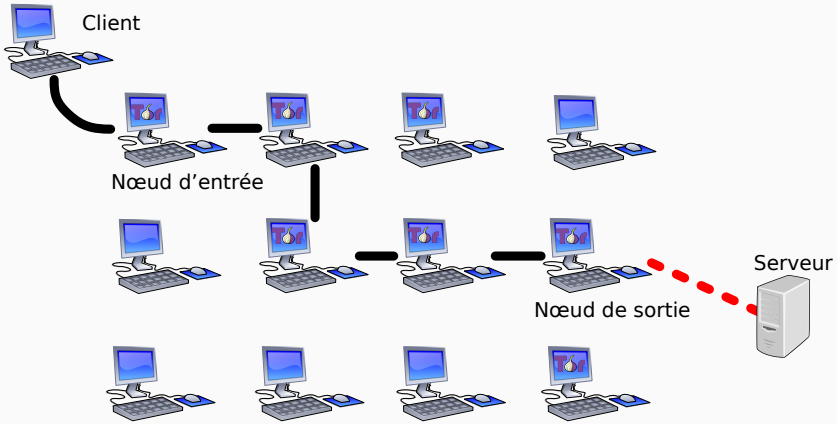
# ICS - Information Security Systems
# Lecture 5: Protocols

2025-2026

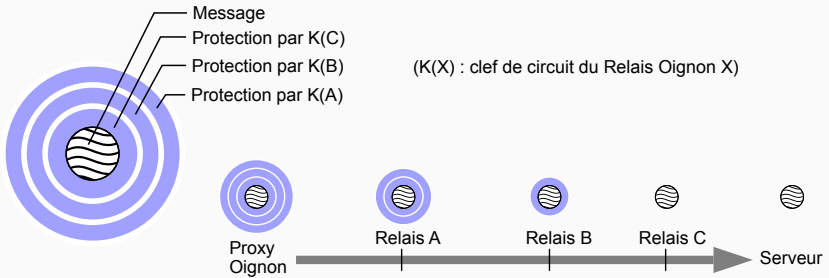

# Outline

# Application : The Onion Router (TOR)




https://www.torproject.org

# Application : Tor



# TOR session initiation :

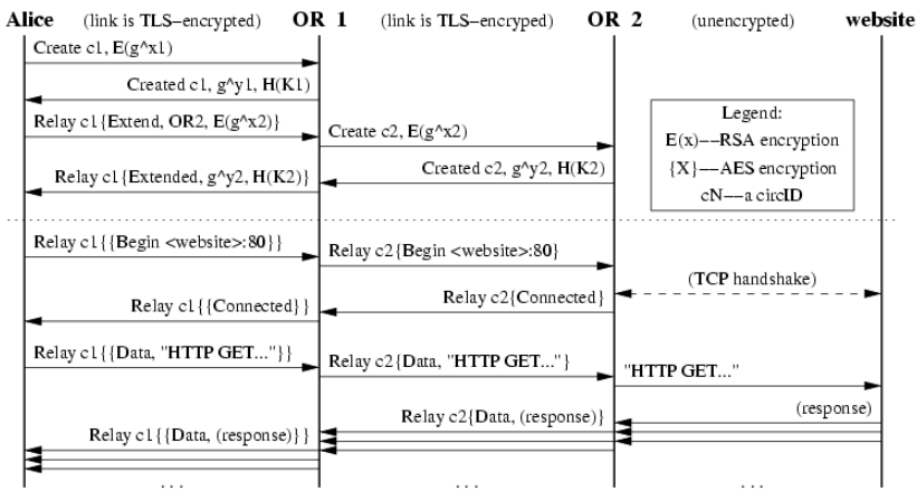



Figure 1: Alice builds a two-hop circuit and begins fetching a web page.

# Outline

# Idea of Zero-Knowledge Proofs

Prover (P)

(P) convinces (V) that he knows something without revealing any additional information.

Verifier (V)

# Idea of Zero-Knowledge Proofs

Prover (P)

(P) convinces (V) that he knows something without revealing any additional information.

Verifier (V)

**Applications:**

- Authentication systems: proving one's identity to someone without revealing the secret.
- Proving that a participant's behaviour is compliant with the protocol (for example, ballot integrity in an election).
- Group signatures, secure multiparty computation, electronic money, etc.

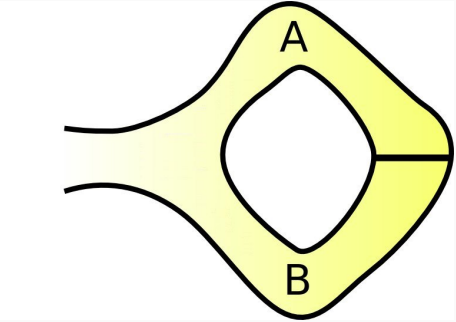
# Interactive Zero Knowledge Proofs

In an interactive zero-knowledge proof, a prover $P$ interacts with a verifier $V$ to demonstrate the validity of an assertion without revealing anything about the assertion to the verifier.

