Introduction to Blockchain Technology and Smart Contracts

2022-2023

1. Introduction

Jesús Correas, Albert Rubio

Departamento de Sistemas Informáticos y Computación
Universidad Complutense de Madrid

Introduction to decentralized systems. Contents

- Dealing with **trust.**
- Examples.
- Centralized systems and their vulnerabilities.
- Distributed ledger technology.
- A payment system without a supervisor. Double spending problem.
- Decentralized network and consensus.
- Attacks to DLT: DDoS, Sybil, 51 % attacks.
- Types of blockchains: Permissionless vs. permissioned.
- Basic components: hashes, cryptography.

Dealing with trust (in a trustless world)

- Blockchain technology started solving some problems related to digital cash systems and digital currencies.
  - Without physical existence.
  - Tamper- and fraud-resistant.
  - Relying on the computational infrastructure to guarantee its **correct behaviour by construction.**
- However, its applications are much wider: for supporting **trustworthy** systems of any kind.

Examples

- We are constantly relying on **trust** in our lives.
- We trust the banking companies where we have our deposits of money,
- We trust that our physical cash preserve their value,
- But we also trust our health system when we are prescribed some medication or treatment,
  - And we trust the protocol for approving medicines,
  - And we trust the pharmaceutical industry, etc.
- Industrial and business processes also require trust as a main ingredient:
  - We trust that transport companies preserve the cold chain of food and medicines.
  - We rely on our mechanic to obtain original parts for our car repair, etc.
- Also intangibles: university degrees, e-recycling, authenticity, blood diamonds, reputation in general, etc.

Centralized systems and their vulnerabilities

- In many cases we rely on a central agency or institution (our savings bank, the European Central Bank, the health system, etc.)
- This centralized approach has important drawbacks:
  - The central agency can be attacked.
  - A failure in that single point can break the whole network.
  - Data leaks may expose sensitive data of a very large number of people.
  - Physical tokens can be forged (banknotes, university degrees, etc.)
  - Users are not anonymous (this may be seen as a feature.)
- There is another approach for handling trust:

distributed ledger technology (DLT)

- Blockchain is based on this technology.

Distributed ledger technology

- The fundamental idea is to provide a **distributed ledger** that keeps a registry of operations.
- The approach of **DLT** is based on two elements:
  - Use a **ledger** (an account book) to record all operations (called **transactions**) performed on the system.
  - This ledger is **immutable:** existing transactions cannot be manipulated (secured with cryptographic techniques).
  - All transactions are **signed** by their parties.
  - No central organization: use of a **network of nodes** instead.
  - Ledger is replicated in all nodes that agree on its contents by a **consensus mechanism**
- This technology was first used in the context of electronic payment systems: Bitcoin.[1]

[1] S. Nakamoto, *Bitcoin: A peer-to-peer electronic cash system*, 2008. `http://bitcoin.org/bitcoin.pdf`

A payment system without a supervisor. Double spending

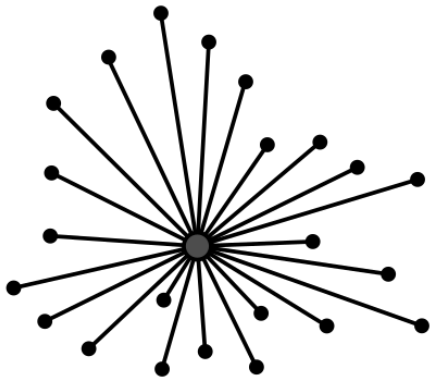
- Bitcoin was the first system that successfully addressed the most relevant problem related to a DLT-based payment system: the **double spending problem:**

> The **double spending problem** is a flaw in a digital cash scheme in which the same single digital token can be spent more than once.
>
> From `https://en.wikipedia.org/wiki/Double-spending`

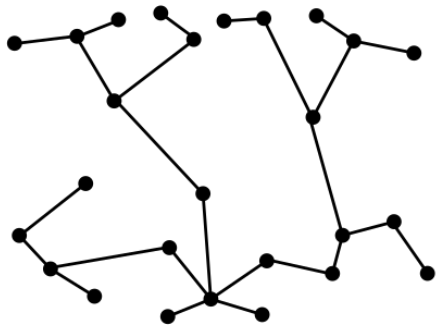
- The double spending problem is traditionally prevented by means of a central trusted third party:
  - It can verify whether a token has been spent.
  - However, this represents **a single point of failure.**
- DLT has a radically different approach.

