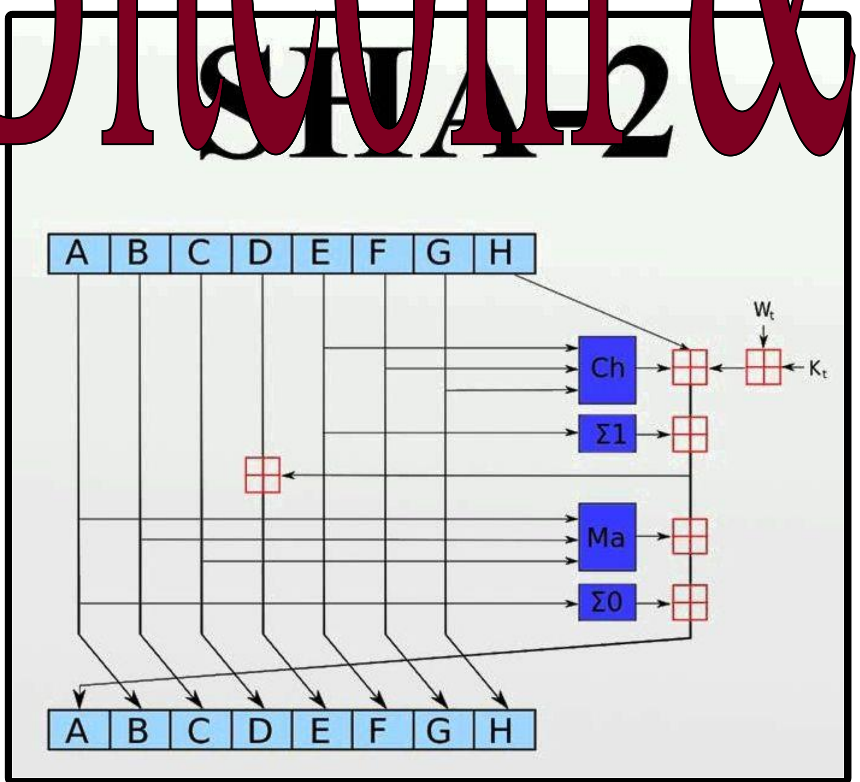
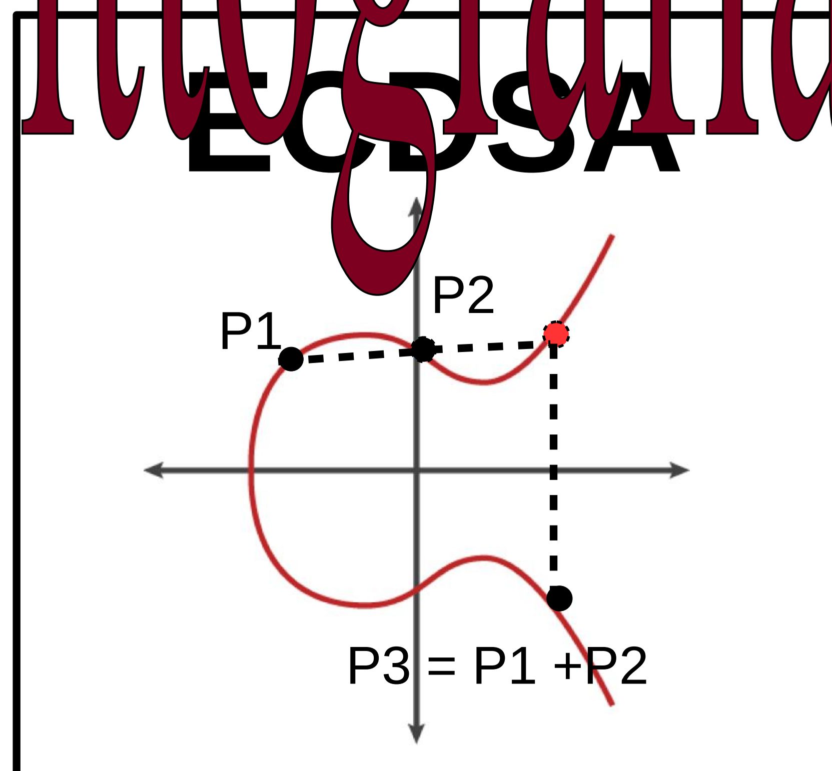
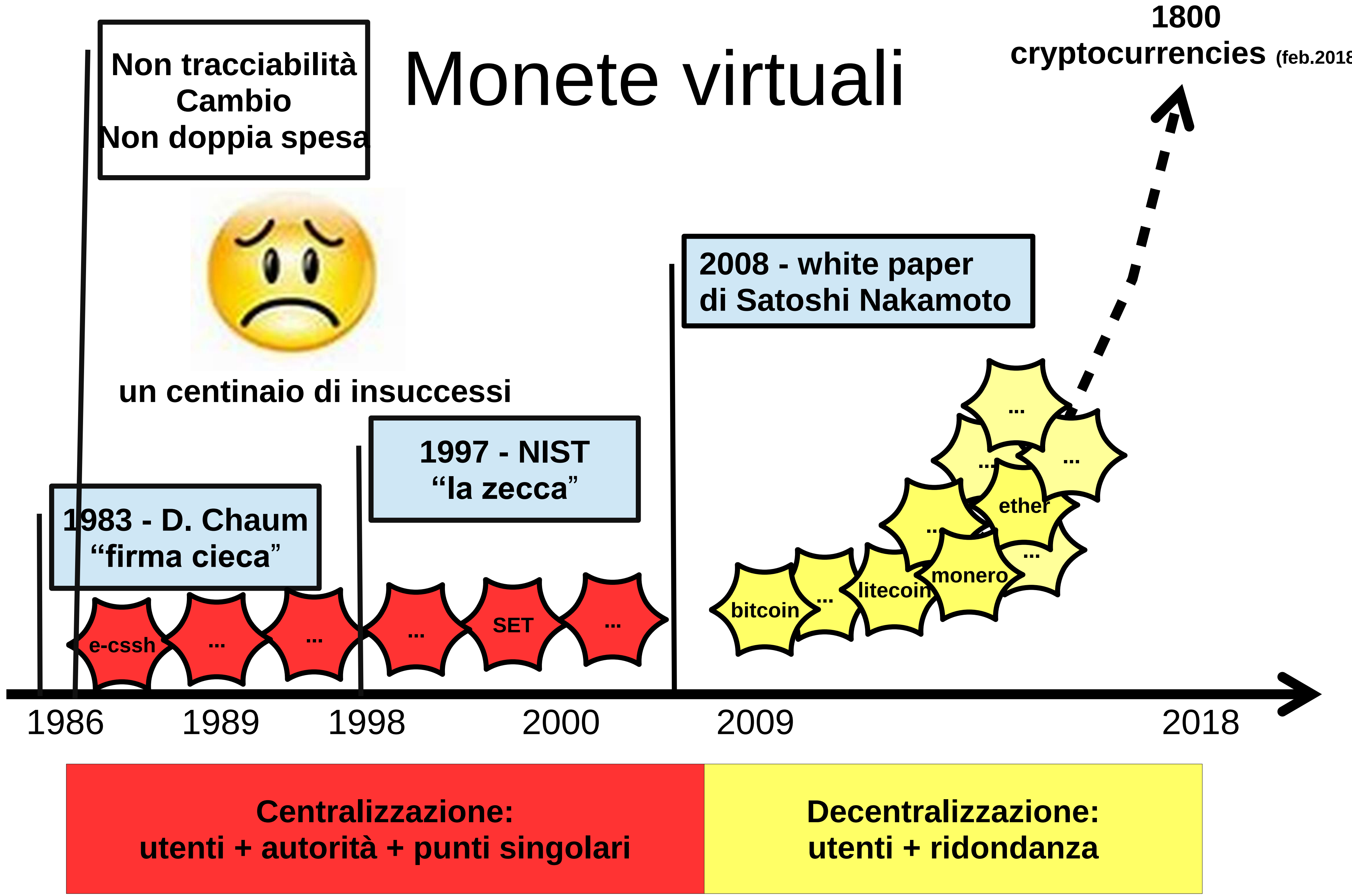
Bitcoin & Crittografia

