Blockchain and Cryptocurrencies

Riccardo Salvalaggio

19th of April, 2021

Contents

1 Introduction

Ledger: record of transaction history, centralized, unforgeable.
Blockchain: Decentralized ledger, replicated, need of consensus (blocks because updating at each transaction would be too expensive).
Cryptocurrency: digital money, cryptographic means, rely on blockchain.
Non-monetary Uses: DApps - Decentralized Apps - Smart contracts: storing and executing programs on the blockchain. (e.g.: crypto assets, smart property, voting).

2 Money

Money formalizes the accounting, keeps track of global favor-granting and is useful as memory.

2.1 Payment Systems

A payment system is based **on representation of money, creation of it, and transfer of ownership**. **Cash:** physical, has storage and transport drawbacks.
Electronic: mainly digital, physical token required (e.g.: credit card), book money, centralized point of failure.
Blockchain-based: no physical, algorithmic creation, decentralized consensus.

2.2 Monetary unit

The need of a unit emerges from social processes and ease of interchanging. **Functions:** mean of exchange (w/o money too much exchanging), unit of accounting (allows an interpretation of prices, to compare valuation of goods and in order to have market transparency)and value storage (possibility of save in order to make larger investments).
Properties: durability, transferability, divisibility, homogeneity, verifiability, stability, scarcity.

2.3 Monetary equivalent

Why money has value? Fundamental value (material), payment promise (what you can buy with that), speculation (based on demand for the object and possibility of variation).

2.3.1 Fiat Money

No fundamental value, no payment promise, value entirely based on expectation (no bounds). Most currencies now are Fiat, the value is maintained by a central bank and have legal tender.

2.4 Monetary Control Structures

Two modes of money creation: **Competitive:** (everybody can, limit: production ¡ market value) two possible scenario: gold mining, constant low creation cost, **Monopolized:** (restricted to government agencies) artificially limited.

2.4.1 Creation by Banks

The central bank creates money by loaning to corporate banks. Temporary loans help regulate the money supply. Corporate banks can create money by lending or deposit currency" (to private customers and businesses) that is not backed by legal tender. If all customers withdraws legal tender the bank goes bankrupt.

2.5 Money representations

Physical: physical ownership, easy to use. Bound to location, integrity, divisibility.
Virtual: digital representation of a value not bounded to a real currency. It is nevertheless accepted as a means of payment and can be transmitted, stored and traded. Ownership achieved by cryptographic certificates.

2.6 Transaction processing

Conditions: Capability: transactions can be started and value can be transferred, Legitimacy: only by the owner, Consensus: process to determine the current balance of everyone.

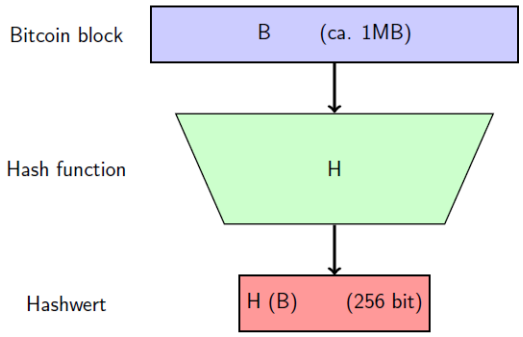
3 Basic Tools

3.1 Blockchain

A blockchain is a growing list of records, called blocks, that are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data.
A transaction is a movement of a monetary item. Transaction history is public, so everyone can know your balance and check every transaction. It's alright unless the history can be rewritten/faked.

3.2 Cryptographic Hash Functions

In the Blockchain, every block consists of the text of the transactions and the hash of the preceding block.



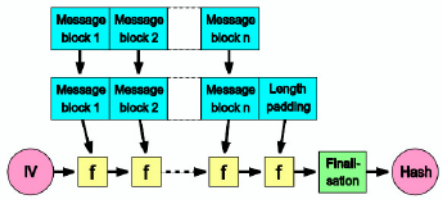
A hash function is a mapping h : B* ->B at n, for some n >0. B = 0, 1.
A compression function is a mapping h : B at m ->B at n, for some m >n >0.
Function h is a one-way function (or preimage resistant) if, given some s in Bn, it is practically impossible to
find a preimage x in D such that h(x) = s.
A collision of h is a pair (x; x0) with x != x0 in D such that h(x) = h(x0).
A function h is second preimage resistant (weakly collision resistant) if, given some x in D, it is practically impossible to
find x0 such that (x; x0) is a collision of h.
A function h is collision resistant if it is practically impossible to fi
nd a collision (x; x0) of h.
Any second preimage resistant function h is also one-way.
Any collision resistant function is also second preimage resistant.(See proofs)

3.2.1 Find a collision

Bitcoin uses SHA-256 so output is B at 256. To find a collision you need more time than universe age. It is practially impossible to
nd a collision (x; x0) for h.
Birthday Paradox: we need to do just $N/2$ in order ot have 0.5 probability of exploiting the correct answer.

