Writing Smart Contracts
01 Introduction

Peter H. Gruber

Supported by the Algorand Foundation

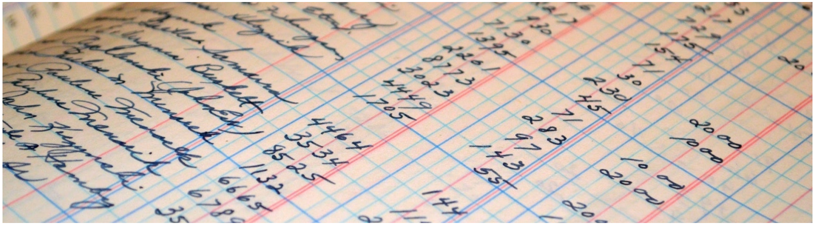
Introduction

"To understand the power of blockchain systems, [. . .] it is important to distinguish between three things that are commonly muddled up, namely

1. the bitcoin currency,
2. the specific blockchain that underpins it and
3. the idea of blockchains in general."

"The Trust Machine", THE ECONOMIST, Oct. 31, 2015

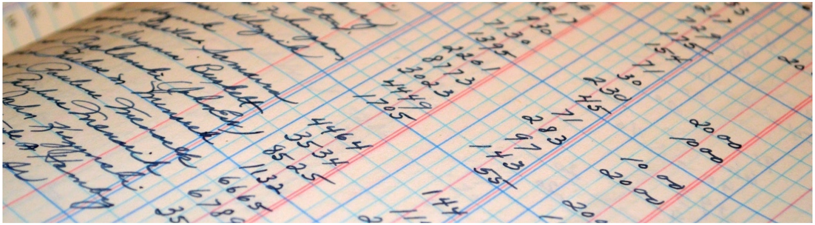
Blockchain = a digital ledger



	Alice	Bob	Charlie
In the beginning, Alice has 100 coins and Bob 50			
Alice pays 10 coins to Bob for bread			
Charlie repairs the house of Alice for 30 coins			
Alice goes to Bobs and buys coffee for 1 coin			

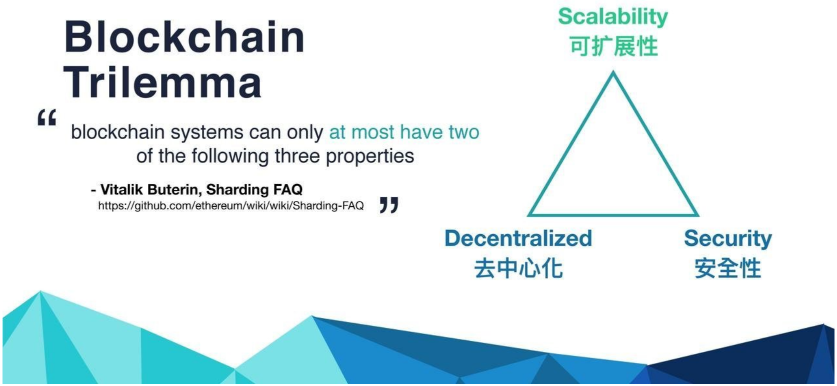
How many coins do Alice, Bob and Charlie have in the end?

Blockchain = a digital ledger



	Alice	Bob	Charlie
In the beginning, Alice has 100 coins and Bob 50	100	50	0
Alice pays 10 coins to Bob for bread	90	60	0
Charlie repairs the house of Alice for 30 coins	60	60	30
Alice goes to Bobs and buys coffee for 1 coin	59	61	30

Challenges: Blockchain trilemma



Blockchain Trilemma

“ blockchain systems can only at most have two of the following three properties

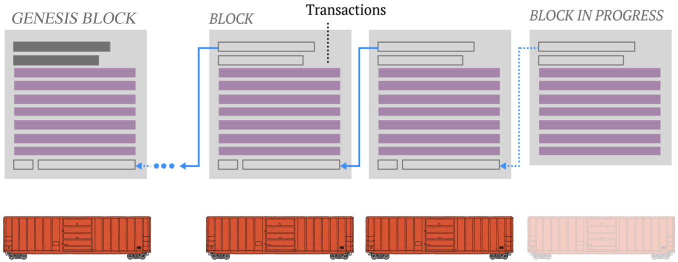
- Vitalik Buterin, Sharding FAQ
https://github.com/ethereum/wiki/wiki/Sharding-FAQ ”

Security

Goal: Make it impossible to "rewrite history"
Why?: Change history ↔ steal (all) money

Solution

- Start from *genesis block*
- Concatenate *all(!)* transactions
  via 1-way cryptographic algorithm
- Make it *costly* for a new block to get accepted



Security (2)

How do we make proposing a new block costly?
$\rightarrow$ link to valuable/limited resource

Physical proofs
– proof of work (BTC, ETH 1.0)
– proof of elapsed time (Intel's trused execution environment)
– proof of (disk) space (or retrievability, data possession; Chia)
– proof of space-time (similar, but over some amount of time)
– proof of capacity (Signum)

Economic proofs
– proof of stake (ETH 2.0, . . .)
– pure proof of stake (Algorand)
– proof of authority (Lugano's triple-A blockchain)
– proof of burn

Algorand's consensus algorithm

Pure proof of stake

- Starting point: Bitcoin and ETH 1.0 are energy hungry = dirty
- Goal:
    - Avoid malicious attacks
    - Without requirement for costly energy burning