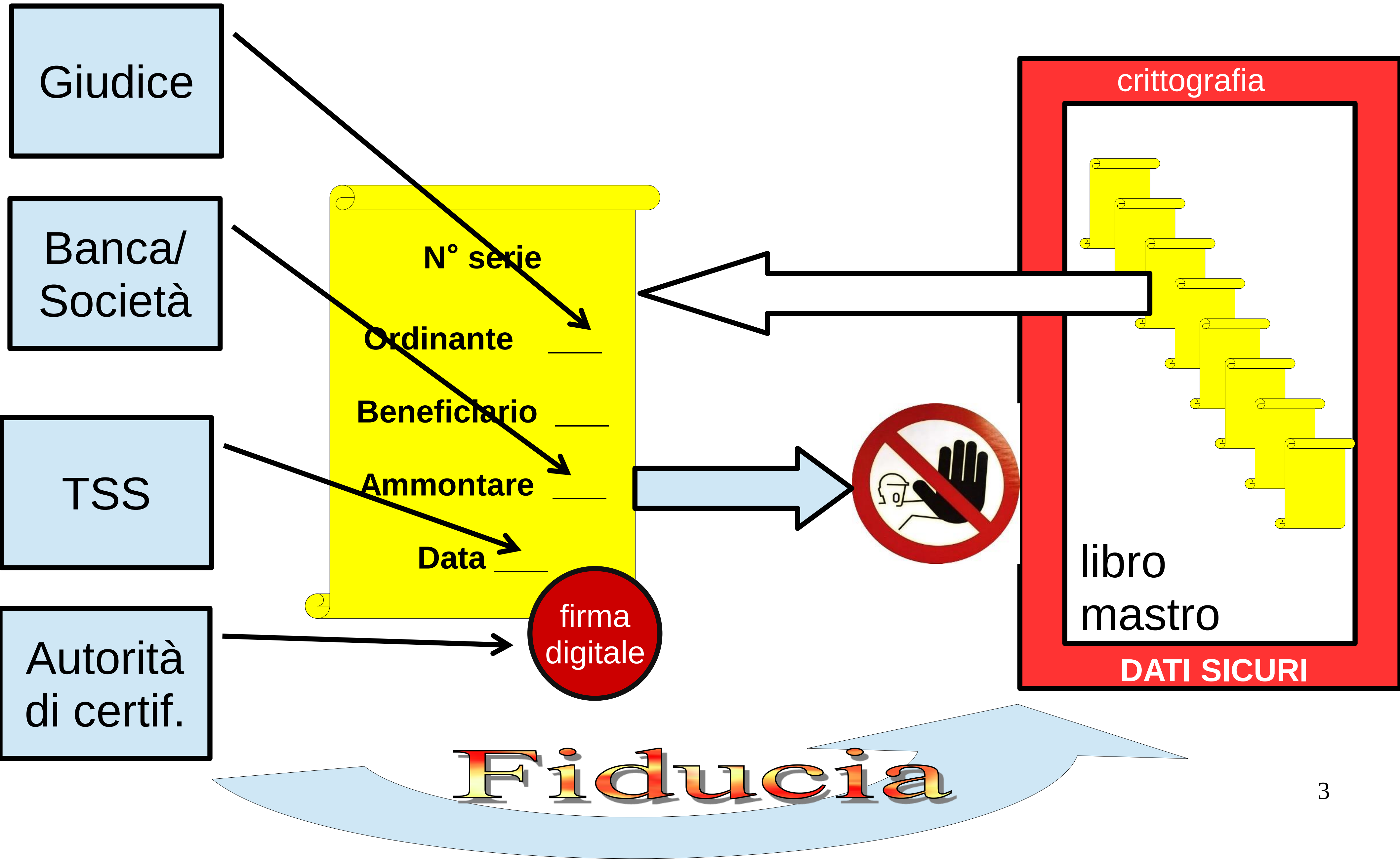




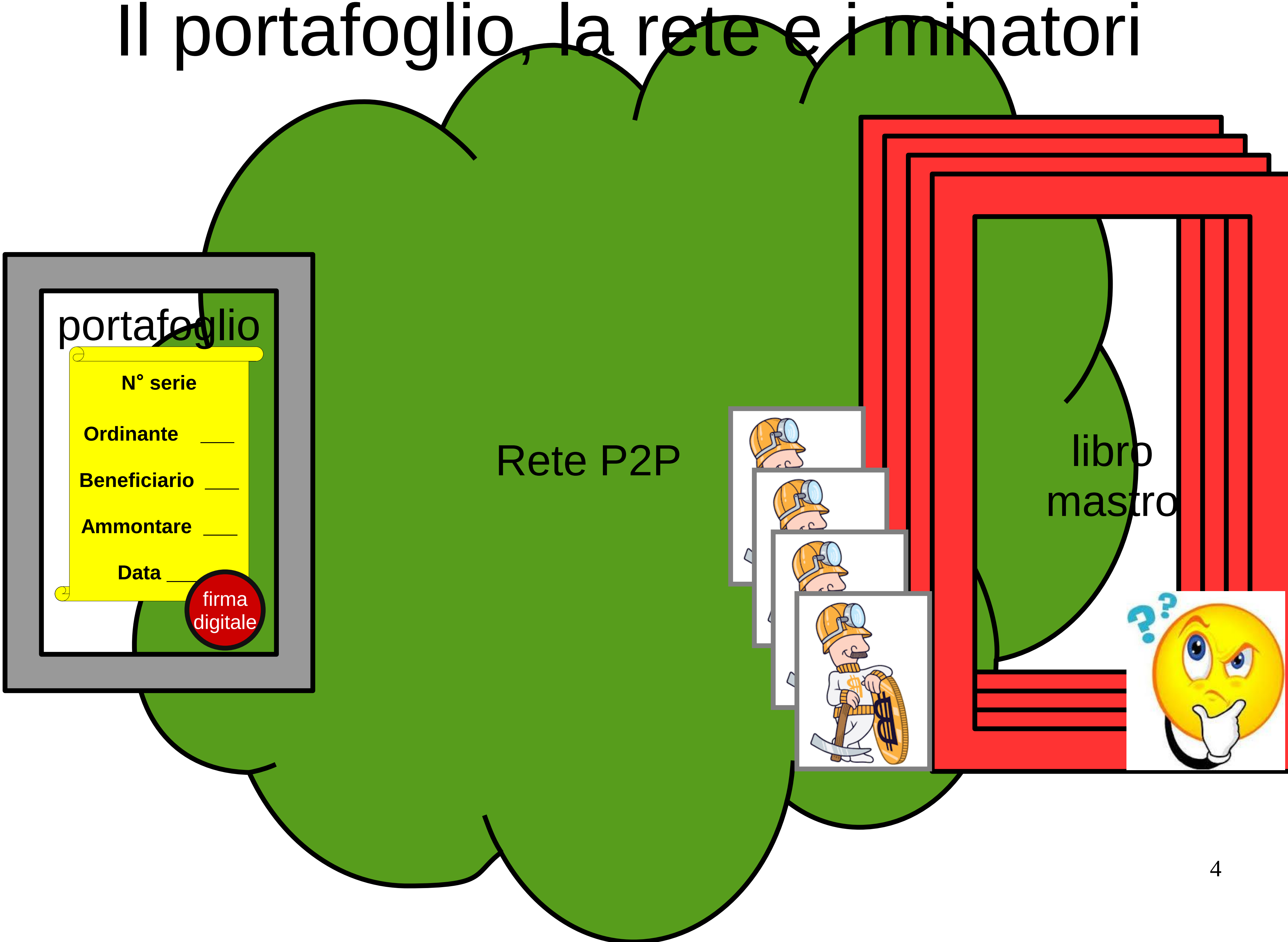
Monete virtuali



autorità fidate *versus* dati sicuri



Il portafoglio, la rete e i minatori

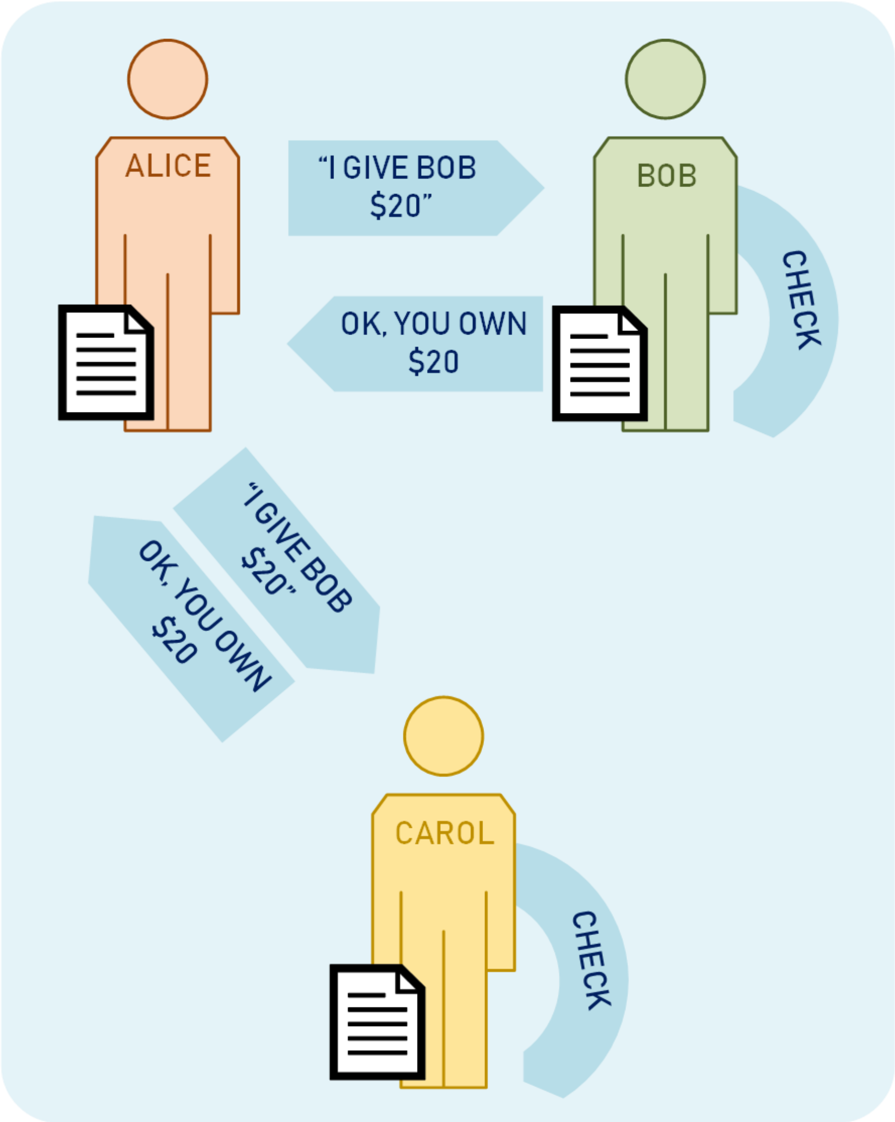


Trust: the new model

The Distributed Ledger

Intuition:

If everyone has a copy of the **ledger** (the book where transactions get registered), it is always possible to verify that **what is promised by one party is true**, without the need for external intermediaries



Trust: distributed ledger

If…

- All participants keep track of each other's current balance
- Each operation is registered by every member of the system
- No operation can be authorized without first being accepted and registered by the other participants

Trust: distributed ledger

Then...

