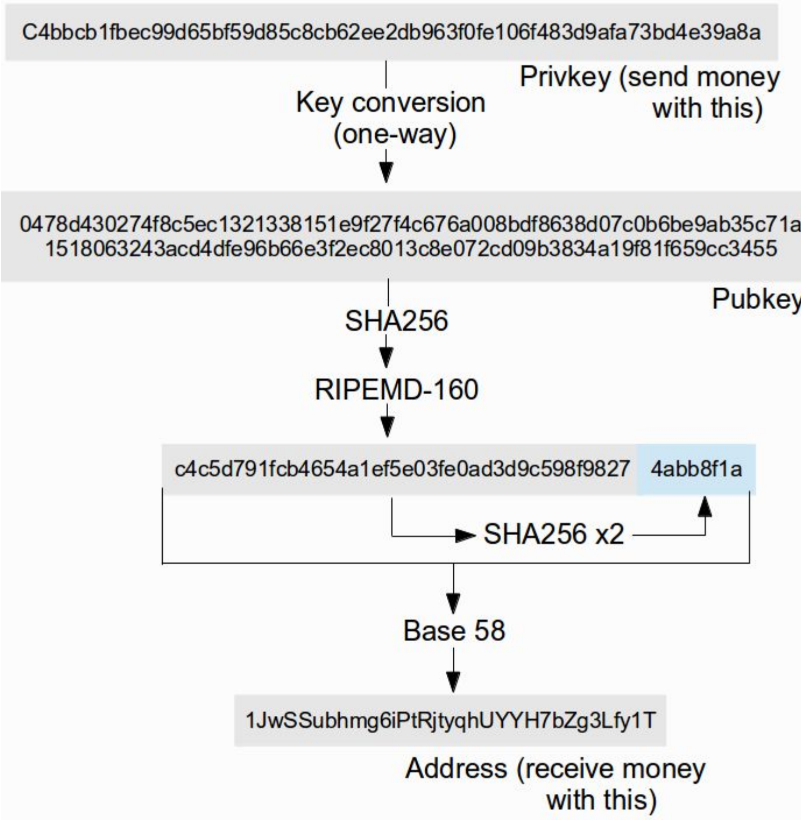
Digital Signatures

- Digital signatures are used to authenticate Bitcoin transactions by **ensuring only the owner of the private key can authorize spending**.
- When making a Bitcoin payment, **a transaction (T) is constructed**, and **a subset (M)** of the transaction data **is signed** using the private key, **generating values R and S**.
- To verify the transaction, **the public key (Kpub)** and the **signature** are used to **calculate a point (P) on the elliptic curve**. If the x-coordinate of P equals R, the signature is valid without revealing the private key.
- Each transaction transfers bitcoin from the current owner's address to the new owner’s address, authorized by a digital signature.
- The Bitcoin network **tracks the entire transaction history**, allowing anyone to verify ownership of bitcoin without access to private keys.
- **Bitcoin wallets store the private-public key pairs**, which are used to access and manage funds, though the wallet itself does not contain any actual bitcoin.



Transactions

- Bitcoin operates over a **peer-to-peer (P2P) network**, where each computer connected is called a **node**, and anyone can run a node by downloading the open-source software.
- **Full nodes** verify and maintain the **complete transaction history** according to consensus rules, while **mining nodes** also process transactions and add new blocks to the blockchain.
- Transactions are recorded in a **public ledger** (the blockchain) that tracks transfers of bitcoin ownership from one address to another.
- A **Bitcoin address** is derived from a user's **public key** using a combination of **SHA-256 and RIPEMD-160 hash functions**, ensuring security and privacy.
- A Bitcoin **transaction** includes **inputs (debits) and outputs (credits).** The difference between them is the transaction fee, which is rewarded to the miner who processes the block.
- Digital signatures are used to **prove ownership** of the bitcoin being transferred, and any node in the network can validate these signatures independently.


C4bbcb1fbec99d65bf59d85c8cb62ee2db963f0fe106f483d9afa73bd4e39a8a
Privkey (send money with this)
Key conversion (one-way)
0478d430274f8c5ec1321338151e9f27f4c676a008bdf8638d07c0b6be9ab35c71a 1518063243acd4dfe96b66e3f2ec8013c8e072cd09b3834a19f81f659cc3455
Pubkey
SHA256
RIPEMD-160
c4c5d791fcb4654a1ef5e03fe0ad3d9c598f9827
4abb8f1a
SHA256 x2
Base 58
1JwSSubhmg6iPtRjtyqhUYYH7bZg3Lfy1T
Address (receive money with this)

Generating a Bitcoin Address

Session: Bitcoin and Digital Money

2. The Byzantine Generals’ Problem

What is the Byzantine Generals’ Problem (BGP)?

The problem of trust in a **distributed system**, with no central control to enforce rules, is not a new one in computer science. The components may fail to reach consensus due to technical failures or misinformation.

