Overview of security and privacy technologies

Claudia Diaz
K.U.Leuven ESAT/COSIC
Outline

◦ Part I: Introduction to security technologies
◦ Part II: Privacy models
◦ Part III: Examples of privacy technologies
◦ Part IV: Why the word “trust” should be better avoided
Part I: Introduction to security technologies
Security properties

- Traditionally: CIA
  - Confidentiality
  - Integrity
  - Availability

- Confidentiality
  - Prevention of unauthorized disclosure of information
Security properties (II)

- **Integrity**
  - prevention of unauthorized modification of information

- **Availability**
  - prevention of unauthorized denial of service
Security properties (III)

- Entity authentication
  - sender is who he is claiming to be

- Data authentication
  - data origin is who it is claimed to be
Security properties (IV)

- **No repudiation (origin)**
  - the sender cannot repudiate having sent a message

- **No repudiation (destination)**
  - the receiver cannot repudiate having received a message

It was not me!

I didn’t receive it

Claudia Diaz (K.U.Leuven)
Symmetric key encryption

- Alice and Bob share keys
- Achieves confidentiality
Data integrity

- Encryption does not protect against modifications
- Replace authenticity of long message by authenticity of short string
- Message Authentication Code (MAC)
  - Provides data origin authentication

\[
\text{One way function} \quad 7CB3410087
\]

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque vestibulum, nibh et hendrerit tincidunt, purus quam pellentesque tortor, in faucibus tellus arcu iaculis ante.
Data integrity

- Manipulation Detection Code (MDC) or Hash function
- MD5, SHA-1, RIPEMD

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque vestibulum, nibh et hendrerit tincidunt, purus quam pellentesque tortor, in faucibus tellus arcu iaculis ante.

Claudia Diaz (K.U.Leuven)
Public Key Cryptography

- Symmetric key limitations
  - How to establish symmetric keys?
  - How to distribute them?
  - How to store them

- Confidentiality

Claudia Diaz (K.U.Leuven)
Public Key Cryptography

- Integrity

- RSA, ElGamal

- Slow, normally combined with Symmetric Key
  - Key agreement, another full lecture...

Claudia Diaz (K.U.Leuven)
Why authentication?

- Before you grant permissions....
- Authentication: binding of identity to subject
  - Alice is....
    - The person she says
    - The computer she says
    - The process she says
    - ...
  - Prove attributes
How do we establish identity?

- One or more of the following
  - What you know: secret
  - What you have: token
  - What you are: fingerprint

- Others
  - Where you are
  - How do you type

- In practice, authentication requires more than strong cryptography (the whole environment is important)
Access control: key concepts

- Main actors
  - Principals/users (subject)
  - Resources (object)
  - Operations (action: read, write, append, execute...)

- Only authorized principals should perform authorized operations on authorized resources

Access Control Matrix

<table>
<thead>
<tr>
<th>User</th>
<th>foo1.txt</th>
<th>foo2.txt</th>
<th>foo3.txt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>write</td>
<td>read, execute</td>
<td>-</td>
</tr>
<tr>
<td>Bob</td>
<td>-</td>
<td>read, write</td>
<td>-</td>
</tr>
<tr>
<td>Charlie</td>
<td>execute</td>
<td>-</td>
<td>read, write, execute</td>
</tr>
</tbody>
</table>
Not that easy...

- What/who are the principals?
  - User = physical person or process?
  - Accountability: are users responsible for their programs?
  - Intentionality: what if there is a bug?

- Granularity of the resources: processors, files, sockets, DB records, user accounts
  - At which level do we implement security?

- Who sets/manages/updates the policy?
  - Creator of information (discretionary), central authority (mandatory)
  - Dynamic environments (users join/leave the system, change roles, etc.)
Summary Part I

- Security is a property of the overall design
- What are you trying to protect against whom?
  - Requirements specification is not trivial
  - Neither is to implement them
- Ensuring that “only authorized principals should perform authorized operations on authorized resources” is not easy
  - Each system has its own requirements, that depend on the environment: there is no perfect recipe for security
  - Even if there was... translate into implementation is not trivial
  - In some cases checking that the policy is not violated might be undecidable
- The human is often the weakest link in the chain
  - Social engineering, lack of usability...
Part II: Privacy models
Perspectives on privacy

- Popular definitions:
  - “The right to be let alone”
  - “Informational self-determination”
  - “The freedom from unreasonable constraints on the construction of one's own identity”

- Solove:
  - Identifies 16 privacy threats relating to information collection, processing and dissemination, and invasion

- Data protection:
  - Purpose, proportionality, consent, data subject’s rights, data security obligations, ...