- **Bob can always be sure that Alice has $ 20**
- **Alice can not spend the same money twice *(double spending)***

Systems based on blockchain, such as **Bitcoin** (that we use in the following as a notable seminal example), successfully implement a distributed ledger that can provide those guarantees

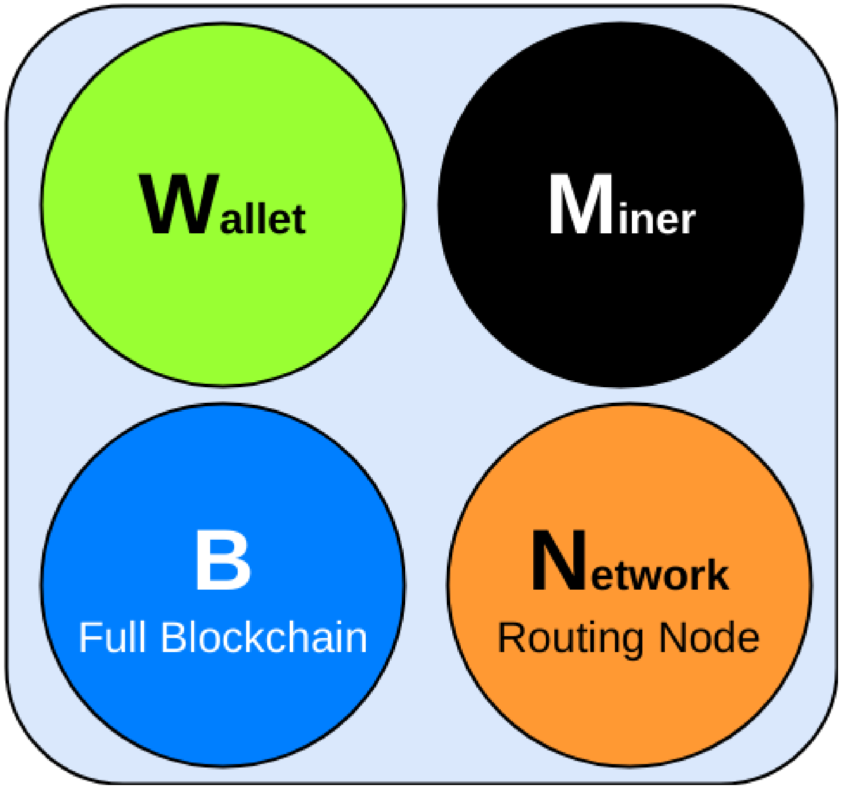
Bitcoin: a first glossary

- **Address:** an alphanumeric string that represents an identity on the ledger, used as a reference for sending currencies; each identity is associated with a **pair of keys** (one private and one public) which allows transactions to be carried out with one's own cryptocurrency
- **Transaction:** represents the sending of cryptocurrency **from one address to another** (may involve more than one address)
- **Block:** the basic entity of the blockchain, which contains a set of valid, immutable and irreversible transactions

Bitcoin: a first glossary (cont'ed…)

- **Network:** is a peer-to-peer network, in which each participant in the network is both a client and a server (generically called **node**)
- **Miner:** node belonging to the network that generates new bitcoins and records new transactions (to do this it must solve the so-called **Proof of Work**, thus acquiring the right to insert **a new block** to the network, as detailed below)
- **Wallet:** node belonging to the network that maintains identities

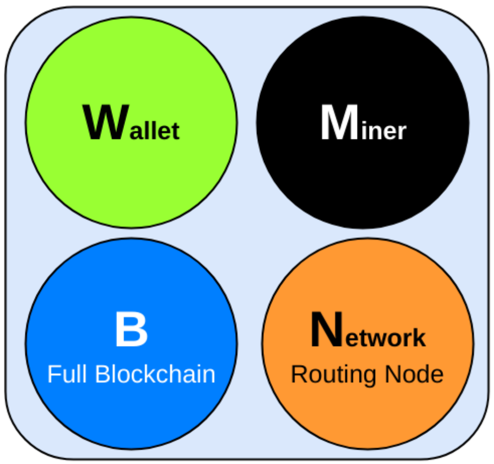
Bitcoin: node functionalities



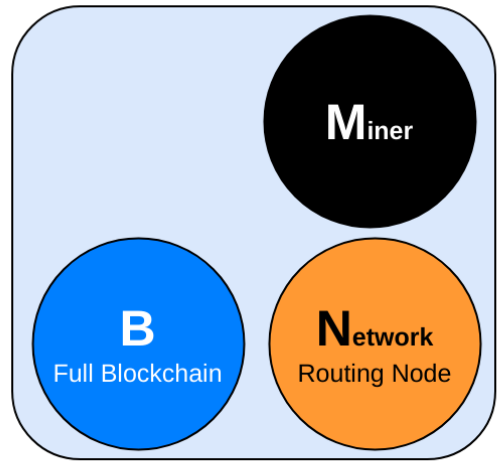
Bitcoin nodes can implement four main functionalities:

- **Wallet:** nodes able to store identities
- **Miner:** nodes able to execute the mining operation
- **Full Blockchain:** nodes that store the whole ledger state
- **Network:** nodes that spread **new blocks** and **transactions** received **over the network**

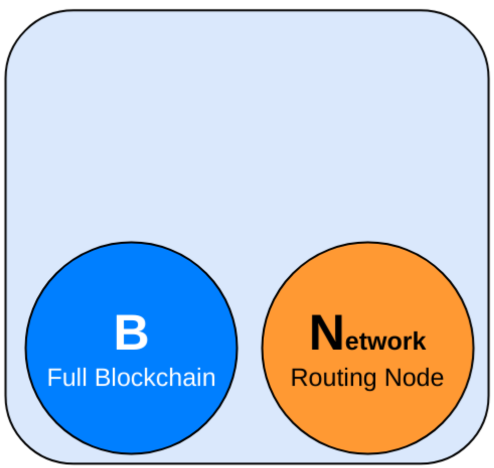
Bitcoin: node roles



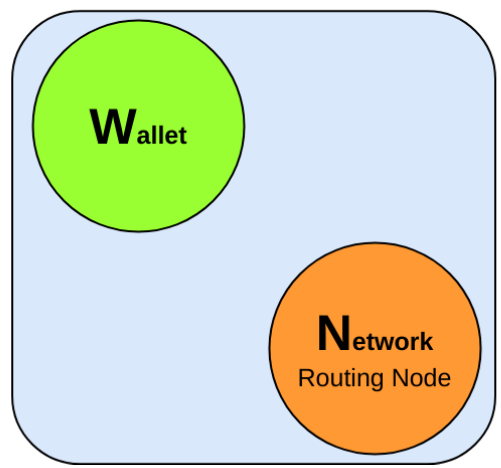
Reference Client (Bitcoin Core)



Miner Node



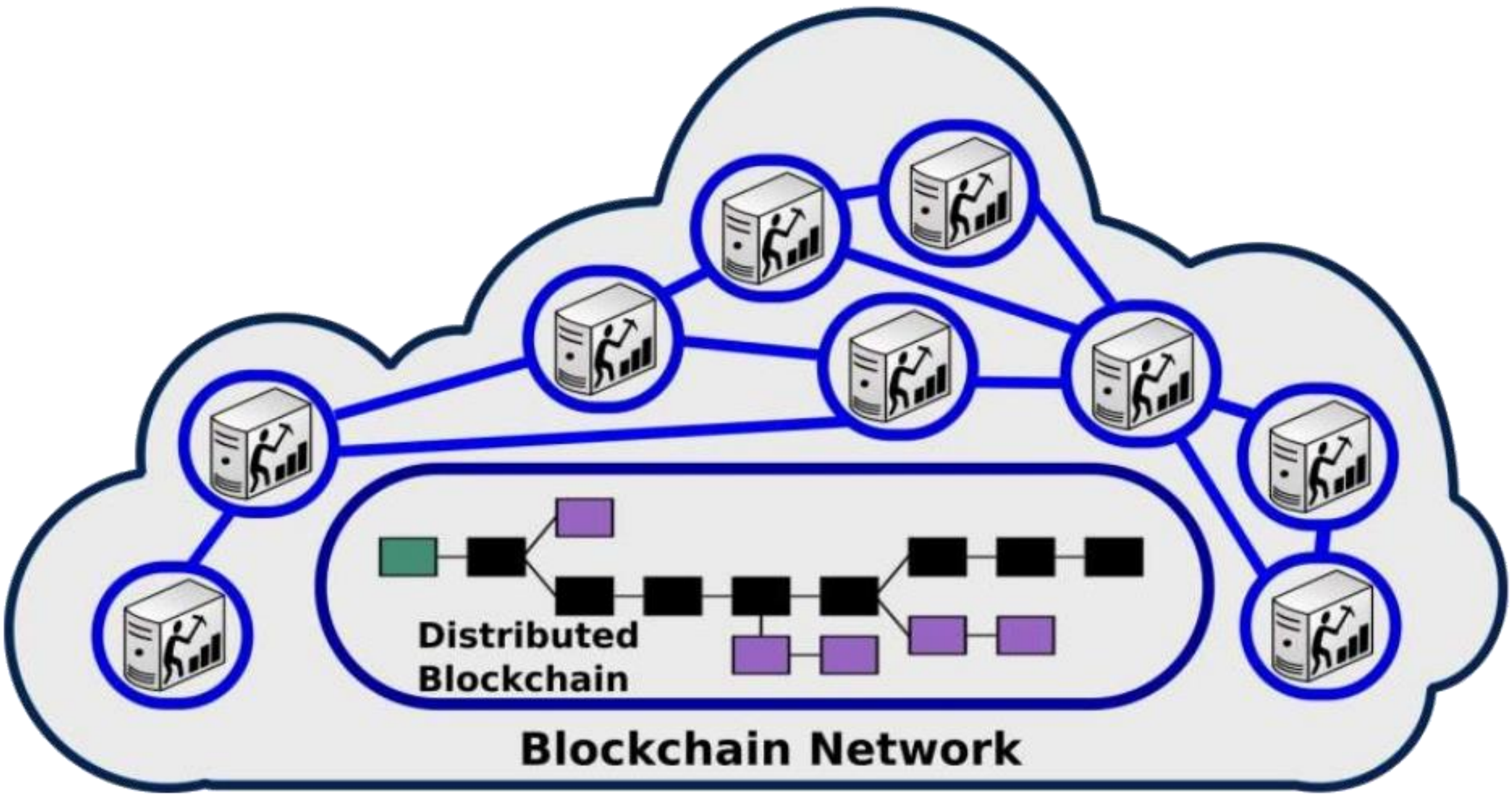
Full Bitcoin Node



Lightweight Wallet Node

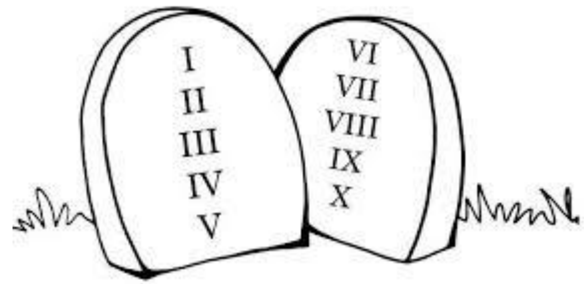
à

Bitcoin Blockchain



*The **miner nodes** collectively collaborate to maintain and add new blocks to the blockchain*

