Lesson 1

Introduction

Course Plan

Week 1 - Introduction to Blockchain, Solana and Rust
Week 2 - Rust / dev tools / token program
Week 3 - Anchor framework
Week 4 - Solana Program Library / Security

Practical Details

All lessons will be conducted online.
The usual format will usually be 45 mins of theory followed by 45 mins practical
You will be able to work in teams

How to ask questions ?

We have channels for questions

- Sli.do : link
- Discord channel

Put your questions in channels rather than asking individuals

How do practicals work ?

The lessons will typically be split 50/50 into theory and practical

During the practical half of the lesson you can work on exercises and ask questions in the support channel

You do not need to submit the homeworks, we suggest you put your answers into a repo.
We will review the exercises once most students have finished them

---

About us

Free Developer Workshops

- Basic
- Enterprise
- Advanced EVM
- Zero Knowledge Proofs

Business Workshops

Website : https://extropy.io

Email : info@extropy.io

Twitter : @extropy

Decentralised Systems

Problems with centralised systems

Monetary System

- Bank closure / insufficient capital reserves
- greek debt crisis in 2015 ? banks closed and people lost savings, insurance schemes meant nothing, lead to an increase in Bitcoin use in Greece
- Availability of banks
- Inflation - money supply controlled by central authority
- Merchant accounts may be shut down
- Control of money for political reasons - wikileaks funding shutdown

There are layers of access control built into our banking systems to prevent fraudulent transactions, effectively security is achieved by closing the network.

Goals of decentralisation

- Participation
- Diversity
- Conflict resolution
- Flexibility
- Moving power to the edge (user)

Introduction to Blockchain

Gossip network

**

Shared public ledger

Cryptography

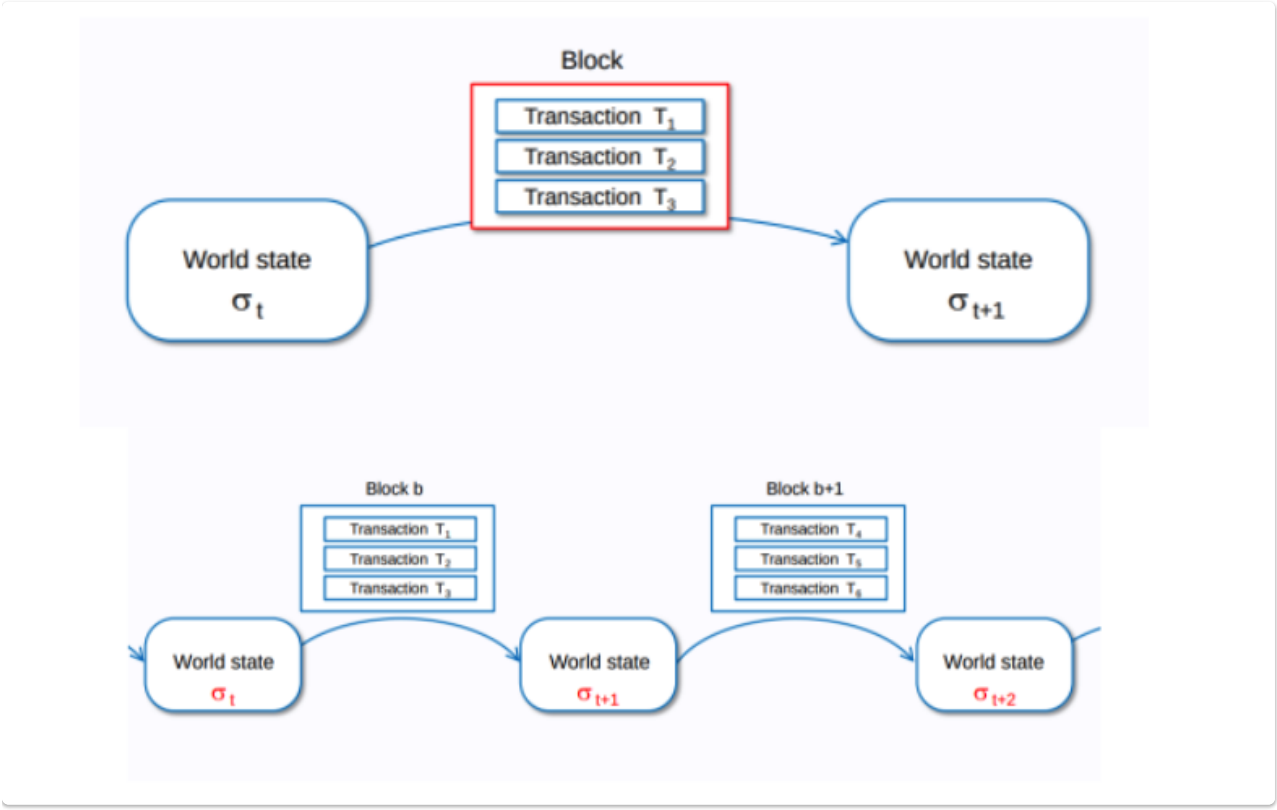
These components give the blockchain

- Transparency and verifiable state based on consensus
- Resilience
- Censorship resistance
- Tamper proof interactions