Decentralized network and consensus

- In a centralized system it is crystal clear *where the truth lies.*
  - If there are inconsistencies in the data of some components with respect to the central node, the latter is always right.
- In a decentralized system things are not so simple: there is *no absolute truth.*
- The nodes in a DLT network have a procedure **to reach an agreement** on the global state, which is **the network truth.**



CENTRALIZED



DECENTRALIZED

Source: Kes47 (?), CC BY-SA 3.0 via Wikimedia Commons

Decentralized network and consensus

- This procedure is based on a **consensus** among the nodes in the network.
- In this approach, nodes in the network vote for what each node considers to be the truth.
- They can change their minds if they receive updated information from other nodes.
- The **Byzantine generals problem** is a typical vulnerability of these systems.

Decentralized network and consensus

The Byzantine Generals Problem[1]

"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.
B. A small number of traitors cannot cause the loyal generals to adopt a bad plan."

[1] L. Lamport et al. The Byzantine Generals Problem. *ACM TOPLAS*, 1982.

Decentralized network and consensus

- This problem was initially described to design a reliable computer system even when one or several of its components fail.
- There are several complex algorithms to solve this problem (for example PBFT).
- In a public DLT (as Bitcoin or Ethereum) anyone can participate as a node: a perfect scenario for this problem.
- Both Bitcoin and Ethereum (1.0)[2] use a consensus mechanism based on **proof of work.**
- This blockchain solution provides some level of **byzantine fault tolerance,** although it is not immune.

[2] Ethereum moved to a Proof-of-Stake consensus on 15 Sep 2022.

Attacks to DLT: DDoS, Sybil, 51 %

- There are several kinds of attacks, both general to payment systems and specific for spending on DLT systems.
- The most relevant are:
  1. Distributed Denial of Service (DDoS).
  2. The **sybil attack.**
  3. The **51 % attack.**

Distributed DoS attack

- A **DDoS** attack intends to slow down or collapse a system.
- A malicious user tries to flood a server or network with requests and traffic.
- a blockchain server can lose connectivity to any crypto exchanges, online crypto wallets, or any other connected applications.
- Several attacks to blockchain networks have been reported, including an attack to the Ethereum mainnet.
- This is a typical attack to centralized systems, since they have a single point of failure.
- The decentralized nature of DLTs make these attacks harder:
  - A DLT can be further decentralized with more nodes: the attacker has to attack a large number of nodes simultaneously.
  - Other mechanisms, like *gas* in Ethereum, can be modulated to face DDoS attacks.

Sybil attack

- A fundamental weakness of anonymous decentralised systems: they have no arbiters representing **the ground truth.**
- Nodes use a procedure based on a **consensus** to reach an agreement on the global state, **the network truth.**
- The **sybil attack** consists in using a number of malicious nodes (or *identities*) in the network trying to control it.
  - The system is doomed if the number of malicious nodes is large enough.
  - Its effectiveness depends on how cheap is to create multiple identities.
  - It attacks the systems where the consensus is based on **one-node-one-vote**.
- There are some approaches to prevent sybil attacks.
- DLT systems use **identity validation** (*e.g.*, in permissioned DLT) or making voting expensive, *e.g.*, **proof of work.**

51 % attack

- The **51 % attack** consists in using attacker nodes with more than 50 % of the computing capacity of the network.
- These nodes are capable to generate data (blocks) with fraudulent transactions at a higher rate than the rest of the network.
- Open (*permissionless*) DLT systems are not immune to these attacks.
- In particular, **proof of work**-based systems are vulnerable to it, as we will see in next lesson.
- There have been real cases of this attack:
  - Gold (a fork of Bitcoin) suffered 51 % attacks in 2018 and 2020.
  - Ethereum Classic was attacked three times in a month in summer 2020.
- Some systems limit the cpu capacity of some participants in the network to reduce this kind of attacks.
- Other approaches such as **proof of stake** can also prevent this type of attacks.

