Blockchain and Privacy

Integrity, confidentiality and Blockchains

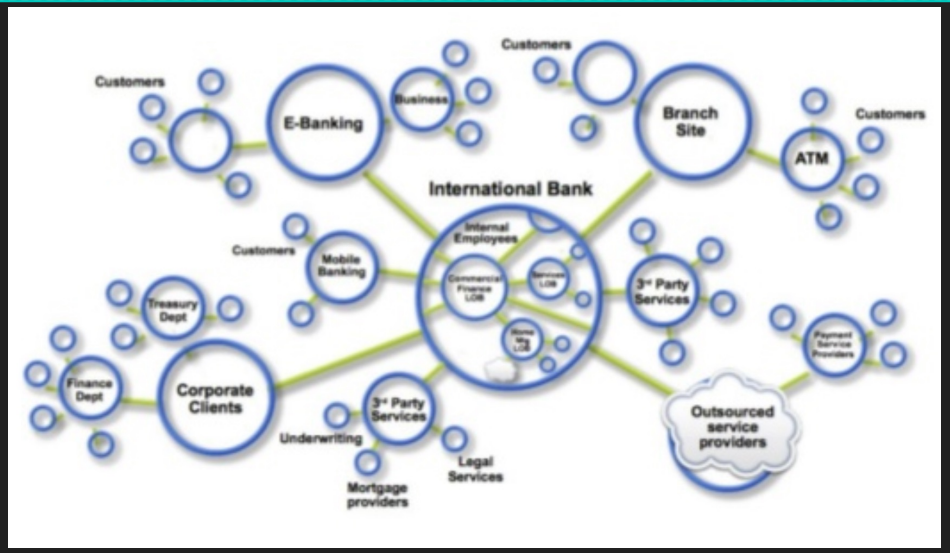
Yves ROUDIER
Université Côte d'Azur / I3S / Polytech Nice Sophia
SI5 / M2 Informatique parcours Ingénierie

From centralized applications …

- Web applications are central
  - Every second: 50000 Google searches, 10000 tweets, 2000 Skype calls …
  - Every month: more than 5 billion hours of video watched on Youtube
- Many applications are centralized
  - Major players dwarfing competition
    - E.g., Facebook accounts for more than 70% of social network traffic
  - Very sensitive: privacy matters, …
    - Tim Berners-Lee: "The Web's future relies on individuals owning their data"
  - Technically single points of failure
    - History of extortion attempts through DoS blackmail …

… to connected worldwide markets

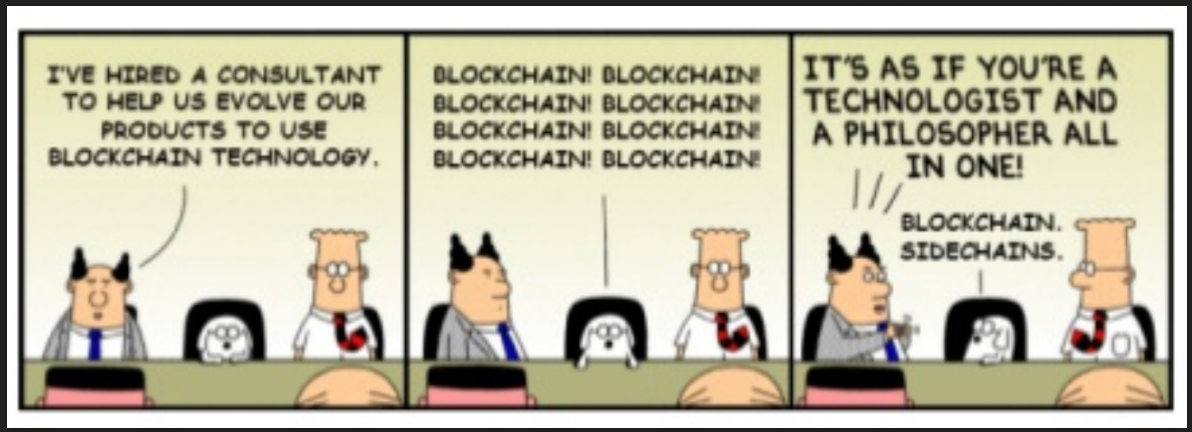
- Markets organize trades for exchanging assets
  - Physical assets (computers, cars)
  - Immaterial assets (services, bonds, patents)
- Transactions exchange assets
- Networks connect economic agents
  - customers, suppliers, financial organizations
- Already heavily reliant on cryptography
  - Payment, ATM, smart cards, online banking, clearing …