The idea

- The Byzantine General's problem:
  Whom can I trust? Who has been bribed?
- If 51% of nodes are compromised,
  a majority vote would confirm fake transactions.
- So don't do a pure majority vote!
    - Randomly choose nodes that participate in a majority vote
    - Attacker does not know whom to bribe
    - Threshold requirement for compromised nodes increases

Consensus algorithms – discussion

Desirable properties of consensus algorithms

- Robustness against attacks
- Minimize waste (minimize signaling)
- Provide a competition-friendly framework for participants
- Incentivize participation in network (network effects)
- Cost efficient incentivisation to provide common goods
  - Provision of infrastructure (work/capacity/connectivity)
  - Participation in consensus
  - Participation in governance

The three cryptographic problems

- Encrypting
- Signing
- Hashing

Encrypting – keep a secret

- Keep a message from Alice to Bob secret, even if transported over public channel.
    - Alice encrypts using the public key of Bob.
    - Bob decrypts using this private key.
- Usually **not** used on the Blockchain
- Example: PGP algorithm

The three cryptographic problems (2)

Signing – ensuring authenticity/identity

- Ensure authority of sender of a message/transaction
  - Alice signs a message using her private key.
  - Bob can verify the authenticity using the public key of Alice.
- Verify authorisation of blockchain transactions
- Example: Ed25519 signature algorithm

Hashing – ensure immutability

- Ensure immutability of blockchain
  - Alice calculates the hash of a message and communicates it securely to Bob.
  - Bob calculates the hash of the received message and verifies that it has not been modified.
- Link blocks on the blockchain
- Example: SHA-256 algorithm

Language

- **Blockchain:** a digital ledger or growing list of records/blocks, that are linked together using cryptography.
  Properties: decentralized, distributed, often public.
- **Key pair:** Two keys (public–private) in public key cryptography.
- **Address:** Public key, encoded for better readability.
- **Mnemonic:** Set of English words representing the private key.
- **Wallet:** Collection of several (public/private) keys.
- **Hash function:** Injective function that is difficult to invert.
- **Digital Signature function:** Injective, verifiable cryptographic signature. Sign with private key, verify with public key.
- **Token:** Class of entries in a blockchain that represents an absolute claim, often ownership.
- **Layer 1:** Base or infrastructure layer of a blockchain, e.g. Bitcoin, Ether, Algorand.
- **Smart Signature:** Logic that can approve (or not) a proposed transaction.
- **Smart Contract:** Logic that can interact (read/write) with the blockchain.

Official terms and definitions: `https://www.federalregister.gov/d/2022-05471/p-58`

A brief history of Algorand

- 2017 work started; goal to improve over Bitcoin's inefficiencies
- 2017 company founded by Silvio Micali (MIT, Turing price)
- 2018-Feb 4M USD seed funding from Pillar, Union Sq. Ventures
- 2018-July Launch of testnet
- 2018-Oct 62M USD venture capital funding
- 2019-May launch of Algorand University program
- 2019-June auction of ALGO token
- 2019-July launch of mainnet
- 2020-Feb first version of PyTEAL
- 2021 First carbon negative blockchain

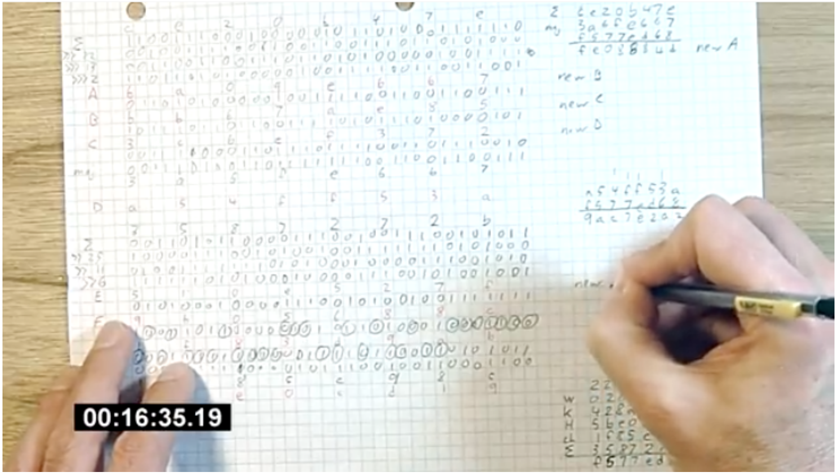
See also
`https://arxiv.org/abs/1607.01341`
`https://dl.acm.org/doi/10.1145/3132747.3132757` (first video)
`https://www.algorand.com/about/our-history`
`https://www.algorand.com/about/sustainability`
`https://github.com/algorand/pyteal/releases`

Algorand and USI

Global University Program
Massachusetts Institute of Technology
Stony Brook University
University of Waterloo
Università degli Studi di Salerno
Sapienza Università di Roma
Tel Aviv University
Università della Svizzera italiana
北京大学 Peking University
Berkeley University of California
清華大学 Tsinghua University

- USI joined the Algorand Global University Program in May 2019
- Algorand supports the USI foundation
- USI runs a validation node
- USI collaborates on research and teaching

SHA-256 Hashing Algorithm



How to calcualte an SHA-256 hash by hand in 16 minutes
https://www.youtube.com/watch?v=y3dqhixzGVo