regole di corretto impiego



Utenti interessati
Minatori incentivati

tamper resistant
fault tolerant

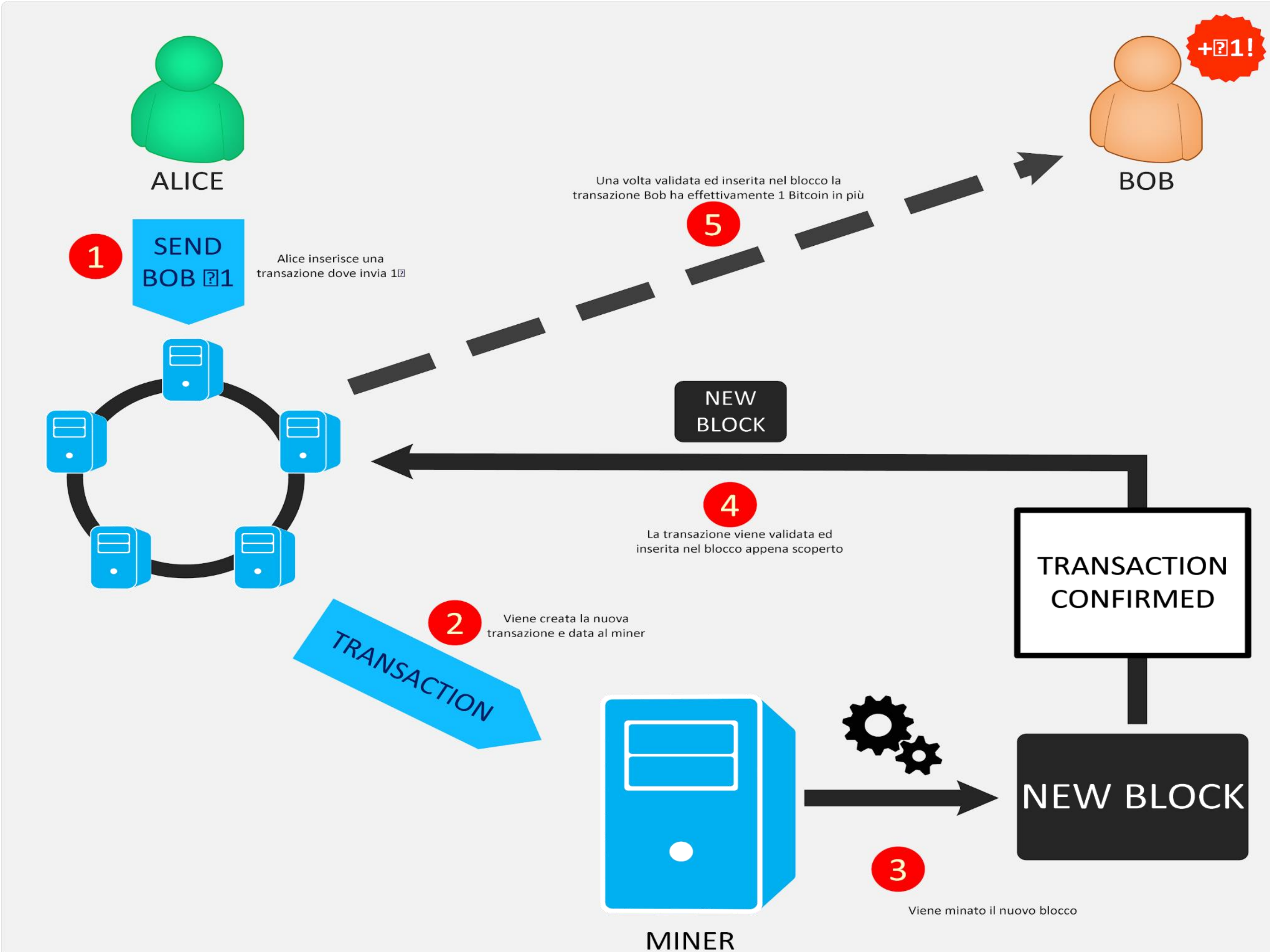
decisioni a maggioranza

Malintenzionati

<50%

Bitcoin: mining and transactions

- When a user wishes to send BTCs to another user, she generates a new **transaction** using **her address** and publishes it, **protecting it with an asymmetric key mechanism**, sending it to the miners she knows that propagate it to the whole network (in addition to the desired amount, a commission fee must be added)
- Miners **collect transactions**, while seeking a value that satisfy the *Proof-of-Work* challenge
  - **The first miner that succeeds in this intent, creates and publishes a new block**
  - The other miners validate the block (verifies the puzzle and the hash pointer) and add the block to their ledger
  - Transactions are **immutably** added to the block, they are an integral part of it and can no longer be changed
  - The miner **earns the reward** for the operation (currently ₿12.5), as well as the **commissions** included in the mined transactions



Miners create new blocks and
add pending transactions to the chain

Bitcoin: *Proof-of-Work*

The search for the value of **nonce** is the Bitcoin *Proof-of-Work* challenge that the miners must win:

$$H(nonce \,||\, prev\_hash \,||\, tx \,||\, tx \,||\, \ldots \,||\, tx) < target$$

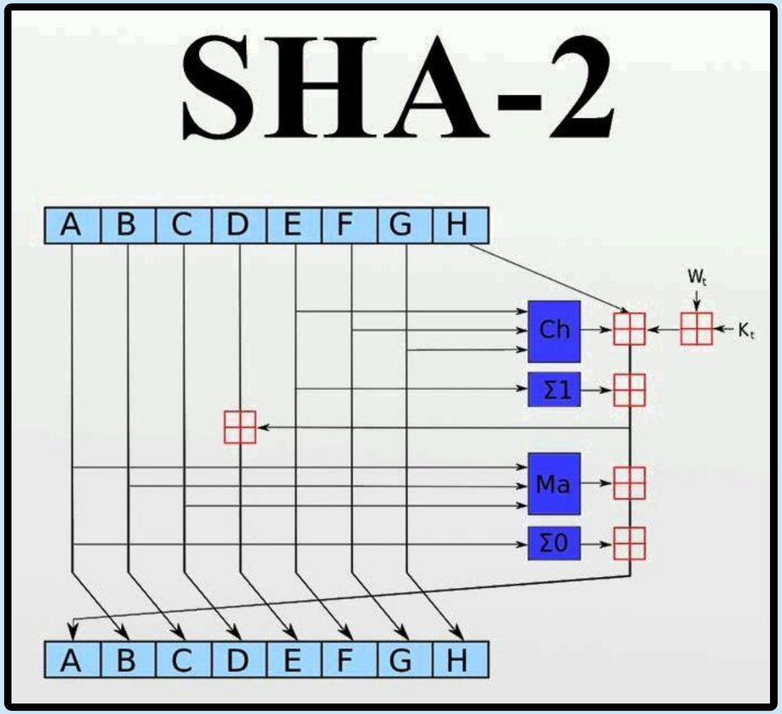
- After adding the transactions it has received, the miner searches for a **nonce** value such that **the hash of the whole block is less than a given value (target)**; this operation requires a lot of resources and luck
- The **level of difficulty** is chosen each 2016 blocks (about 2 weeks) through an algorithm known to all, to **maintain the average rhythm of a new block generation every 10 minutes**

