Blockchains & Distributed Ledgers

Lecture 01

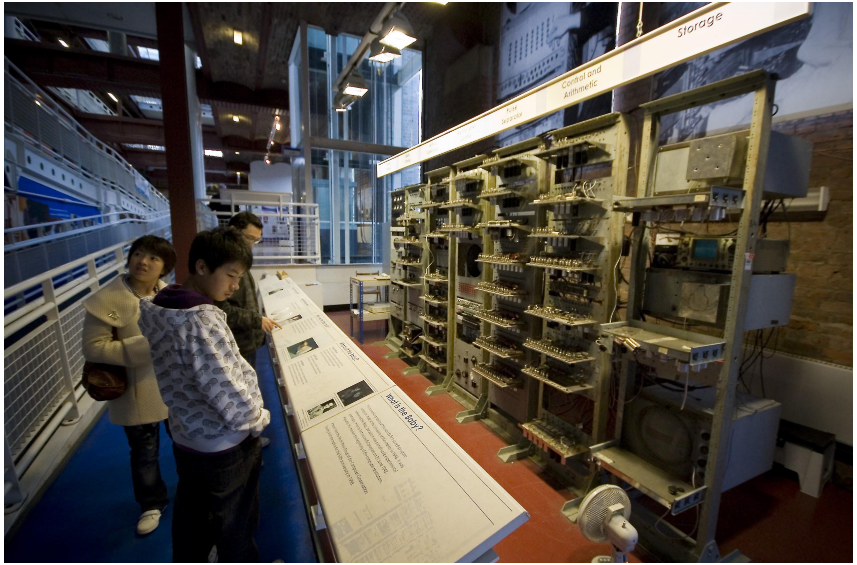
Dimitris Karakostas

Slide credits: DK, Aggelos Kiayias, Dionysis Zindros, Christos Nasikas

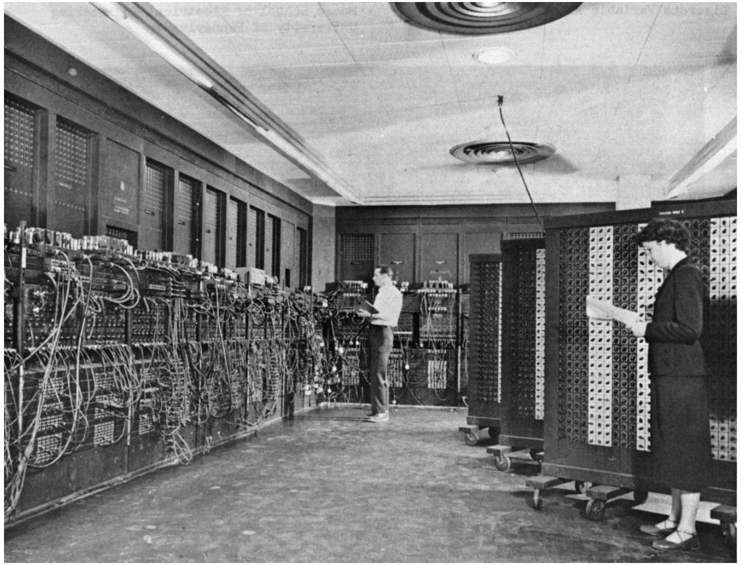
Introduction

- Historical background
- Introduction to decentralized ledgers
- The never-ending book parable
- The first blockchain application
- Hash functions
- Digital signatures
- Why blockchains?

Once upon a time...



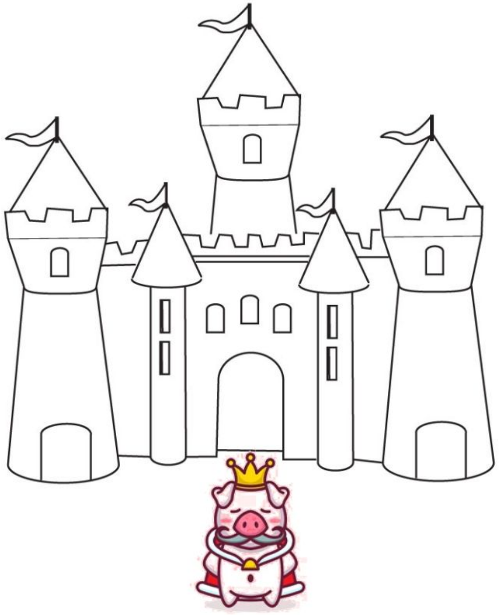
[Manchester baby]



[ENIAC]

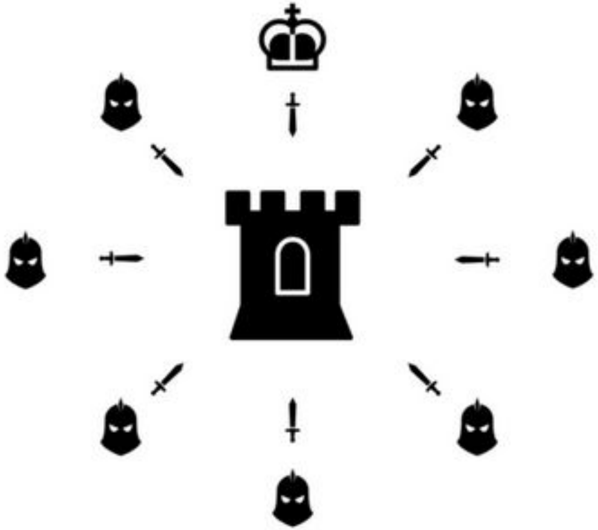
Centrally-controlled systems

- A single person (party/node) controls who can read/write/delete data
- If the person/party/node dies/is dishonest/crashes, the system crashes



Controlled-access distributed systems

- Nodes collectively control the system
- If only few nodes faulty, system remains operational
- Controlled participation - only authorized parties



Open-access distributed systems?

- Nodes collectively control the system
- If only few nodes faulty, system remains operational
- Anyone can participate, join or leave as they please

What is a blockchain ?

- A blockchain is a distributed database that satisfies a unique set of safety and liveness properties
- Distributed ledgers use a blockchain protocol as one means of implementation
- To understand it, we can focus on its first application

Why study blockchains?

Why study blockchains?

- Good foundations for exploring security of information systems in general
- Explore decentralisation, a property of increasing importance in the design of modern information systems
- Solid understanding of many security critical components, including:
  - Key management
  - Software security
  - Privacy preserving technologies
  - Public Key Infrastructure
- Novel opportunities for applications on various aspects of societal organisation
- It’s fun!