An Example: The Cave Story (1)

- a cave shaped like a circle
- a magic door on one side
- an entrance on the other side
- Victor will pay Peggy only if she knows the secret
- Peggy will not reveal the secret until she has been paid

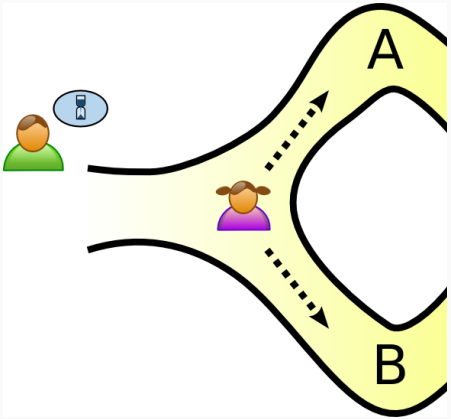
# Cave Example (0)




Door with a secret code

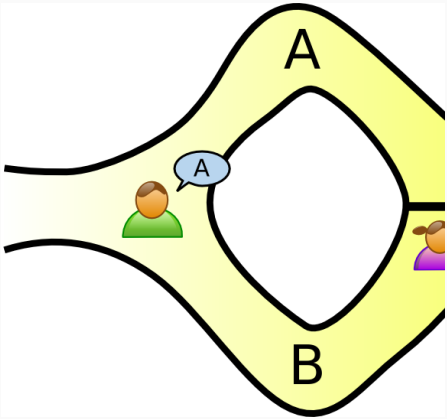
# Cave Example (I)

V waits outside while P chooses a path



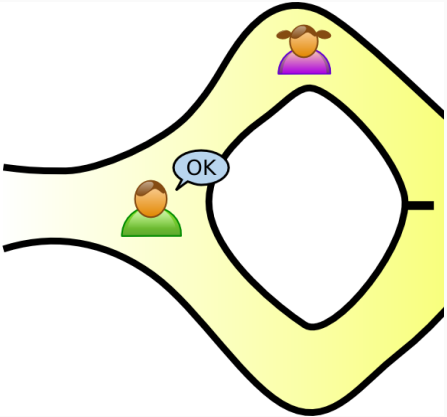
# Cave Example (II)

V enters and shouts the name of a path



# Cave Example (III)

P returns along the desired path (using the secret if necessary)



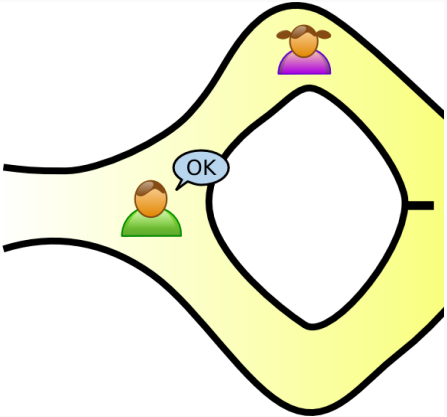
# Cave Example (III)

P returns along the desired path (using the secret if necessary)

$A$ = “P does not know the secret”

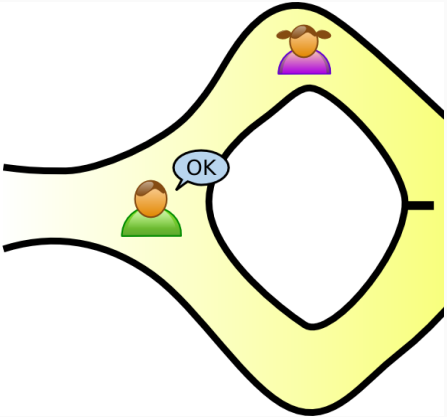
is equivalent to saying “P is lucky”

$$Pr[A] = \frac{1}{2}$$



# Cave Example (III)

P returns along the desired path (using the secret if necessary)




$A$ = “P does not know the secret”

is equivalent to saying “P is lucky”

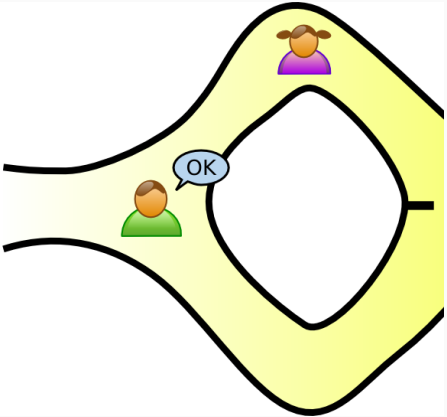
$$Pr[A] = \frac{1}{2}$$

After $k$ attempts,

$$Pr[A] = \left(\frac{1}{2}\right)^k$$

# Cave Example (III)

P returns along the desired path (using the secret if necessary)




$A$ = “P does not know the secret”

is equivalent to saying “P is lucky”

$$Pr[A] = \frac{1}{2}$$

After $k$ attempts,

$$Pr[A] = \left(\frac{1}{2}\right)^k$$

$\overline{A}$ = “P knows the secret”, then

$$Pr[\overline{A}] = 1 - Pr[A] = 1 - \left(\frac{1}{2}\right)^k$$

# Properties of ZKP

1. **Completeness**: If $P$ knows a solution, then he can convince $V$.

# Properties of ZKP

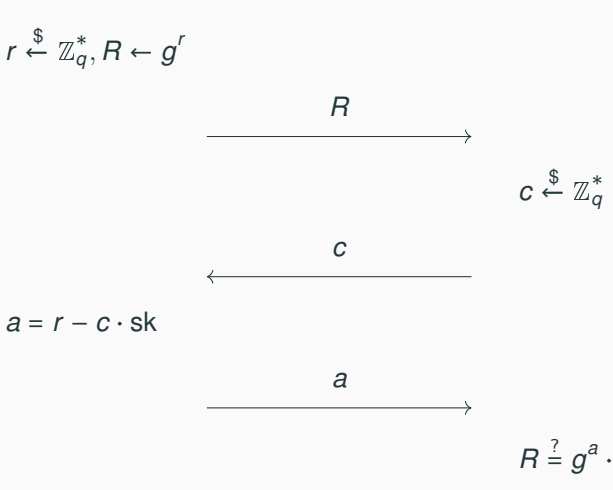
1. **Completeness**: If $P$ knows a solution, then he can convince $V$.
2. **Extractability**: If $P$ does not know the solution, then he cannot convince $V$.