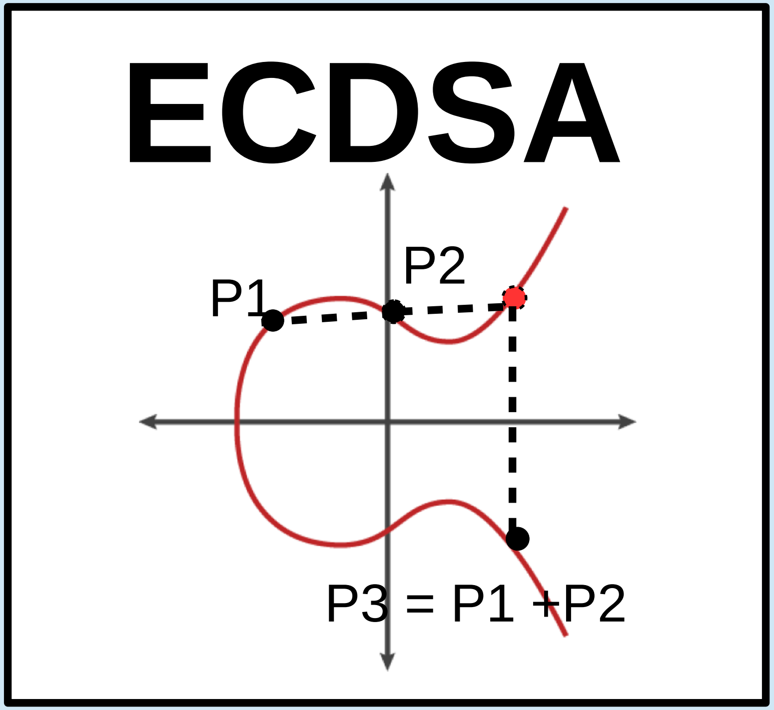
inalterabilità del libro mastro

controllabilità delle transazioni

affidabilità delle chiavi

primitive crittografiche

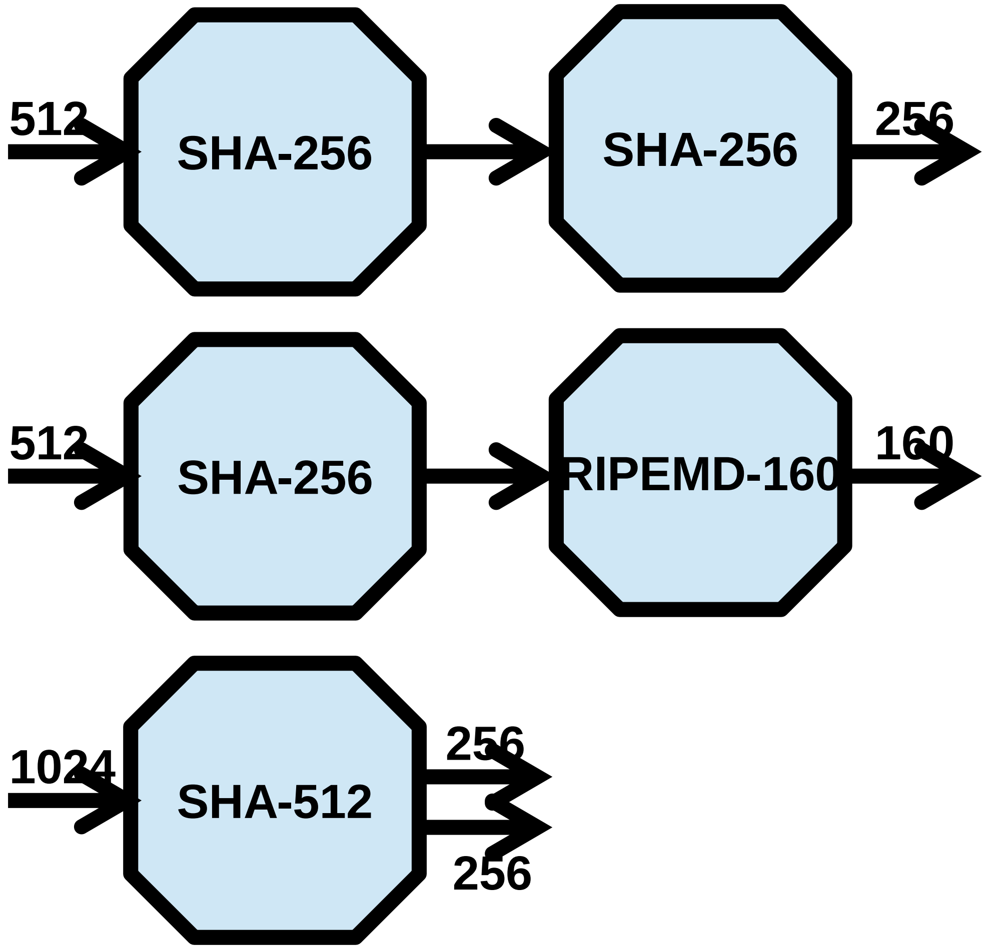




Hash crittografico: problemi e strumenti

- **trovare x,y : $H(x) = H(y)$**
- **dato h, trovare x : $H(x) = h$**
- **dato y, trovare x : $H(x) = H(y)$**

Per SHA-2 e per RIPEMD sono intrattabili: 128 bit è il livello di sicurezza



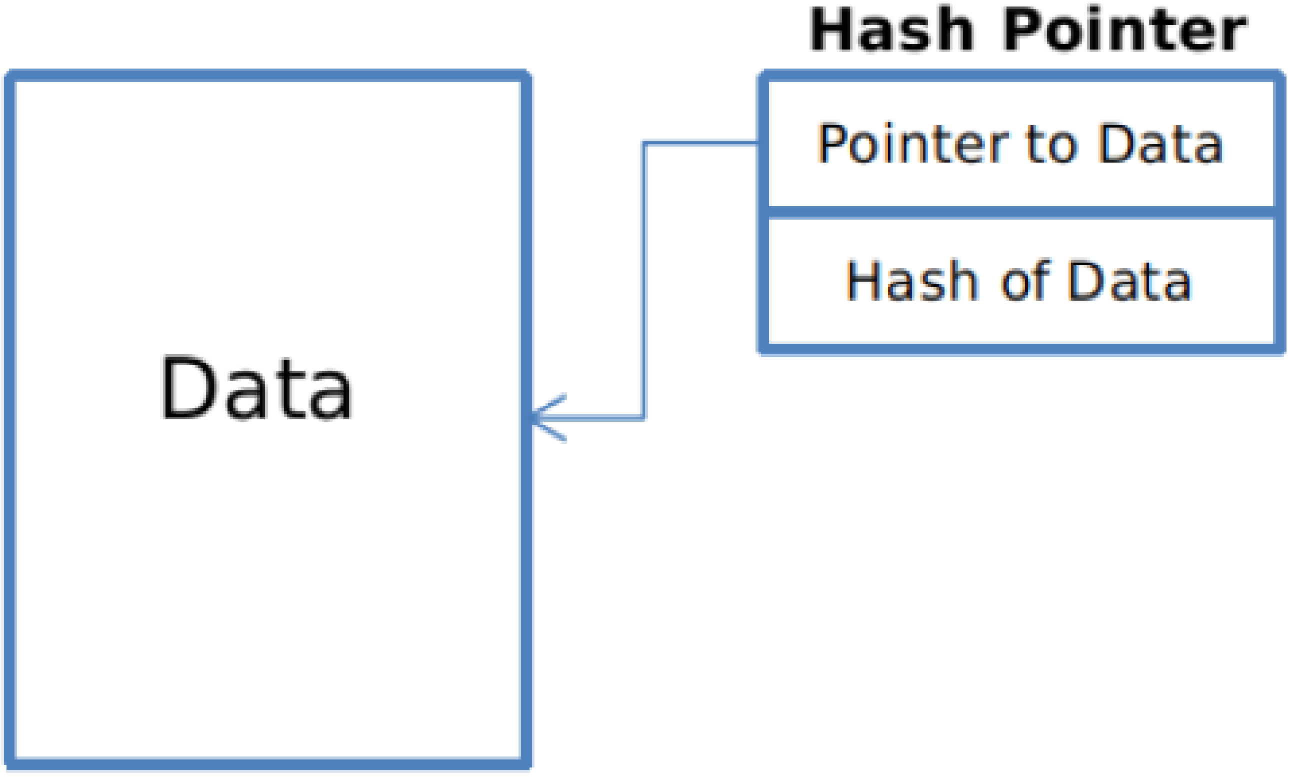
hash pointer
puzzle
checksum
input di ECDSA

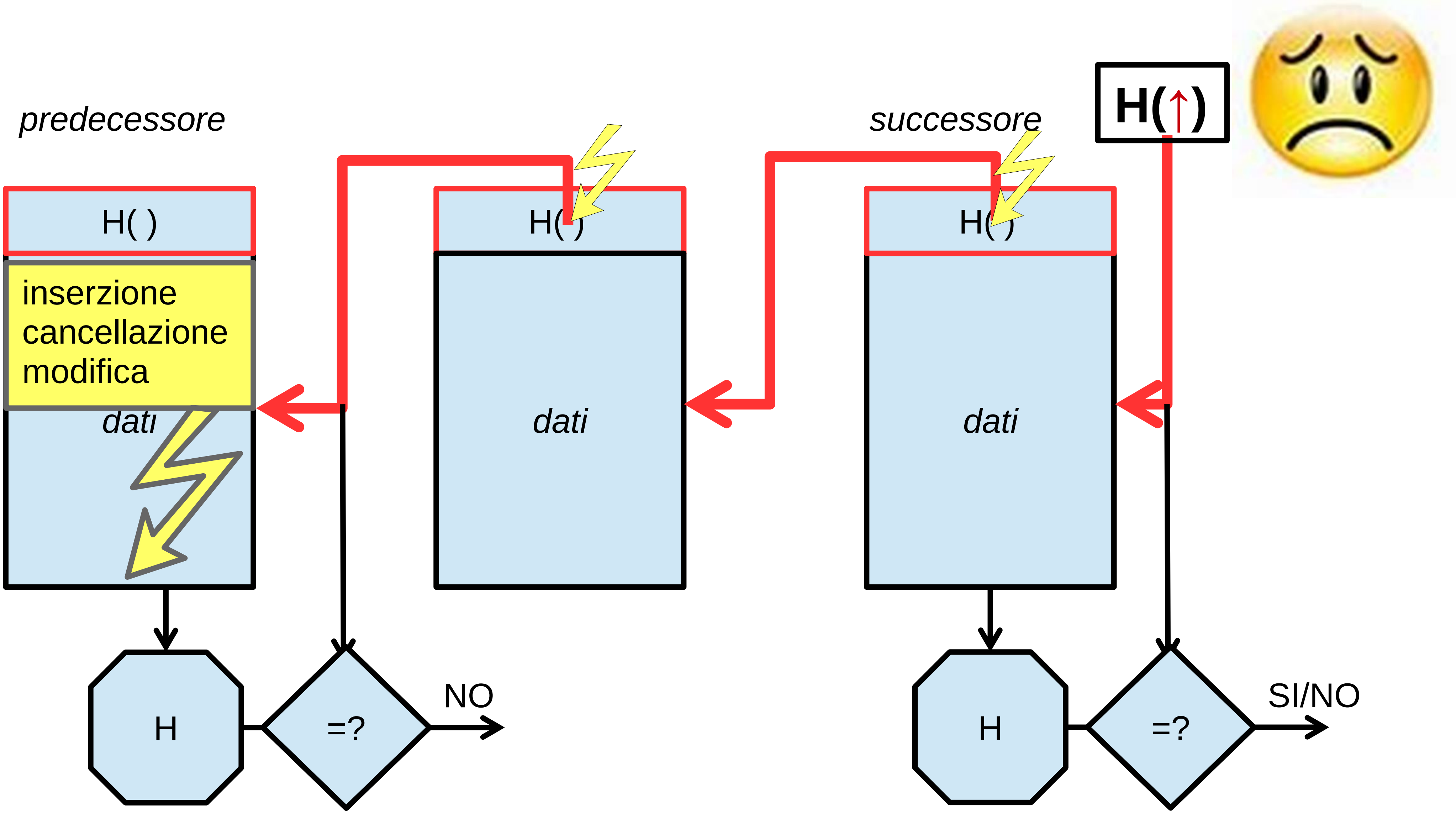
address

key generation

Tamper evidence

Hash pointer (simbolo H(↑)): chiave primaria per accedere a dati e controllarne l'integrità