The never-ending book parable

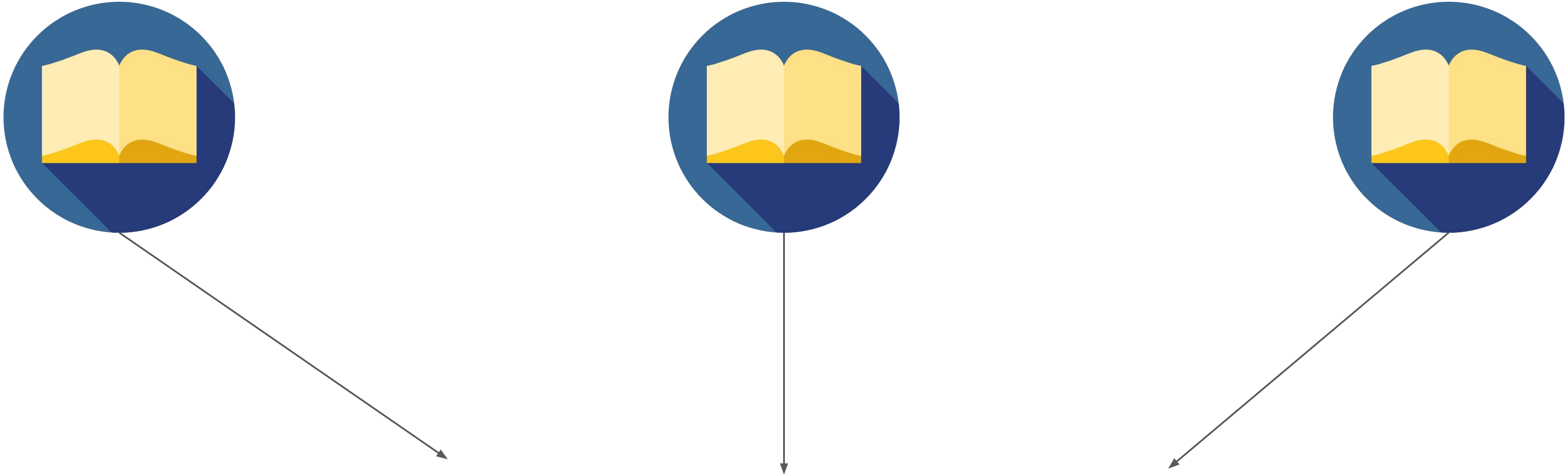
A book of data

- Anyone can be a scribe and produce a page
- New pages are produced indefinitely, as long as scribes are interested in doing so
- Each new page requires some effort to produce

Importance of consensus

- If multiple conflicting books exist, which is the “right one” ?

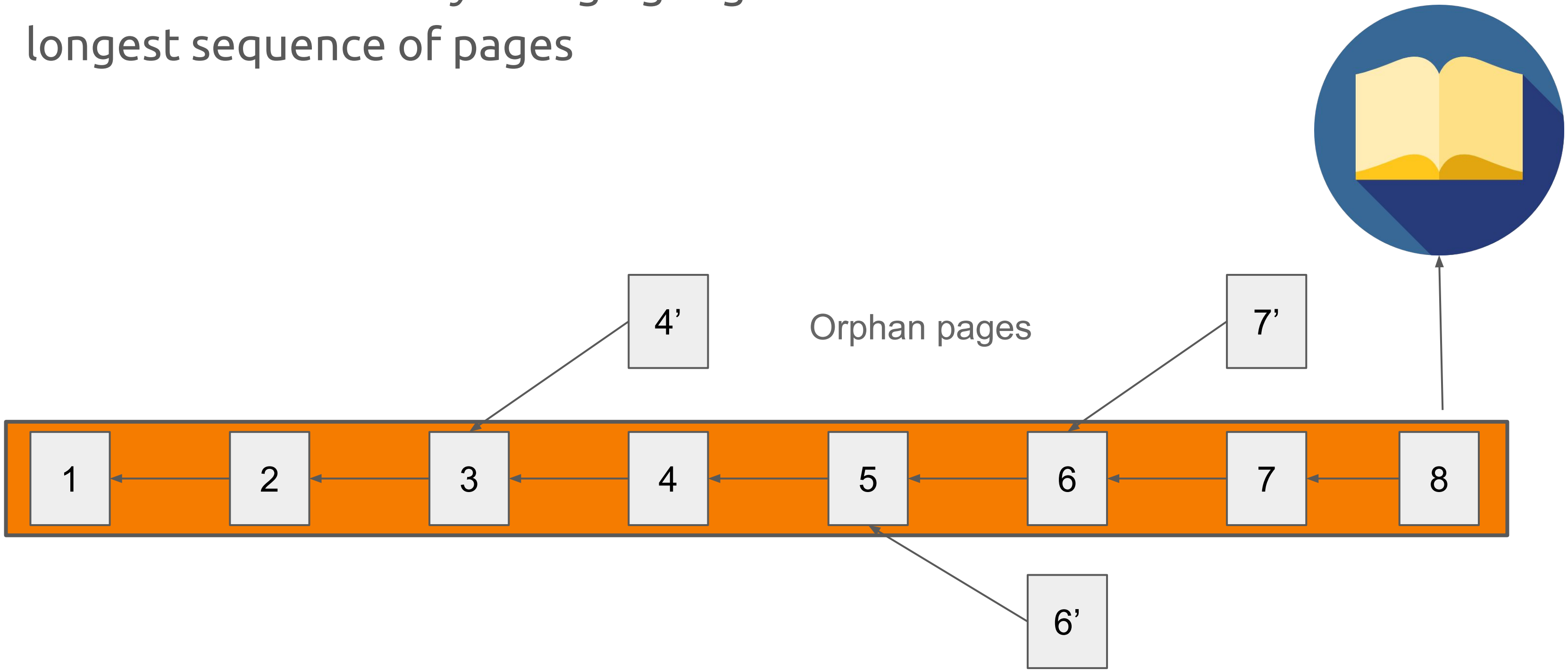
Choosing the correct book ?



The correct book to work on & refer to is the book with the most pages. If multiple exist, just pick one at random.

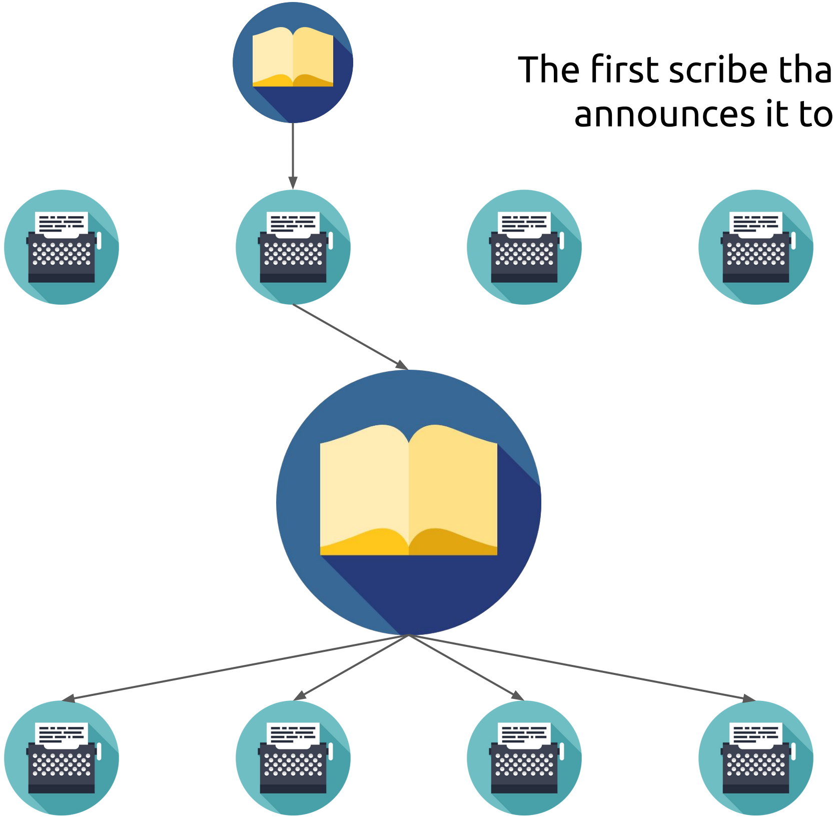
Assembling the current book

- Each page refers only to the previous one
- Current assembled by stringing together the longest sequence of pages