“We imagine that several divisions of the Byzantine army are camped outside an enemy city, each division commanded by its own general. The generals can communicate with one another only by messenger. After observing the enemy, they must decide upon a common plan of action. However, some of the generals may be traitors, trying to prevent the loyal generals from reaching agreement. The generals must have an algorithm to guarantee that (A) All loyal generals decide upon the same plan of action and (B) A small number of traitors cannot cause the loyal generals to adopt a bad plan”

– The Byzantine Generals Problem, 1982

Source: The Byzantine Generals’ Problem, Lamport, Shostak, Pease, 1982 Image Source: Wikimedia Commons.

The Byzantine Generals’ Problem

The Byzantine Generals' Problem (BGP) was first proposed by Marshall Pease, Robert Shostak and Leslie Lamport in 1982, "expressed abstractly in terms of a group of generals of the Byzantine army camped with their troops around an enemy city."

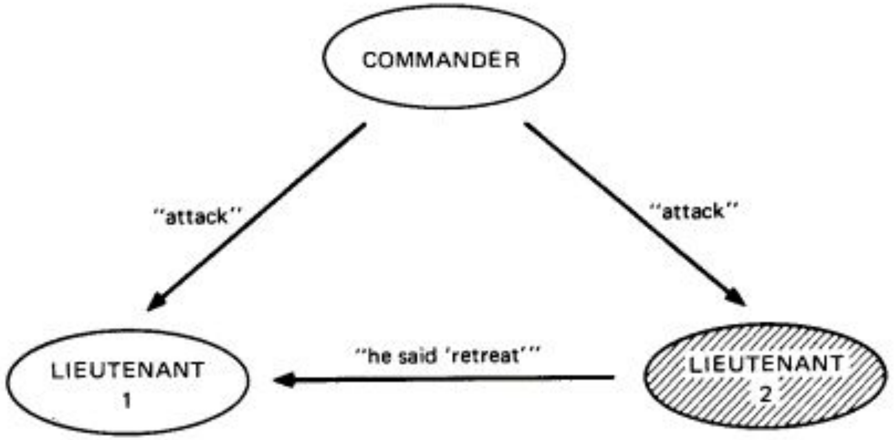


Fig. 1. Lieutenant 2 a traitor.

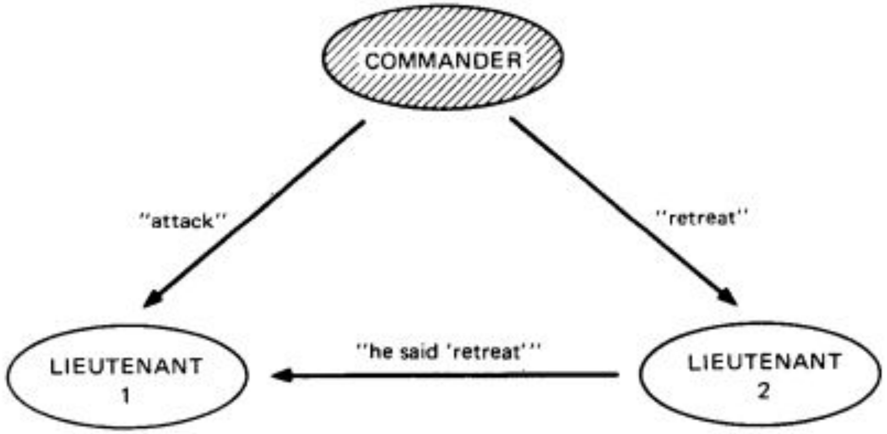


Fig. 2. The commander a traitor.

This is important to understanding why Bitcoin and other open blockchains are designed the way they are!

In this scenario, a traitor (either the Commander or Lieutenant) prevents the group from reaching consensus. In a financial ledger, think of the traitor as a malicious party that aims to facilitate fraudulent transactions.

Source: https://people.eecs.berkeley.edu/~kubitron/cs262/lectures/lec19-Byzantine.pdf The Byzantine Generals’ Problem, Lamport, Shostak, Pease, 1982

Problem Formulation

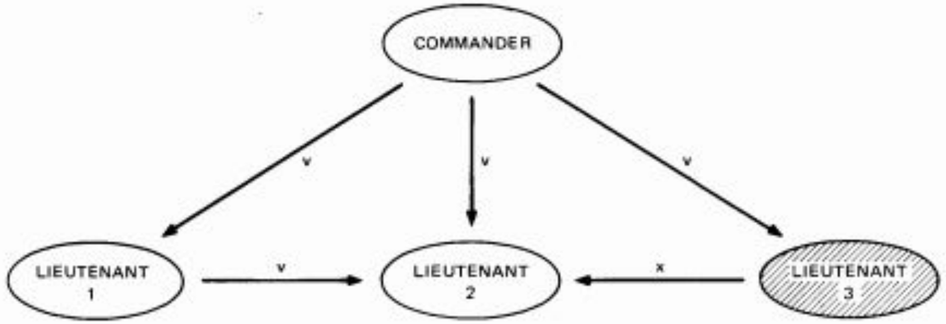


Fig. 3. Algorithm OM(1); Lieutenant 3 a traitor.

