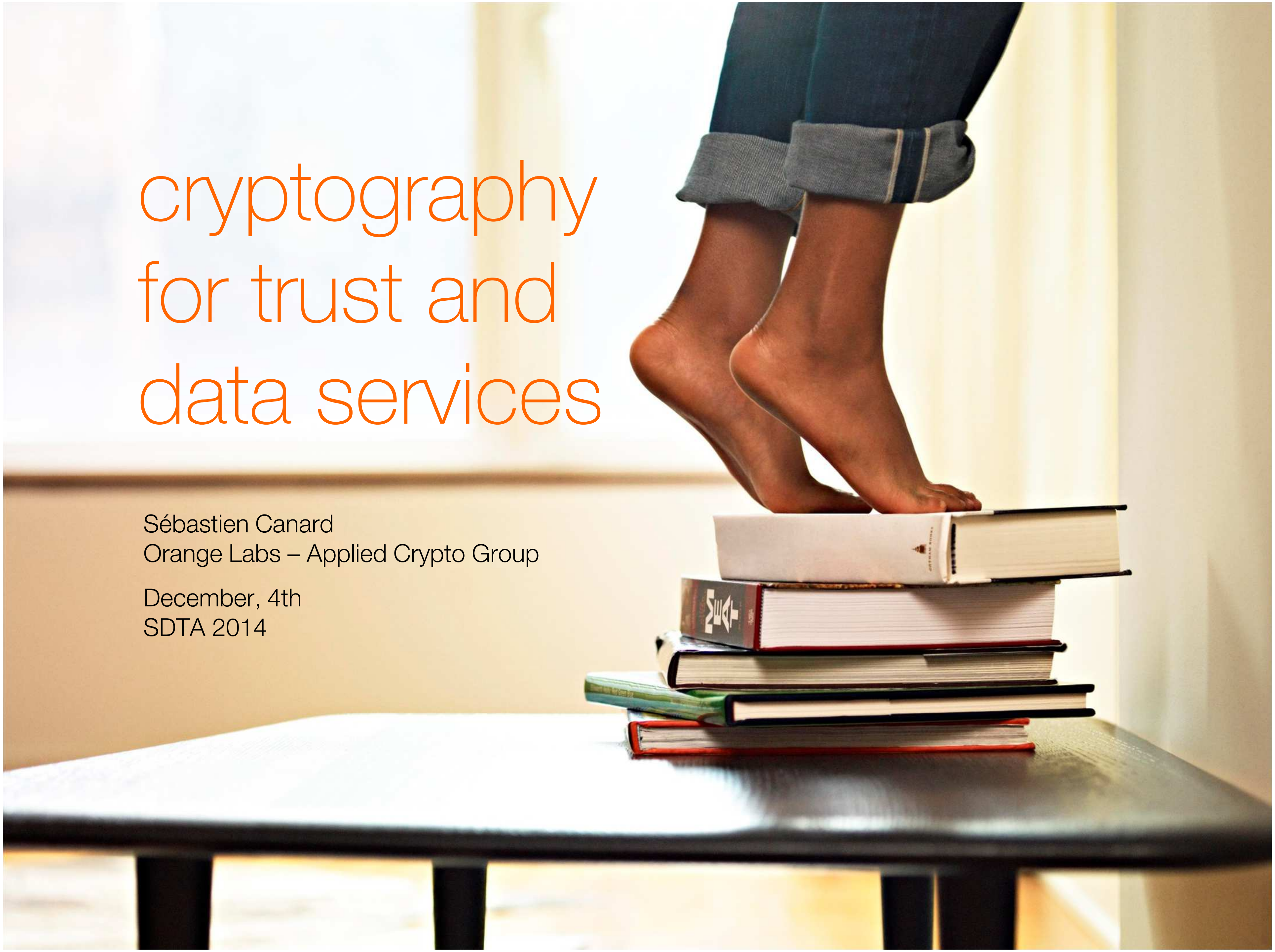


cryptography for trust and data services

Sébastien Canard
Orange Labs – Applied Crypto Group

December, 4th
SDTA 2014

let me introduce you Alice…

- she has a smartphone
- she works for a small company
- she makes use of public transportations
- she likes cinema and theatre
- she lives in a place where cultural activities are well funding
- she likes using new technologies… but not at any price

two modern services Alice can use

contactless services

- in France(*)
  - more than 3 millions of connected users
- transportation (several experimentations in France)
- payment (some bank cards, Orange Cash, Apple Pay, …)
- loyalty cards, tag reading, …

cloud computing

- in France(**)
  - 29% of companies use cloud computing
  - 5000 M€ in 2014 (+100% in 2 years)
- IaaS, PaaS, SaaS services
- storage and/or compute

can Alice make use these services in trust?

(*)source AFSCM

(**)source http://www.cloudindex.fr/

confidentiality of her companies' data

- to protect and preserve the confidentiality of information means to ensure that it is not made available or disclosed to unauthorized entities
- these services need to manipulate sensitive data
  - administrative documents
  - sensitive data related to competitiveness
- what a service provider can do to give confidence?
  - do they have access to the data…
  - …while ensuring a good and appropriate service?