Orphan pages

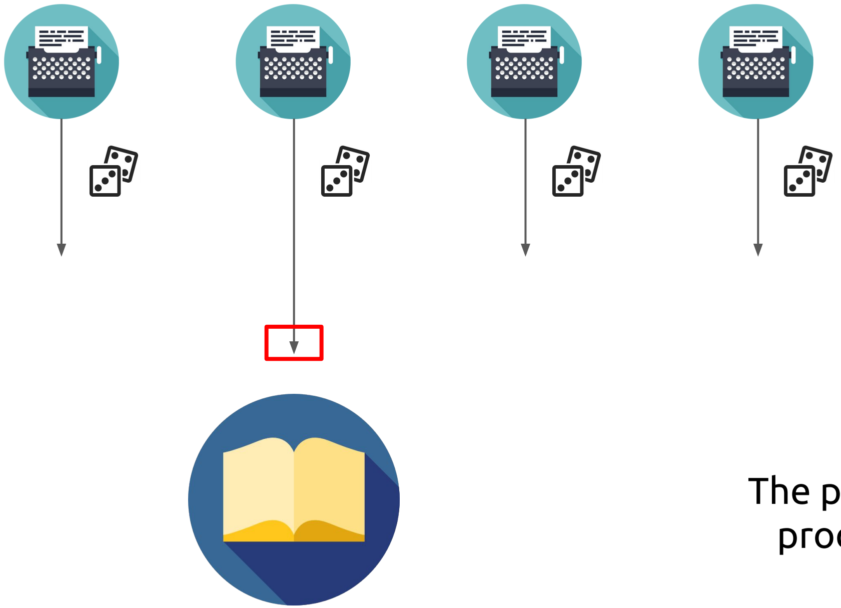
Rules of extending the book



The first scribe that discovers a page announces it to everyone else

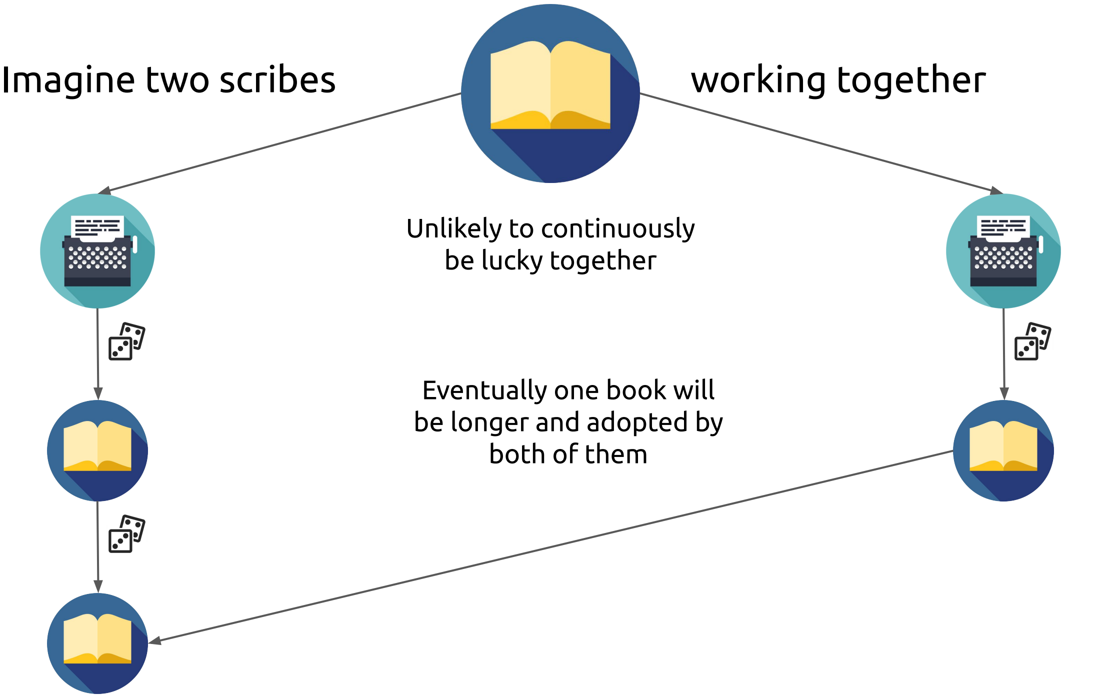
Effort is needed to produce a page

Equivalent to: each page needs a special combination from a set of dice to be rolled.



The probabilistic nature of the process is paramount to its security

The benefits of randomness



Imagine two scribes

working together

Unlikely to continuously be lucky together

Eventually one book will be longer and adopted by both of them

Symmetry Breaking

Being a scribe

- Anyone can be a scribe for the book
- As long as one has a set of dice
- The more dice one has, the higher the likelihood to produce the winning combination to make a page

Parable & Reality

📖	The “blockchain”
⌨	“Miners” / Computer systems that organize transactions in blocks
	Solving a cryptographic puzzle that is moderate hard to solve
🎲	Using a computer to test for a solution from a large space of candidate solutions

The first blockchain application

Blockchain applications

- Effort is needed to produce a page
    - Why would anyone care to constantly roll dices?
- Decentralization is complex
    - What kind of application could benefit from a completely decentralized database?

Bitcoin

What is bitcoin ?

Why would people participate?

- Utility: “that property in any object, whereby it tends to produce benefit, advantage, pleasure, good, or happiness” (Jeremy Bentham, 1870)
- Homo Economicus:
    - Constantly acts rationally and optimally
    - Aims to maximize its financial gain

Why would people participate?

- Utility: “that property in any object, whereby it tends to produce benefit, advantage, pleasure, good, or happiness” (Jeremy Bentham, 1870)
- Homo Economicus:
  - Constantly acts rationally and optimally
  - Aims to maximize its financial gain
- People would participate in Bitcoin, if they are financially rewarded
- Users would pay others to run the system
- Free market (everywhere)

What is Bitcoin (trying to be)?

- Money
  - Competing with GBP £, USD $, EUR €, etc
  - *Medium of exchange*: give money to get goods and vice versa
  - *Unit of account*: price goods, for accounting/debt purposes
  - *Short/Medium-term store of value*: can be exchanged for the same amount of goods in the (not so distant) future

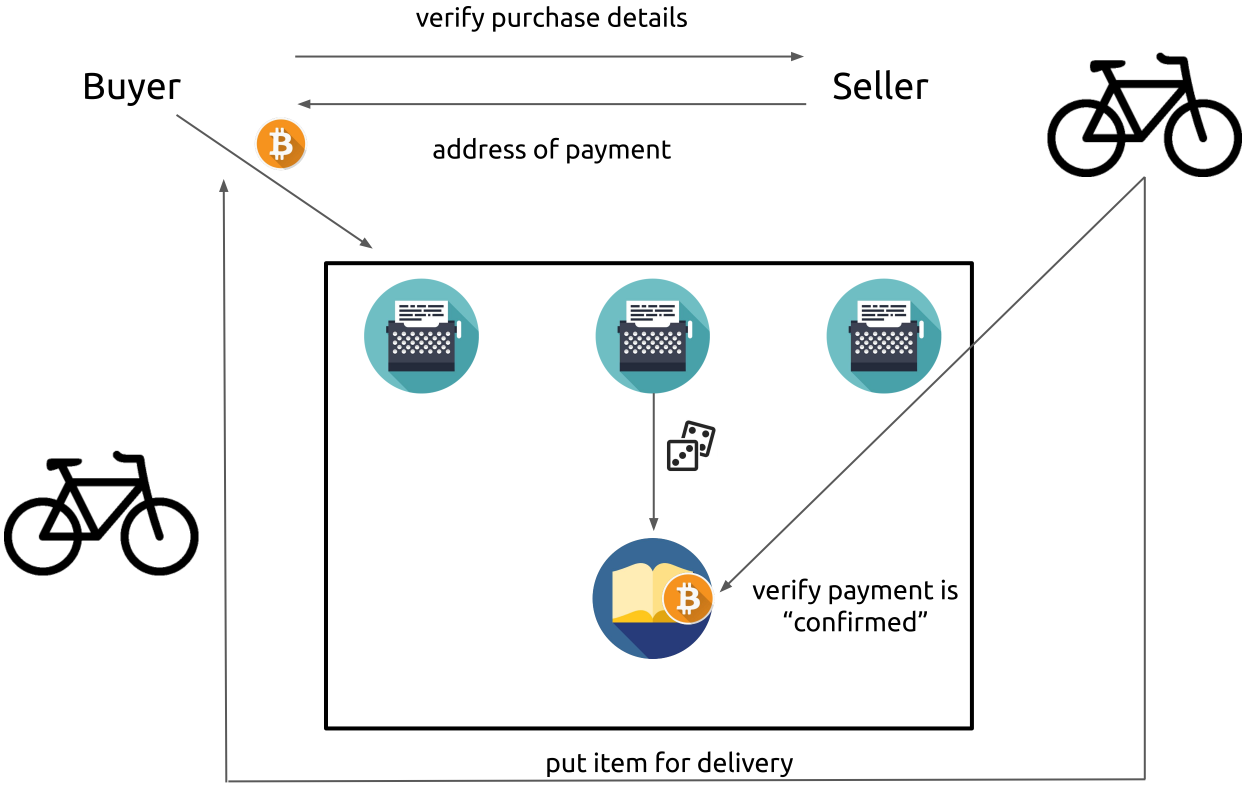
What is Bitcoin (trying to be)?