3.2.2 Hash from Compression

Merkle-Damgaard procedure: Let f : Bm ->Bn be a compression function and let r = m - n >= 2. The goal is to construct a hash function h : B* ->Bn from f .
Preprocessing: becomes b'——0 at k——x——0 at r of length t*r.



If f is collision resistant, then the Merkle-Damgaard construction yields a function h that is also collision resistant.

3.2.3 Properties and Applications

Use of MAC for integrity of Data.
- Hiding: consequence of using a one-way. Small inputs can be a problem. To solve: Instead of computing h(x), compute h(rkx) where r is a suitably chosen random number.
- Commitment scheme (MAC scheme): commit and verification. Reqs: Hiding (given com it is infeasible to find msg) Binding (it is infeasible two find two mesages tht return same commit). To enforce, use nonce, number used once: it add perturbation.
- Def. Puzzle friendly: if for every possible n-bit output value and every k chosen from a distribution with high min-entropy, then it is infeasible to and x such that h(k——x) = y in time significantly less than 2 at n.
SHA-256 is defined via the Merkle-Damgaard construction from a compression function the compression function is designed such that flipping a bit in the input changes at least 50% of the bits in the output. Not know to be compromised, but successors exist since 2012: SHA-3 (Keccak).

4 Cryptography and Digital Signatures

4.1 Public key cryptography

- Symmetric encryption: Encode, Decode. **Decode(k,encode(k,m)) = m.**
Partners needs same key, if it is exposed, confidentiality is compromised.
- Asymmetric encryption: Pair of different keys: Pub_K, Pri_K. Public key is freely shared, the private is known only by the owner. Asymmetric is more expensive than symmetric, so most of the times is only used for key exchanging.
Decode(Pri_K,encode(Pub_k,m)) = m. It is a one-way encryption.

- Cryptosystem: Key pair, keygenerator, encode, decode (polynomial time algorithm).

4.1.1 RSA

Choose two primes p,q: n= pq of k bits.
Choose 1¡e¡(p-1)(q-1): e is coprime with (p-1)(q-1).
Choose 1¡d¡(p-1)(q-1): de=1mod(p-1)(q-1). (thanks to EEA)
Pub_key = (n,e), Pri_key = d.

Encode

- Given the public key (n, e) and message $0 \leq m < n$
- Compute $c = \textbf{encode}((n, e), m) = m^e \mod n$

Decode

- Given the public key (n, e), secret key d, and ciphertext c.
- let $\textbf{decode}((n, d), c) = c^d \mod n$.

This is a decoding because $ed = 1 \mod (p-1)(q-1)$ means that there exists some $l \in N$ such that $ed = 1 + l(p-1)(q-1)$. Hence

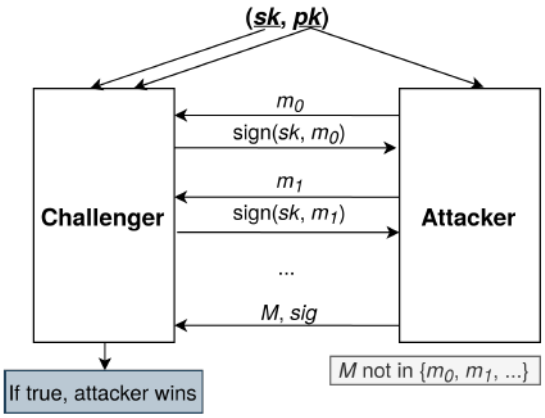
$$(m^e)^d = m^{ed} = m^{1+l(p-1)(q-1)} = m \cdot m^{l(p-1)(q-1)} = m \mod n$$

by Fermat's theorem.

Bitcoin uses a different scheme based on Elliptic curves.

4.2 Digital Signatures

Signature is a handwritten depiction of someone's name, nickname etc.
Functions: proof that signer has seen the content of the document, integrity of the document, signature has to be difficult to forge, but easy to verify.
Same mechanism of Public-key cryptography but sign with private and decrypt with public, so everyone can decrypt but only the owner can sign.



- Practical concerns: many algorithms are probabilistic, limit on message size (possible solution, sign the hash).
Bitcoin uses ECDSA (Elliptic Curve Digital Signature Algorithm) that provide 128 bits: Pri_K = 256 bits, Pub_K = 512 bits (compressed 257 bits), m_size = 256 bits, signature = 512 bits.

4.3 Identities

Identity consists of a pair (Pri,Pub).
Decentralized Identity Management: new identities can be created at any time, good randomness source is required, no need for central user registry, but transactions may reveal behaviour and connections.