Types of blockchains: Permissionless vs. permissioned

- **permisionless** (or public) DLTs are those DLTs in which anyone can participate in the system and become part of the network.
  - Participants do not have to identify themselves to enter.
  - Bitcoin or Ethereum are permissionless DLTs: anyone can make transactions or act as nodes of the network.
- In a **permissioned** (or private) DLT only those participants that have permission can access the system.
- Both types use the same technology, although there are some peculiarities specific to some of them:
  - Both use a **consensus** mechanism, although most current public DLTs are based on **proof of work.**
  - This makes permissionless DLTs very slow and much less scalable.
  - Permissioned DLTs often use more lightweight mechanisms such as **proof of stake.**

Types of blockchains: Permissionless vs. permissioned

- Examples of permissionless blockchains:
  - `https://bitcoin.org/`
  - `https://ethereum.org/`
- Examples of permissioned blockchains:
  - `https://www.everledger.com/`
  - `https://www.ibm.com/blockchain/solutions/food-trust`
  - etc. etc.
- Initiatives and platforms for implementing blockchain technologies:
  - EU EBSI:
    `https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/EBSI`
  - Open source platform from the Linux Foundation (and support from companies such as IBM):
    `https://www.hyperledger.org/`

Basic components.

- Hash algorithms.
- Asymmetric cryptography.

Basic components: Hash algorithms.

- DLT systems heavily rely on cryptographic technology.
- We will briefly see the most relevant techniques used.
- The first technique is the use of **hash algorithms.**
- A **cryptographic hash function** maps any piece of data to a bit array of fixed length with some properties:
  - **Deterministic.**
  - **Quick** to compute (low computational cost).
  - **Not reversible:** infeasible to obtain the data that produced the hash value (**"one-way hash function"**).
  - **Collision resistant:** Infeasible to produce two pieces of data with the same hash.
  - **Avalanche effect:** small changes to the input data produce a completely different result.

Basic components: Hash algorithms.

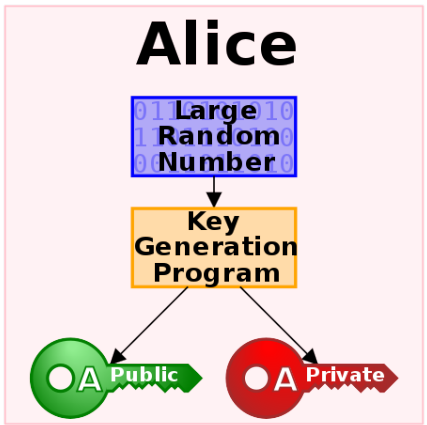
- **Example:**
  - **`sha256("hello, world!")`** returns:
    `0x68e656b251e67e8358bef8483ab0d51`
    `c6619f3e7a1a9f0e75838d41ff368f728`
  - In contrast, **`sha256("Hello, world!")`** returns:
    `0x315f5bdb76d078c43b8ac0064e4a016`
    `4612b1fce77c869345bfc94c75894edd3`
- You can play with SHA: `https://xorbin.com/`
- Hashes are mostly used to **certify that a piece of data** (a *"message"*) **has not been manipulated**, but can also be used for password encryption.
- **SHA** ("Secure Hash Algorithm") is a collection of algorithms developed by the NSA. There are three: SHA-1 to SHA-3.
- **SHA-1** is no longer considered secure, and even **SHA-2** should not be used for password encryption.