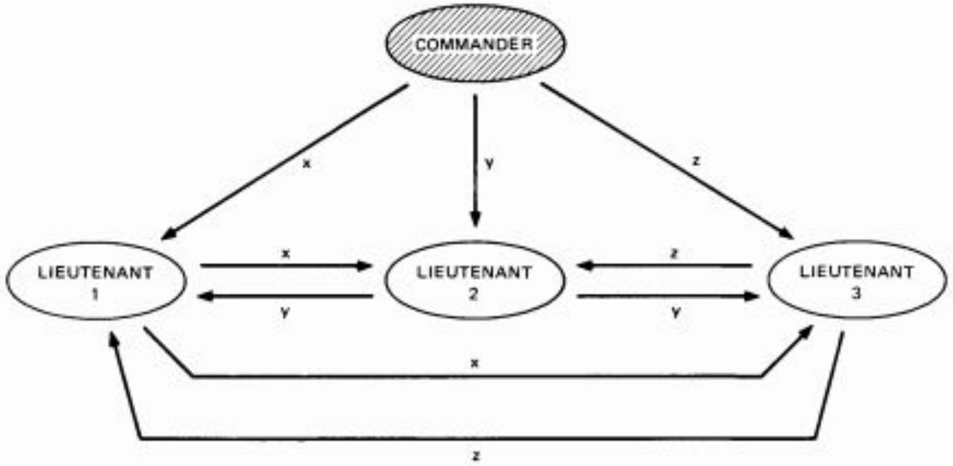


Fig. 4. Algorithm OM(1); the commander a traitor.

As the number of parties in the system grows, the number of communication channels -and potential points of mistrust- increases exponentially.

Now, imagine the complexity of achieving consensus when thousands or even millions of parties are involved.

Source: The Byzantine Generals' Problem, Lamport, Shostak, Pease, 1982

Past Attempts to Solve the Byzantine Generals Problem

Past attempts at solving this problem in digital currencies include the following:

- Chaum, D., 1984. Blind Signature System, in: Chaum, D. (Ed.), Advances in Cryptology. Springer US, pp. 153–153.
- Chaum, D., Fiat, A., Naor, M., 1990. Untraceable Electronic Cash, in: Proceedings of the 8th Annual International Cryptology Conference on Advances in Cryptology, CRYPTO ’88. Springer-Verlag, London, UK, UK, pp. 319–327.
- Okamoto, T., Ohta, K., 1992. Universal Electronic Cash, in: Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology, CRYPTO ’91. Springer-Verlag, London, UK, UK, pp. 324–337.
- Wei Dai’s B-Money Wei Dai, 1998, http://www.weidai.com/bmoney.txt

Bitcoin was first proposed on October 31st 2008, by an individual or group under the pseudonym “Satoshi Nakamoto.” It is the best solution to date and has had – by far – the broadest adoption.

This is a significant milestone for digital currencies.

Session: Bitcoin and Digital Money

3. Proof-of-Work and Mining in Bitcoin

The Bitcoin Ledger: The Blockchain

- Want to witness and store the Bitcoin blockchain? The starting point:
  1. A Bitcoin user **downloads a piece of software** (the Bitcoin reference client software)
  2. This client software will **initially download the blockchain**, the ledger of all transactions in the history of Bitcoin.
  3. Each Bitcoin full client **stores the complete record of all bitcoin transactions from all time**. There is no central record-keeper, just a set of copies distributed among all the clients.

- Once the blockchain history is downloaded and validated, the issue of **synchronization** emerges: How are these copies of the blockchain (ledger) kept in sync with each other?
  - In other words, how do they reach **distributed consensus** without a definitive central party?
  - When a client receives conflicting messages about a transaction, which one should it accept and which one should it ignore? Which one is truthful, which one is a traitor?

- By now, you should realize that **keeping the blockchain copies in sync is a manifestation of the Byzantine Generals' Problem**.

- Keep in mind that downloading the Bitcoin client software **is not mandatory** in order to use bitcoin. It is recommended for users wishing to **validate** their transactions and participate in the consensus process.