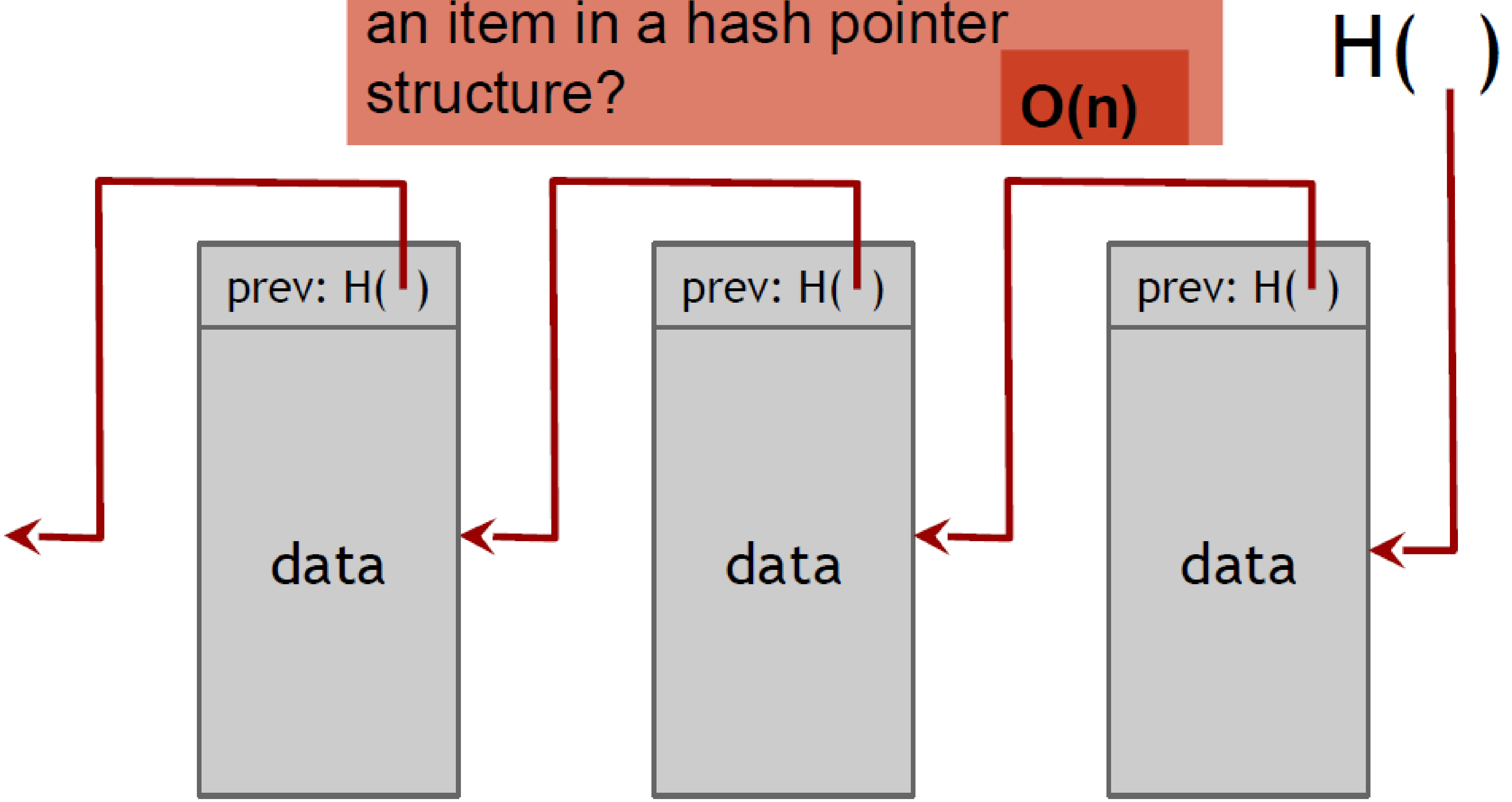




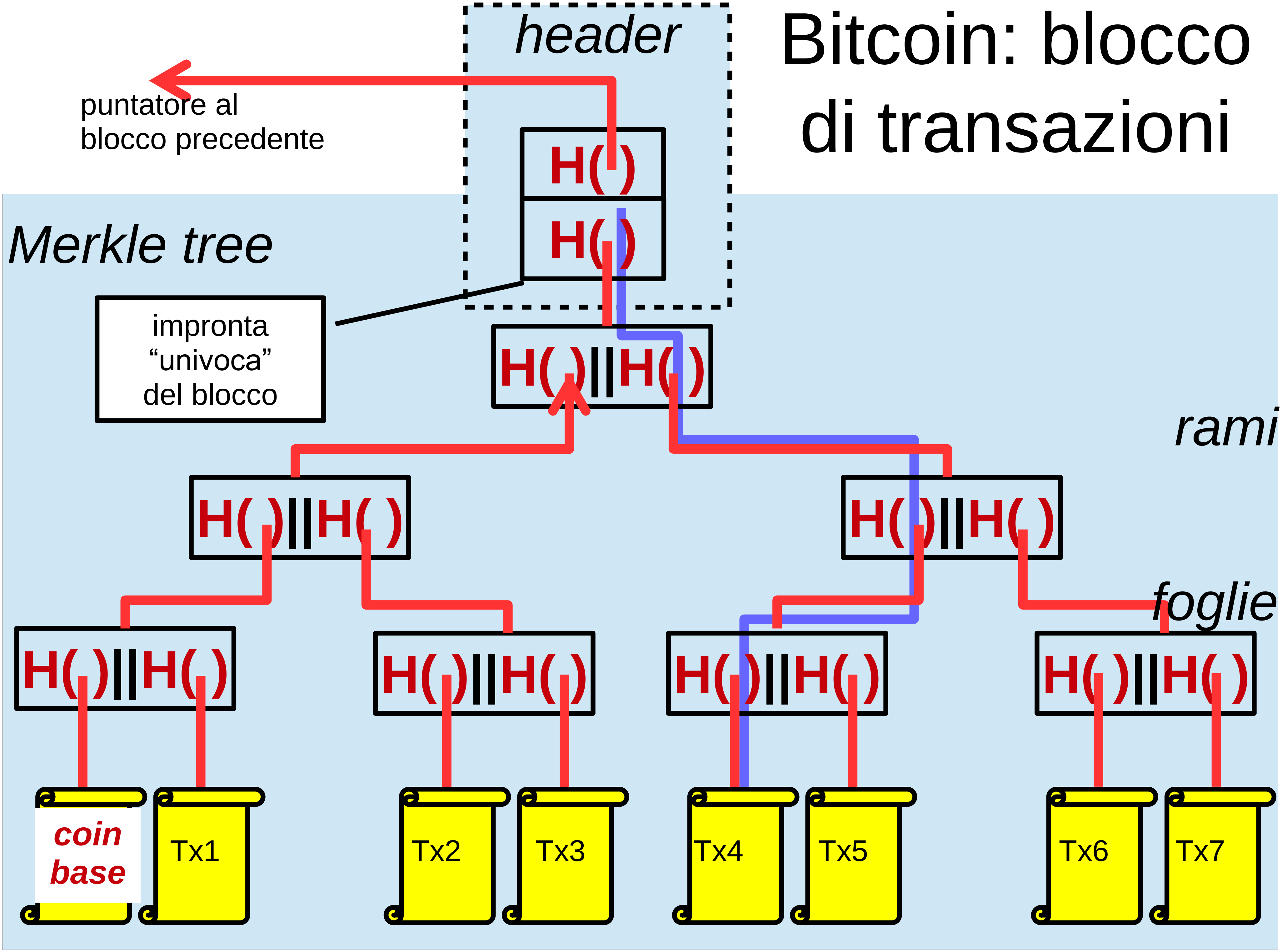
Efficienza: blocchi contenenti più documenti

Given n items, how much does it cost to prove membership of an item in a hash pointer structure?

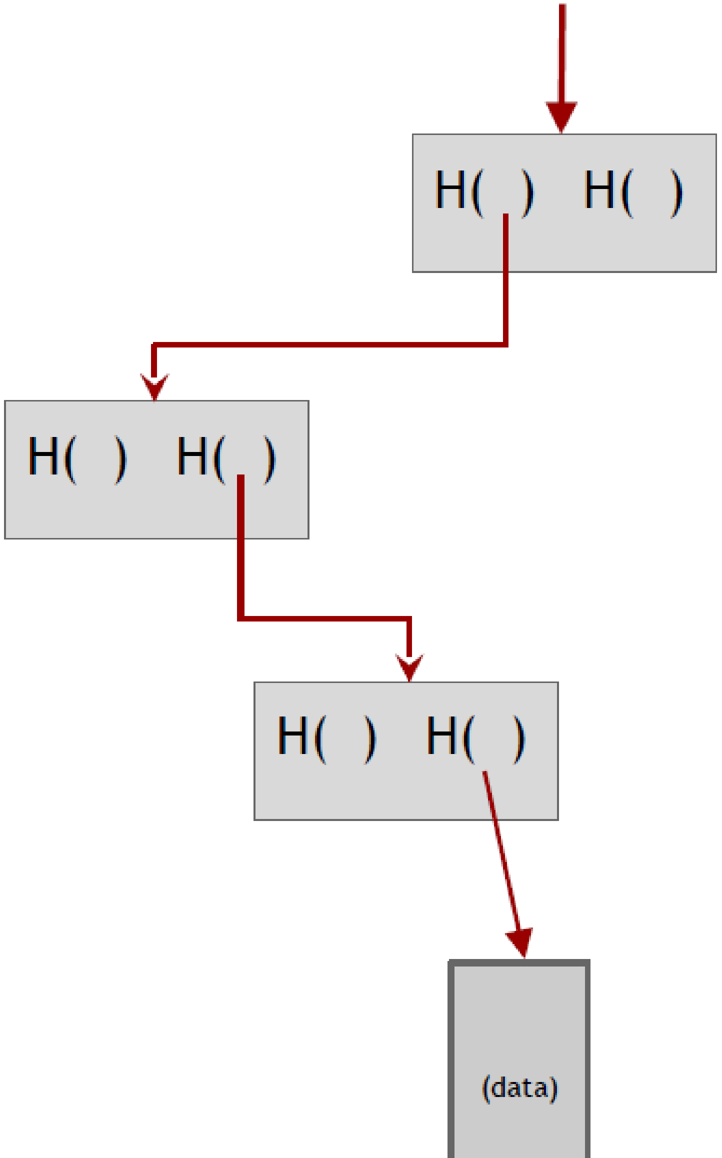
O(n)