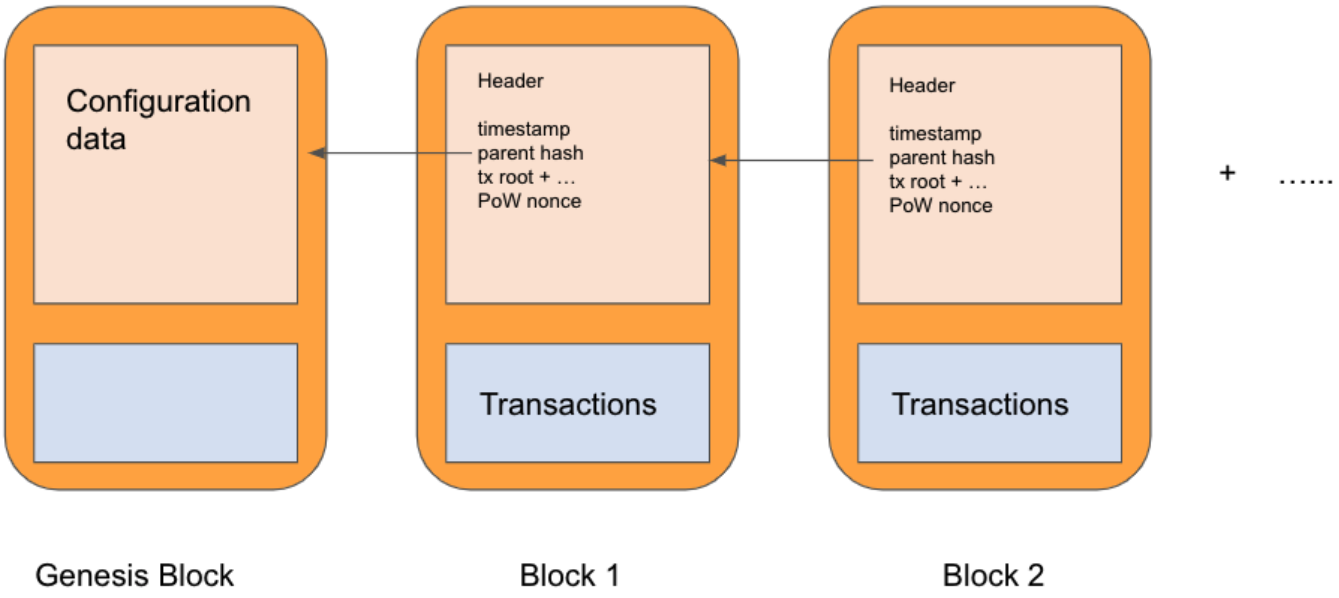
Blockchain components in more detail

- A peer-to-peer (P2P) network connecting participants and propagating transactions and blocks of verified transactions, based on a standardized "gossip" protocol
- Messages, in the form of transactions, representing state transitions
- A set of consensus rules, governing what constitutes a transaction and what makes for a valid state transition
- A state machine that processes transactions according to the consensus rules
- A chain of cryptographically secured blocks that acts as a journal of all the verified and accepted state transitions
- A consensus algorithm that decentralizes control over the blockchain, by forcing participants to cooperate in the enforcement of the consensus rules
- A game-theoretically sound incentivization scheme to economically secure the state machine in an open environment
- One or more open source software implementations of the above ("clients")