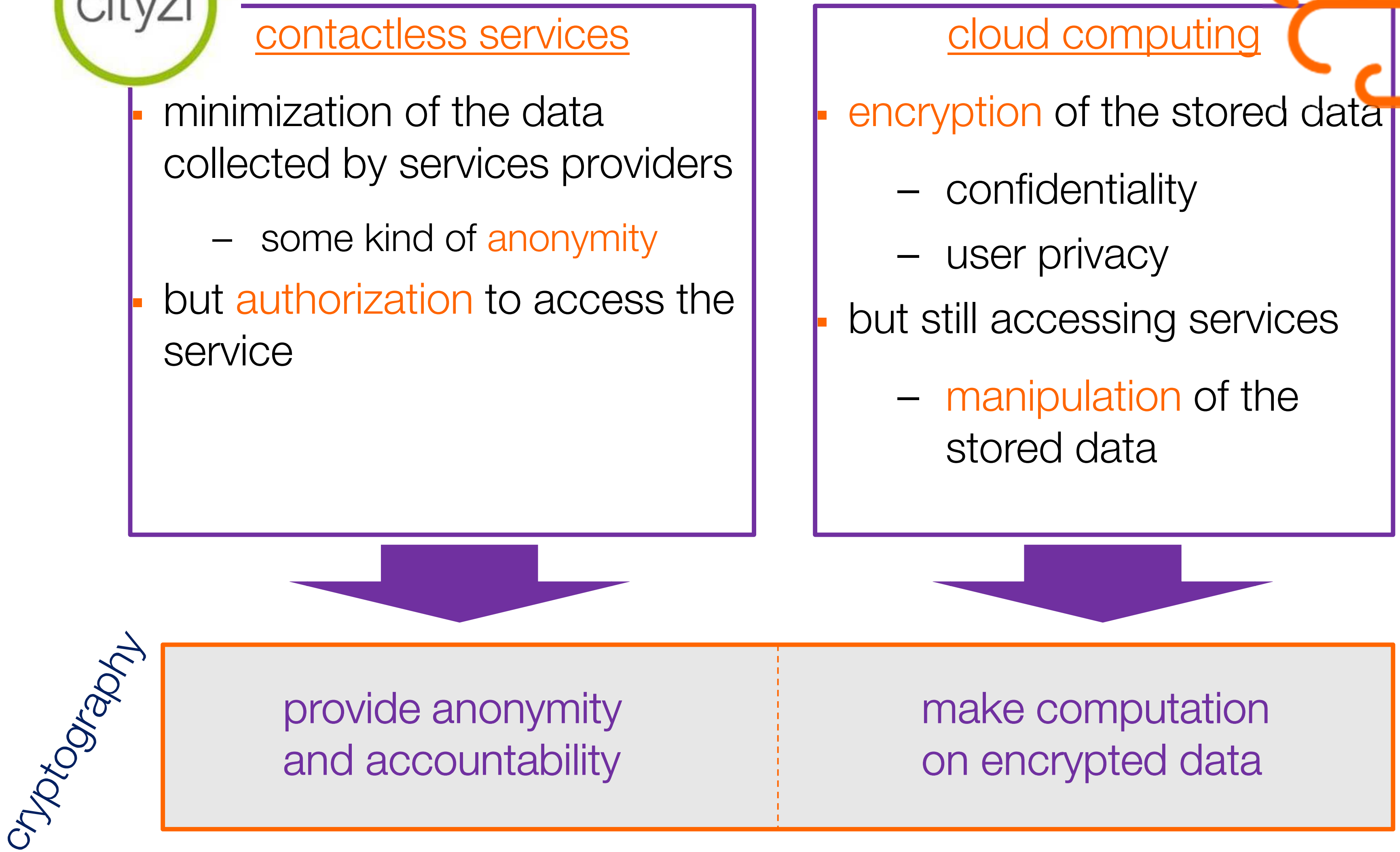
protection of her privacy

- in France, these services should work in accordance to the "loi informatique et liberté"
  - transparency of the data gathering
  - use of the data should be clear
  - relevant data gathering
  - data precision
  - right to oblivion
- what a service provider can do to give confidence?
  - verify the sensitivity of data, supervise data transfer
  - provide solutions to protect the privacy of customers
  - how to protect the privacy of customers…
  - … while offering them the best possible service?

can cryptography be useful?

- historical objectives
  - confidentiality
  - (data) authentication
  - integrity
  - non repudiation
- new objectives
  - provide tools to obtain conflicting properties
  - including data protection

cryptography and trust in new services



cityzi

contactless services

- minimization of the data collected by services providers
  - some kind of anonymity
- but authorization to access the service

cloud computing

- encryption of the stored data
  - confidentiality
  - user privacy
- but still accessing services
  - manipulation of the stored data

cryptography

provide anonymity and accountability

make computation on encrypted data

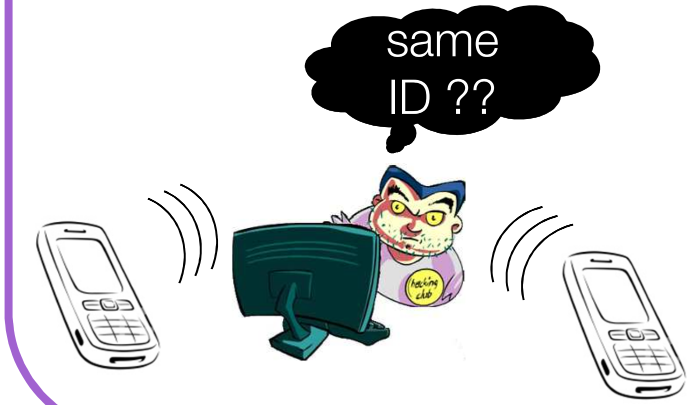
anonymity and accountability

ANONYMITY

- having one communication log
- infeasibility to link such communication with an identity