- Technical privacy properties:
  - Anonymity, Pseudonymity, Unlinkability, Unobservability, Plausible deniability (OTR), Location privacy...

Claudia Diaz (K.U.Leuven)
“Trust-based” or “Soft” privacy

- System model: data protection oriented
  - Data subject provides her data
  - Data controller responsible for its protection

- Threat model
  - External parties, errors, malicious insider

Claudia Diaz (K.U.Leuven)
Soft privacy

- Controller: main security “user”
- Policies, access control, trust, audits (liability)
- Goal (data protection): purpose, consent, data security
Soft privacy

- Data subject has already lost control of her data
  - In practice, very difficult for data subject to verify how her data is collected and processed
Soft privacy

- Data subject has already lost control of her data
  - In practice, very difficult for data subject to verify how her data is collected and processed
  - Need to trust data controllers (honesty, competence) and hope for the best
  - Weak enforcement, low penalties

TRUST ASSUMPTIONS?

INCENTIVES?

TECHNOLOGICALLY ENFORCED?

Claudia Diaz (K.U.Leuven)
Hard privacy

- System model
  - Subject provides as little data as possible
  - Reduce as much as possible the need to “trust” other entities

- Threat model
  - Adversarial environment: communication provider, data holder
  - Strategic adversary with certain resources motivated to breach privacy (similar to security systems)
Hard privacy

- Subject is an active security “user”
- Goal (data protection): data minimization (trust minimization)
- Technologies:
  - Solutions based on anonymity/hiding identity: anonymous communications, anonymous credential protocols
  - Solutions based on hiding actions/data/content: oblivious transfer, commitments, private information retrieval, keep user data at user side

Claudia Diaz (K.U.Leuven)
Privacy = Security Property

- **Governments / Military**
  - Protection of national secrets, confidentiality of law enforcement investigations, diplomatic activities, political negotiations

- **Companies**
  - Protection of trade secrets, business strategy, internal operations, access to patents

- **Individuals**
  - Freedom from intrusion, profiling and manipulation, protection against crime/identity theft, control over one’s information

- **Shared infrastructure**
  - Despite varying capabilities infrastructure is shared
  - Telecommunications, operating systems, search engines, on-line shops, software...
  - Denying security to some, means denying it to all!

Claudia Diaz (K.U.Leuven)
Two main approaches

- **Anonymity**
  - Service provider can observe access to the service
  - Cannot observe the identity of the user

- **Oblivious Transfer (OT) / Private Information Retrieval (PIR)**
  - Service provider can identify user
  - Cannot observe details of the access to the service
    - Which records were accessed
    - Which search keywords were used
    - Which content was downloaded
    - ...

- All parties have assurance that the other participants in the protocol are cannot cheat
Anonymous authentication

Are anonymity and authentication incompatible?

Many transactions involve attribute certificates
- ID docs: state certifies name, birth dates, address
- Letter reference: employer certifies salary
- Student card: university certifies student status

Do you want to show all attributes for each transaction?

Credential: token certifying attributes
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
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.
Part III: Examples of (hard) privacy technologies
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
e-petitions

- Formal requests addressed to an authority and signed by numerous individuals

- Typically citizens provide
  - Unique identifier (name, national ID number)
  - Signature

- Verification:
  - Validating that the signatures correspond to the identifiers
  - Discarding multiple/invalid signatures

- Benefits of going electronic:
  - Many resources are needed in order to physically collect the signatures
  - Manual signature verification is a costly and tedious process

Claudia Diaz (K.U.Leuven)
The straightforward e-petition implementation

- Have users sign the petitions with their e-ID
  1. Select petition
  2. Sign using the e-ID (2-factor authentication)
  3. Check that the petition has not yet been signed with that e-ID
  4. Count (or discard) the signature

- Privacy risks
  - Leak sensitive information on political beliefs, religious inclinations, etc.
  - Through unique identifiers, petition signatures can be linked to other data