- Money
  - Competing with GBP £, USD $, EUR €, etc
  - *Medium of exchange*: give money to get goods and vice versa
  - *Unit of account*: price goods, for accounting/debt purposes
  - *Short/Medium-term store of value*: can be exchanged for the same amount of goods in the (not so distant) future
- Payment system
  - Competing with cash, bank deposit operation network, Visa, Mastercard, etc
  - High throughput (large volume of transactions/sec)
  - Low latency (fast transaction settlement)
  - Uninterrupted service

What is Bitcoin (trying to be)?

- Money
  - Competing with GBP £, USD $, EUR €, etc
  - *Medium of exchange*: give money to get goods and vice versa
  - *Unit of account*: price goods, for accounting/debt purposes
  - *Short/Medium-term store of value*: can be exchanged for the same amount of goods in the (not so distant) future
- Payment system
  - Competing with cash, bank deposit operation network, Visa, Mastercard, etc
  - High throughput (large volume of transactions/sec)
  - Low latency (fast transaction settlement)
  - Uninterrupted service
- Commodity
  - Competing with gold, silver, oil, etc
  - A (useful) material that can be bought/sold

Using the Bitcoin book



verify purchase details

Buyer

Seller

address of payment

verify payment is "confirmed"

put item for delivery

Person to person

Advantages

- Resilience
  - The book is shared across the network
  - Even if some nodes crash or are corrupted, system is operational
- Censorship resistance
  - Anyone can participate
  - Geographical disparity of nodes
  - Good alternative for borderline (or beyond) legal financial transactions
- Digital and Open
  - New applications can be easily built on top of it
  - Programs can be hosted on the ledger itself (smart contracts)

Disadvantages

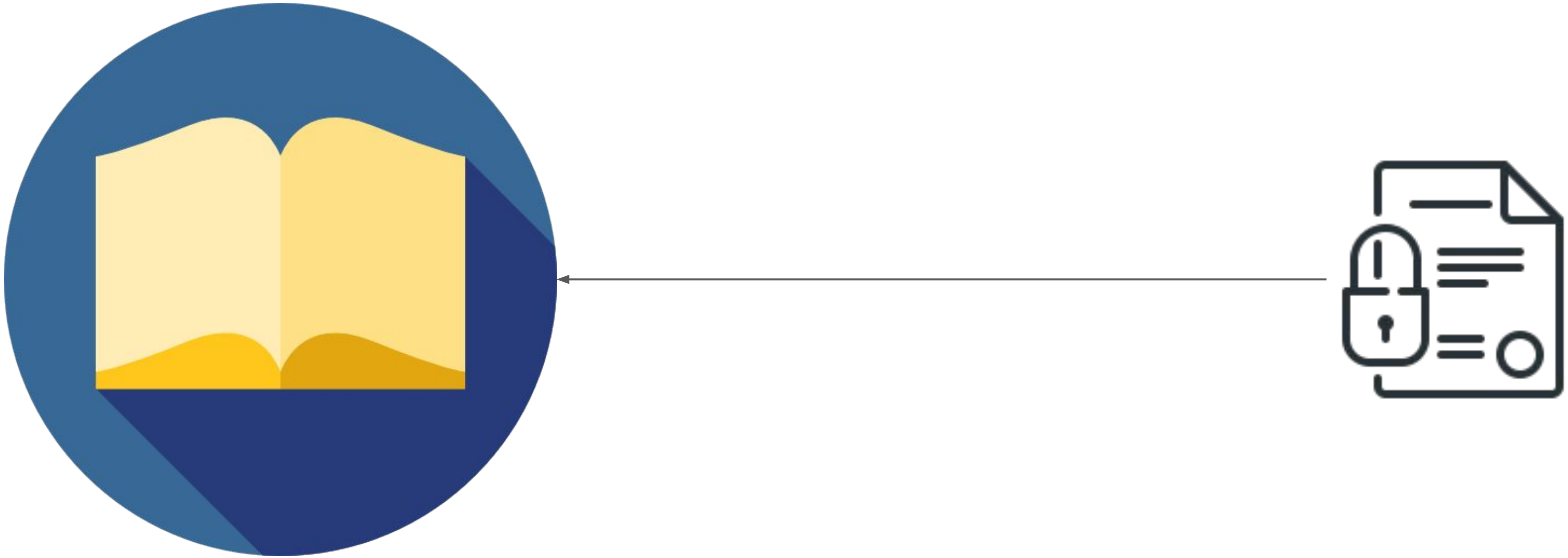
- Bad as money
  - Price fluctuations and circulation does not follow economic growth (bad store of value)
  - Nothing is priced in Bitcoin (bad unit of account)
  - Slow and expensive (bad medium of exchange)
    - Low throughput (~5 tx/sec)
    - High latency (60 mins)
    - High fees[***] (~3$)
- Irreversibility
  - If a transaction is processed, it cannot be deleted/reversed
  - If user's bitcoins are stolen or loses key, no recovery mechanism exists
- Environmental inefficiency
  - Bitcoin CO_2 footprint[*]: 76.44 Mt (~Colombia)
  - *Single* Bitcoin tx CO_2 footprint[*]: 820.84 kg (~1.8M VISA transactions)
  - Single Bitcoin tx e-waste[**]: 242g (~1.5 Iphones)

[*] https://digiconomist.net/bitcoin-energy-consumption

[**] https://www.sciencedirect.com/science/article/pii/S0921344921005103

[***] https://bitinfocharts.com/comparison/bitcoin-transactionfees.html

Smart contract



From Money to Smart Contracts

- Since we have created the book, why stop at recording monetary transactions?
- We can encode in the book’s pages arbitrary relations between accounts
- Scribes can perform tasks and take action, like verifying that stakeholders comply to contractual obligations

Questions to Consider

- How are pages created? Since the book is empty at the beginning, where do the money come from?
- How is it possible to sign something digitally?
- How does a page properly refer to the previous page?

Questions to Consider

- How are pages created? Since the book is empty at the beginning, where do the money come from? - Proof-of-Work
- How is it possible to sign something digitally? - Digital signatures
- How does a page properly refer to the previous page? - Hash functions

Hash Functions

- An algorithm that produces a fingerprint of a file.
- what are the required properties (traditionally):
  a. Efficiency
  b. A good spread for various input distributions.
- What are Security/Cryptographic considerations

$$\mathcal{H} : \{0,1\}^* \to \{0,1\}^\lambda$$

Collision resistance

Collision attack

Find $x, y : \mathcal{H}(x) = \mathcal{H}(y)$

Second pre-image attack

Find $y : \mathcal{H}(x) = \mathcal{H}(y)$

For given x

Birthday paradox

- How many people should be in a room so that the probability that two of them share a birthday becomes larger than 50% ?