(NON) TRACEABILITY

- having 2 distinct communication logs
- infeasibility to know whether both communications are related to the same identity



accountability

- anonymity is a good point for privacy
  - permits data minimization
  - “I belong to the group of authorized users”
- but anonymity should not lead to more fraud
  - money laundering, anonymity of terrorists, etc.
- we also need accountability
  - the user should be authorized
  - necessity to revoke the anonymity in case of fraud
    - by whom? when?
    - it depends on the use case and on legal restrictions
  - be careful on false accusations

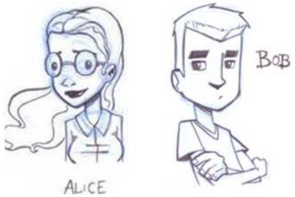
standardized cryptographic solutions

- ISO/IEC SC27 WG2
- group signatures
  - ISO/IEC 20009 Part 2
  - each group member can sign messages on behalf of the group
  - each signature is anonymous, except for a designated opening manager
- blind signatures
  - ISO/IEC 20009 (future Part 3)
  - a signer can sign documents that he does not know
  - the user who obtain the signature of his choice is anonymous in the group of users having obtain a signature from this signer
  - the user is authenticated by the signer when he obtains the signature

actors in a group signature scheme

- issuer
  - manage the group
  - permits addition and deletion of group members

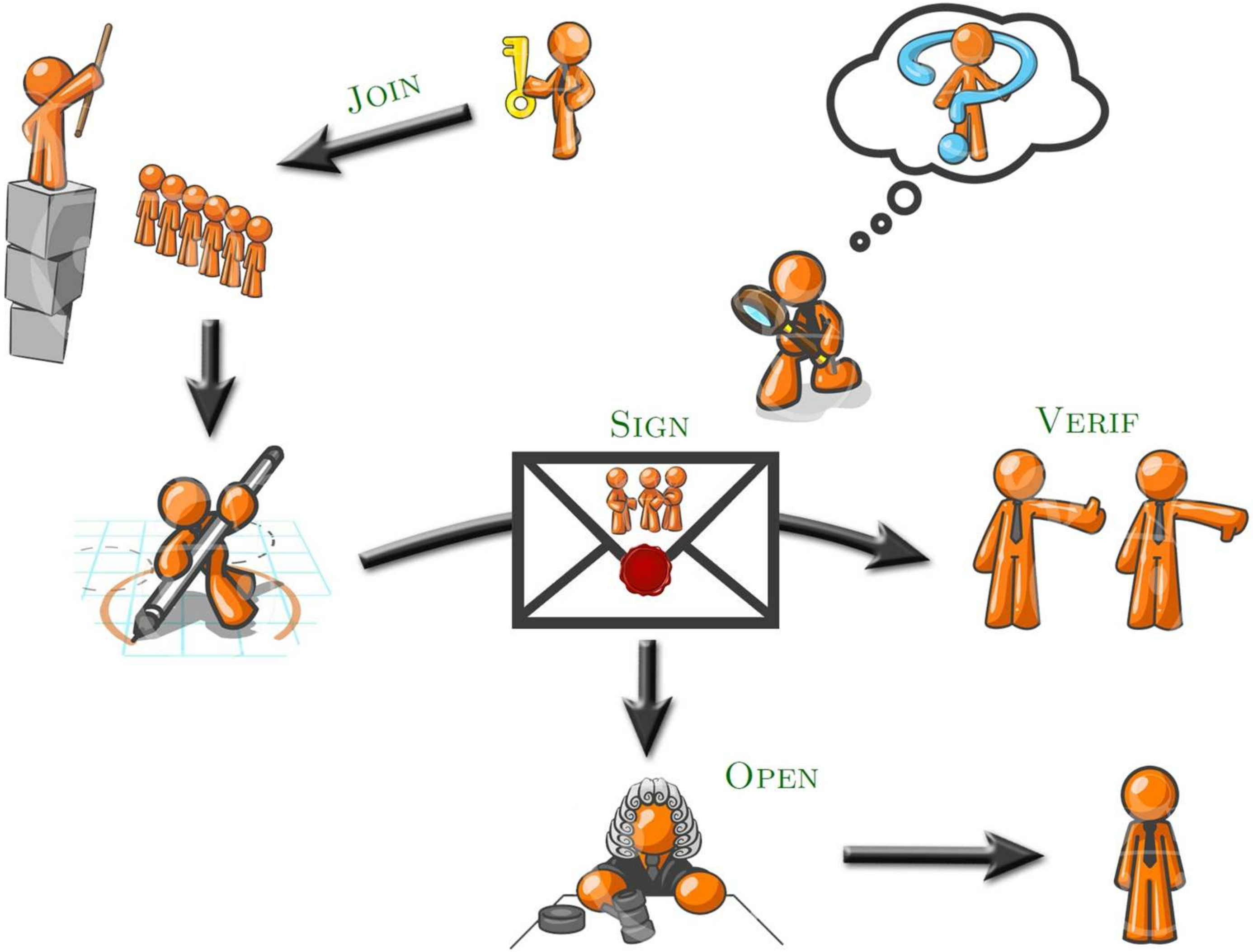
- group members
  - need interaction with the group manager
  - able to sign on behalf of the group



- opener
  - can revoke the anonymity of a signature

- anybody else
  - can verify the correctness of a group signature
  - does not obtain the identity of the signer

main procedures



security properties

- correctness
  - it pertains to signatures generated by honest group members
  - the signature should be valid
  - the opening algorithm should correctly identify the signer
  - the proof returned by the opening algorithm should be accepted

- traceability
  - the attacker is unable to produce a signature such that
    - either the honest opener declares itself unable to identify the origin of the signature, or,
    - the honest opener believes it has identified the origin but is unable to produce a correct proof of its claim

security properties

- anonymity
  - the attacker is unable to recover the identity of a signer from signatures
    - with messages of its choice
    - between two group members of its choice

- non-frameability
  - the attacker is unable to create a judge-accepted proof that
    - an honest user produced a certain valid signature
    - unless this user really did produce this signature

suitable for many use cases

anonymous access control

- authorization to access the place or the service
- anonymity within the group of authorized entities
- case of transportation

Alice's transportation

Alice's payments

e-vote systems

- a voter is a member of the group of authorized voters
- anonymity of the votes
- (without anonymity revocation)

e-cash systems

- a coin is a member of a group of authorized coins
- each spending corresponds to a group signature
- double spending detection

how can it be done in practice?