Claudia Diaz (K.U.Leuven)
e-petition requirements

- Basic requirements
  - Authentication: citizen is who claims to be (i.e., no impersonation)
  - Required attributes: citizen is entitled to sign (e.g., age ≥ 18 and nationality ∈ EU)
  - Uniqueness: citizens sign a petition only once
  - Correctness: all valid signatures are counted

- Privacy requirements
  - Citizen unlinkable to petition (i.e., not possible to identify who are the signers)
Anonymous credentials

- Cryptographic protocols between <Issuer, Prover, Verifier>
  - Prover can prove that he holds a credential with certain attributes
  - or any expression on them (simple arithmetic, boolean) (e.g. salary > 30,000 and contract = permanent)

- Unforgeability and Privacy
  - Verifier gains **no more information**: One party proves to another that a statement is true, without revealing anything other than the veracity of the statement.

- Secure even if Issuer and Verifier collude (single/multiple show)

- Security: cryptographic

Claudia Diaz (K.U.Leuven)
Citizen 

credential issuing protocol  

Credential show + Petition signing  

Multiple signing?  

Yes  

Count signature and store transcript  

No  

Claudia Diaz (K.U.Leuven)
Properties

- Only citizens entitled to sign can do so
  - Possession of e-ID + knowledge of PIN
  - Attribute verification (e.g., age, locality)
  - One credential per citizen

- Citizens can sign only once (multiple signing is detectable so that repeated signatures can be deleted)

- Collusion of credential issuer and e-Petition server does not reveal the identity of a signer

- Need for anonymous communication channel to preserve privacy properties

Claudia Diaz (K.U.Leuven)
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)

Slide credit: George Danezis
**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.

Slide credit: George Danezis
Deployed anonymous communication systems

- Email:
  - Anon.penet.fi (Helsingius 1993)
  - Cypherpunk remailers (Hughes, Finney 1996)
  - Mixmaster (Cottrell, evolving since 1995)
  - Mixminion (Danezis, 2003)

- Low-latency, bidirectional circuits (web browsing)
  - ZK Freedom network (1999-2001)
  - Java Anon Proxy (since 2002)
  - Tor (since 2003)
Electronic Toll Pricing

- Differentiated payment for mobility: Congestion pricing
  - Users will pay depending on their use of the car and roads

- European Electronic Toll Service (EETS) Decision (Oct 2009)
  - Defines EETS architecture and interfaces
  - Within three years for vehicles above 3.5 tons, all other vehicles within five years.

Slide credit: Carmela Troncoso
EETS straightforward implementation

Slide credit: Carmela Troncoso
Privacy for Electronic Toll Pricing

- Privacy issues?
  - Pay as you drive
  - Fine grained GPS data allows for inferences
    - Medical issues (visits to a Cancer specialized clinic)
    - Political affiliation (visits to the headquarters of a political party)
    - Industry espionage (visits to other companies)

- What data is necessary?
  - Final fee that the user must pay to the provider/government
  - This is the actual purpose of the whole system – and not collecting everyone’s detailed location data
    - Enormous reduction of risk and cost by eliminating the need to store all the raw data

- Legal issues
  - Actors must not be able to cheat
  - Actors must be held liable when misusing the system

Claudia Diaz (K.U.Leuven)
Privacy-Friendly Electronic Toll Pricing

- No personal data leaves the domain of the user

Slide credit: Carmela Troncoso

Claudia Diaz (K.U.Leuven)
Enforcement

- OBU in hands of the user
  - Even if the box is tamper-resistant, the input is easy to spoof

- We need to:
  - Detect vehicles with inactive OBUs
  - Detect vehicles reporting false location data
  - Detect vehicles using incorrect road prices
  - Detect vehicles reporting false final fees

- Combination of law + technology
Non-Interactive Commitment Schemes

<table>
<thead>
<tr>
<th></th>
<th>00u00 – 07u00</th>
<th>22u00 – 00u00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Highway</td>
<td>$p_1$</td>
<td>$p_2$</td>
</tr>
<tr>
<td>Primary</td>
<td>$p_3$</td>
<td>$p_4$</td>
</tr>
<tr>
<td>Residential</td>
<td>$p_{n+1}$</td>
<td>$p_n$</td>
</tr>
</tbody>
</table>

Where you at....?