Synching the Blockchain: Mechanics

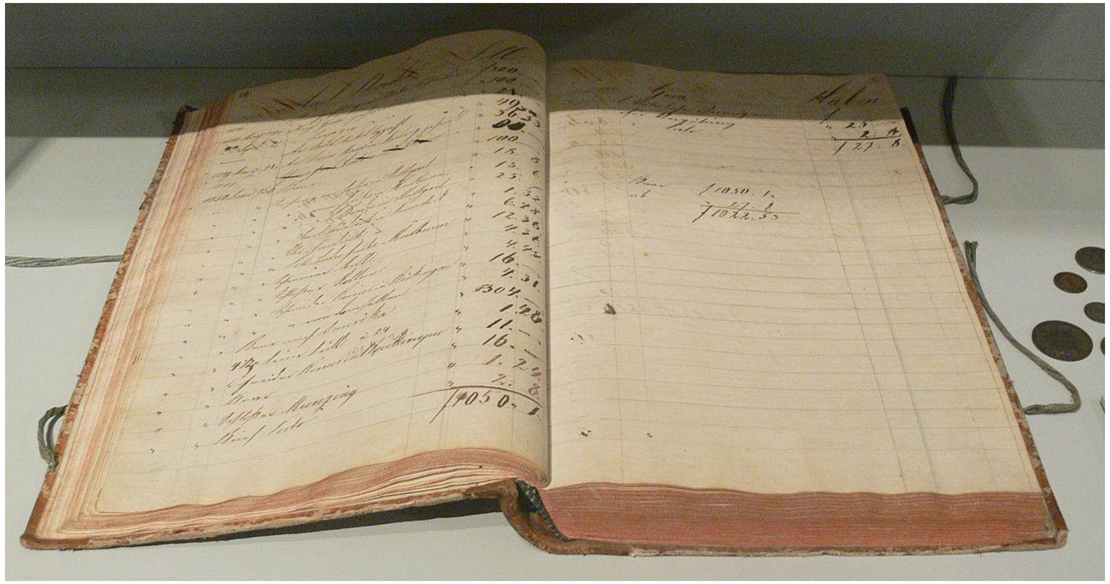
1. When a user executes a transaction (sends bitcoin from one address to another), the transaction is **eventually broadcast to all network nodes in the system**. Within a few seconds, most of the clients in the world see the transaction.

1. At this point, however, the transaction is considered **“unconfirmed.”** Remember the **Byzantine Generals’ Problem**: what if a **dishonest** Bitcoin client sent out two transactions moving the same bitcoin to two different addresses? Which one should the clients accept?

1. Bitcoin confirms transactions and resolves the BGP through a process called **“mining.”**

Read more about mining in Chapter 10 of *Mastering Bitcoin* by Andreas M. Antonopoulos

Synching the Blockchain: Mining



Looks more like this...

Not this.

Mining is a rather misleading analogy for what ‘miners’ do.

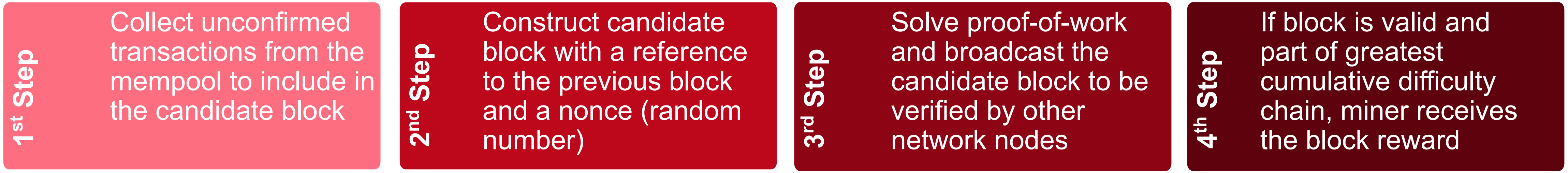
Think of the miners as 'bookkeepers' and it will make much more sense.

Image Source: Wikimedia Commons. Ledger and Coal Strip Mine

Mining and Proof-of-Work

Mining:

- The process by which new blocks are appended to the blockchain, and new units of bitcoin are "minted" according to a deterministic issuance schedule. The total supply is finite, limited to approximately 21 million bitcoin.
- The integrity of transactions and blocks is ensured through a contribution of computational power, e.g. proof-of-work. The candidate block data is repeatedly hashed until it is less than the value of a desired pattern according to the current difficulty rate. (Note: A hash's value is 'less' if it has more leading zeroes.)
- If the process seems complicated, do not worry! It is automated. The setup and maintenance of mining machines is often the only manual work involved.



1st Step: Collect unconfirmed transactions from the mempool to include in the candidate block

2nd Step: Construct candidate block with a reference to the previous block and a nonce (random number)