- how to ensure membership?
  - each group member obtains a signature **s**
    - on a secret value **x**
    - by the Issuer

$\mathbf{s}=\textsc{Sign}(\mathbf{x},\mathbf{isk})$

- how to ensure anonymity?
  - the secret value **x** and the signature **s** are not revealed during the group signature process
  - based on the zero-knowledge paradigm
- how to revoke the anonymity?
  - additional encryption of a component of the signature **s**

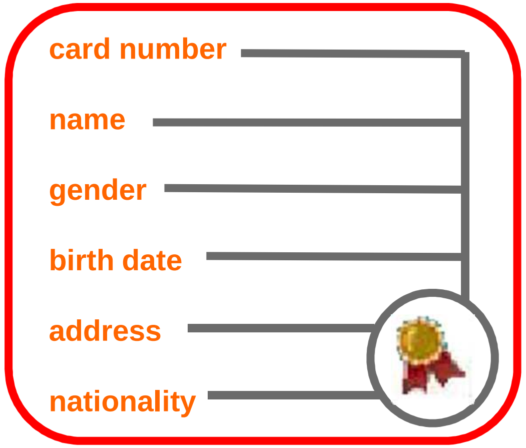
management of user attributes

- case of static attributes…
  - identity card: name, address, birthdate, etc.
  - student card: name, student identification number, University, studies, etc.

- …and non traceability in proximity services…
  - transportation, cinema, access control, etc.
  - refunds, advantages, etc.
- … in a digital world
- we can use anonymous credential systems

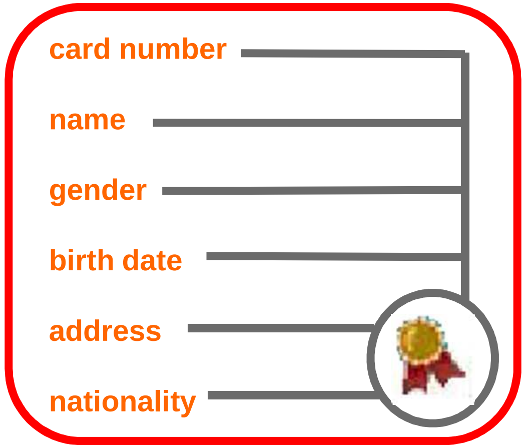
general principle

- objective = minimization of the personal data that are given to third parties
- certification of the attributes by an authorized entity
  - identity card by the local city hall
  - student card by the University
- disclosure of all or part of the certificate when accessing a service
  - « I'm a student in Caen », « I'm under 25 »
  - similar to group signature schemes