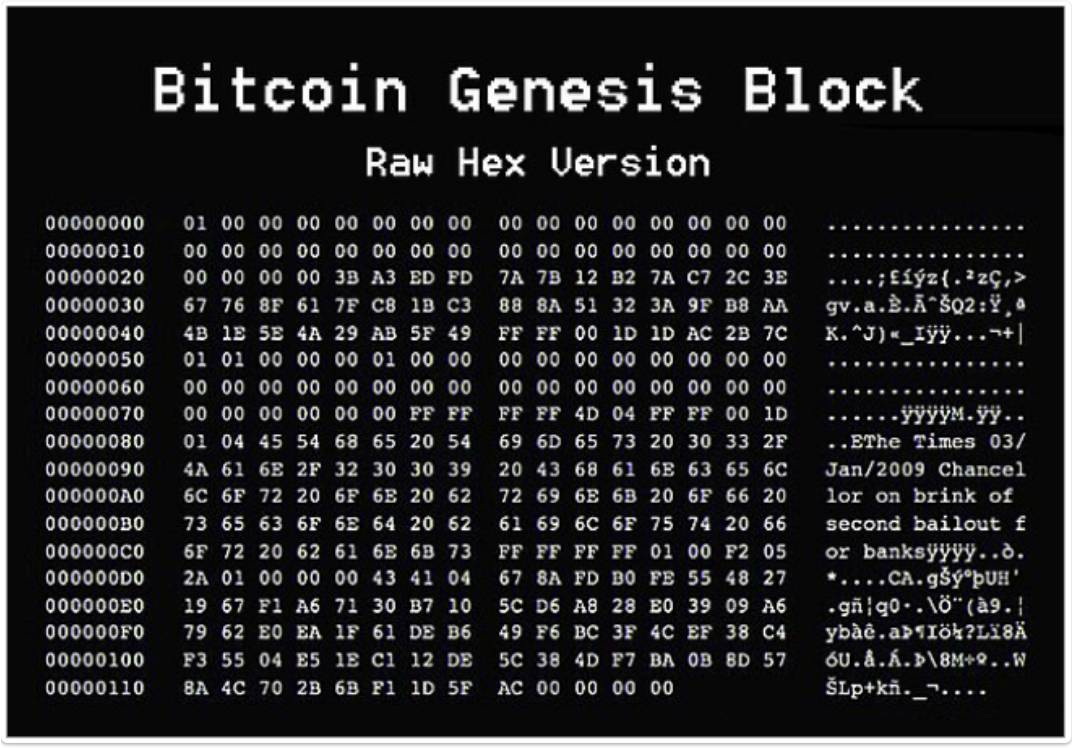
Blockchain as a state machine in Bitcoin



General Blockchain Structure for example Bitcoin



Genesis Block - the starting block



Blockchain Timeline

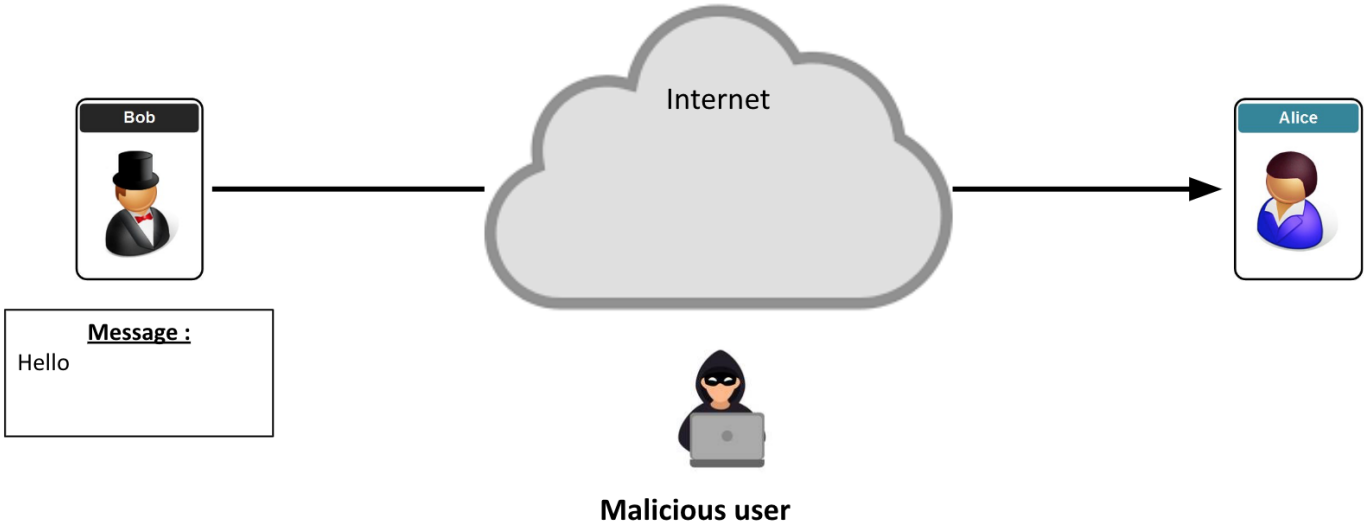
1970s

Problem = Security !