HIDING PROPERTY

BINDING PROPERTY

Slide credit: Carmela Troncoso
Homomorphic commitments

- The content of the vaults can be added up without being known

Slide credit: Carmela Troncoso

Claudia Diaz (K.U.Leuven)
Creating commitments

- Slice trajectory in segments (e.g., 1 Km)

- Each segment has assigned a price per Km $p_i$

- This price is specified by the policy, example:
  - $p_i = f(\text{type road, time day})$

- A commitment per segment is created
How does it work?

Does not reveal information about the trajectory

Cannot be changed

OBU

License Plate Reader

Toll Service Provider

Toll Charger

Slide credit: Carmela Troncoso

Claudia Diaz (K.U.Leuven)
What can we prove?

- OBU was active
  - A commitment with the committed location and time must be available

- OBU used correct prices
  - Prices in the table signed by Toll Service Provider
  - Check correct pricing upon commitment opening

- OBU was at reported location
  - Compare photo location with committed location

- OBU made correct operations
  - Homomorphic commitments: prices in the “vaults” can be added to verify that they correspond to the reported final fee without being opened

Slide credit: Carmela Troncoso
Holistic analysis

- From a theoretic point of view
  - The cryptography in the system ensures both privacy and law enforcement

- From a legal point of view
  - No location data revealed
  - Data minimization by design

- From a practical point of view
  - Prototype (ARM7)
  - Performance analysis
    - Computation
    - Communication

Slide credit: Carmela Troncoso
Privacy is not “opposed” to security, but rather a security property.

Soft and hard privacy: two privacy paradigms.

Current privacy technologies are able to reconcile requirements that seem (intuitively) incompatible.

Privacy properties do not compose: need for taking into account multiple system layers.

Available technologies for implementing systems with hard privacy guarantees are less understood outside of the privacy research community.

Enormous reduction of risk and cost by avoiding mass data collection (that then needs to be secured).

Claudia Diaz (K.U.Leuven)
Part IV: Why the word “trust” should be better avoided

“Trust” has become a fashionable word, but what does it mean?

What do you understand as “trust”?
“Trust” in common language

- “Firm reliance on the integrity, ability, or character of a person or thing”
- “have confidence or faith in”
- “reliance: certainty based on past experience”
- “To have or place reliance; depend”
- “To expect with assurance; assume”
- “To be confident; hope”
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 based on policy enforcement
- Trust that a device or process will behave in a particular way based on its design
- “Trust me, because you do not need to”
Reliance on trust

- Consider an election system
- Everybody tells their vote to a person
- That person computes the result of the election
- Need to trust that person to behave honestly
- Would you trust such a system?
- Or would you rather trust a system in which even if people are dishonest there is no way they can cheat?
Trusted Computing Base

The Trusted Computer System Evaluation Criteria\(^1\) define a Trusted Computing Base (TCB) as:

- “The totality of protection mechanisms within a computer system – including hardware, firmware, and software – the combination of which is responsible for enforcing a security policy.”

Meaning: “trusted denotes everything one has to rely on for the system to remain secure”

- If a trusted component fails, security can be violated
- Trusted components are those that can hurt you


Claudia Díaz - K.U.Leuven
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

- Where legal obligations are concerned, trust would imply saying: “Instead of signing a contract, let’s just trust each other”

Claudia Diaz - K.U.Leuven
Trustworthy components

- Components that have to work as expected for the system to meet its advertised purpose are called trusted.
- Components that provide evidence that they will not fail are called trustworthy.
- Attestation: mechanism whereby one can check that a remote system is configured as advertised.
- Equates trust with expected behavior.
Trust based on reputation

- Attempt to indicate the level of trustworthiness of an entity based on its prior behavior
  - Assumes that if an entity has behaved honestly in the past, it will continue to do so in the future
  - Incentives for establishing and maintaining a good reputation
  - Example: eBay

- Collaborative systems: users share information with each other

- Challenges:
  - Ensuring that ratings come from real and distinct users (prevention of Sybil attacks)
  - Privacy of users providing the ratings
Trusted Computing

- Goal of the TCG (Trusted Computing Group): “to make the Web a safer place for surfers”