Birthday paradox

- How many people should be in a room so that the probability that two of them share a birthday becomes larger than 50%?

n possible dates $\qquad$ k people

$$\Pr[\neg Col] =$$

$$\frac{n}{n}\frac{n-1}{n}\frac{n-2}{n}\dots\frac{n-k+1}{n} = \prod_{\ell=1}^{k}(1-\frac{\ell}{n})$$

$$\leq \exp(-\frac{1}{n}\sum_{\ell=1}^{k}\ell) = \exp(-k(k+1)/2n)$$

$$\Pr[Col] = \frac{1}{2} \Rightarrow k \approx 1.177\sqrt{n}$$

What do we learn about collision finding?

Describe an algorithm that finds collisions taking advantage of the Birthday paradox.

Pre-image attack

Given $\mathcal{H}(m)$ $m \in \{0,1\}^t$

Find an element of $\mathcal{H}^{-1}(\mathcal{H}(m))$

Generic algorithm tries all possible candidates
Complexity: ?

One-way functions

$$f : X \to Y$$

easy : given x find $f(x)$

hard : given $f(x)$ sample $f^{-1}(f(x))$

Do one-way functions exist?

Relates to most important open question in computer science right now:

$$P \neq NP$$

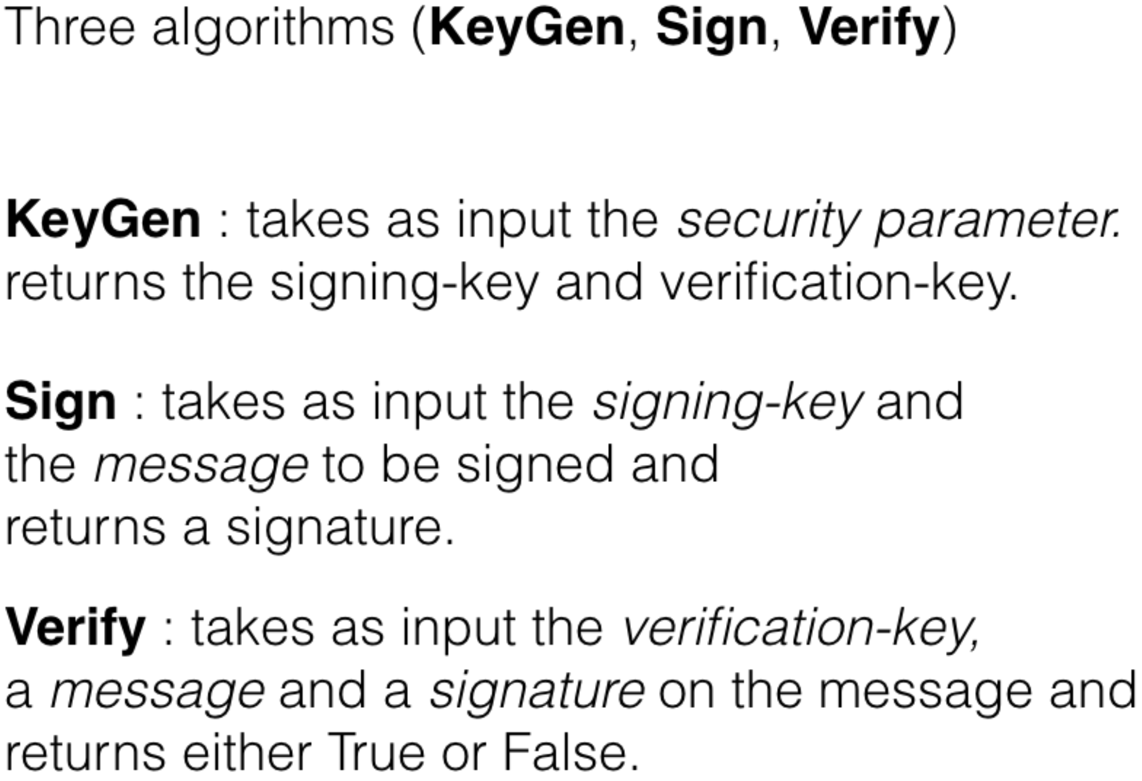
Hash function instantiations

- **Retired.** MD5, SHA1.
- **Current.** SHA2, SHA3, available for 224,256,384,512 bits fingerprints.
- **Bitcoin.** Uses SHA2 with 256 bits output, SHA-256.

Digital Signatures

- Can be produced by one specified entity.
- Can be verified by anyone (that is suitably “equipped" and “initialised”).
- Cannot be forged on a new message even if multiple signatures have been transmitted.

Digital Signatures



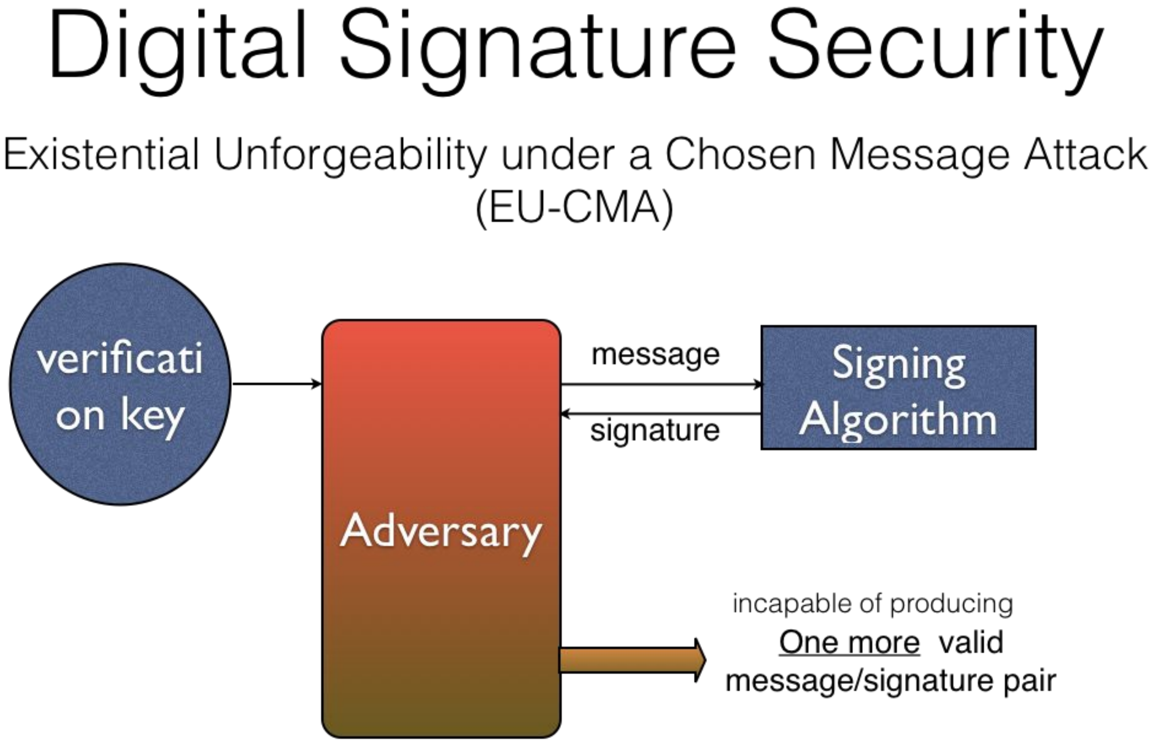
Three algorithms (**KeyGen**, **Sign**, **Verify**)

KeyGen : takes as input the *security parameter.*
returns the signing-key and verification-key.

Sign : takes as input the *signing-key* and
the *message* to be signed and
returns a signature.

Verify : takes as input the *verification-key,*
a *message* and a *signature* on the message and
returns either True or False.