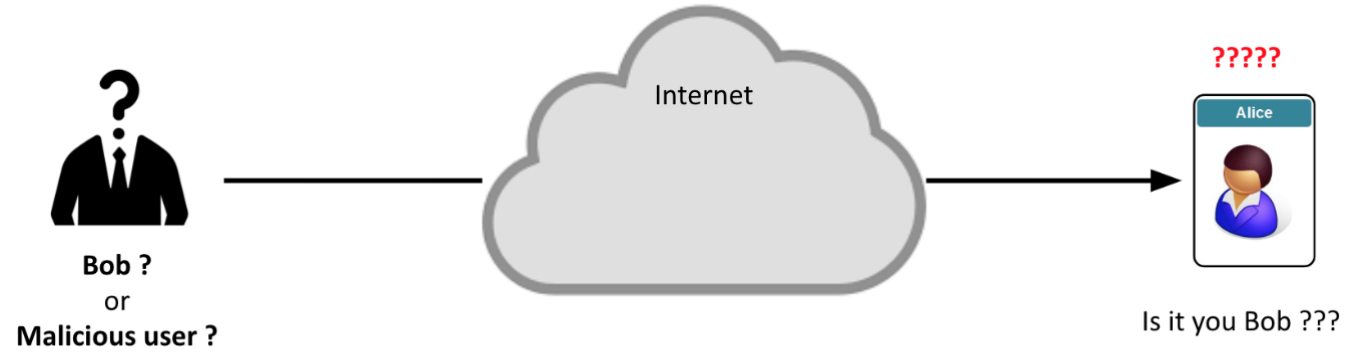
- How do I ensure that my message has not been modified ?
- How do I ensure that the message comes from a legitimate person ?

Secure Communication over Insecure Channel

Problem 1 : How do I ensure that my message has not been modified ?



Problem 2 : How do I ensure that the message comes from a legitimate person ?



pre 1970s solution : Symmetric Cryptography !

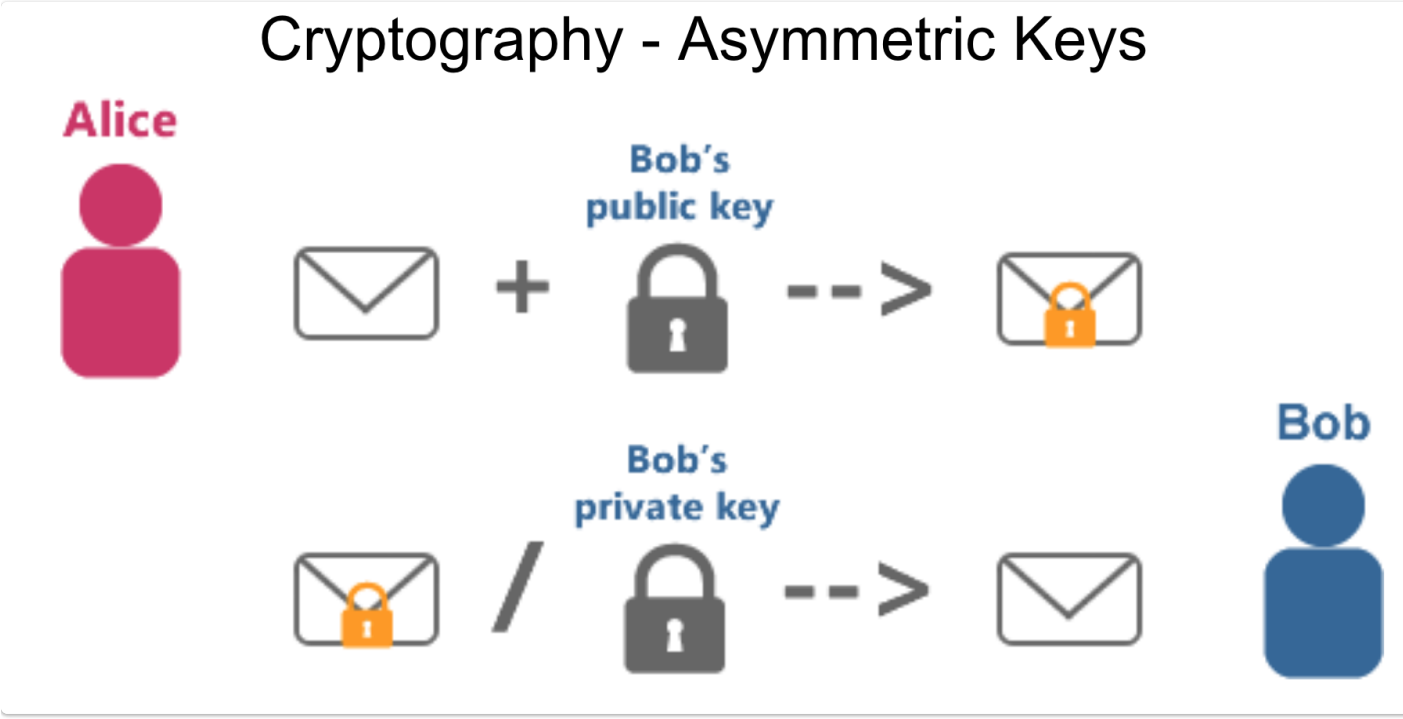
- Alice and Bob share the same key.
- One key for both encryption and decryption of messages

But what about key management ?