- Problem: every market relies on its own infrastructure for transactions
  - Often incompatible, except for proprietary – and again SPOF – solutions (e.g. VISA, Paypal, Amazon …)

Introducing the Blockchain…

… Block what ?!



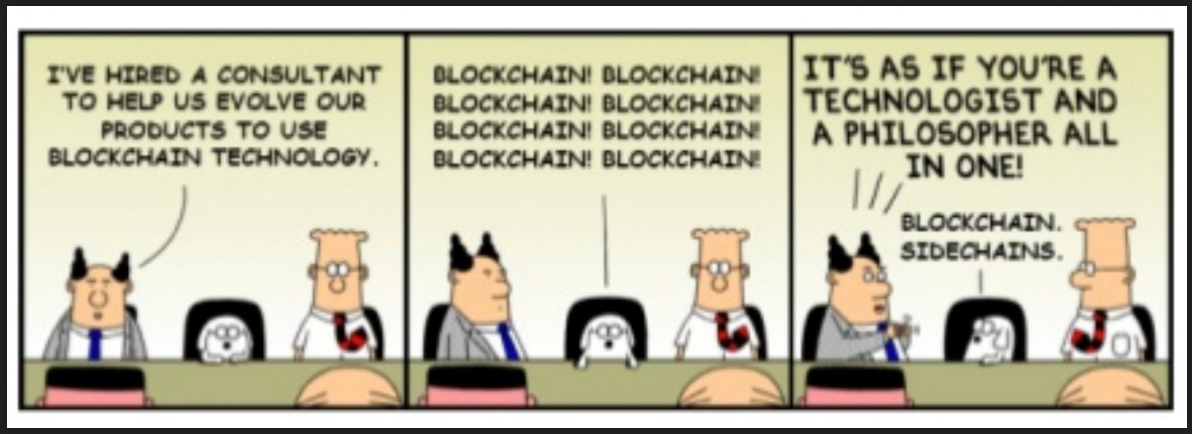
Introducing the Blockchain…

Digital currency

Virtual currency

Cryptocurrency

Bitcoin



NFTs

Distributed Ledger

Ethereum

Smart contracts

ICO

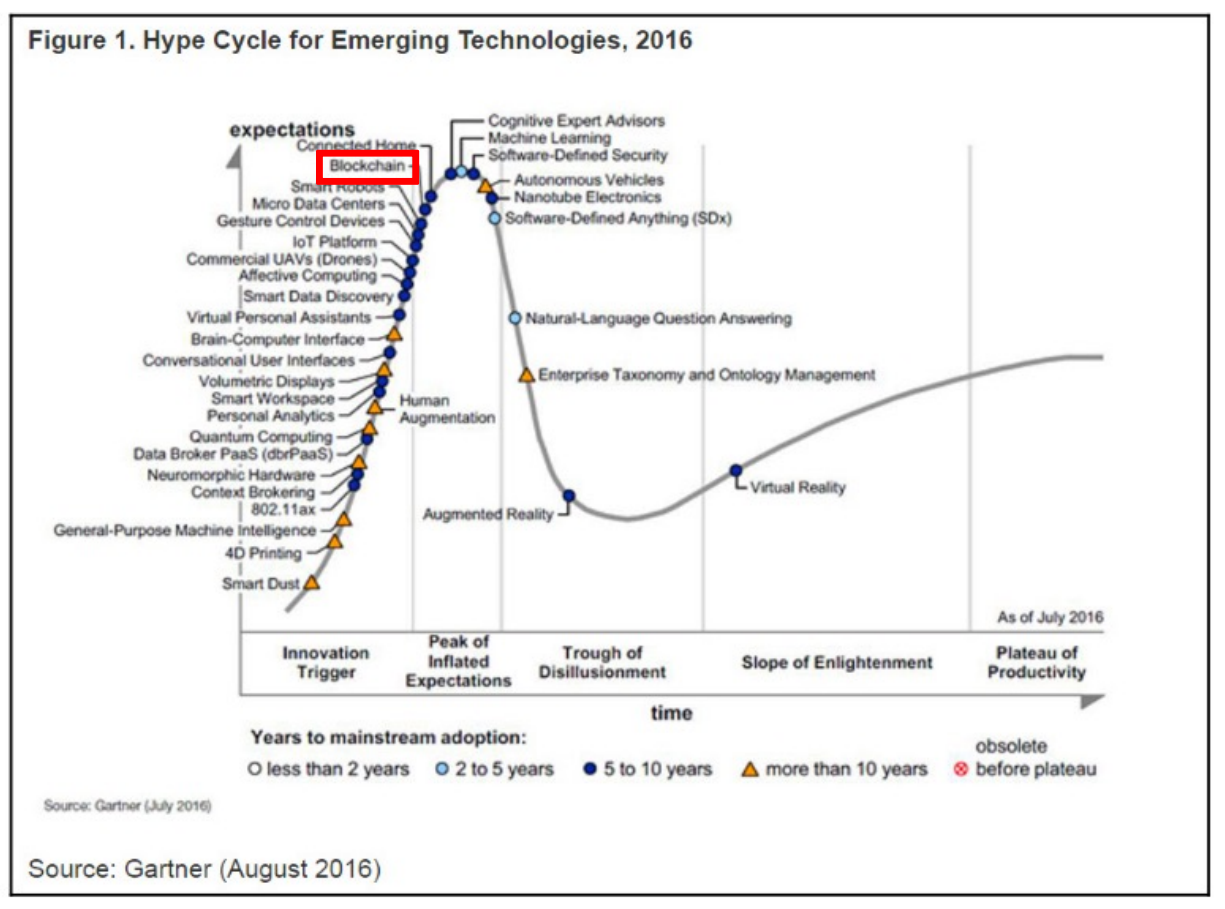
Blockchain tomorrow: decentralized trust

- Suppress single points of failures through a distributed and dynamic infrastructure
  - P2P interactions (truly distributed applications)
  - Availability
  - Monopoly prevention
  - Privacy
- The promise: reduce trust towards infrastructure /participants and replace it by technology
  - Ensure data availability despite dynamicity and attacks through large-scale replication
  - Enforce data integrity through cryptographic protocols
  - Protect data confidentiality when required through cryptographic encryption

Blockchain: domains of application

- Blockchain 1.0: Cryptocurrencies
- Blockchain 2.0: Financial services and contracts
- Blockchain 3.0: Services relying on "notarization" (data storage with integrity, and possibly confidentiality)
  - Computer security (DPKI, Internet of Things, …)
  - government, health, justice, education
  - NFTs (arts, games, mass media …)

Blockchain innovation lifecycle



Source: Gartner (August 2016)

Blockchain trends

2008: Bitcoin infrastructure – forges the term “Blockchain”

« Blockchain » search according to Google Trends (January 2023)

Blockchain trends

2008: Bitcoin infrastructure – forges the term “Blockchain”

« Bitcoin » BTC/EUR price (January 2023)

Types of blockchains

- Public blockchains
  - Permissionless: open to the public and anyone can participate as a node in the decision-making process
  - Users may or may not be rewarded for their participation
- Private blockchains
  - Permissioned: private and open only to a consortium or group of individuals or organizations that has decided to share the ledger among themselves
  - Enforced through thorough access control policies and mechanisms
- Semi-Private blockchains
  - Permissioned: private part is controlled by a group of individuals
  - Permissionless: public part is open for participation by anyone (with different rights)