Digital Signature Security



Digital Signature Security

Existential Unforgeability under a Chosen Message Attack (EU-CMA)

verificati on key

Adversary

message

signature

Signing Algorithm

incapable of producing

<u>One more</u> valid message/signature pair

Constructing Digital Signatures

- Major challenge:
  - what prevents the adversary from learning how to *sign* messages by analyzing the *verification-key?*
- Exercise: construct a digital signature based on a hash-function that is one-time secure (i.e., it is secure for signing only a single message)

Digital Signature Implementations

- Based on the RSA (Rivest Shamir Adleman), one way trapdoor function (with hardness that relates to the factoring problem)
  - The RSA algorithm
- Based on the discrete-logarithm problem
  - the DSA algorithm
- Bitcoin. Uses ECDSA, a DSA variant over elliptic curve groups

Proof-of-Work

- Objective: given some *data*, ensure that some amount of work has been invested for computing them

```
int counter;
counter = 0
while Hash(data, counter) > Target
        increment counter
return counter
```

- In this case: Proof-of-Work of *data* equals to a value *w* with the property Hash(*data*, *w*) <= Target

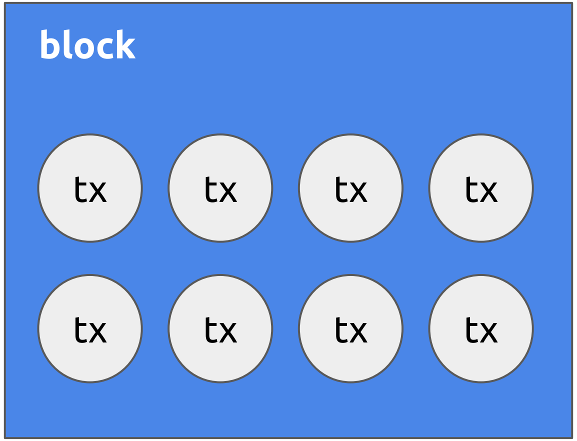
- (Informal) Properties:
  - efficient verification
  - no computational shortcuts (i.e., independent of algorithm that computes it complexity is proportional to Target)
  - independence for symmetry-breaking

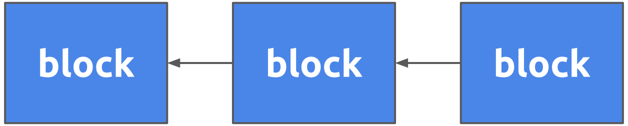
Proof-of-Work Algorithms

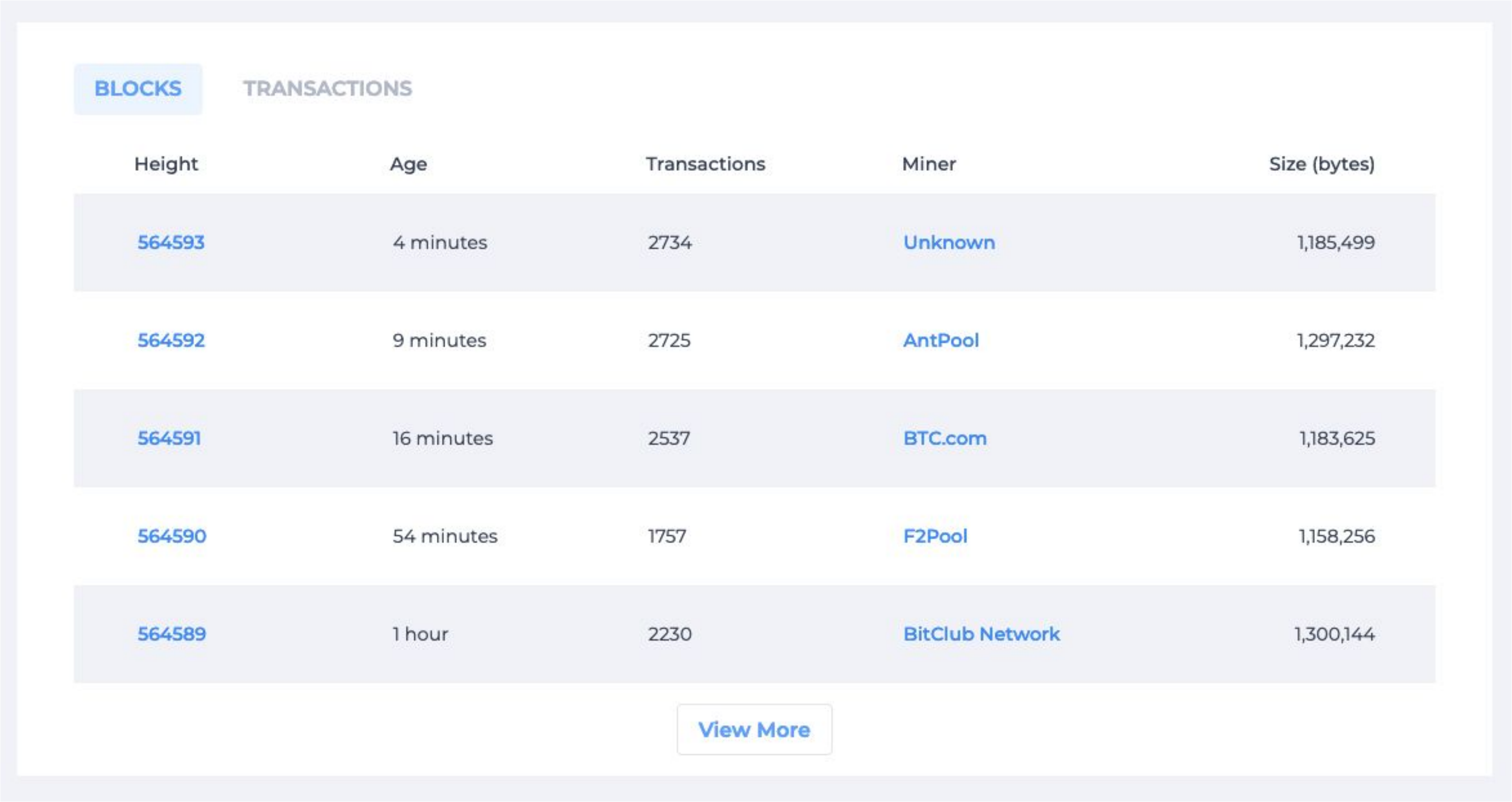
- Hashcash (as in previous slide)
- Memory hardness
  - ASIC resistance (ASIC = Application Specific Integrated Circuit)
  - A number of algorithms proposed: scrypt, argon, progpow