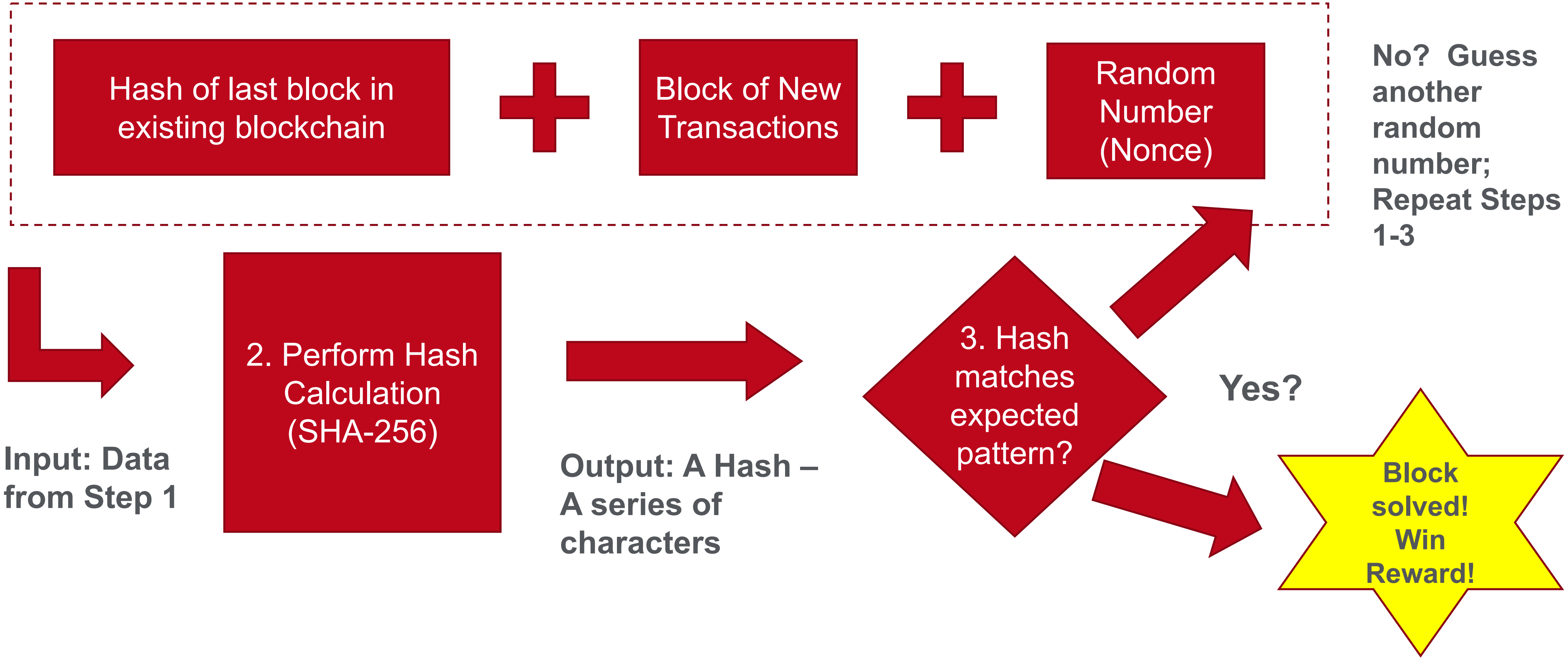
3rd Step: Solve proof-of-work and broadcast the candidate block to be verified by other network nodes

4th Step: If block is valid and part of greatest cumulative difficulty chain, miner receives the block reward

https://github.com/bitcoinbook/bitcoinbook/blob/develop/ch10.asciidoc. https://www.coindesk.com/information/how-to-set-up-a-miner/

Mining in 3 Steps

1. Compile Some Data To Be The Input To A Calculation


Hash of last block in existing blockchain
Block of New Transactions
Random Number (Nonce)
No? Guess another random number; Repeat Steps 1-3
Input: Data from Step 1
2. Perform Hash Calculation (SHA-256)
Output: A Hash – A series of characters
3. Hash matches expected pattern?
Yes?
Block solved! Win Reward!

Mining: Winning a Prize?

Once a miner has a winning block, it **broadcasts** it to the other clients:

- The client nodes verify that the hash matches the expected pattern and accept the new block, adding it to their own copy of the blockchain. Note: Blockchain = a chain of blocks (!)
- After that, all miners start working on finding the next block, incorporating the new previous block hash as their starting point in Step 1

The miner is allowed to collect as part of having a winning block:

- The current block subsidy, which increases the circulating supply of bitcoin
- The fees from all the transactions that were included in the block

The block reward started at 50 bitcoin per block and halves every 210,000 blocks, about every 4 years.

New block rewards, except transaction fees, will stop once the network reaches Block 6,930,000 (sometime around the year 2140). The total number of bitcoin issued by then will be about 21 million https://en.bitcoin.it/wiki/Controlled_supply

Mining: Auto-Adjusting Puzzles

This winning of prizes sounds very tempting, but why hasn't a powerful computer mined all the Bitcoin yet?

Fortunately, the mining difficulty **automatically adjusts** to match the amount of computing power in the Bitcoin network, ensuring that new blocks are mined at a consistent rate:

- Difficulty adjustments occur every 2,016 blocks, or approximately every two weeks.
- If the average block time over this period exceeds 10 minutes (i.e., “too hard”), the difficulty is decreased.
- If the average block time is less than 10 minutes (i.e., “too easy”), the difficulty is increased.

Regardless of the mining power in the network, new blocks are still mined approximately every 10 minutes, keeping the Bitcoin issuance rate stable.

Bitcoin Mining Today

- **Dominated by ASIC Miners:** Bitcoin mining is now primarily performed by large-scale operations using ASIC (Application-Specific Integrated Circuit) miners. These machines are highly specialized and designed for optimal performance and energy efficiency.
- **Increasing Hardware Efficiency:** The hardware used in Bitcoin mining continues to evolve. Modern ASIC miners are much more energy-efficient than earlier models, drastically improving performance and reducing operational costs. These machines are **optimized for high hash rates** and minimal energy consumption.
- **Adapting to Hardware Evolution:** As ASIC models evolve and their lifespan extends, miners have more opportunities to optimize operations with older models, enhancing profitability and operational efficiency in an increasingly competitive landscape.
- **Energy Considerations:** Mining profitability is heavily dependent on energy costs, with a growing emphasis on using renewable energy sources to reduce both costs and environmental impact.
- **Challenges for Home Mining:** Solo mining at home is no longer profitable for most people due to the high costs of energy and competition from large mining farms. Home miners now typically rely on cheap or renewable energy to remain competitive.