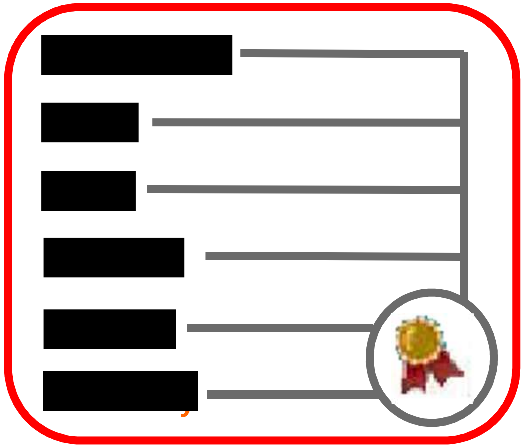


how to use a credential

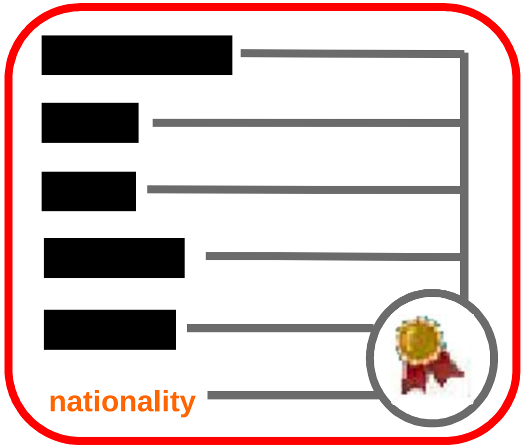
reveal all attributes



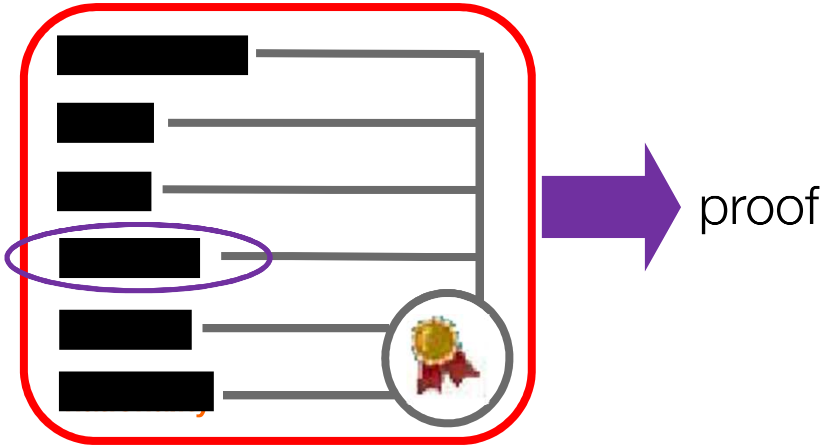
hide all attributes



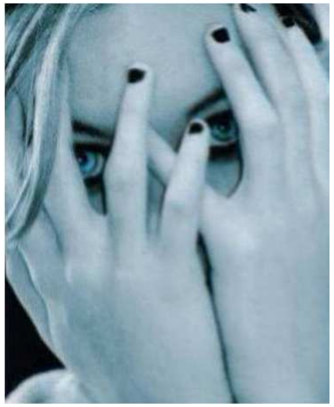
reveal some attributes
and hide others



prove some statements
on an attribute

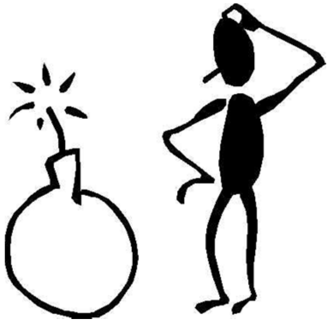


what kind of proof

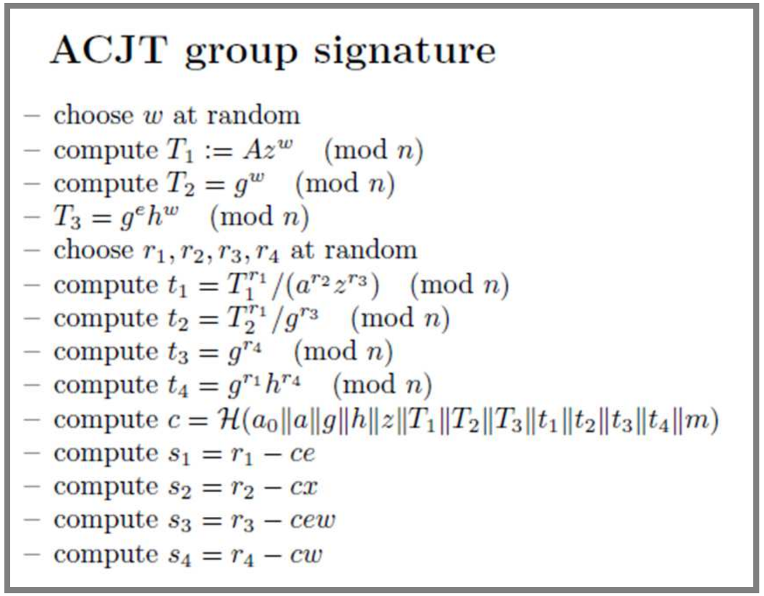


- an attribute is greater or lower than a public value
  - « I’m more than 65 »
- an attribute is in a public interval
  - « I’m between 18 and 25 »
- an attribute has a public size
- two certificates contain the same attribute
  - « I’m a student and under 25 »
  - using both student and identity cards

other problems



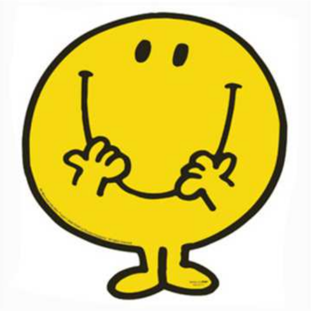
- the attributes should not be all revealed request after request
- how to prove that this my identity card?
  - we can use a photo
- efficiency of an implementation in a smart card or a mobile phone
  - equivalent to a dozen of RSA signatures
  - can it be implemented practically?
  - can we improve efficiency?



ACJT group signature

- choose w at random
- compute $T_1 := Az^w \pmod{n}$
- compute $T_2 = g^w \pmod{n}$
- $T_3 = g^e h^w \pmod{n}$
- choose r_1, r_2, r_3, r_4 at random
- compute $t_1 = T_1^{r_1}/(a^{r_2} z^{r_3}) \pmod{n}$
- compute $t_2 = T_2^{r_1}/g^{r_3} \pmod{n}$
- compute $t_3 = g^{r_4} \pmod{n}$
- compute $t_4 = g^{r_1} h^{r_4} \pmod{n}$
- compute $c = \mathcal{H}(a_0 \| a \| g \| h \| z \| T_1 \| T_2 \| T_3 \| t_1 \| t_2 \| t_3 \| t_4 \| m)$
- compute $s_1 = r_1 - ce$
- compute $s_2 = r_2 - cx$
- compute $s_3 = r_3 - cew$
- compute $s_4 = r_4 - cw$

we can do it efficiently



- do pre-computations
  - all modular exponentiations can be pre-computed
  - necessitates storage (most of time possible)
- delegation of computations
  - part of the computations can be delegated to a more powerful entity
    - SIM card vs. mobile phone
    - PC vs. server