Bitcoin in practice

Blocks

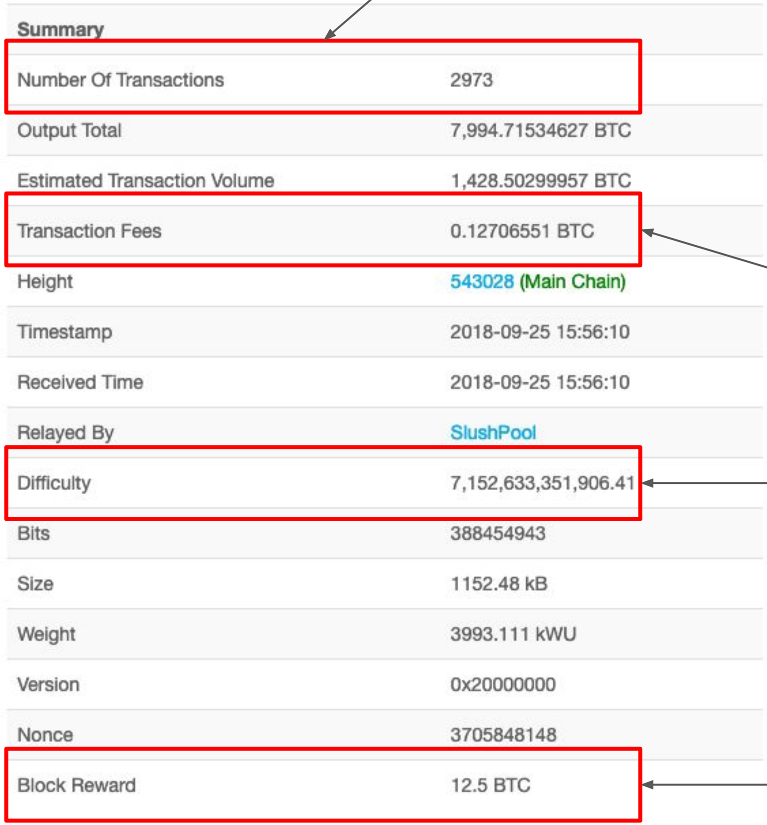




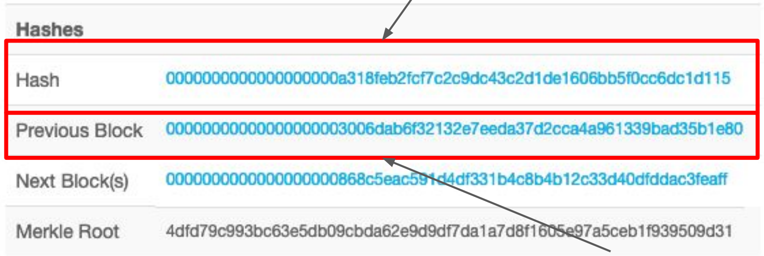


Height	Age	Transactions	Miner	Size (bytes)
564593	4 minutes	2734	Unknown	1,185,499
564592	9 minutes	2725	AntPool	1,297,232
564591	16 minutes	2537	BTC.com	1,183,625
564590	54 minutes	1757	F2Pool	1,158,256
564589	1 hour	2230	BitClub Network	1,300,144

Blocks

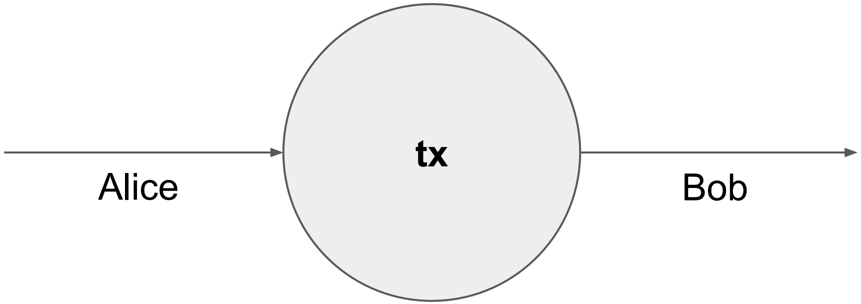


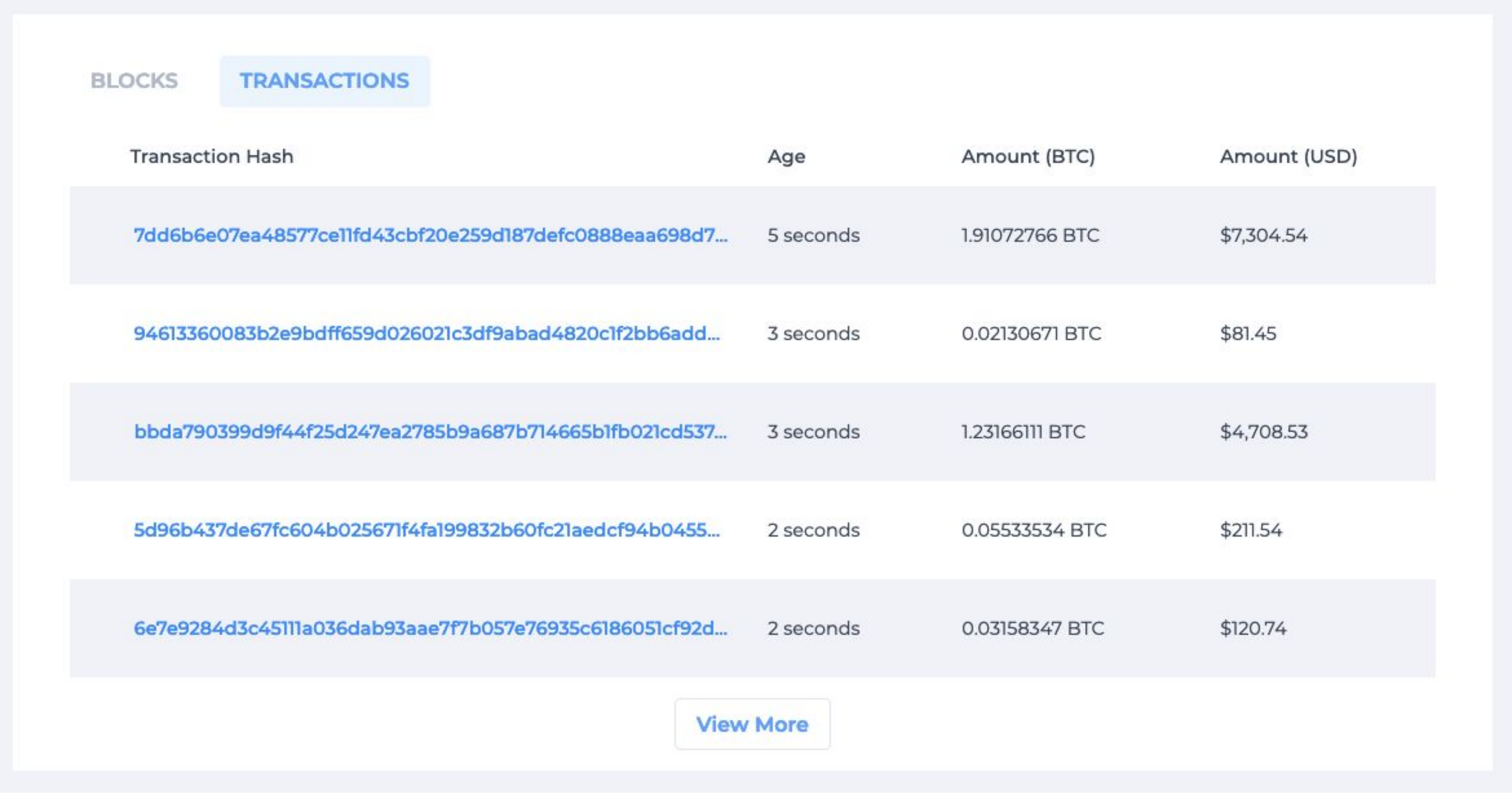
Summary	
Number Of Transactions	2973
Output Total	7,994.71534627 BTC
Estimated Transaction Volume	1,428.50299957 BTC
Transaction Fees	0.12706551 BTC
Height	543028 (Main Chain)
Timestamp	2018-09-25 15:56:10
Received Time	2018-09-25 15:56:10
Relayed By	SlushPool
Difficulty	7,152,633,351,906.41
Bits	388454943
Size	1152.48 kB
Weight	3993.111 kWU
Version	0x20000000
Nonce	3705848148
Block Reward	12.5 BTC



Hashes	
Hash	0000000000000000000a318feb2fcf7c2c9dc43c2d1de1606bb5f0cc6dc1d115
Previous Block	00000000000000000003006dab6f32132e7eeda37d2cca4a961339bad35b1e80
Next Block(s)	0000000000000000000868c5eac591d4df331b4c8b4b12c33d40dfddac3feaff
Merkle Root	4dfd79c993bc63e5db09cbda62e9d9df7da1a7d8f1605e97a5ceb1f939509d31