Bitcoin: blocco di transazioni



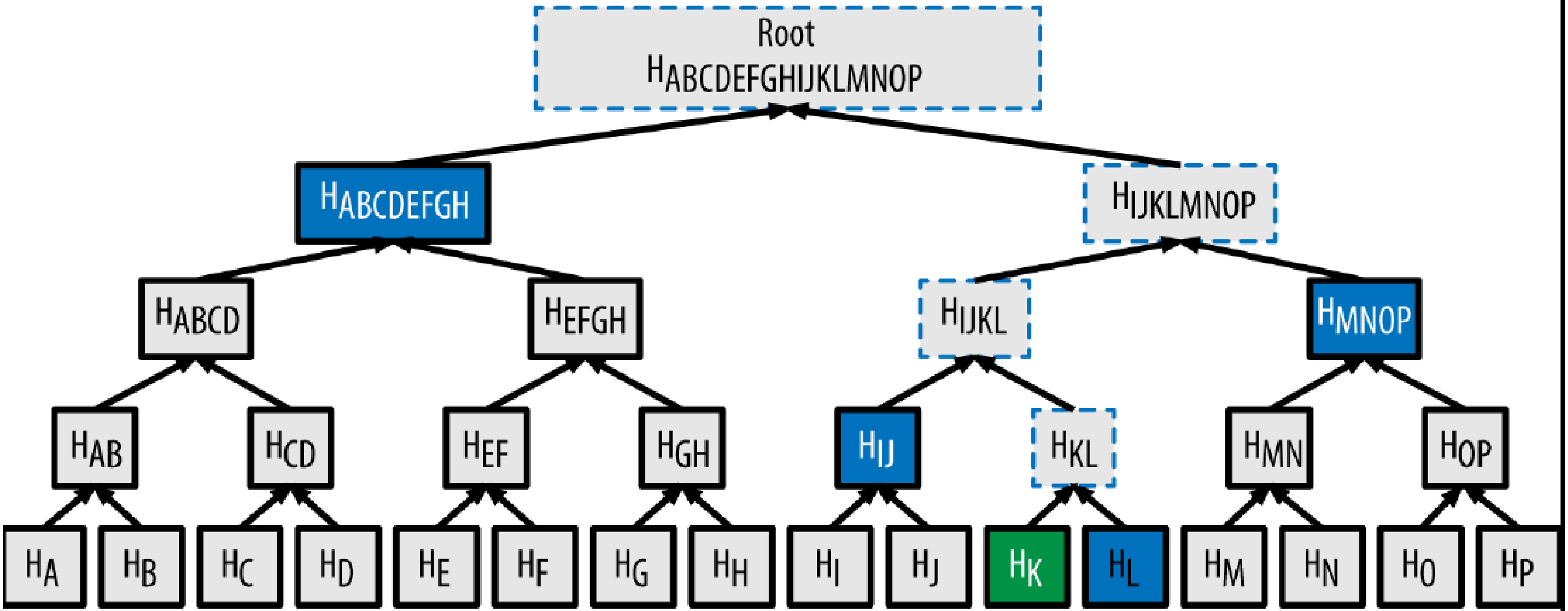
Proving Membership in a Merkle tree



Given n items, how much does it cost to prove membership?

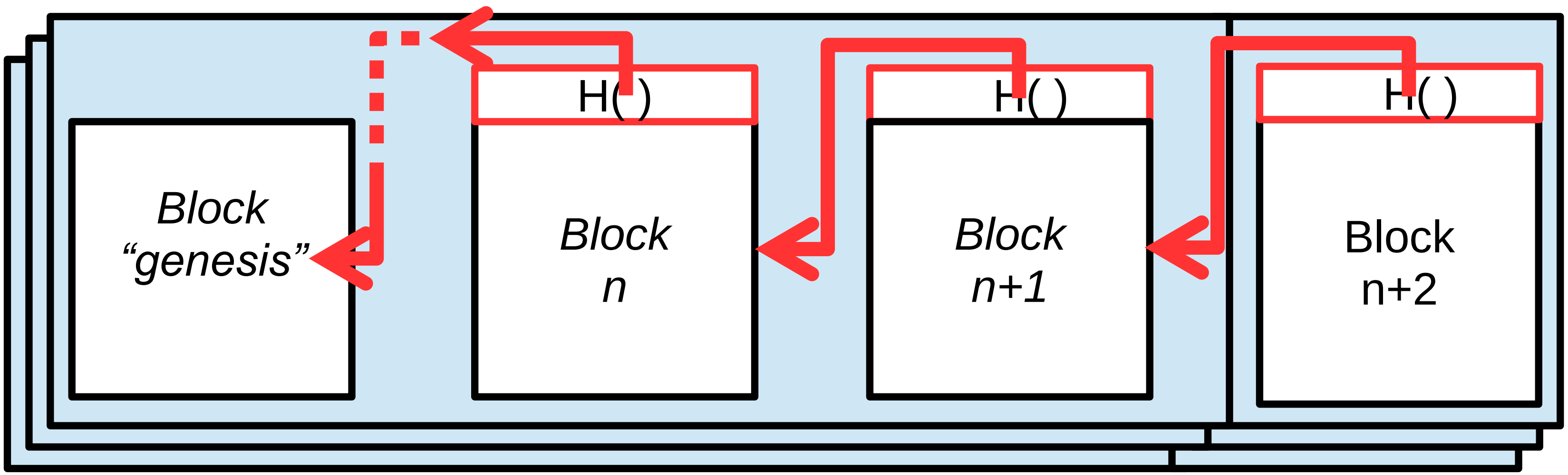
O(log(n))

Proving Membership in a Merkle tree



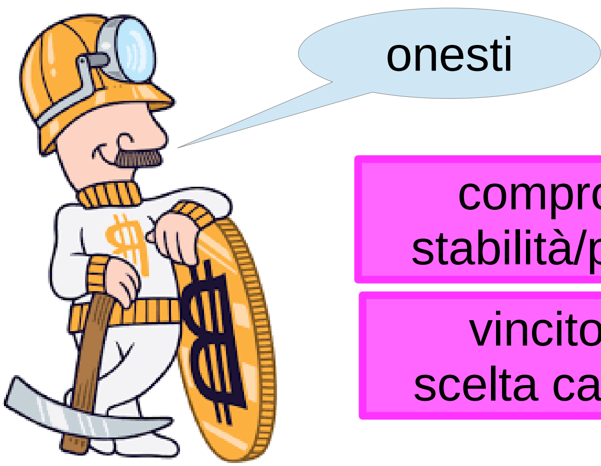
Struttura di un Blocco

Field name	**Type (Size)**	**Description**
nVersion	int (4 bytes)	Block format version (currently 2).
HashPrevBlock	uint256 (32 bytes)	Hash of previous block header $SHA256^2(nVersion\|\|\dots\|\|nNonce)$.
HashMerkleRoot	uint256 (32 bytes)	Top hash of the Merkle tree built from all transactions.
nTime	unsigned int (4 bytes)	Timestamp in UNIX-format of approximate block creation time.
nBits	unsigned int (4 bytes)	Target T for the proof of work problem in compact format. Full target value is derived as: $T = 0xh_2h_3h_4h_5h_6h_7 * 2^{8*(0xh_0h_1-3)}$
nNonce	unsigned int (4 bytes)	Nonce allowing variations for solving the proof of work problem.
#vtx	VarInt (1-9 bytes)	Number of transaction entries in *vtx*.
vtx[]	Transaction (Variable)	Vector of transactions.



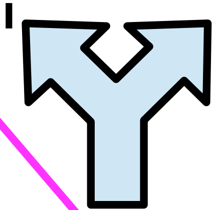
5:ricezione
e controllo del nu

Il consenso implicito dei minatori



4b:stop
a calcoli

4a:invio
del blocco



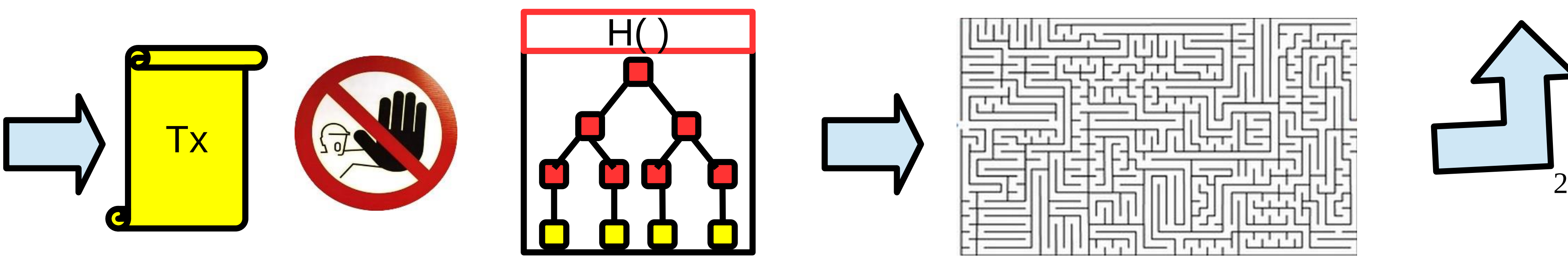
compromesso
stabilità/precisione

vincitore/
scelta casuale

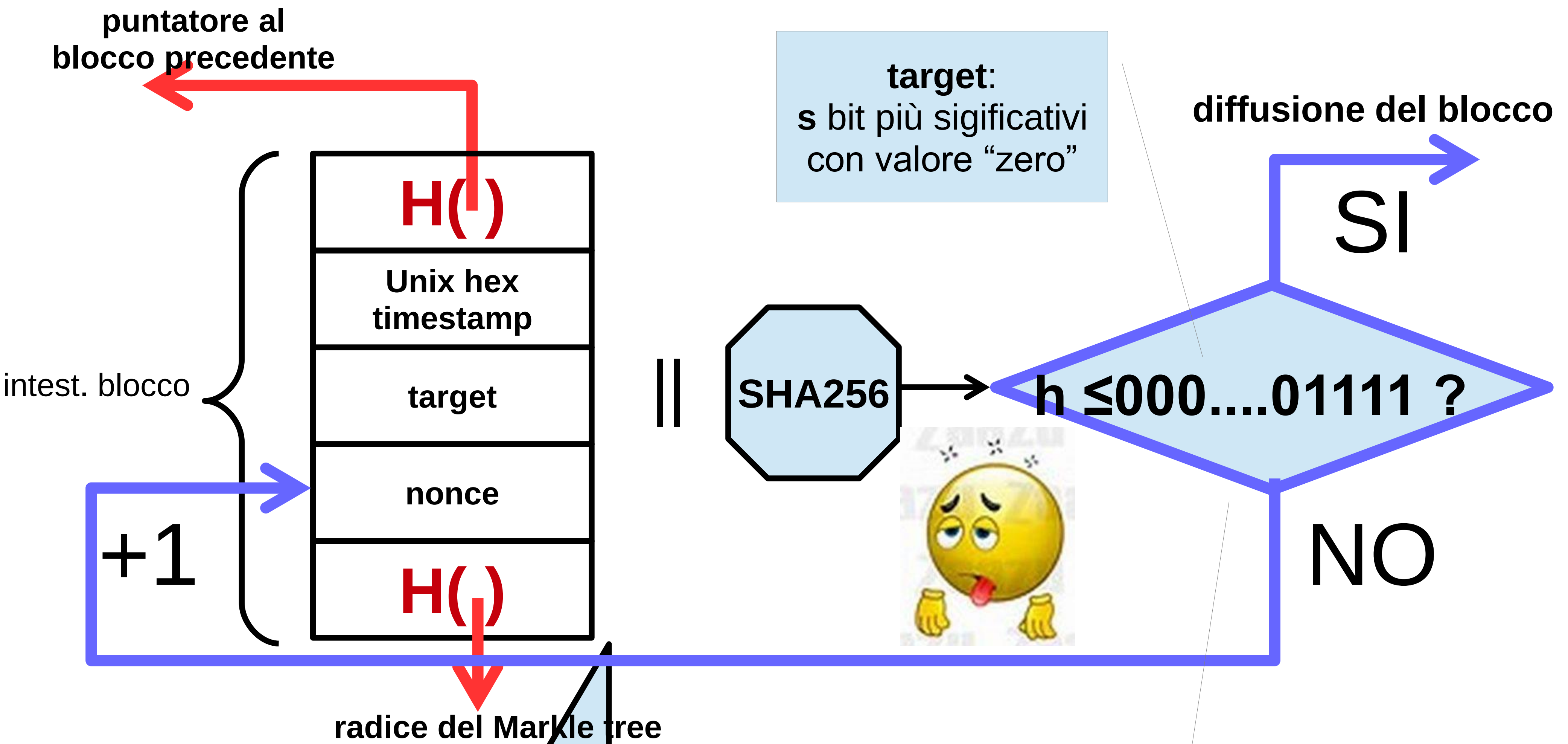
1:ricezione
e controllo delle transaz.