Basic components: Hash algorithms.

- Bitcoin uses **SHA-256** (SHA-2 that generates 256-bit hashes)
- Ethereum uses SHA-256 and Keccak-256 (uses the same algorithm as SHA-3).
- Both blockchain systems use hashes a lot:
  - For verifying the integrity of a piece of data: as a **fingerprint** of the data.
    It can be used as a small fingerprint of a very large amount of data (for example, an entire block containing hundreds or thousands of transactions).
  - For implementing the **proof of work** that we will see in next lesson.
  - Addresses on the blockchain are derived from hashing.
  - Ethereum contracts written in Solidity implement mappings as a data structure in which the storage addresses are computed as hashes of the data.

Basic components: Asymmetric cryptography.

- The second basic element is **asymmetric cryptography** (also known as **public-key cryptography**).
- We will not see the internals of this technique, just its use in DLT.
- It was developed in the 70s to overcome the complexities of traditional single-key cryptographic systems:
  - The key had to be exchanged between the communicating parties in some secure way.
- In a public-key system, a pair of keys is generated:



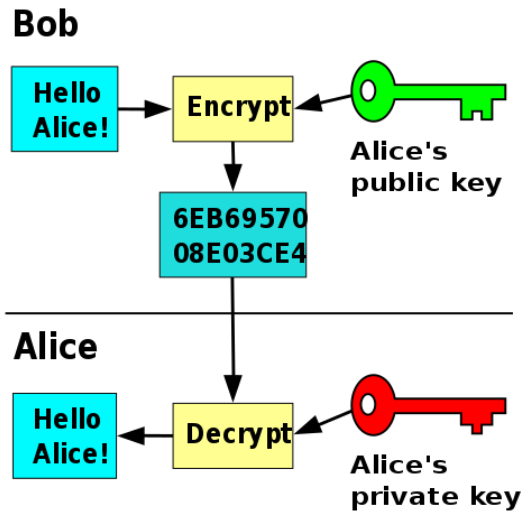
Source: Wikimedia Commons

- You can play with public-key generation here:

`https://andersbrownworth.com/blockchain/public-private-keys/keys`

Basic components: Asymmetric cryptography.

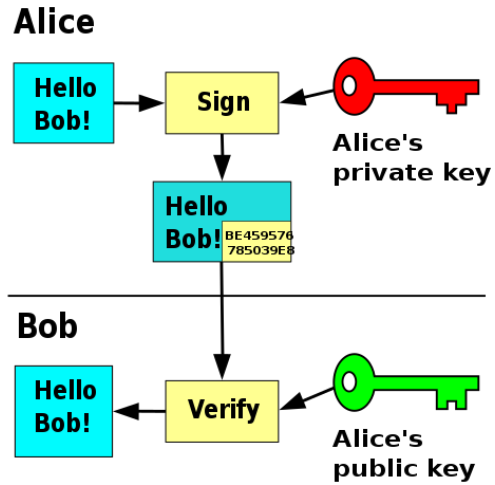
- The **public key** can be openly distributed without compromising security.
- The owner of the keys must keep the **private key** private.
- These keys have a fundamental property: any message encrypted with one of the keys, **can only be decrypted with the other one.**
- That is extremely useful. It can be used for **message encryption:**
- Any person can encrypt a message addressed to Alice, **but it can only be decrypted with Alice's private key.**



Source: Wikimedia Commons

Basic components: Asymmetric cryptography.

- Asymmetric cryptography can also be used for **digital signatures:**



Source: Wikimedia Commons

- Alice can encrypt a document **with her private key.**
- The encrypted document is **appended** as a signature.
- **Anyone can verify** it was signed with Alice's private key: decrypting the signature with Alice's public key and checking that it is identical to the original document.
- This also ensures that the message has not been **tampered with.**
- You can play with signatures here:

`https://andersbrownworth.com/blockchain/public-private-keys/signatures`