



Privacy & trust in architectures

Claudia Diaz
K.U.Leuven ESAT/COSIC

Trust based on...

- Blind trust: belief, confidence, hope, reliance, depends on personal judgement or instinct
- Trust based on a good reputation
- Trust based on control and punishment, contractual agreements
- Trust that a device or process will behave in a particular way based on its design (verifiability)
- Trust based on good PR
- Need-to-know
- “Trust me, because you do not need to”
- **Lots of different meanings !!**

- Is “social trust” (trust between people) comparable to trusting code or trusting Facebook?
 - Example: “Trusted code” or “untrusted code”?
- Should we try to have as much trust as possible or as little as possible?
 - Trust as a goal vs trust as an assumption
 - Trustworthiness: device behaves as described in specification, but that does not mean that the specification is privacy friendly
- Trust can help dealing with (reducing) complexity
 - But: mass data collection increases the complexity of securing the system
- Trust for what? I trust mobile phone providers to route my calls, but why should I trust them not to use/share data about me to my disadvantage?
 - Incentives? Perceptions? Privacy violations quite invisible
- Careful with deriving trust from usage: “if people do not trust a technology, they will not use it” => “if people use a technology, it implies that they trust it”
 - Willingness to take a risk: obvious advantage vs non-obvious disadvantage

Trusted entities/components

- Trusted entities/components are **undesirable** in computer security
 - System 1: the security relies on a trusted entity/component
 - System 2: same functionality without a trusted entity/component
 - System 2 is superior to System 1 from a security point of view
- Security: the less you need to trust entities/components, the better
 - “Trust me, because you do not need to”
 - Minimize trust / disclosure of information to reduce privacy risks -> Privacy Enhancing Technologies

Two main approaches

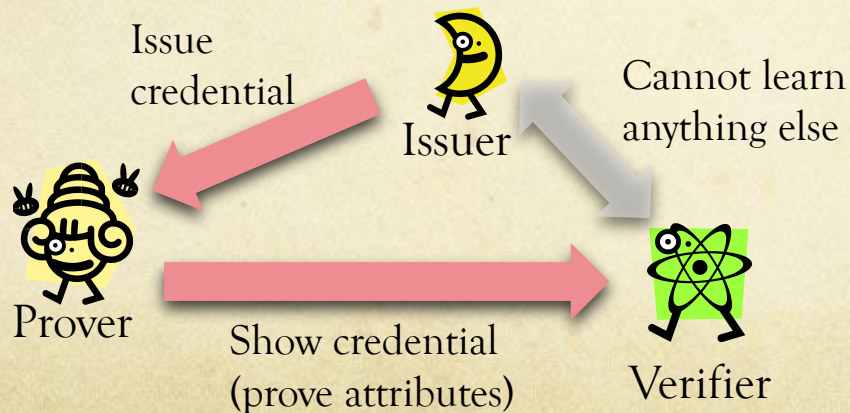
- Anonymity
 - Service provider can observe access to the service
 - Cannot observe the identity of the user
- Private computations
 - Service provider can identify user
 - Cannot observe details of the access to the service
 - Which records were accessed
 - Which search keywords were used
- All parties have assurance that the other participants in the protocol are cannot cheat

Useful distinctions?

- What is the concern?
 - Privacy towards other individuals relevant to our lives
 - Concentration of power in organizations who perform mass data collection and analysis
- What type of data?
 - Gives: the content you upload or share
 - Give-offs: clicks, search queries, behavior

Anonymous credentials

- Properties:
 - The prover convinces the verifier that he holds a credential with (certified) attributes that satisfy some conditions:
 - Example “salary>30.000 AND contract= permanent”
 - Prover cannot lie
 - Verifier cannot infer anything else aside the formula
 - Anonymity maintained despite collusion of V & I



Protection at all layers

- Easy to defeat by “changing” abstraction layer
 - Privacy properties (e.g., anonymity) do not compose
- Example: previous protocols are useless if the adversary can link transactions based on traffic data (e.g., IP address)
- Secure and private channels: protection against traffic analysis