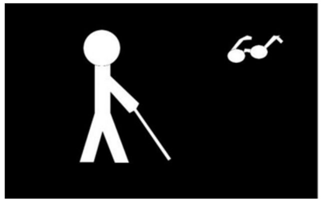
  - need to find a compromise between security and efficiency
    - SIM card: secure but not very powerful
    - smart phone: powerful but not enough secure
- an anonymous credential system can be executed in less than 300 ms in a commercialized SIM card (helped by a smartphone)

intermediate conclusion

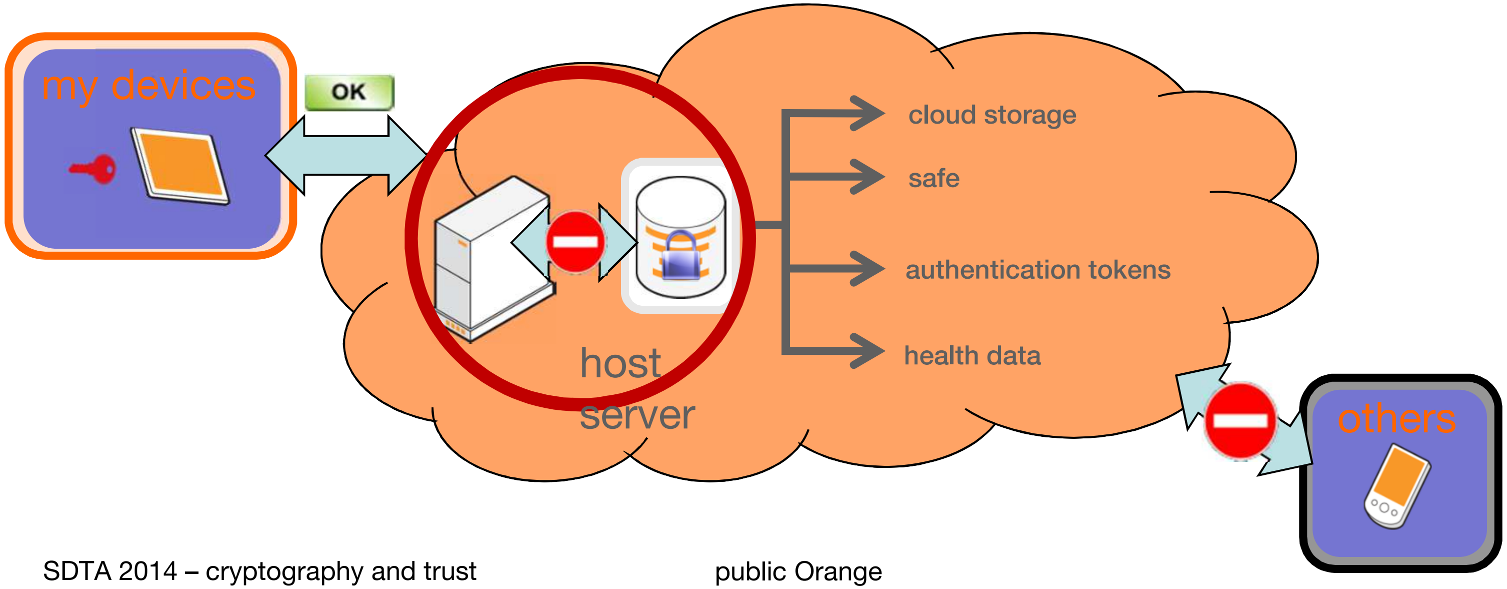
- the way to efficiently protect the privacy of Alice in contactless service is now a reality
- cryptography can help
  - ISO standards
  - efficient implementations
  - big companies are working (IBM, Microsoft, Orange, …)
- customers want to protect their privacy…
- … but not always service providers
  - partial traceability is possible (e.g. for a given service provider)
  - anonymous profiling can be done
- we need to show again and again how powerful cryptography is…

computation on encrypted data

the concept of blind storage



- data storage
  - confidential documents, administrative documents
  - digital safes, cloud storage, …
- confidentiality of data ⇒ encryption of the data
  - the host server CANNOT obtain the data in clear
  - it stores the data « in blind »



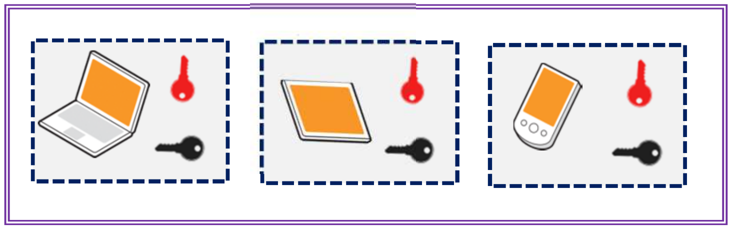
but what if we need additional services?

- share of data,
  - between devices, people/collaborators
  - with the administration
  - in a hierarchical structure
  - inside a group
- word indexation,
  - to make a search on documents related to a keyword
- or more complicated computations
  - spam filtering, targeted advertising and pricing, medical applications, private “Google” search, code compiling, …

we need encryption schemes with new features

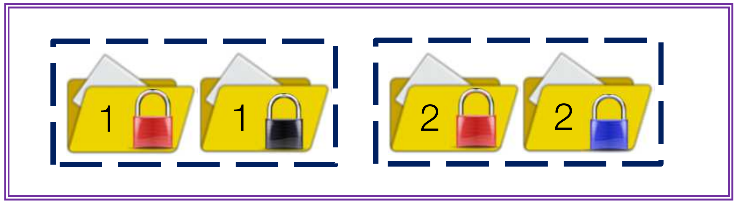
possible solutions to share data

SHARE OF THE KEY



- security hole if key compromising
- such compromising necessitates a key update for all authorized devices

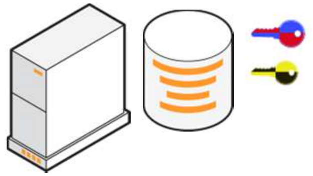
DUPLICATION OF FILES



- good security, less flexibility
- a lot of keys to manage
- additional work when withdrawing an access right

VS.