# Properties of ZKP

1. **Completeness**: If $P$ knows a solution, then he can convince $V$.
2. **Extractability**: If $P$ does not know the solution, then he cannot convince $V$.
3. **Zero-Knowledge**: $V$ learns nothing about the solution.

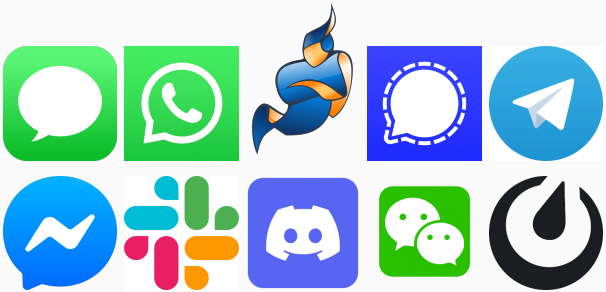
# Schnorr Identification Protocol (1989)

Goal: prove knowledge of a secret key *sk* without revealing it, with $\mathrm{pk} = g^{\mathrm{sk}}$.



# Outline

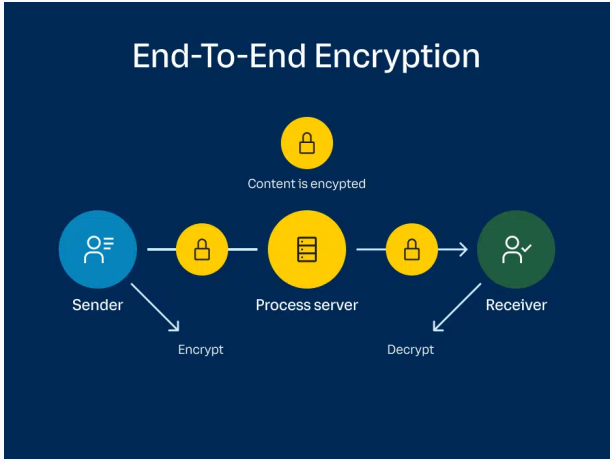
# Application: Chat



iMessage, WhatsApp, Jitsi, Signal, Telegram, Messenger, Slack, Discord, WeChat, Mattermost

# End-to-end Encryption

Encryption: on the sender's device
Decryption: on the receiver's device



# Security Properties

- Confidentiality
- Authenticity
- PFS: Perfect Forward Secrecy
- PCS: Post-Compromise Security

# PFS & PCS

PFS: Protects a session **before** a key compromise.


PCS: Protects a session **after** a key compromise.

# PFS & PCS

PFS: Useful when you lose your smartphone/PC.

$\Rightarrow$ all prior communications remain secure.

PCS: Does not protect against a full compromise...

$\Rightarrow$ for example, during a U.S. customs check: your smartphone is scanned.

# Stage

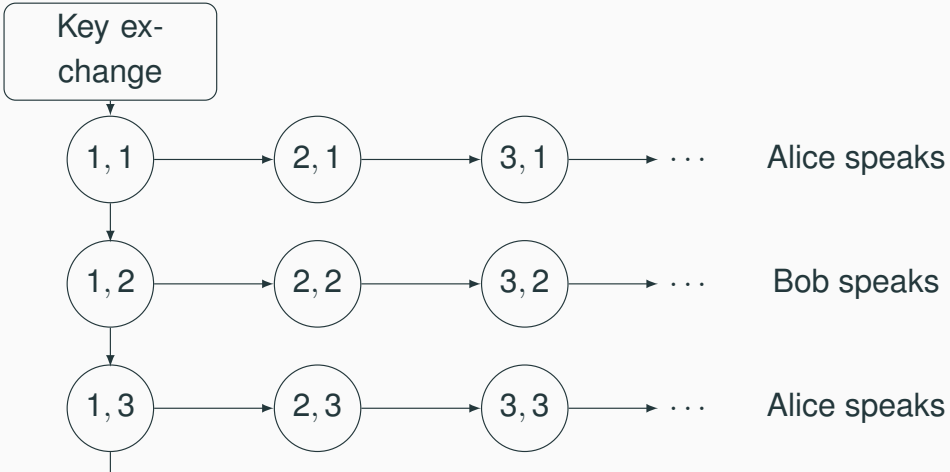
A stage corresponds to establishing a (symmetric) encryption key.

$\Rightarrow$ One key per message!

# Stage

A stage corresponds to establishing a (symmetric) encryption key.

$\Rightarrow$ One key per message!



# Stage for Signal

**Chain 1**

1. Hello
2. How are you?

⋮

**Chain 2**

1. Good, and you?

⋮

**Chain 3**

1. Great!

⋮

Stage $(1,1)$: “Hello”
Stage $(2,1)$: “How are you?”
Stage $(1,2)$: “Good, and you?”

Stage $(x,y)$: $x^{\text{th}}$ message of chain $y$

## Key Derivation Function (KDF)

Used to add entropy (randomness) to a key.

Writing $y = \mathrm{KDF}(x)$, the only way to obtain $y$ is to know $x$. Moreover, knowing only $y$ does not allow one to "go back" to recover $x$.

Example of KDF: HMAC

# Cryptographic Tools

## Key Derivation Function (KDF)

Used to add entropy (randomness) to a key.

Writing $y = \mathrm{KDF}(x)$, the only way to obtain $y$ is to know $x$. Moreover, knowing only $y$ does not allow one to "go back" to recover $x$.