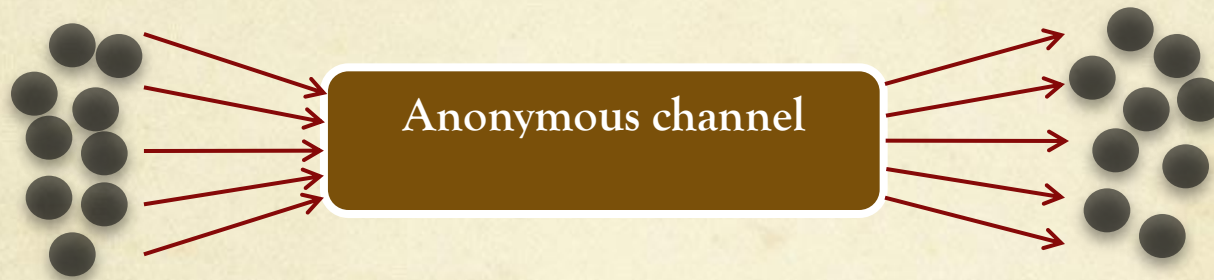


Anonymous communications

- Anonymity / unlinkability **not** provided by default by the communication infrastructure
- **Traffic data** (origin, destination, time, volume): side channel information
 - Less volume than content: coarser, but highly valuable information
 - Formats that are easy to process for machines
 - Can be used to select targets for more intensive surveillance
 - Hard to conceal
- **Adversarial:**
 - **Third party** with access to the communication channels
 - **Recipient:** adversarial or trusted (subject can authenticate over the anonymous channel)

Anonymous communications: abstract model

- Objective: hide the identity of the sender (or receiver, or both)



- Make the bit patterns of inputs and outputs different (bitwise unlinkability)
- Destroy the timing characteristics (traffic analysis resistance)

Basic Anonymity Properties

- 3rd party anonymity
 - Alice and Bob trust each other but do not want other parties to learn that they are communicating
- Sender anonymity
 - Alice sends to Bob, and Bob cannot trace Alice's identity
- Receiver Anonymity
 - Bob can contact Alice, without knowing her identity.
- Bi-directional Anonymity
 - Alice and Bob communicate without knowing each other's identities.
- Anonymity sets!

Systems for anonymous communications

- Theoretical / Research
 - Mix networks (1981)
 - DC-networks (1985)
 - ISDN mixes (1992)
 - Onion Routing (1996)
 - Crowds (1998)
- Real world systems
 - Single proxy (90s): anon.penet.fi, Anonymizer, SafeWeb
 - Remailers: Cipherpunk Type 0, Type 1, Mixmaster(1994), Mixminion (2003)
 - Low-latency communication: Freedom Network (1999-2001), JAP (2000), Tor (2005)
- Number of powerful traffic analysis attacks
- Research on anonymous communications / traffic analysis countermeasures is underfunded

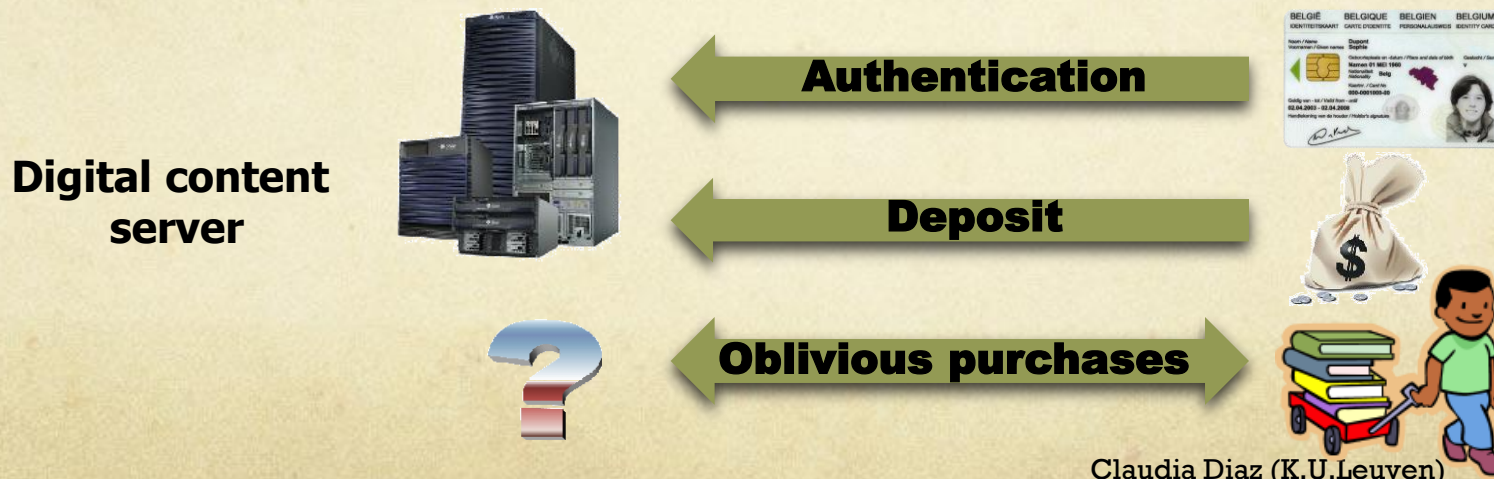
Oblivious Transfer (OT)