Cryptography and security concepts

- **Security Properties / Security services according to ITU-T X-800 (OSI context) :**
  - **Confidentiality**
    - unauthorized users cannot read information or access message)
  - **Integrity**
    - unauthorized users cannot alter information
  - **Availability**
    - authorized users can always access information
  - **Authentication**
    - Of data (prevent message replay) or of entities (prevent identity theft)
  - **Authorization / Access Control**
    - only authorized users may perform certain actions
  - **Non-repudiation**
    - User cannot deny his transactions (proof of origin or proof of delivery)
  - **Privacy and anonymity:**
    - Degrees of anonymity: pseudonymity, unlinkability, unobservability

From Hash Functions …

- A hash function h is a function which associates
  - to a message M
  - a message h(M) of a shorter length.
- M can be arbitrarily large while h(M) has a constant length

- Example: file system hashcodes, error detection codes

… to Cryptographic Hash Functions

- Still a hash function:
  - Takes data (e.g. string) as an input
  - Generates a fixed-size output "summary" (assume 256 bits for Bitcoin for instance with SHA-256)
  - Easy to compute

- **Cryptographic** hash function = Hard to reverse - 3 properties:
  - Preimage resistance: given K, it is computationally hard to find M such that h(M)=K
  - 2nd Preimage resistance: given M, it is computationally hard to find M' distinct from M such that h(M)=h(M')=K
  - Collision-free: it is computationally hard to find M and M' distinct from M such that h(M)=h(M')
  - … yet Birthday attacks …

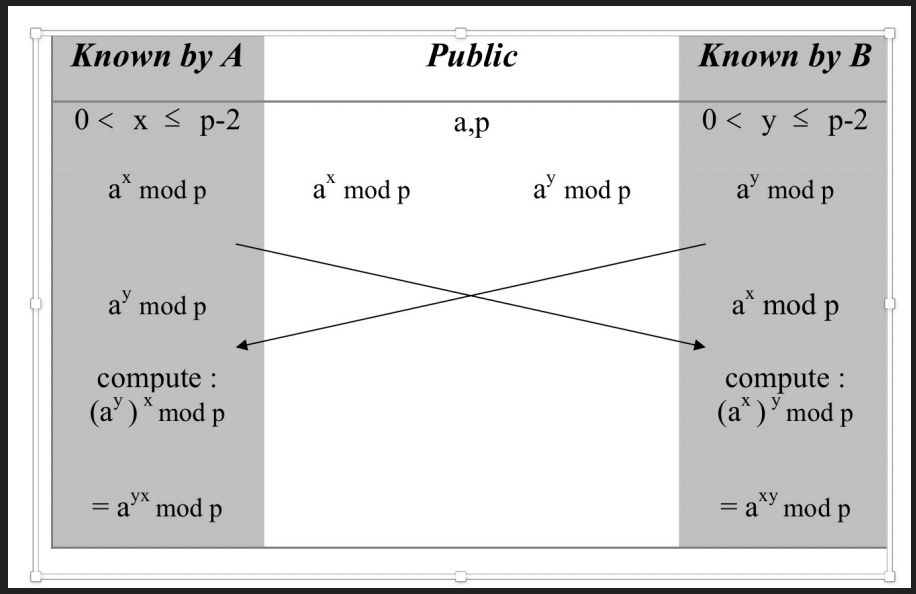
- It is keyed if its computation depends on a secret information
  - termed MAC – Message Authentication Code

Asymmetric encryption

- Encryption based on the difficulty to find the inverse solution to a mathematical problem
  - Much more costly than symmetric key encryption
- Diffie-Hellman (1977)
  - Secret sharing : $(g^a)^b=(g^b)^a=g^{ab} \bmod p$

Diffie-Hellman encryption algorithm

p is a large prime, *a* a primitive element of Z^*_p



- A and B establish a shared secret (a^{xy} mod p) without exchanging any secret information
- a^{xy} mod p may be used as a secret key with a symmetric key algorithm in order to encrypt data
- r relies on the hardness to compute the discrete logarithm