Example of KDF: HMAC

## AEAD

Each message is encrypted in an authenticated manner (Authenticated Encryption).

Additional information (Additional Data) is attached to every message.

# Signal > WhatsApp

- Signal is a non-profit organization.
- Signal is the pioneer of secure messaging protocols.
- WhatsApp cannot access your messages, but it can access all metadata (who you talk to, when, how often...).

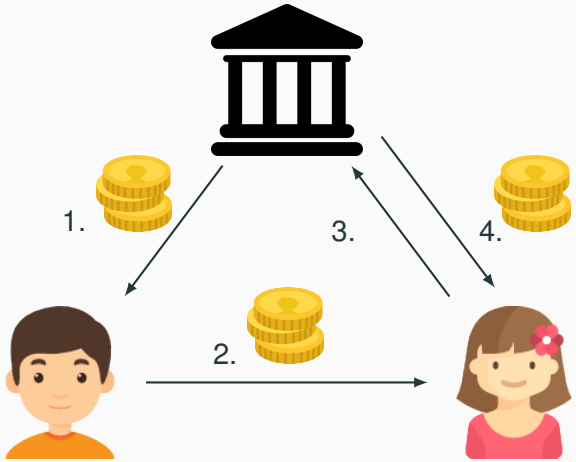
# Outline

# Money as a functionality: a currency

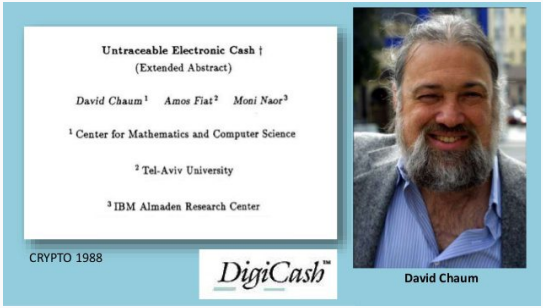
1. Medium of Exchange of goods and services between people
2. Store of Value
3. Unit of Account (measure of value)

# Many currencies

# Principle: central Bank



# 1988: Digitcash




- Preserves privacy
- Use cryptographic primitives
- Requires a third party (bank)

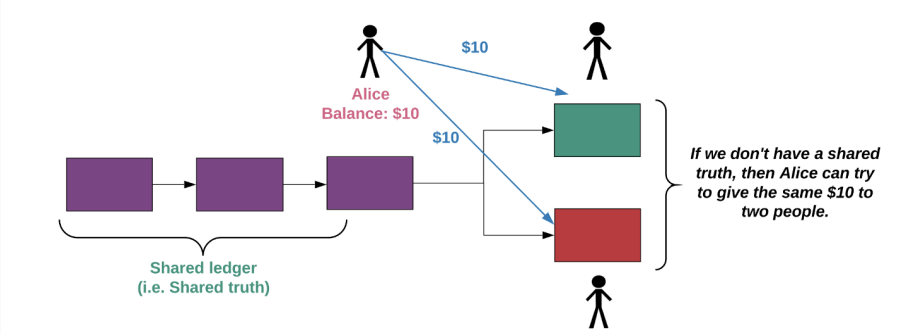
# Cryptocurrency

- Money
  1. Medium of exchange
  2. Store of value
  3. Unit of account

- Cryptocurrency: electronic cash, without a third party
  4. Preserving privacy
  5. Unforgeable
  6. Preventing double-spending

# Properties: Unforgeability

# Properties: Preventing double spending




- Fraud identification
- Presumption of innocence

# Properties: Preserving privacy

- Weak anonymity: non identification of a buyer
- Strong anonymity: non tractability of a buyer

# SCA, KYC and anonymity properties of the different payment methods

| Payment method | SCA | | KYC | | Anonymity | | |
|---|---|---|---|---|---|---|---|
| | Issuer | Proxy | Issuer | Proxy | Issuer ($\neg\,\mathcal{P}\mathtt{An}$ and $\neg\,\mathtt{Unlnk}$ always holds) | Merchant ($\neg\,\mathcal{M}\mathtt{An}$ always holds) | Proxy |
| Cash | no | × | × | × | $\mathcal{M}\mathtt{An}$ | $\mathcal{P}\mathtt{An}, \mathtt{Unlnk}$ | × |
| Physical cards | yes | × | yes | × | $\neg\,\mathcal{M}\mathtt{An}$ | $\neg\,\mathcal{P}\mathtt{An}, \neg\,\mathtt{Unlnk}$ | × |
| Google, Apple Pay, etc. | yes | × | yes | × | $\neg\,\mathcal{M}\mathtt{An}$ | $\mathcal{P}\mathtt{An}, \neg\,\mathtt{Unlnk}$ | × |
| Top-up cards | yes | × | yes | × | $\mathcal{M}\mathtt{An}$ | $\mathcal{P}\mathtt{An}, \neg\,\mathtt{Unlnk}$ | $\neg\,\mathcal{P}\mathtt{An}, \neg\,\mathtt{Unlnk}, \neg\,\mathcal{M}\mathtt{An}$ |
| Pre-Paid/gifts cards | no | × | no | × | $\mathcal{M}\mathtt{An}/\neg\,\mathcal{M}\mathtt{An}$ | $\mathcal{P}\mathtt{An}, \mathtt{Unlnk}/\neg\,\mathtt{Unlnk}$ | $(\neg\,\mathcal{P}\mathtt{An}, \neg\,\mathtt{Unlnk}, \neg\,\mathcal{M}\mathtt{An})$ |
| PayPal | yes | yes/no | yes | yes/no | $\mathcal{M}\mathtt{An}$ | $\mathcal{P}\mathtt{An}, \neg\,\mathtt{Unlnk}$ | $\neg\,\mathcal{P}\mathtt{An}, \neg\,\mathtt{Unlnk}, \neg\,\mathcal{M}\mathtt{An}$ |