- A inputs two information items, B inputs the index of one of A's items
- B learns his chosen item, A learns nothing
 - A does not learn which item B has chosen;
 - B does not learn the value of the item that he did not choose
- Generalizes M instead of 2, etc.
- Example: retrieving location-based content

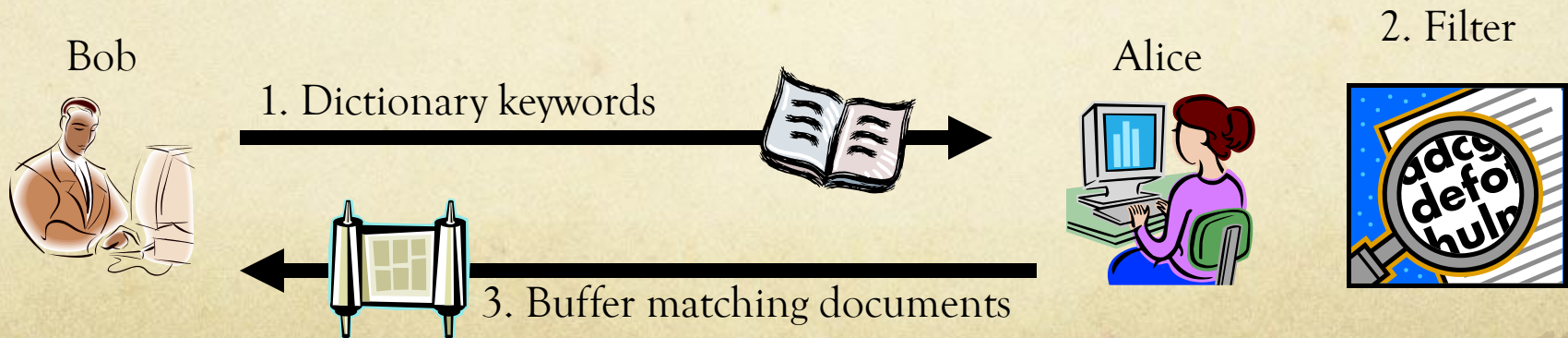
Buying digital goods (Priced Oblivious Transfer)

- Privacy of the buyer:
 - Vendor does not learn which particular item she buys
 - Vendor learns neither the amount of money paid nor the new value of the deposit ($\text{NewDeposit} = \text{OldDeposit} - \text{price}$) – only that $\text{NewDeposit} > 0$
- The vendor is assured that:
 - Buyer does not learn anything about content for which she did not pay.
 - Buyer pays the right price for the item she buys and updates the deposit correctly.