Public key encryption

Asymmetric encryption relies on an operation whose inverse computation is assumed NP-hard (Discrete Logarithm problem for Diffie-Hellman)

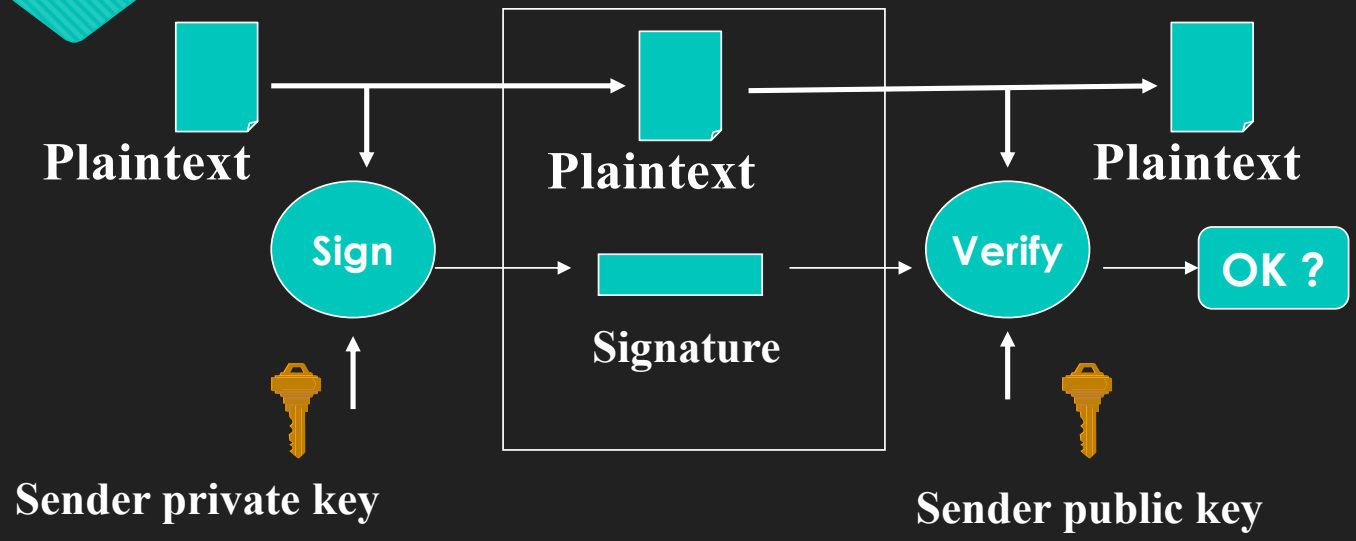
Modulus:

X	1	2	3	4	5	6
3^x	3	9	27	81	243	729
$3^x \bmod 7$	3	2	6	4	5	1

Asymmetric encryption

- Encryption based on the difficulty to find the inverse solution to a mathematical problem
  - Much costly than symmetric key encryption
- Diffie-Hellman (1977)
  - Secret sharing : $(g^a)^b=(g^b)^a=g^{ab} \bmod p$
- RSA : Rivest-Shamir-Adleman (1978)
  - Standard solution today
  - $E_{KP}(P)=M^e \bmod n$ et $D_{KS}(C)=M^d \bmod n$
  - Les clés KP=(e,n) et KS=(d,n) are connected by a mathematical relationship
- Elliptic curves: the new breed

Digital Signature



- Comes with a message (which can be encrypted or in cleartext) and ensures:
  - The authentication of the origin of a message
  - The protection of the integrity of a message
  - The non-repudiation of a message

Signature using a public key algorithm

E, D : public key algorithm

Generating the signature of A over message M :

$$S = E_{KSa}(h(M))$$

Verifying the signature :