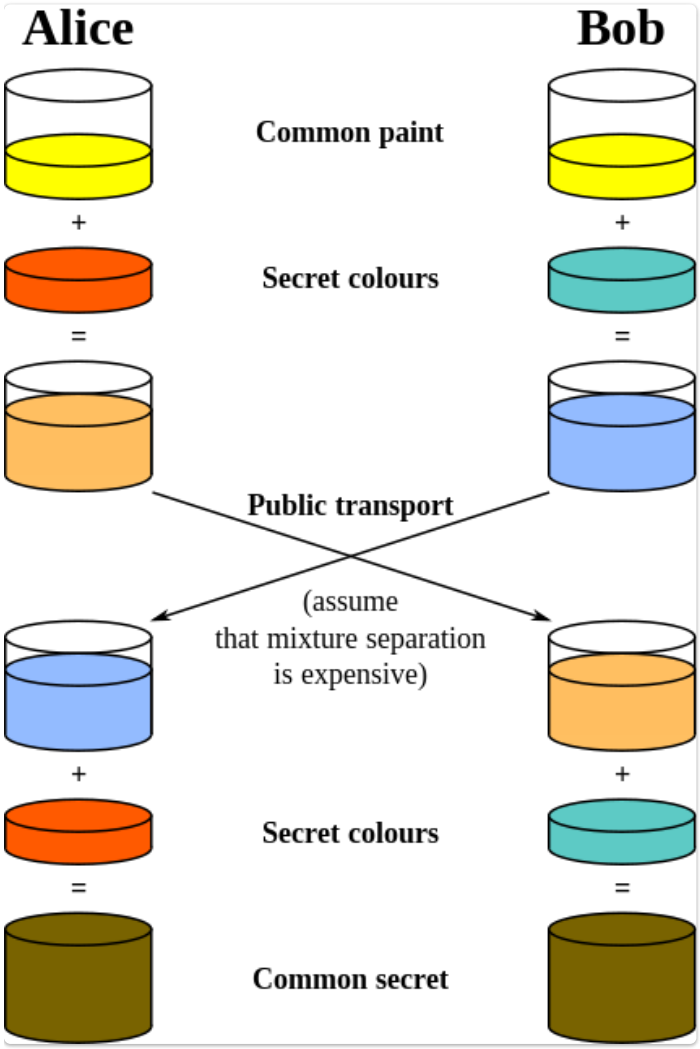
Can Alice and Bob share a key

- Without meeting
- Across a potentially hostile network

In the 1970s there was a major step forward with the invention of asymmetric cryptography.
Here 2 paired keys are used, one for encryption and one for decryption.



A method was also designed to allow the production of a shared secret across an insecure channel, via Diffie Helman Key Exchange. An analogy of the process is provided by thinking of colours being mixed. See [Guide] (https://github.com/archit-p/simplest-oblivious-transfer)



Public Key Encryption solves :

Problem 1 : Key Management

- If I use a 3rd party to share my key, do I trust him ?

Problem 2 : Integrity

- How do I ensure that my message has not been modified ?

Problem 3 : Authenticity

- How do I ensure that the message comes from a legitimate person ?