PROXY RE-ENCRYPTION



- + lost of a device
- + fine-grained rights

- best alliance of security and flexibility

a cryptographic solution

- based on a public key encryption system
  - a public key to encrypt data
  - a private key to decrypt data
- additional role (a blind storage back-end)
  - transform a message encrypted for Alice into a message encrypted for Bob
    - if Alice agrees
    - without obtaining any knowledge on Alice and Bob's keys
    - without obtaining any knowledge of the encrypted message
  - for this purpose, manage a particular cryptographic re-encryption key
- we encrypt an data specific secret key to manage big files

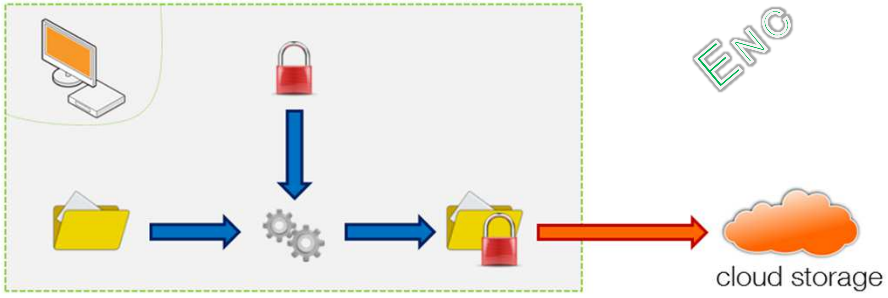
main steps

private keys

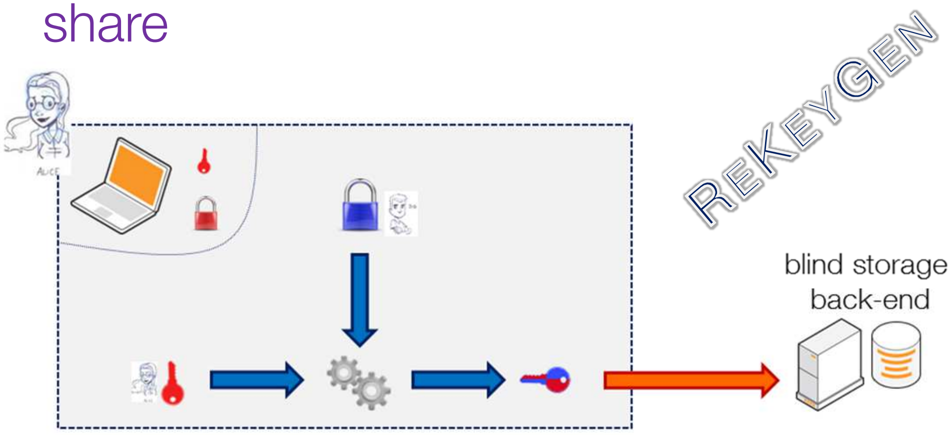
public keys

re-encryption keys

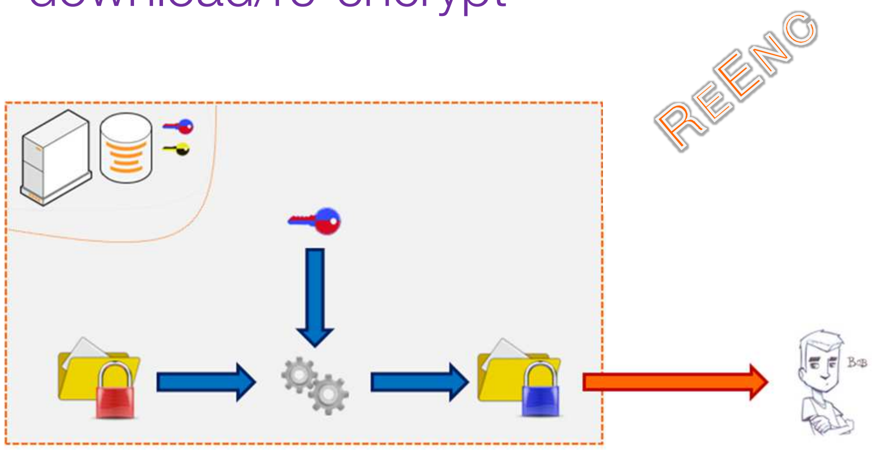
upload/encrypt



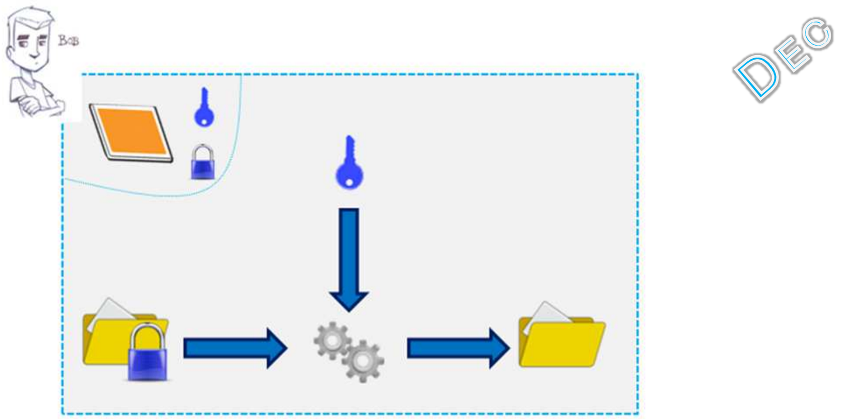
share



download/re-encrypt



download/decrypt



security and efficiency

- the decryption key is not shared between several devices
- the data is not duplicated on servers
- the owner is contacted only once for the creation of the re-encryption keys
- the cloud storage provider is not trust
- no need to know a priori the persons with which you will share data
- each device owns a key pair
  - the private key never goes outside the device
- the data is never sent outside a device in a non-encrypted form