- compute h(M)
- verify whether $D_{KPa}(S) = h(M)$

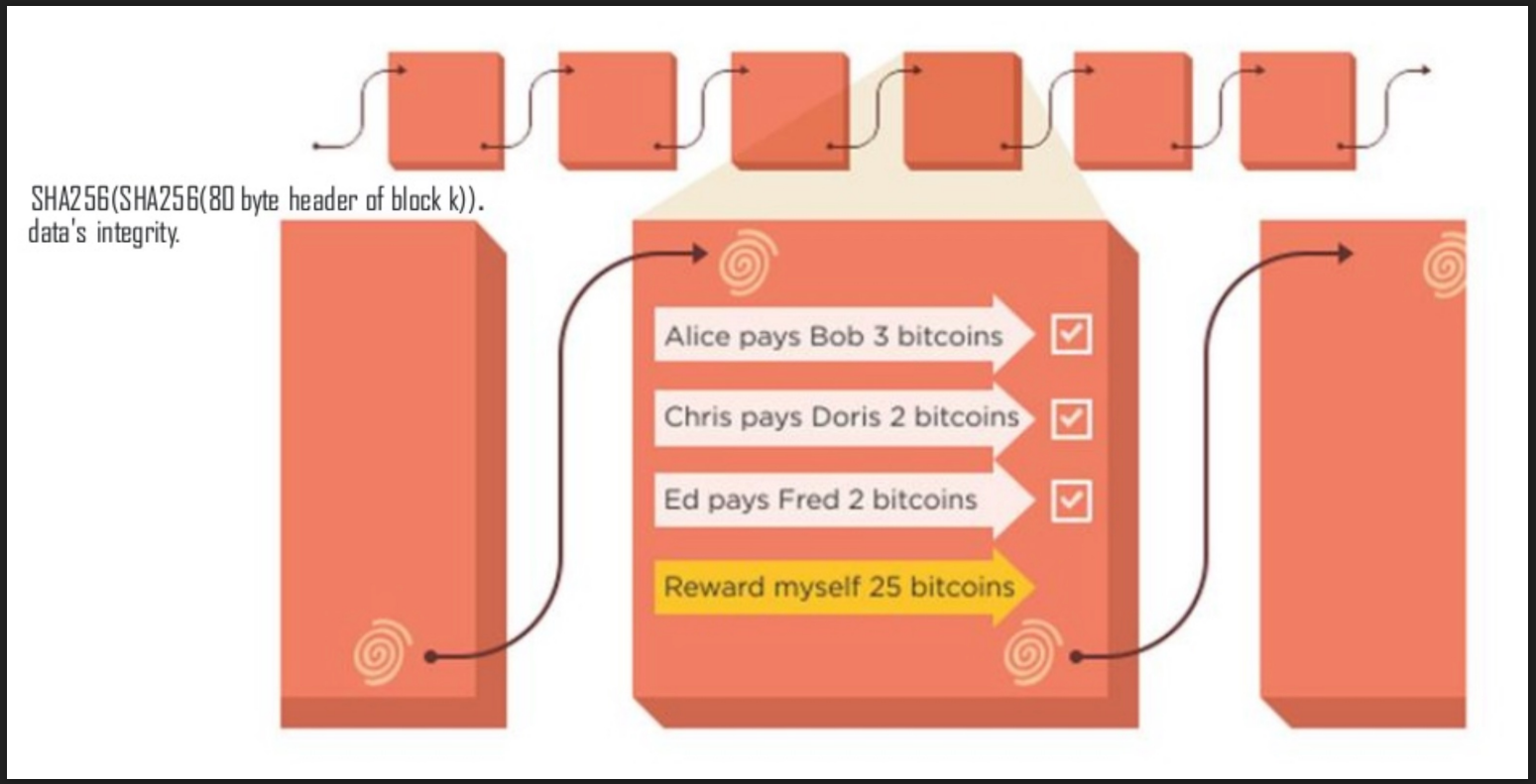
Digital signatures

- Properties:
  - Authenticity: guarantees that the data are signed by the owner of the private key
    - A honest signer is always accepted
    - A dishonest signer is always rejected
  - Non-repudiation: the data submitted are imputable to the owner of the private key
  - Public verifiability: anyone can verify a signature (using the owner's public key)