For a recent exploration of the profitability of bitcoin mining see:
https://www.bitcoinmagazinepro.com/blog/is-bitcoin-mining-profitable-a-deep-dive-into-costs-and-rewards/

Solo and Pool Mining

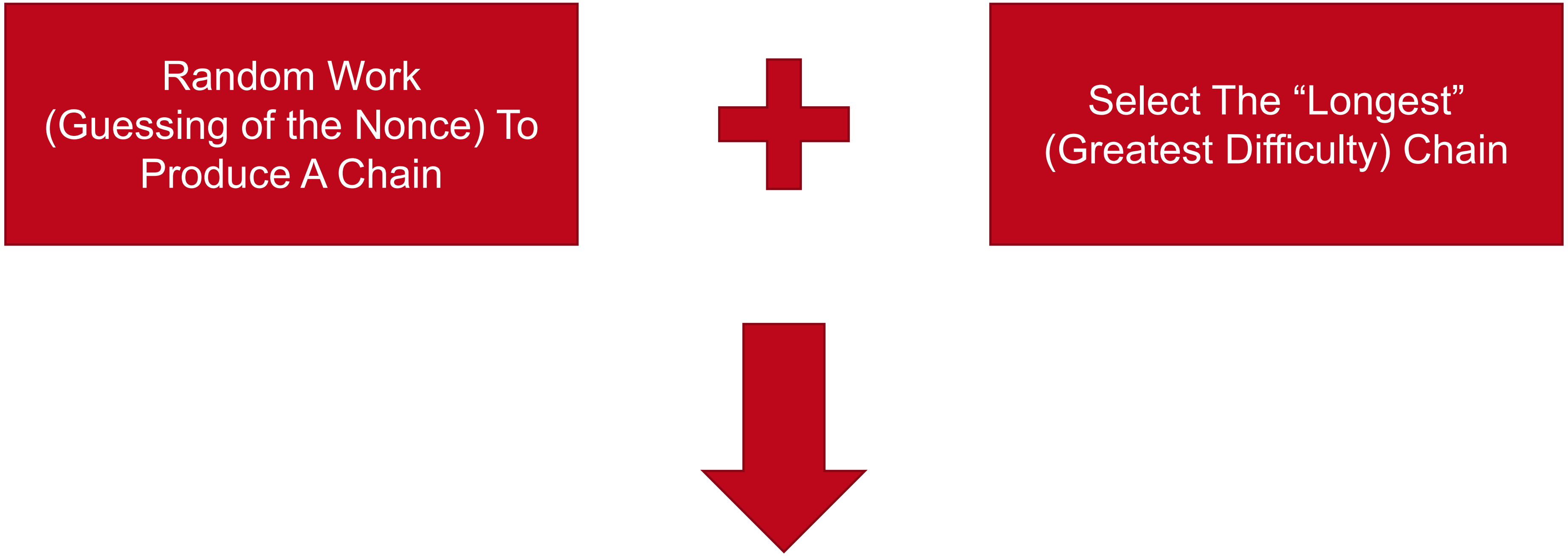
- **Solo mining:** In solo mining, individuals use their own hardware to independently search for and solve blocks. Payment is only received if the miner successfully solves a block, making it highly competitive and unlikely for individuals without significant computational resources.
- **Pool mining:** Most miners today participate in mining pools, where multiple miners combine their hash power to solve blocks collaboratively. When a block is solved, rewards are distributed among miners based on their contribution to the pool's hash power.
- Pool mining **increases the likelihood of earning consistent rewards**, as miners receive a share of the block reward relative to their contribution, rather than waiting for a solo success.
- Pool mining algorithms distribute rewards based on methods such as:
  - **PPS (Pay Per Share):** Miners receive a fixed reward for every valid share submitted.
  - **PPLNS (Pay Per Last N Shares):** Miners are paid based on their contribution to the last set of shares after a block is found.
  - **Proportional:** Miners are paid based on the proportion of shares they contributed since the last block was solved.

- **Solo mining is less viable for individuals** due to Bitcoin's high hash rate, while **pool mining allows smaller miners to contribute and earn more regularly**.

Back to the Byzantine Generals’ Problem

- “But, you have not yet solved the BGP, just moved it to the miners. **What if two miners send out blocks with different information** (i.e. different transactions within the block)? How do the clients choose **which one to include?**”
- The answer is that when a client is trying to decide which block history to accept, it must choose the one that is not only the **“longest” (in the number of blocks)**, but the one that has the **“greatest cumulative difficulty”** (total proof-of-work used to create it). In other words, the chain that took the **most computation power** to build.
    - Blocks that are **invalid**, or a version of the chain that has **less cumulative proof-of-work**, will become **“orphaned,”** and those transactions will need to be reprocessed.
- Given this system, **a traitor/dishonest node cannot keep broadcasting bad signals into the Bitcoin network**, such as attempting to double-spend by including a transaction in one block and then erasing it in the next. Unless he or she controls a **significant majority of the hashing power** and can sustain that control, which would be very hard to do.
- One useful mental model of the block reward scheme is a **lottery**. The miner who “wins,” is a matter of probabilities. This prevents any one party from taking control.