Private Search

- Alice stores documents
- Bob wants to retrieve documents matching some keywords
- Properties:
 - Bob gets documents containing the keywords
 - Alice does not learn Bob's keywords
 - Alice does not learn the results of the search



Other systems

- PrETP: Privacy enhanced Electronic Toll Pricing
 - Guarantee correct payments while disclosing *minimal* location data (spot checks) to the toll service provider
 - http://cosic.esat.kuleuven.be/road_charging/
- Privacy preserving smart metering
 - Guarantee correct payments without disclosing *detailed* energy consumption data to the utility
 - http://research.microsoft.com/en-us/projects/privacy_in_metering/
- No need to “trust” providers to protect detailed, rich, potentially sensitive data (because they do not have the data!)
 - Reduction of privacy breach risks
 - Reduction of costs in securing large databases

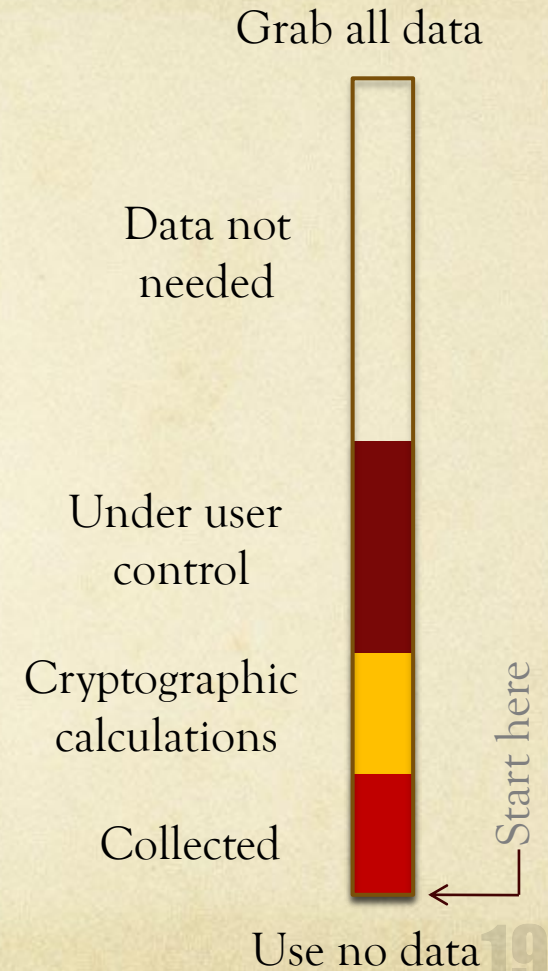
How not to engineer for privacy

A step by step guide to current practices

1. Think of a vague service – no matter how implausible
2. Engineer it to grab and store as such information from users and third parties as possible
3. Hope no one notices or complains
4. When the scandals break out fix your terms of service or do some PR
5. If the scandals persist make your privacy controls more complex
6. When DPAs are after you explain there is no other way
7. Sit on data you have no idea what to do with until your company is sold

Privacy Engineering Principles

- Define clearly what you want to do (functional)
 - Is this by itself privacy invasive?
 - Mechanisms to prevent abuse?
- Define the minimum private inputs necessary to achieve the functionality
- Build a solution to guarantee the integrity of the service that discloses no more information than necessary.
 - Push processing of private information to user devices
 - Use advanced cryptography for integrity and privacy



Conclusions

- Trust is an overloaded term with lots of intertwined meanings
 - Overloaded concepts cannot promote precise discussions and clear understanding
 - Trust is often used as a placeholder when we can't say precisely what we mean
 - A problem in interactions with the general public and between different communities
 - Security needs precision and clarity
 - Reliance on trusted entities: bad for security
 - Trust for what? Incentives? Do not confuse trusting entities to respect privacy with trusting them to provide the service

- Privacy not about perception management only, or a declaration of intention (eg, DNT)
- Solid design principles to support most functionalities
 - anonymity
 - Does not compose, network-layer
 - private computations
 - combinations of both?
- Integrate privacy and integrity – not balance
 - Identity escrow? codify misbehaviour to enable automatic deanonymization (eg, double spending) – But careful with leaving the decision to the “judgement” of some entity
 - Criminals will find ways to not be identifiable (eg, use a compromised computer)
- Privacy design methodology / privacy engineering practices needed
 - Not about checklists
 - Lots of expertise in security/privacy needed
 - Engineering Privacy by Design. S.Gürses, C.Troncoso, and C. Diaz
 - <https://www.cosic.esat.kuleuven.be/publications/article-1542.pdf>