# What is a currency? (money creation)

- Standard currency
  - Guaranteed by gold,
  - Then by US Dollars (Bretton-Woods),
  - Then simply fiduciary:
    - Legal tender & forced tender
    - Creation by authorized institutions
    - Guaranteed by a central bank

Nowadays: only debt

# What is a currency? (money creation)

- Standard currency
  - Guaranteed by gold,
  - Then by US Dollars (Bretton-Woods),
  - Then simply fiduciary:
    - Legal tender & forced tender
    - Creation by authorized institutions
    - Guaranteed by a central bank

- Cryptocurrency
  - Various creation mechanisms:
    - Fixed number
    - Capped/linear/bounded growth / year
    - etc.

  Gathering of mining farms (money supply)

Nowadays: only debt

# Classical and cryptographic currencies

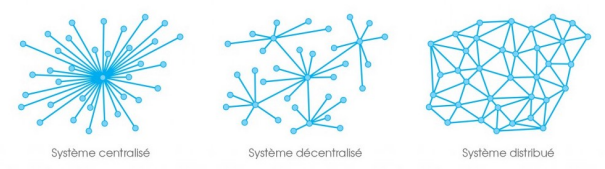
| | Classical currency | | Cryptocurrency |
|---|---|---|---|
| | Cash | Electronic | |
| Medium of exchange | ✓ | ✓ | ✓ |
| Store of value | ✓ | ✓ | ✓ |
| Unit of account | ✓ | ✓ | ✓ |
| Creation | central Bank | Debt | Automatic |
| Privacy | ✓ | x | ✓ |
| Peer 2 peer | x | x | ✓ |
| Legal guaranty, stabilization | ✓ | ✓ | X |

# The Bitcoin revolution 2009

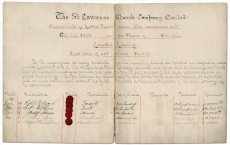

21 millions BTC

# Bitcoin

- Decentralized and distributed cryptocurrency



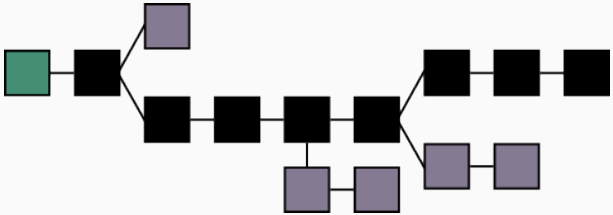

- Uses a blockchain: a shared ledger, known to all participants



# Unstoppable, because distributed


- Broadcast of all the transactions to all the nodes of the network

# Unforgeable



- A fingerprint (hash) of each transaction block is added to each new block of transaction (chain of hashes) ...

# Auditable

- Every participant owns, locally, a copy of the complete history of every transactions

# Bitcoins: main characteristics

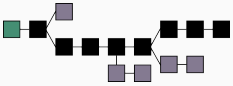
- the total number of bitcoins is finite

  21 millions BTC

- Transactions use electronic signatures
- Account number:

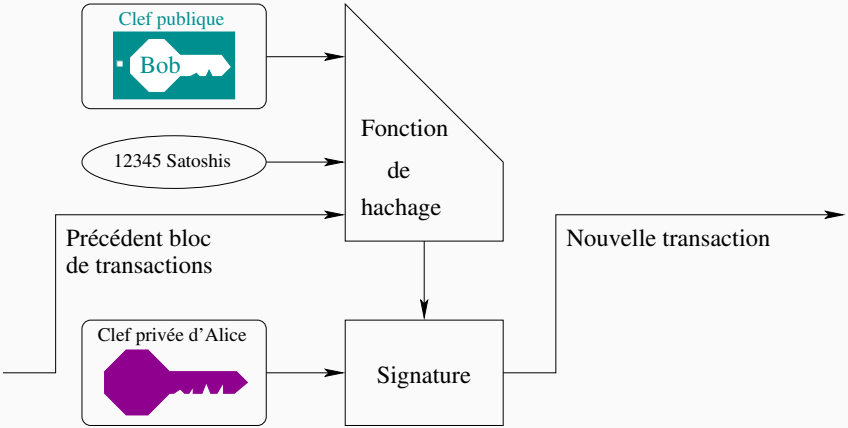
  $$\text{RIPEMD-160}(\text{SHA-256}(\text{ECDSA}_{pub}))$$

- All the transactions are public
- Blockchain: a peer-to-peer system guaranteeing the validity of transactions



# How to perform a transaction?

Alice gives 12345 Satochis (a few cents) to Bob.



# Mining: hashing target as a Proof-of-work

Example target:

`0000 0000 0000 0000 002e 8fcc c211 838c 7d12 c913 d13b 9686 e8f6 3127 cb57 e712`

Find a number *n* such that:

$$\text{SHA-256}(\text{SHA-256}(\text{Transactions}, \textit{nonce})) = x < \text{Target}$$

Must have at least 18 zeroes (even $18 * 4 + 2 = 74$ bits) for the msb of $x$

# Mining: hashing target as a Proof-of-work

Example target:

`0000 0000 0000 0000 002e 8fcc c211 838c 7d12 c913 d13b 9686 e8f6 3127 cb57 e712`

Find a number *n* such that:

$$\text{SHA-256}(\text{SHA-256}(\text{Transactions}, \textit{nonce})) = x < \text{Target}$$

Must have at least 18 zeroes (even $18 * 4 + 2 = 74$ bits) for the msb of $x$

**Strategy: brute force**
Randomly try the possible values for *nonce*

# Electronic wallet

- Consultation of the balance
- Completion of a transaction
- Coins storage management
- Creation of new account keys

Where are my private keys?