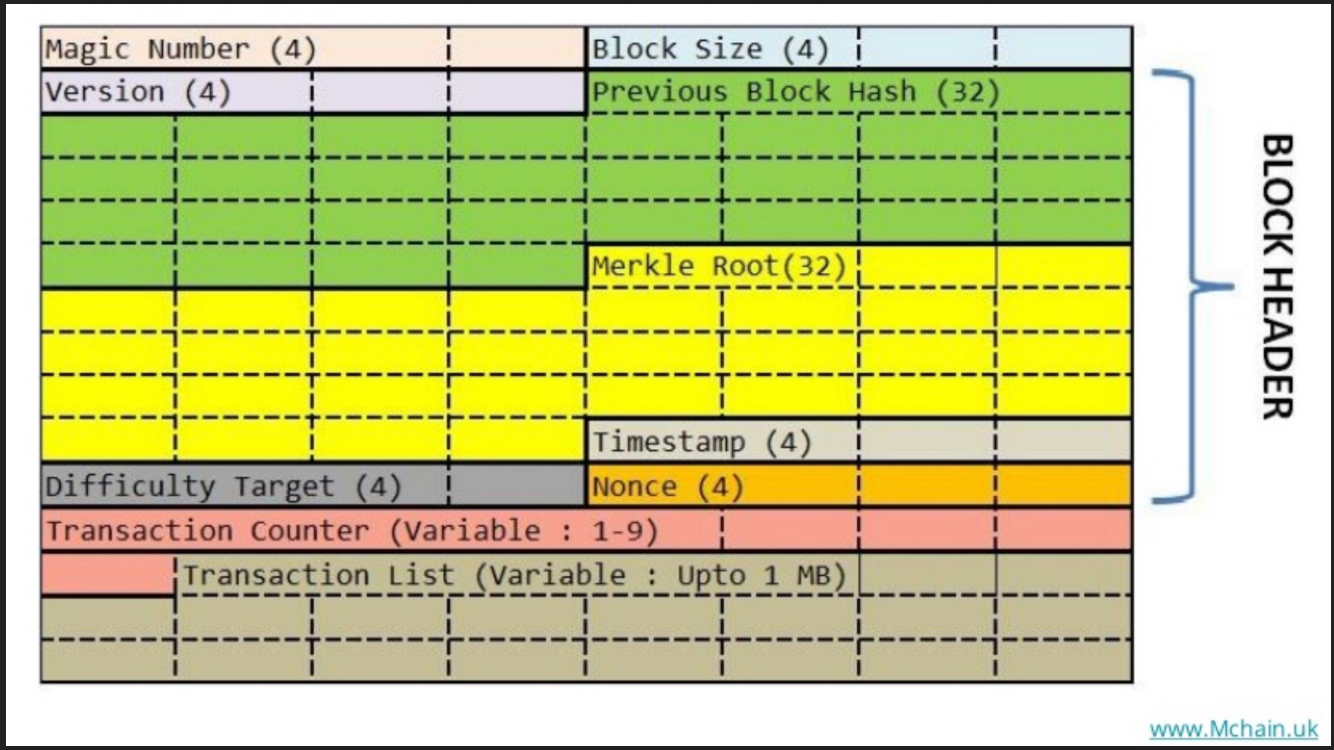
Blockchain as a secure data structure

- 2008: the research paper *Bitcoin: A Peer-to-Peer Electronic Cash System* – by so-called Satoshi Nakamoto introduces the term "chain of blocks"
- Transactions
  - Users remain pseudonymous – linked through their cryptographic keys (encrypted transactions)
  - Bitcoin: user are known by their pseudonym, H(PK), which can be used to transfer coins (through digital signature)
- Blocks
  - Records multiple data (granularity of storage)
  - First block in chain, created with the blockchain (= genesis block in Bitcoin blockchain)
- Chain
  - Multiple blocks connected in a chainlike fashion
  - Each block contains a hash (summary) of previous block
  - Immutable data: nobody can tamper with content of previous block without being noticed when inspecting next block hash