some possible additional features

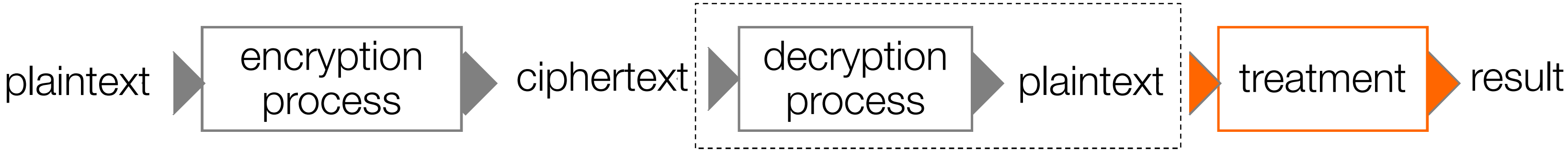
- multi-device setting
  - share with a group of devices
  - share with other users
- fine grain management of the rights
  - to manage files and folders
- possibility to share a document with a group
- what about a practical implementation?
  - performances: 10% loss w.r.t. no encryption
  - about 10 ms for encryption/decryption in a modern smartphone

legal aspects

- the case of a digital safe from the CNIL point of view
  - the service provide should not have access to the data
- obligation to give the data if requested by legal authorities
- it seems contradictory
  - but cryptography can help
  - possibility to share a “file opening” with authorities
  - no unique actor can obtain the data in clear

what about more complicated operations?

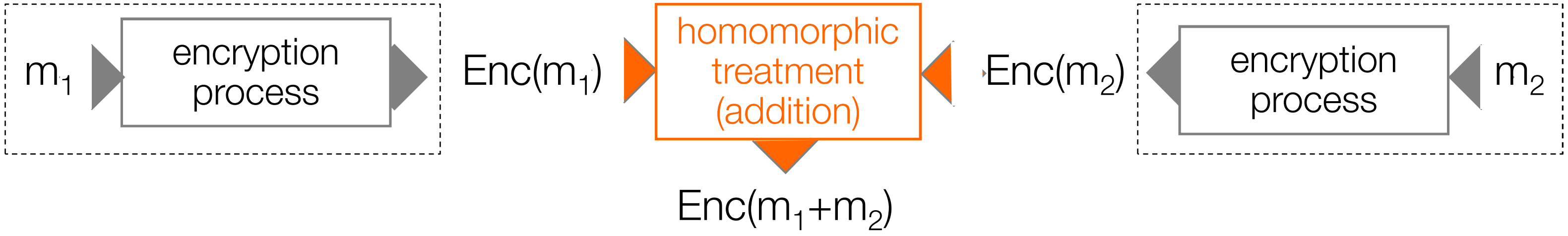
- conventional encryption



plaintext → encryption process → ciphertext → decryption process → plaintext → treatment → result

- what if the treatment could not be performed by the same entity?
  - The latter obtains the information in clear ⇒ Privacy/confidentiality threat

- (fully) homomorphic encryption allows to perform (arbitrary) specific computations on plaintexts while manipulating only the corresponding ciphertexts



m_1 → encryption process → $Enc(m_1)$ → homomorphic treatment (addition) ← $Enc(m_2)$ ← encryption process ← m_2

$Enc(m_1+m_2)$

example: addition of encrypted data without ever decrypting them!

any kind of treatment

- addition ⇒ secret ballot elections
- means / statistics ⇒ medical applications

- word search ⇒ spam filtering , private Google search

- greater than ⇒ sealed-bid auctions

- comparison ⇒ private database queries
- code compiling ⇒ cloud computing
- current homomorphic encryption schemes support either <u>addition</u> or <u>multiplication</u> but not both!
- fully homomorphic encryption schemes can handle both operations on encrypted data and thus perform arbitrary computations.

can (fully) homomorphic encryption be practical?

(*)source Coron et al., Eurocrypt 2012

security parameter	public key size	multiplication	bootstrapping
52 bits	1692 KB	0.59 sec	100 sec
62 bits	7.9 MB	9.1 sec	30 min
72 bits	18 MB	41 sec	2 h 30 min

- partially homomorphic encryption (in comparison)
  - supports only addition (Paillier) or multiplication (ElGamal)
  - size of the public key: less than 1 kb
  - time for a treatment : some ms

128 bits of security

- in practice, do we really need fully homomorphism?

how to improve the efficiency

- parameters of the scheme can depend on the evaluated circuit's depth
  - notion of leveled FHE
  - no more need to use a bootstrapping
- loss of generality
  - need to know a priori an upper bound of the circuit depth
  - but much more efficient
- best implementations necessitates less than 1 sec for a 128 bits security level(*)

(*)source HELib

can we do even better?

proxy re-encryption

broadcast encryption

identity based encryption

attribute based encryption

distributed decryption

functional encryption

searchable encryption

…

intermediate and final conclusion

- the way to efficiently protect the sensitive and personal data of Alice in cloud computing is now a reality
- cryptography can help
  - adaptive solutions
  - efficient implementations
  - big companies are working (IBM, Microsoft, Orange, …)
- the professional world seems more ready
  - but they do not want to lose their useful services
- we need to show again and again how powerful cryptography is…
  - and also some future work on cryptography, but also on the other technical and legal aspects

thank you

Orangeexpert
security

orange™