# Digital wallet solutions

1. Security
2. Availability
3. Ease of access

# Digital wallet solutions

1. Security
2. Availability
3. Ease of access

Hardware

Digital

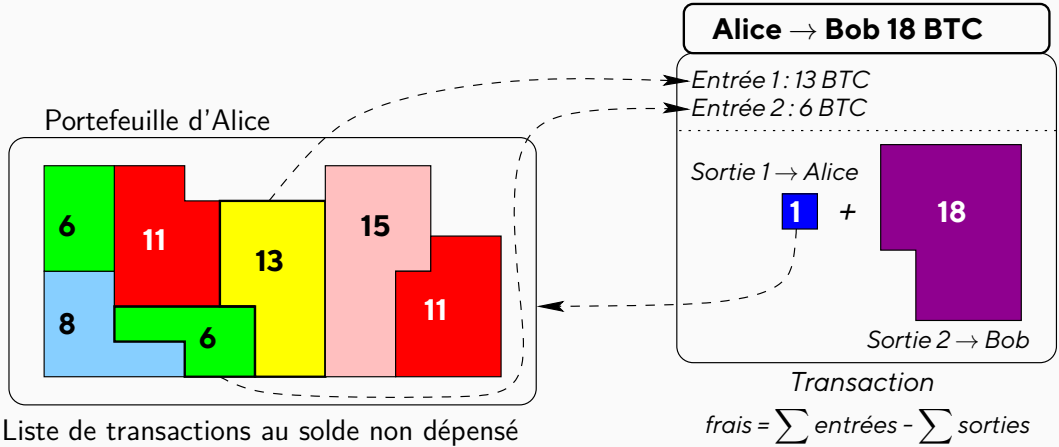
Clouded

# Main digital wallets

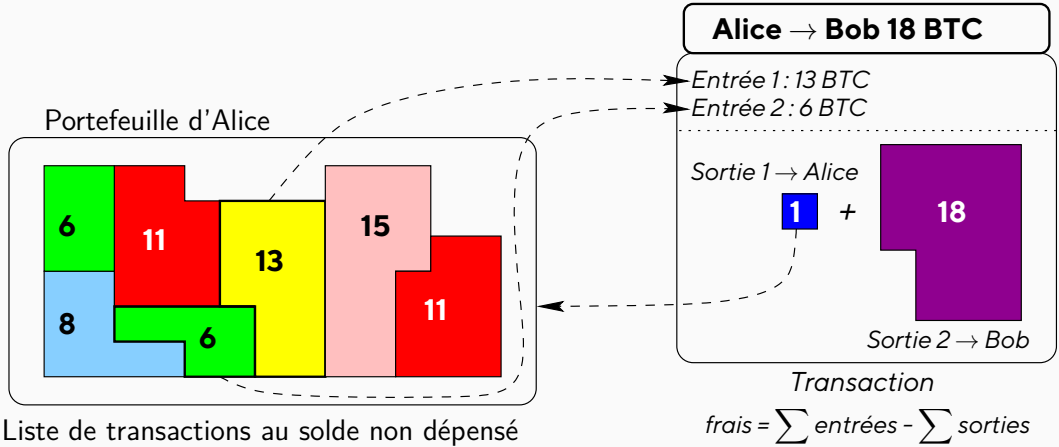
| Type | Storage | Security | Example | #currencies | mobile version |
|---|---|---|---|---|---|
| **Hardware** | physical (cold wallet) | +++ | Ledger Nano S | 27 | n/a |
| | | | KeepKey | 7 | n/a |
| | | | Trezor | 8 | n/a |
| **Software** | local | + | Bither | 1 | ✓ |
| | | | Coplay | 150 | ✓ |
| | | | Electrum | 1 | ✓(Android) |
| | | | Exodus | 30 | × |
| | | | Jaxx | 60 | ✓ |
| **Cloud** | distant | – | Blockchain.info | 2 | n/a |
| | | | Coinbase | 3 | n/a |
| | | | Kraken | 1 | n/a |
| | | | Bittrex | 190 | n/a |

(march 2018)

# Pay 18 BTC with coins



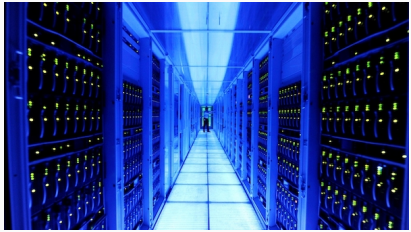
# Pay 18 BTC with coins



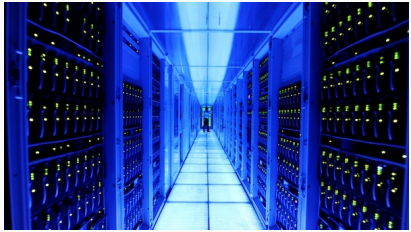

- Only owned bitcoins can be spend

# Outline

# Should we trust our remote storage?



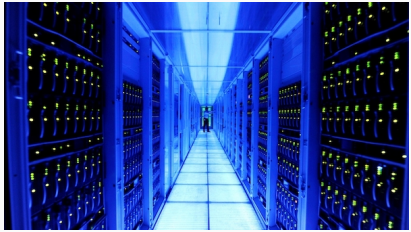
# Should we trust our remote storage?