2:costruzione
del blocco

3:risoluzione
del puzzle



Il puzzle: dato un valore di soglia S = $2^{(256-s)}$-1 trovare un nonce tale che SHA256(intestazione_blocco)≤S



p = P(N=1)=$2^{(256-s)}/2^{256}=2^{-s}$

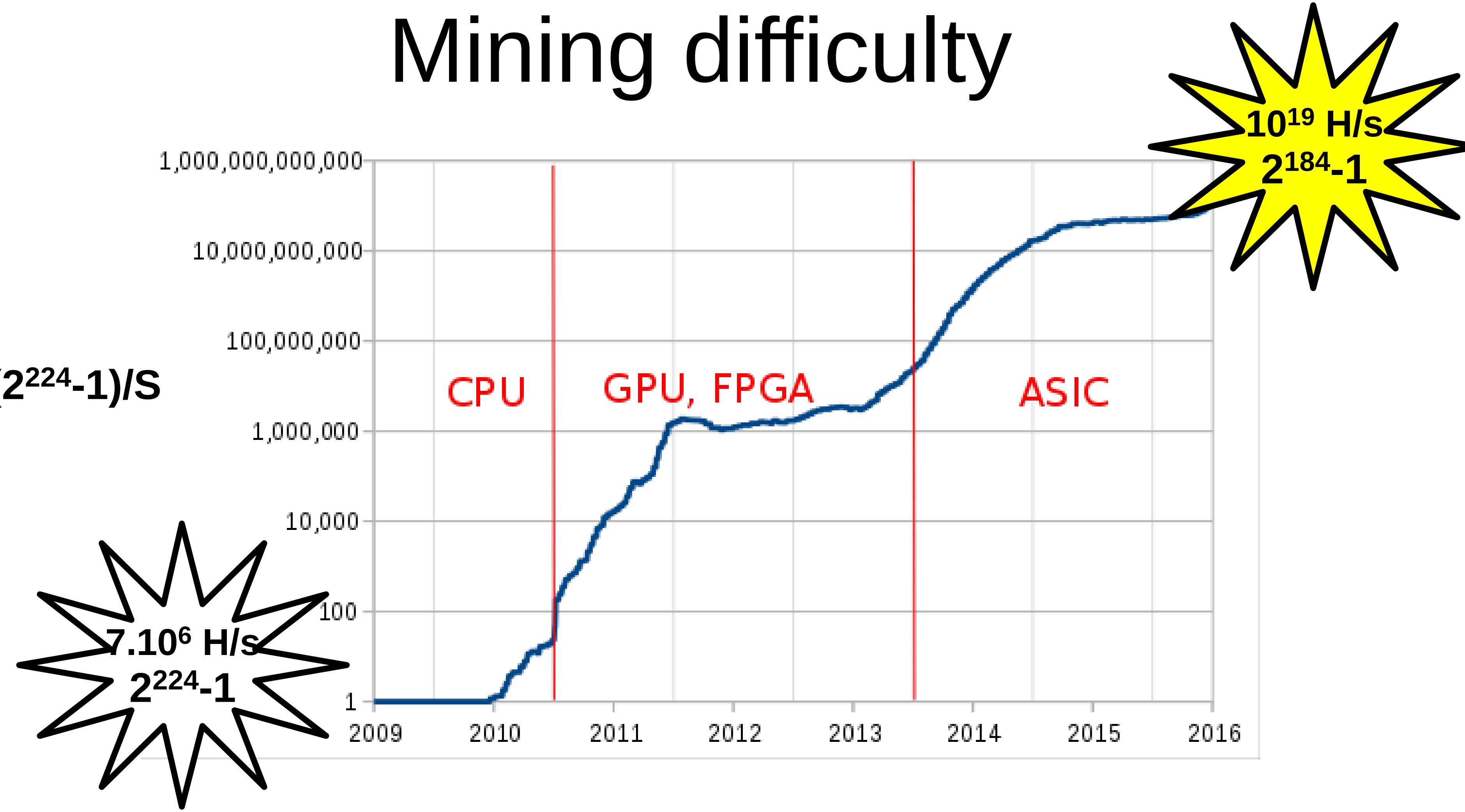
P(N=k)=p.$(1-p)^{k-1}$

E[N] =2^s

modificato
in più o in meno
ogni 2016 blocchi
(2 settimane)

Mining difficulty

$D = (2^{224}-1)/S$

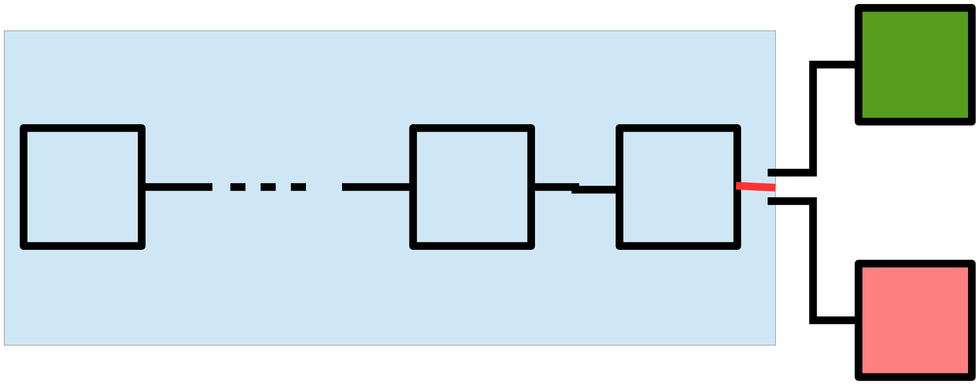


premi e contributi garantiti dalla “coin base” hanno spinto a calcolare SHA256 con dispositivi sempre più veloci e costosi

pericolo di centralizzazione

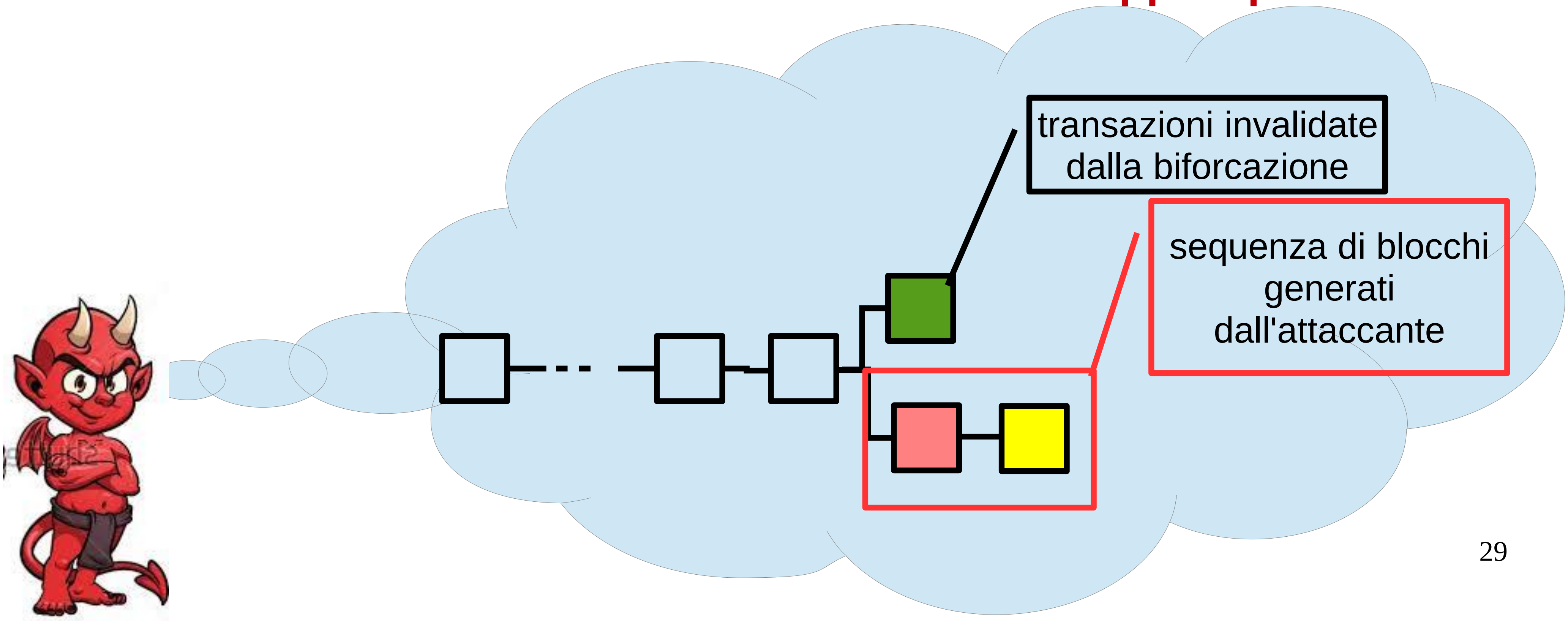
contributo alla sicurezza

biforcazione e speranza di doppia spesa



blocchi formalmente validi e con uguale “hash pointer” al top della blockchain:

- puzzle risolto quasi simultaneamente
- nuova versione del software
- **attacco di doppia spesa**



biforcazione e riallineamento

1)