Back to the Byzantine Generals’ Problem



Random Work (Guessing of the Nonce) To Produce A Chain

Select The “Longest” (Greatest Difficulty) Chain

Solution to the Byzantine Generals' Problem

Session: Bitcoin and Digital Money

4. Unspent Transaction Output (UTXO) Model

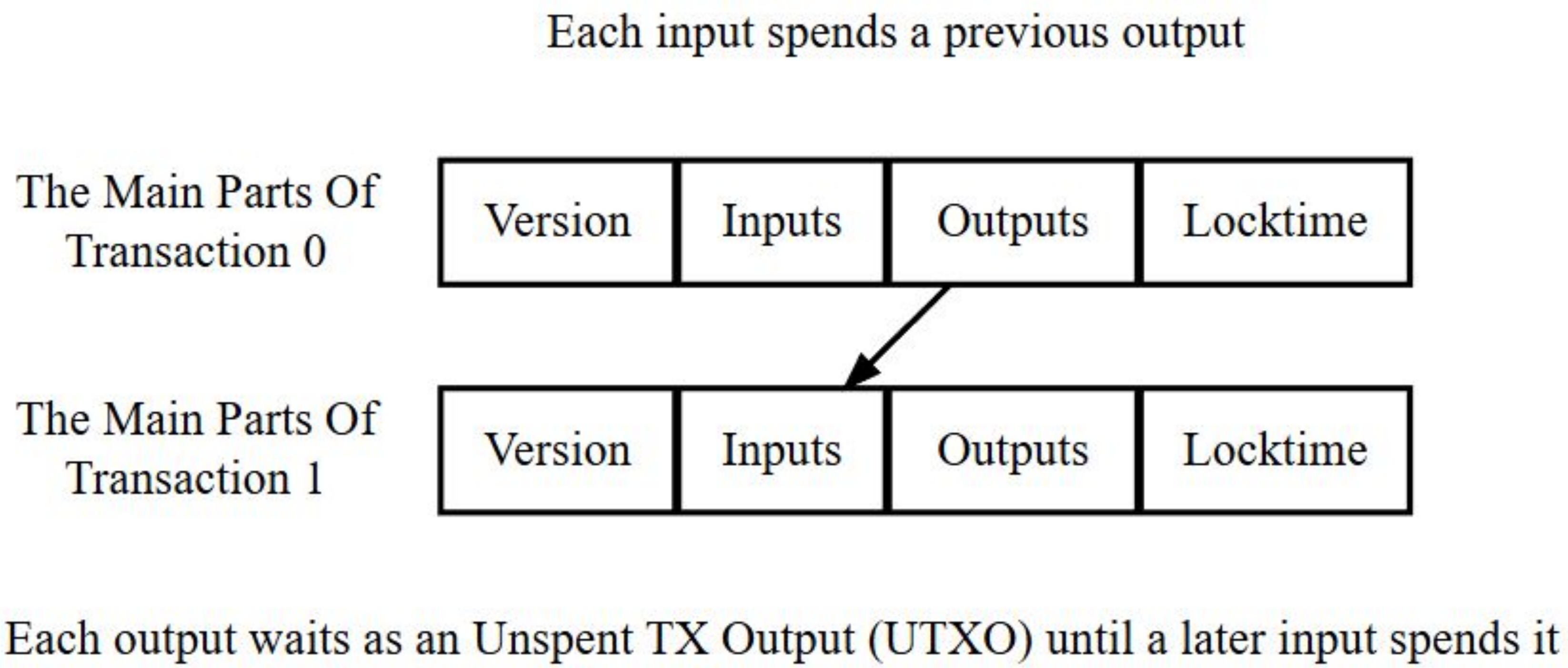
UTXO Model: Core Concepts

Bitcoin uses the Unspent Transaction Output (UTXO) model to represent and process transactions

- Each transaction consumes **one or more existing UTXOs as inputs** and **creates new UTXOs as outputs**
- A UTXO represents a specific amount of **satoshis** locked by a spending condition (script)
- Inputs reference UTXOs by transaction ID and output index
- Wallet balances are derived by summing all UTXOs that the wallet is able to spend
- Once a UTXO is spent, **it is permanently removed from the UTXO set and cannot be reused**
- Transactions **must spend the entire value of their input UTXOs**, with any excess returned to the sender as change