Chain of blocks

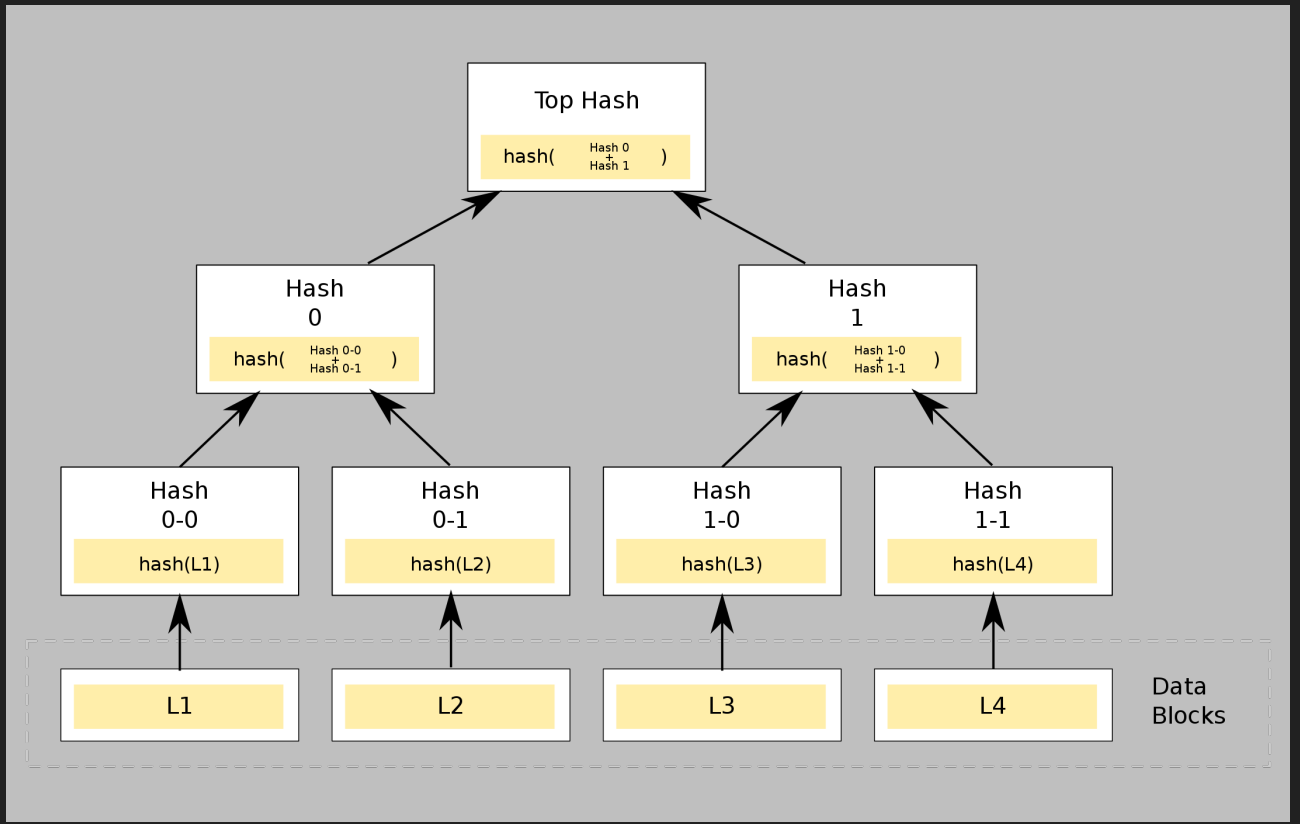


Bitcoin block structure



Bitcoin block structure: Merkle root

- A Merkle tree is used to compute the Merkle root in every block
- Quick summary for all the transaction hashes
  - Li = transactions



Cryptographic chain verification

- The log is append-only
  - Transactions tx_1, tx_2, …
- Its content is verifiable from the most recent element
  - $h_t \leftarrow$ Hash ([tx_1, tx_2, …] | h_{t-1} | t)
  - Or actually : $h_t \leftarrow$ Hash (Merkle root | h_{t-1} | t)