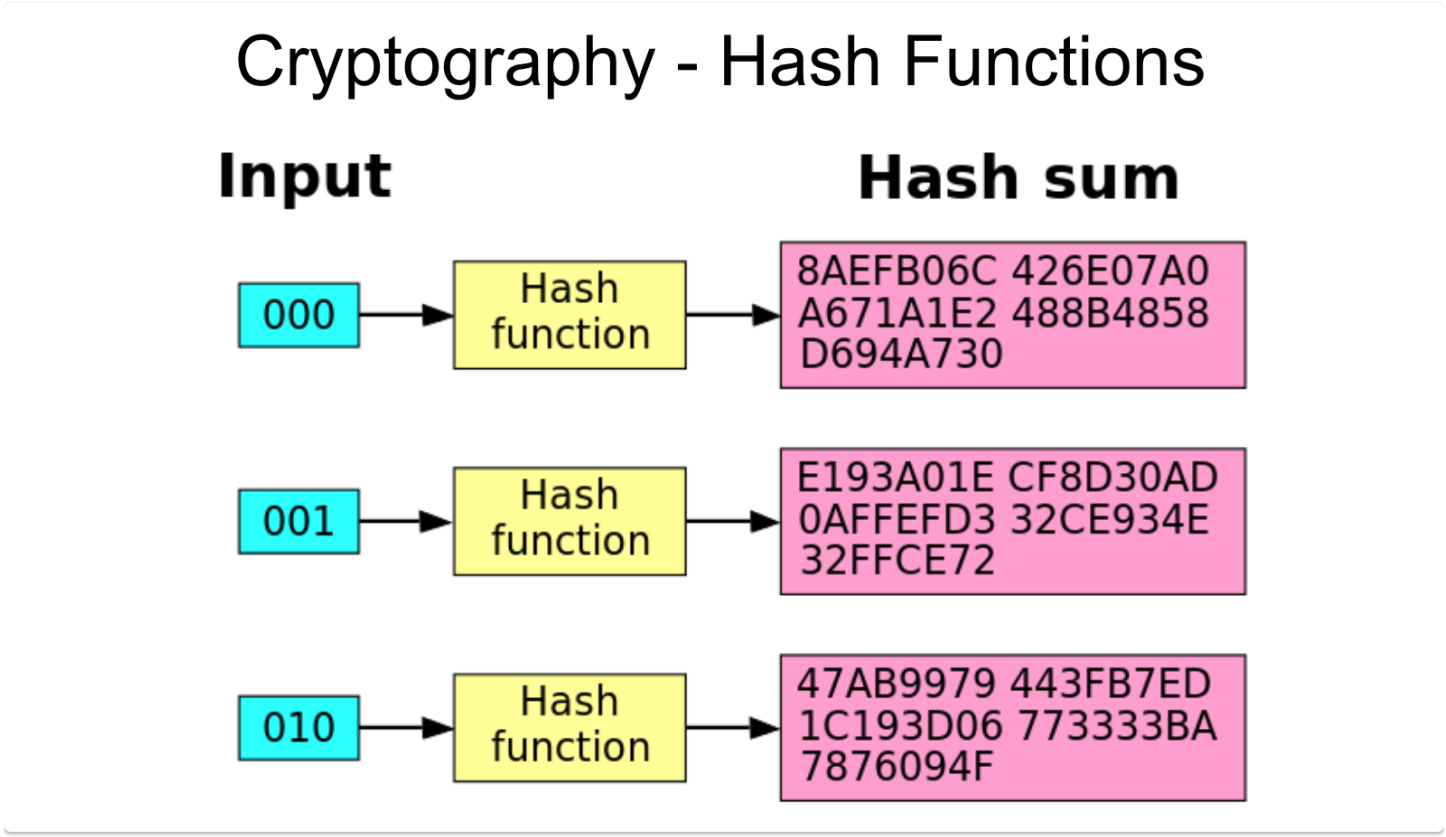
This final problem was solved by the invention of digital signatures.