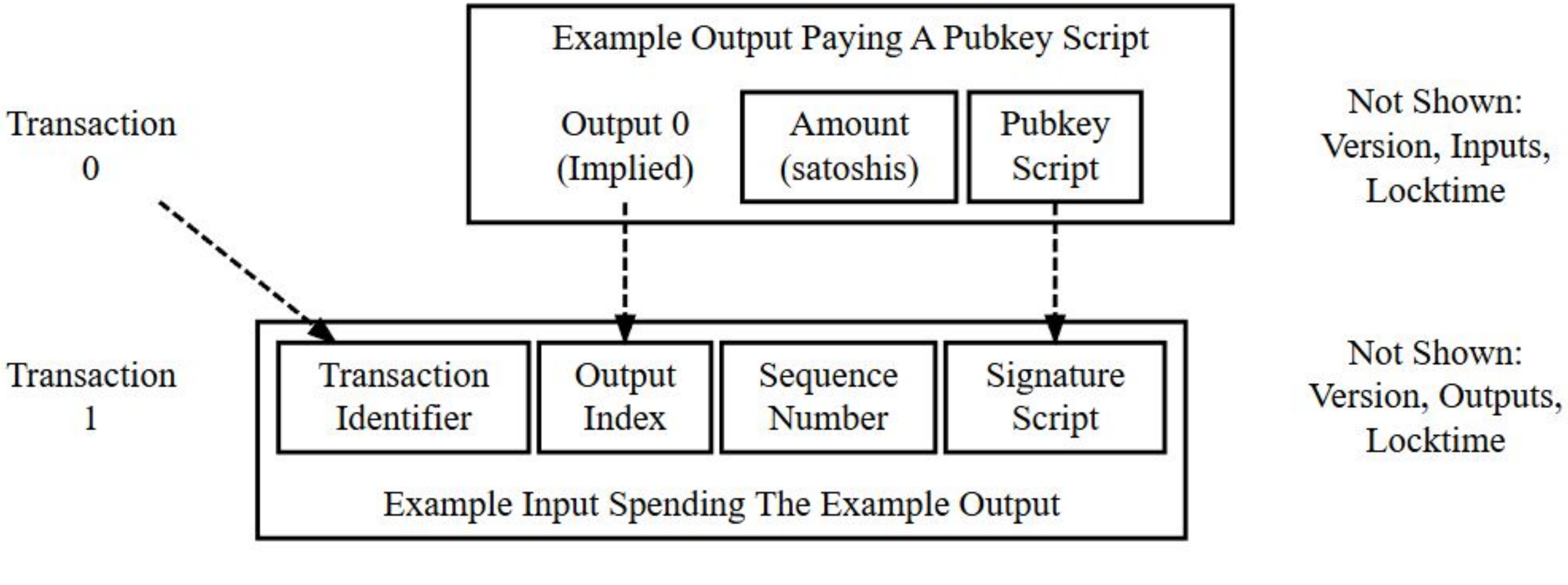
Transaction fees = total inputs − total outputs

The Parts Of A Transaction (bitcoin.org)


Each input spends a previous output
The Main Parts Of Transaction 0
Version
Inputs
Outputs
Locktime
The Main Parts Of Transaction 1
Version
Inputs
Outputs
Locktime
Each output waits as an Unspent TX Output (UTXO) until a later input spends it

Source: https://developer.bitcoin.org/devguide/transactions.html

Spending An Output (bitcoin.org)


Example Output Paying A Pubkey Script
Transaction 0
Output 0 (Implied)
Amount (satoshis)
Pubkey Script
Not Shown: Version, Inputs, Locktime
Transaction 1
Transaction Identifier
Output Index
Sequence Number
Signature Script
Not Shown: Version, Outputs, Locktime
Example Input Spending The Example Output

Overview Of Transaction Spending

Source: https://developer.bitcoin.org/devguide/transactions.html

Session: Bitcoin and Digital Money

5. Conclusions

Conclusions

- Bitcoin combines cryptographic technologies and decentralized networks to create a peer-to-peer digital currency system.
- Public key cryptography ensures secure ownership and transfer of Bitcoin, while digital signatures verify transactions.
- The blockchain serves as a public ledger, maintaining the integrity and history of all transactions.
- Mining is essential for adding new blocks to the blockchain and relies on solving complex Proof-of-Work puzzles.
- Bitcoin's solution to the Byzantine Generals' Problem ensures trust and consensus in a decentralized environment.

Session: Bitcoin and Digital Money

5. Further Reading

Further Reading

- How the Byzantine General Sacked the Castle: A Look Into Blockchain
- Bitcoin Mining
- The Byzantine Generals’ Problem (Leslie Lamport, Robert Shostak, Marshall Pease) (the first paper that defined the Byzantine Generals ‘ Problem in those terms)
- Majority is not Enough: Bitcoin Mining is Vulnerable
- The Economics of Cryptocurrencies – Bitcoin and Beyond
- Bitcoin Mining, CoinDesk
- Decentralizing money: Bitcoin prices and blockchain security

UNIVERSITY *of* NICOSIA

Questions?

Contact Us:

Twitter: @mscdigital
Course Support:
digitalcurrency@unic.ac.cy
IT & Live Session Support: dl.it@unic.ac.cy