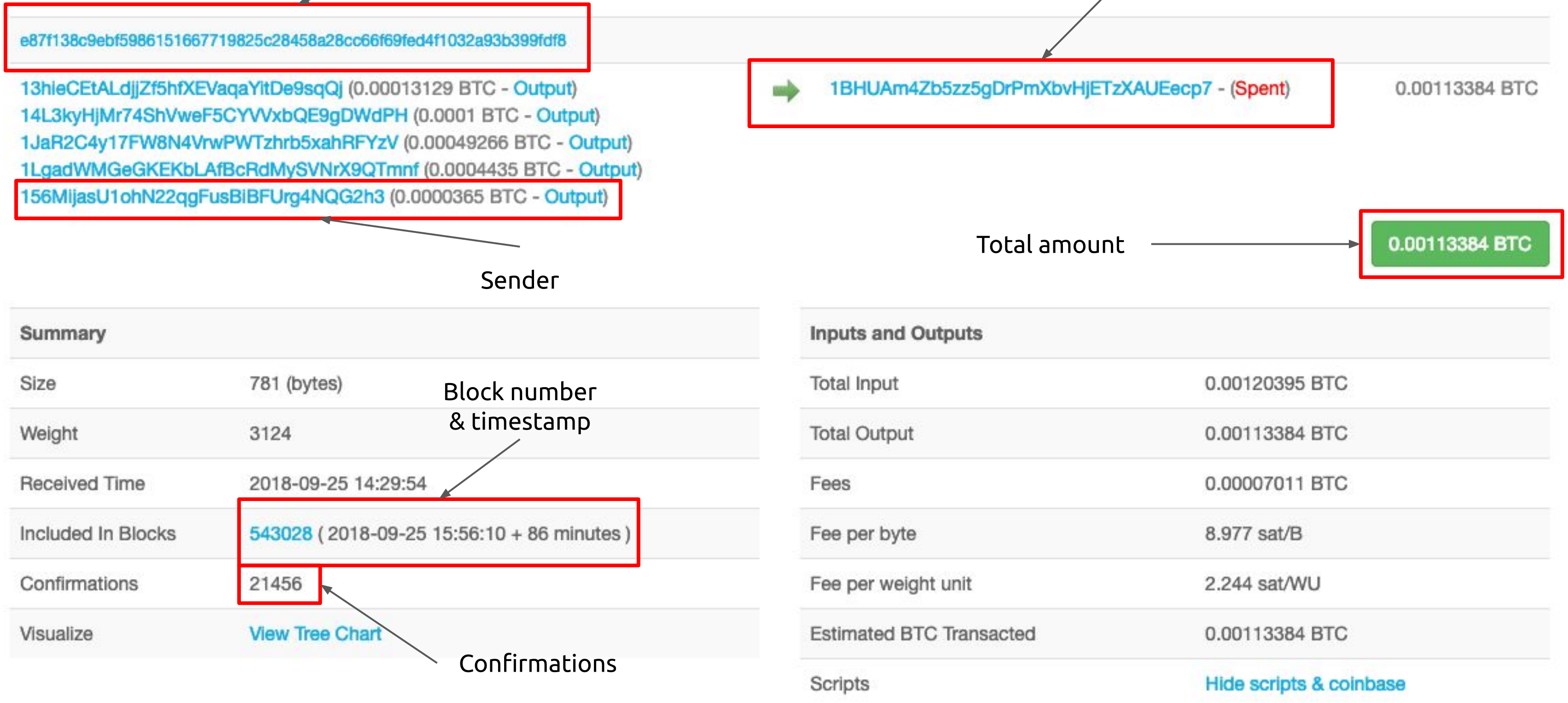
Transactions





Transaction Hash	Age	Amount (BTC)	Amount (USD)
7dd6b6e07ea48577ce11fd43cbf20e259d187defc0888eaa698d7...	5 seconds	1.91072766 BTC	$7,304.54
94613360083b2e9bdff659d026021c3df9abad4820c1f2bb6add...	3 seconds	0.02130671 BTC	$81.45
bbda790399d9f44f25d247ea2785b9a687b714665b1fb021cd537...	3 seconds	1.23166111 BTC	$4,708.53
5d96b437de67fc604b025671f4fa199832b60fc21aedcf94b0455...	2 seconds	0.05533534 BTC	$211.54
6e7e9284d3c45111a036dab93aae7f7b057e76935c6186051cf92d...	2 seconds	0.03158347 BTC	$120.74

Transactions



e87f138c9ebf5986151667719825c28458a28cc66f69fed4f1032a93b399fdf8

13hieCEtALdjjZf5hfXEVaqaYitDe9sqQj (0.00013129 BTC - Output)
14L3kyHjMr74ShVweF5CYVVxbQE9gDWdPH (0.0001 BTC - Output)
1JaR2C4y17FW8N4VrwPWTzhrb5xahRFYzV (0.00049266 BTC - Output)
1LgadWMGeGKEKbLAfBcRdMySVNrX9QTmnf (0.0004435 BTC - Output)
156MijasU1ohN22qgFusBiBFUrg4NQG2h3 (0.0000365 BTC - Output)

1BHUAm4Zb5zz5gDrPmXbvHjETzXAUEecp7 - (Spent) 0.00113384 BTC

0.00113384 BTC

Summary	
Size	781 (bytes)
Weight	3124
Received Time	2018-09-25 14:29:54
Included In Blocks	543028 (2018-09-25 15:56:10 + 86 minutes)
Confirmations	21456
Visualize	View Tree Chart

Inputs and Outputs	
Total Input	0.00120395 BTC
Total Output	0.00113384 BTC
Fees	0.00007011 BTC
Fee per byte	8.977 sat/B
Fee per weight unit	2.244 sat/WU
Estimated BTC Transacted	0.00113384 BTC
Scripts	Hide scripts & coinbase

Transaction ID

Receiver

Sender

Total amount

Block number
& timestamp

Confirmations

Addresses

- Like an IBAN (or email)
- You send bitcoins to a person by sending bitcoins to one of their addresses
- You can have as many addresses as you want
- No need to be online to create an address
- Pseudonymous: a unique address used for each transaction
- Wallet: the application that controls a user’s addresses

Digital wallet: Bitcoins in your computer or phone

Development

- Local blockchains:
    - Used for local development
    - Instant mining
    - Very small in size
  - You can use a local Ethereum blockchain online with Remix
- Testnets:
    - Used for testing and experimenting
    - Very useful, specifically for smart contract development
    - Different blockchain and different genesis block
    - Coins with no real value, separated and distinct from actual coins
    - Different ports and DNS seeds
    - Ethereum: Rinkeby, Ropsten, Kovan
  - In class we will use our own Ethereum testnet
- Main net (production):
    - Blockchains are immutable and irrevertible
    - You cannot simply update your code once deployed!

Explorers

- An online blockchain browser
- Displays the contents of individual blocks and transactions
- Displays the transaction histories and balances of addresses
- Quick way to see if your transactions are confirmed
- Bitcoin:
  - https://www.blockchain.com/explorer (Mainnet)
  - https://testnet.blockexplorer.com/ (Testnet)
- Ethereum:
  - https://etherscan.io/ (Mainnet)
  - https://ropsten.etherscan.io/ (Testnet)
  - https://rinkeby.etherscan.io/ (Testnet)

Faucet

- A way to get test coins necessary for any testing
- Ethereum:
  - https://faucet.rinkeby.io/
  - https://faucet.metamask.io/
  - https://faucet.ropsten.be/
- Bitcoin:
  - http://tbtc.bitaps.com/
  - https://bitcoinfaucet.uo1.net/
  - https://testnet-faucet.mempool.co/
  - https://block.io/ (Online testnet wallet)