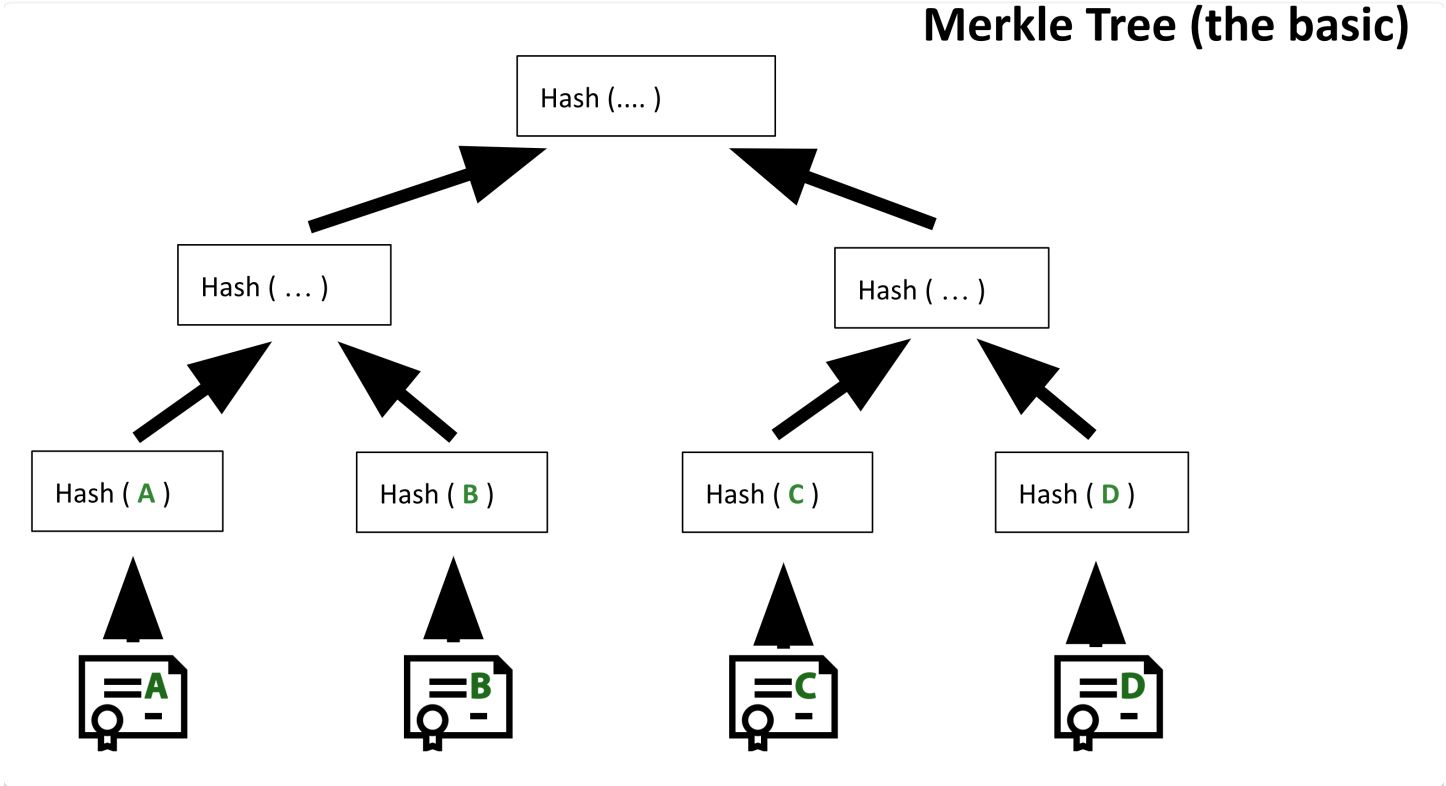
Digital Signatures

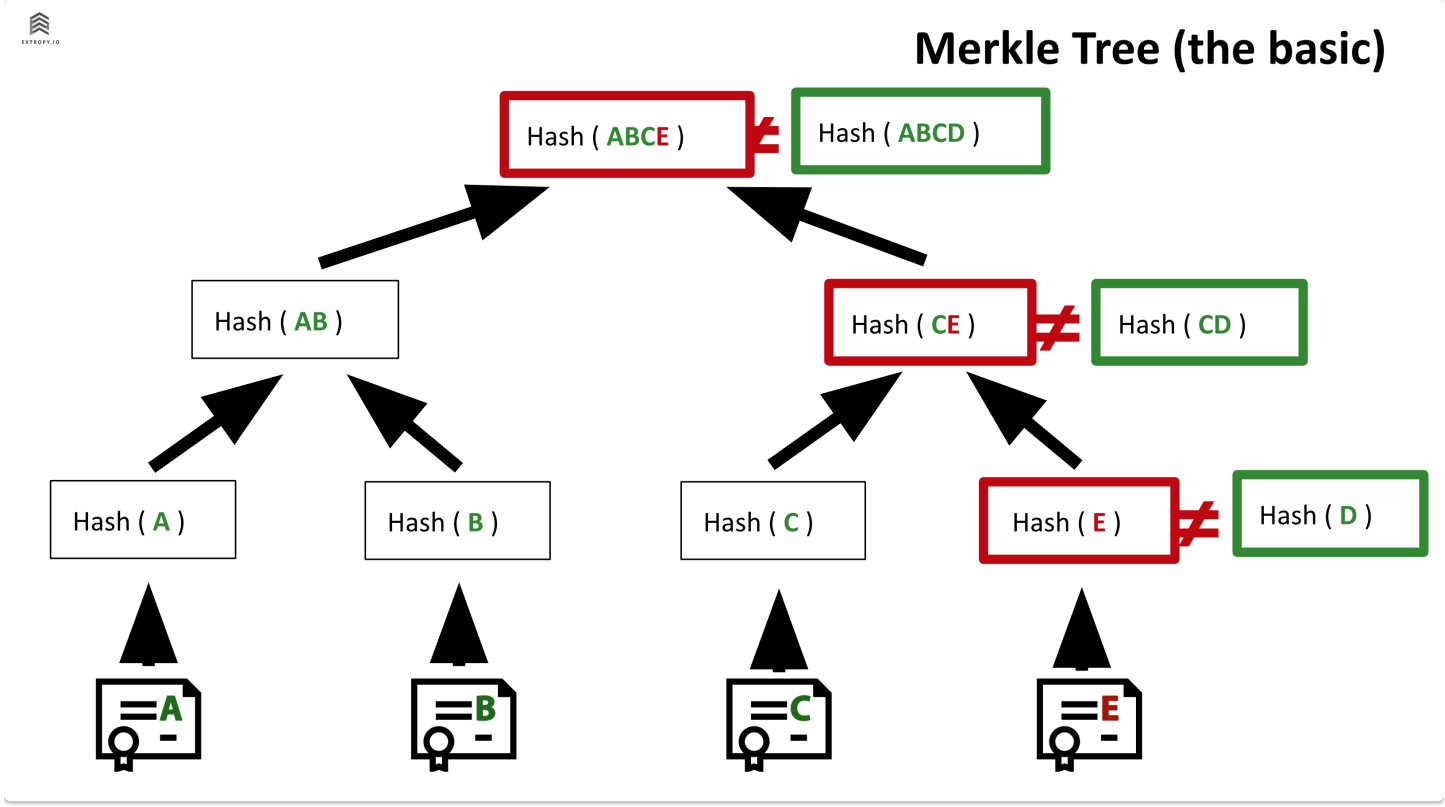
We use digital signatures as a way to show that a message came from a particular person (or holder of a key)
For example imagine Bob is signing a document and sending it to Alice.