- A core element in the security architecture developed by the TCG is a set of so-called *roots of trust*, defined as:
  - “Components that must always behave in the expected manner, because their misbehavior cannot be detected.”
  - **Root of trust** = trusted (in the sense that it can violate security / hurt you)
  - “The complete set of Roots of Trust has at least the minimum set of functions to enable a description of the platform characteristics that affect the **trustworthiness** of the platform.”
  - Leap from *trusted components* to **trustworthiness platforms**. Destroys the careful distinction between “trusted” and “trustworthy” systems

Claudia Diaz - K.U.Leuven
Decentralized access control

- Evolution towards more complex systems:
  - Multiple authorities and decision points, complex policies, delegation of access rights
  - Policy enforcement amounts to more than checking an access control list stored with a protected resource
  - Access rights are often given because they are required to do a job, not to express trust:
    - Need-to-know principle: grant access to resources based on operational needs (nothing to do with “trust”)
Trust management

- **Trust management**, as introduced in KeyNote\(^1\) and PolicyMaker\(^2\) was used as a term to distinguish a new and more general decentralized approach to access control
- “Trust management is supposed to be an incredibly **vague** and provocative term invented by Matt Blaze” – Joan Feigenbaum
- “Fancy name for (distributed) access control systems where access control decisions can be ‘delegated’” – Dieter Gollmann

- Complexity is the enemy of security
  - With multiple authorities setting policies and delegation of access rights, it is very hard to define enforcement mechanisms
  - Gollmann: “‘Precise’ mechanisms that explicitly handle each possible case might become unwieldy, if not altogether **impossible to design.**”

---

\(^1\) Matt Blaze, Joan Feigenbaum, John Ioannidis, and Angelos D. Keromytis. The KeyNote Trust-Management System Version 2, September 1999. RFC 2704.
PKIs and trust

○ Bruce Schneier: “PKIs are supposed to provide authentication, but they don’t even do that”

○ Redirection to another entity’s site for performing the payment is a common practice, indistinguishable from man-in-the-middle attacks

○ Expired certificates might indicate that the server is compromised, but many legitimate sites also have expired certificates

○ If the user wants to carry out a transaction, they will just click “OK” to any warning pop-ups

○ Registration process to get the certificate?

○ Privacy problem: default disclosure of all attributes in the certificate and full linkability of transactions
Trusted Code

- In code-based access control, access privileges are assigned directly to code, not to users.
- Trusted code: code running with many privileges
- Untrusted code: code running with very few privileges
- Example: code restricted to a Java sandbox would be untrusted
Trusted Code

- A flaw in code running with systems privileges might be exploited by an attacker to take over the victim’s system.

- The same flaw in code that runs with limited privileges would have less serious implications.

- Trusted (= privileged) code is a component that can hurt you.

- However, calling code “trusted” may also insinuate that this is code you can trust, i.e. trustworthy code, which can easily lead to confusion.
Policies and Enforcement

- Access control system:
  - Policy the system is supposed to enforce: specific to a given application
  - Mechanisms that make the access control decisions: generic

- The rationales for setting policies may relate to:
  - Trust in a person or trust in a technology
  - Need-to-know principle
  - Contractual agreements between cooperating entities

- Trusted code or trusted subjects
  - Gollmann: “One might be led to assume that these entities have been given access rights because they are trusted in some way. Maybe even worse, restricting someone’s access rights could be read as an indication of a lack of trust (rather than a simple application of the need-to-know principle)”
Trust is a fashionable but overloaded term with lots of intertwined meanings.

- Overloaded concepts cannot promote precise discussions and clear understanding.
- Security needs precision and clarity.
- 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.
- We would be better off if we avoided using the term trust – if we mean security, let’s just call it security.

Claudia Diaz (K.U.Leuven)
Reliance on trusted entities/components: bad for security

TCBs do not guarantee trust, they ask for it

Trust management does not manage trust but access control

PKIs do not create trust but support existing contractual relationships

Trust paradigms from a privacy perspective:

- Soft privacy: “Trust us, because we care about your security and privacy” (give us your data and rely on our honesty/competence)

- Hard privacy: “Trust us, because you do not need to” (minimize disclosed data)

Claudia Diaz (K.U.Leuven)