Many reasons not to

- Outsourced backups and storage
- Sysadmins have root access
- Hackers breaking in

# Should we trust our remote storage?



Many reasons not to

- Outsourced backups and storage
- Sysadmins have root access
- Hackers breaking in

Solution:

# Clouds



# Clouds





# Properties

Acces from everywhere

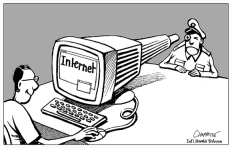
Avaible for everything:

- Store documents, photos, etc
- Share them with colleagues, friends, family
- Process the data
- Ask queries on the data

# Current solutions

Cloud provider knows the content and claims to actually

- identify users and apply access rights
- safely store the data
- securely process the data
- protect privacy

Users need more Storage and Privacy guarantees

- confidentiality of the data
- anonymity of the users
- obliviousness of the queries

# Broadcast encryption (Fiat-Noar 1994)

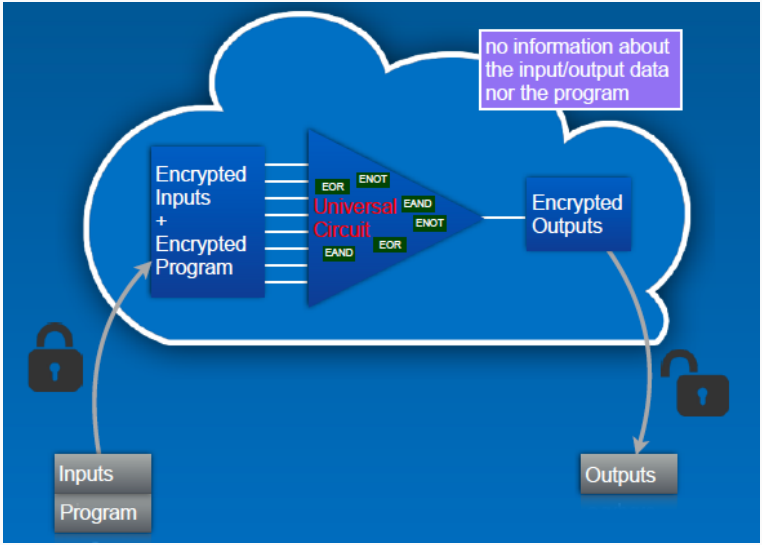
The sender can select the target group of receivers to control who access to the data like in PAYTV

# Functional encryption [Boneh-Sahai-Waters 2011]

The user generates sub-keys $K_y$ according to the input $y$ to control the amount of shared data.

From $C = Encrypt(x)$, then $Decrypt(K_y, C)$, outputs $f(x, y)$

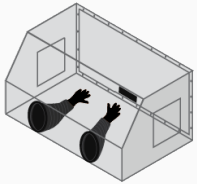
# Fully Homomorphic Encryption [Gentry 2009]



# Fully Homomorphic Encryption [Gentry 2009]

FHE: encrypt data, allow manipulation over data.

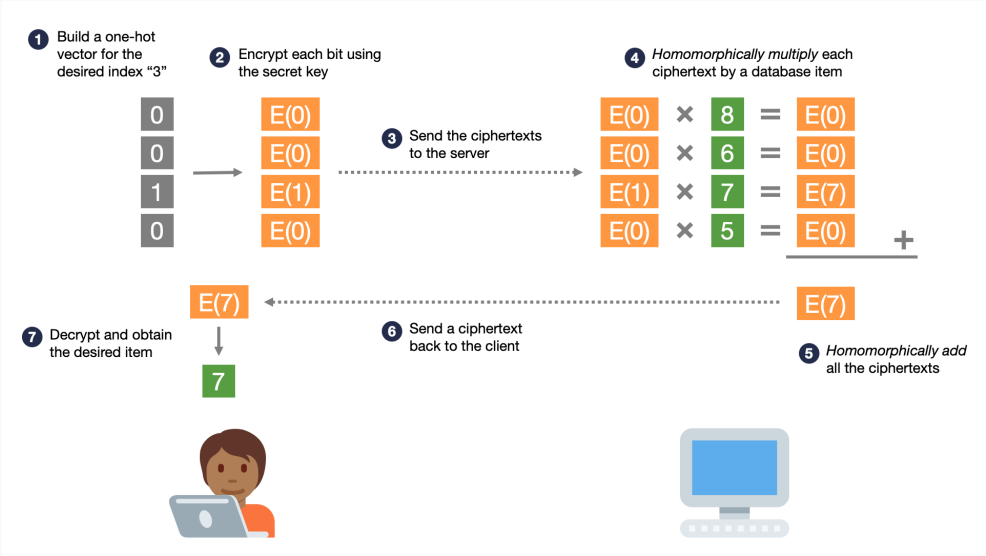
Symmetric Encryption (secret key) is enough



$$f(\{x_1\}_K, \{x_2\}_K, \ldots, \{x_n\}_K) = \{f(x_1, x_2, \ldots, x_n)\}_K$$

- Allows private storage
- Allows private computations
- Private queries in an encrypted database
- Private search: without leaking the content, queries and answers.

# Private Information Retrival



# Outline

# Privacy vs. Confidentiality

**Confidentiality**
Prevent disclosure of information to unauthorized users

**Privacy**
- Prevent disclosure of personal information to unauthorized users
- Control of how personal information is collected and used

# Data Privacy and Security Measures

**Access control**
Restrict access to the (subset or view of) data to authorized users

**Inference control**
Restrict inference from accessible data to additional data

**Flow control**
Prevent information flow from authorized use to unauthorized use

**Encryption**
Use cryptography to protect information from unauthorized disclosure while in transmit and in storage

# 2 kinds of data

- Personal data
- Anonymous data

**CNIL:**
“Dès lors qu’elles concernent des personnes physiques identifiées directement ou indirectement.”

**French Law:**
“Pour déterminer si une personne est identifiable, il convient de considérer l’ensemble des moyens en vue de permettre son identification dont dispose ou auxquels peut avoir accès le responsable du traitement ou toute autre personne.”