A digital signature will have 4 properties

- Authenticity : when Alice verifies the message with Bob's public key, she knows that he signed the message.
- Unforgeable : only Bob knows his private key.
- Not reusable : the signature is tightly bound with the document, it cannot be transferred to any other document.
- Unalterable : If there is any alteration to the document, it will no longer be verifiable with Bob's public key.


Cryptography - Hash Functions
Input
Hash sum
000
Hash function
8AEFB06C 426E07A0 A671A1E2 488B4858 D694A730
001
Hash function
E193A01E CF8D30AD 0AFFEFD3 32CE934E 32FFCE72
010
Hash function
47AB9979 443FB7ED 1C193D06 773333BA 7876094F


Merkle Tree (the basic)
Hash (....)
Hash (…)
Hash (…)
Hash (A)
Hash (B)
Hash (C)
Hash (D)
A
B
C
D


Merkle Tree (the basic)
Hash (ABCE)
≠
Hash (ABCD)
Hash (AB)
Hash (CE)
≠
Hash (CD)
Hash (A)
Hash (B)
Hash (C)
Hash (E)
≠
Hash (D)
A
B
C
E

That is the cryptographic background, how did people try to use this technology ?

The development of

- Electronic cash
- Timestamping
- P2P Systems
- Consensus systems

1980s

David Chaum - Blind Signatures
David Chaum - Digicash

1990s

Timestamping records
Adam Back - HashCash
Wei Dai - B-Money

2000s

Peer to peer networks

- Freenet / Gnutella / Bit Torrent

Further Attempts at Electronic Cash

"the one thing that's missing is a reliable e-cash, whereby on the internet you can transfer funds from A to B without A knowing B or B knowing A" - Milton Friedman 1999

1998 - b-money - Wei Dai (http://www.weidai.com/bmoney.txt)
1998 - Bit Gold - Nick Szabo (https://nakamotoinstitute.org/bit-gold/)

Bitcoin

Satoshi Nakamoto is the name used by the presumed pseudonymous person or persons who developed bitcoin, authored the bitcoin white paper, and created and deployed bitcoin's original reference implementation

Early Bitcoin History

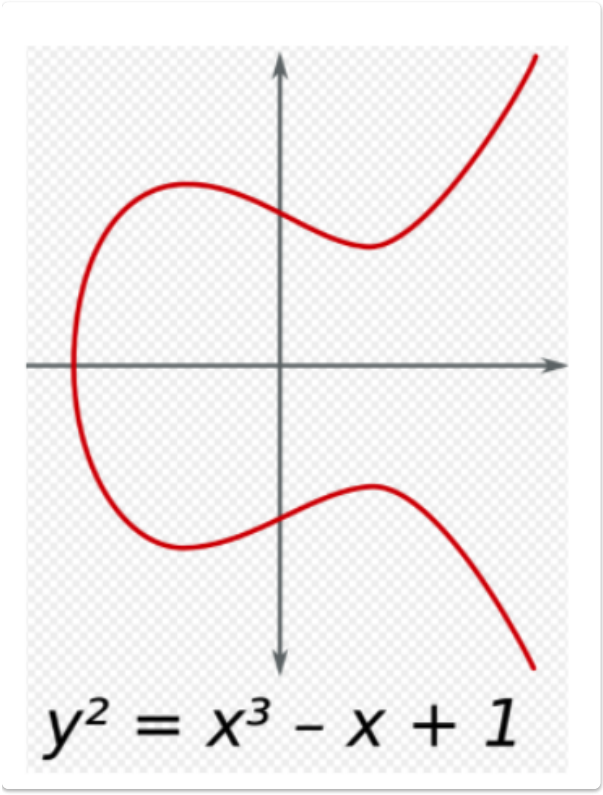
August 2008 - the domain name bitcoin.org registered
October 2008 - A Peer-to-Peer Electronic Cash System posted to a cryptography mailing list
January 2009 - Software implementation released as open source

2010, the first known commercial transaction using bitcoin occurred when programmer Laszlo Hanyecz bought two Papa John's pizzas for 10,000 BTC

General Blockchain events since 2009

2014 - Ethereum created
2017 - ICO Boom / Alternatives to Ethereum
2018 - Crypto winter
2020 - DeFi summer
2021 - Rise of NFTs / Gaming
2022 - Ethereum Merge
2022 - Another crypto winter

Key Cryptography - Elliptic curves



Solana uses EdDSA (Edwards-curve Digital Signature Algorithm)

It uses the curve25519 curve.
Elliptic curves have a shorter key length for the same level of security as RSA

Keys and addresses

The key may be

- an ed25519 public key
- a program-derived account address (32 byte value from the ed25519 curve)
- a hash of an ed25519 public key with a 32 character string