# How to evaluate the security?

**Three criteria of robustness:**

- is it still possible to single out an individual ?
  **Singling out (Individualisation):** the possibility to isolate some or all records which identify an individual in the dataset
- is it still possible to link records relating to an individual ?
  **Linkability (Correlation):** ability to link, at least, two records concerning the same data subject or a group of data subjects.
- can information be inferred concerning an individual?
  **Inference (Deduction):** deduce, with significant probability, the value of an attribute from the values of a set of other attributes

# Example

| ID | Age | CP | Sex | Pathology |
|---|---|---|---|---|
| Paul Sésame | 75 | 75000 | F | Cancer |
| Pierre Richard | 55 | 78000 | F | Cancer |
| Henri Poincarré | 40 | 71000 | M | Influe |

# Randomization

Alter veracity of the DB to remove the link

- **Noise addition:** modifying attributes in the dataset such that they are less accurate whilst retaining the overall distribution
- **Permutation:** shuffling the values of attributes in a table so that some of them are artificially linked to different data subjects,
- **Differential Privacy:** requires the outcome to be formally indistinguishable when run with and without any particular record in the data set.

**Example**
Q = select count() where Age = [20,30] and Diagnosis=B

Answer to Q on D1 and D2 should be indistinguishable, if Bob in D1 or Bob out D2.

# Differential Privacy

C. Dwork : “Differential Privacy”, International Colloquium on Automata, Languages and Programming , 2006.

**Definition**
Let $\epsilon$ be a positive real number and $\mathcal{A}$ be a randomized algorithm that takes a dataset as input (representing the actions of the trusted party holding the data). The algorithm $\mathcal{A}$ is $\epsilon$-differentially private if for all datasets $D_1$ and $D_2$ that differ on a single element (i.e., the data of one person), and all subsets $S$ of $\mathrm{im}\mathcal{A}$,

$$\Pr[\mathcal{A}(D_1) \in S] \leq e^{\epsilon} \times \Pr[\mathcal{A}(D_2) \in S]$$

where the probability is taken over the randomness used by the algorithm.

# Pseudonymisation

| ID | Age | CP | Sex | Pathology |
|---|---|---|---|---|
| 1 | 75 | 75000 | F | Cancer |
| 2 | 55 | 78000 | F | Cancer |
| 3 | 40 | 71000 | M | Influe |

Replace identifier field by a new one called pseudonym.

Using Hash function

It does not ensure anonymity. Using several fields you can recover name like it has benn done by Sweeney in 2001.

**Example**
Sex + birthday date + Zip code are unique for 80 % of USA citizens. (record linkage attack)

# k-Anonymity

- Identify the possible fields that can be used to recover data (generalisation).
- Modify them in order to have at least $k$ different lines having the same identifiers.

It reduce the probabolity to guess something to $1/k$

Advantage: Analysis of data still give the same information that the orginal data base.

# Example: k-Anonymity

| Activity | Age | Pathology |
|---|---|---|
| M2 | [22,23] | Cancer |
| M2 | [22,23] | Blind |
| M2 | [22,23] | VIH |
| PhD | [24,27] | Cancer |
| PhD | [24,27] | Allergies |
| PhD | [24,27] | Allergies |
| L | [20,21] | Cancer |
| L | [20,21] | Cancer |
| L | [20,21] | Cancer |

3-Anonymity

Activity for student can be Master licence or PhD instead of name and activty, age can be ranged.

# Disadvantages: k-Anonymity

- It leaks negative information. For instance you are not in all the other catergories.
- If all personn have the same value then the value is leaked.
- Main problem is to determine the right generalisation (it is difficult and expensive).

Minimum Cost 3-Anonymity is NP-Hard for $|\Sigma| = 2$ (Dondi et al. 2007)

# l-diversity

Aims at avoiding that all person have the same values once they have been generalized.

*l* values souhld be inside each field after generalisation. It allows to recover information by mixing information with some probability

| Activity | Age | Pathology |
|---|---|---|
| M2 | [22,23] | Cancer |
| M2 | [22,23] | Allergies |
| M2 | [22,23] | VIH |
| PhD | [24,27] | Cancer |
| PhD | [24,27] | VIH |
| PhD | [24,27] | Allergies |
| L | [20,21] | VIH |
| L | [20,21] | Allergies |

# t-closeness

Knowledge of global distribution of sensitive data of a class of equivalence.

It tries to reduce the weaknesses introduced by the l-diversity.

$t$ is the factor that says how we are far from a global distribution.

- How to split data into partion to obtain all the same distribution.
- If all class of equivalence have the same number of data, what is the utility of any analysis of the data basis ?

# Summary

| Is Risky | Singling out | Linkability | Inference |
|---|---|---|---|
| Pseudonymisation | Yes | Yes | Yes |
| Noise addition | Yes | May not | May not |
| Substitution | Yes | Yes | May not |
| Aggregation or K-anonymity | No | Yes | Yes |
| L-diversity | No | Yes | May not |
| Differential privacy | May not | May not | May not |

# Outline

# Conclusion

- Introduction to cryptography
- Security
- Presentation of attacks
- Next semester Security Models