Modeling Privacy as Control

Pablo Rauzy

pablo.rauzy@inria.fr

pablo.rauzy.name

Joint work with Daniel Le Métayer



Privatics seminar

20th of January, 2016 @ Corrençon-en-Vercors.

- In 1890, *privacy* is defined as "the right to be let alone".
- Nowadays that is not an option anymore,
  people do not seem to want that either.
- Privacy tends to be defined as *the ability for individuals to control their personal data.*
- The notion of *privacy as control* is predominant in the literature.
- Clear definition of its meaning is still missing.

Challenges

1. Identify what defines control.
2. Propose a language which can model a system in terms of control, and express privacy policies.
3. Define a formal semantics for this language.
4. Develop a tool that evaluates a system's model against a policy.

- In order to understand what is control over personal data
  - I read papers about access control, policy languages, formal models, contextual integrity, ...
  - and documents on the CNIL website,
  - and discussed with Daniel to clarify my ideas :).
- And came up with a few simple notions that enable to capture the concept of control over personal data.

Capacities

- We distinguish 3 axes of control, that we name **capacities**:
  - *usage* of personal data,
  - *authorization* of usage, and
  - *information* on usage.

Operations

- We distinguish 6 types of usage, that we name **operations**:
  - *create*,
  - *access*,
  - *modify*,
  - *delete*,
  - *store*, and
  - *process*.

Purposes

- These 6 generic operations cover all possible data usage.
- However, we need to know the **purpose** of the usage, typically for authorizations.
- This is also a legal requirement in many cases (e.g., cookies).

Locations

- All personal data usages involve treatments on machines.
- These machines are not always under the user's control.
- We need the **location** of each usage to be specified in the model.

Exposure

- Accurate **exposure** information is necessary for informed consent.
- Exposure has two dimensions:
  - information on the number of usages,
  - required effort to be informed of usage.

- We introduce formal definitions of
  - *verbs*: capacity, operation,
  - *agent*: user or service,
  - *relation between agents*: name, agent, agents in the relation,
  - *personal data*: type, referenced agents, embedded personal data,
  - *purpose*: name, can be specialized for an agent,
    (a partial order may be defined on purposes).
  - *location*: name, set of agents that control it,
  - *distance*: name,
    (a total order must be defined on distances).
  - *exposure*: number.

- These notions are used to write *statements*, which describe the control in the modeled system.
- There are 3 types of statements, corresponding to the 3 capacities:
  - Usage statements:
    - agent act·*operation* data, purpose, location
  - Authorization statements:
    - agent grant·*operation* subject, data, purpose, location
  - Information statements:
    - agent know·*operation* subject, data, purpose, location, distance
    - agent know·*operation* #n, data, purpose, location, distance

Example: newsletter service subscription

- Agent u subscribes to a weekly newsletter from agent Service:
  - u grant·access Service, email(u), weekly-mail, laptop(u).
  - u grant·store Service, email(u), weekly-mail, server(Service).
  - u know·access Service, email(u), weekly-mail, server(Service), easy.
  - u know·store Service, email(u), commercial-use, server(Service), hard.

System Policy

- A policy is a Boolean expression over statements.
- In a policy, statements can use wildcards.

Example: commercial use of email address needs consent

- If a user's email address is sold to a partner of Service,
  the user must have agreed to commercial use of their data,
  and it must be obvious to them:
  - $\forall \mathrm{p} \in \mathit{partners}(\mathrm{Service}), \forall \mathrm{u} \in \mathit{subscribers}(\mathrm{Service}),$
    p act·access email(u), *, server(Service)
    $\Longrightarrow$ u know·store Service, email(u), commercial-use, server(Service), easy
    $\wedge$ u grant·store Service, email(u), commercial-use, server(Service).

- The semantics is given by translation to SQL.
- Models are given kind of a denotational semantics, they are translated to `INSERT` queries.
- Policies are given kind of an operational semantics, they are translated to `SELECT` queries.

- The model is translated to `INSERT` queries that are executed against an empty database to populate it.

relations
label
subject
agent

pdata
id
datatype

pdatum_agents
pdatum
agent

pdatum_data
pdatum
datum

purposes
id
label
agent

purposes_cmp
inf
sup

locations
label
agent

distances
label
distance

usages
operation
agent
pdatum
purpose
location

authorizations
operation
agent
subject
pdatum
purpose
location

informations1
operation
agent
subject
pdatum
purpose
location
distance

informations2
operation
agent
exposure
pdatum
purpose
location
distance

Policy

- Each statement in the policy is translated to a `SELECT` query
  - verify if at least a matching statement can be found in the model,
  - if so, the statement is valued to true in the policy.
- In the case of implications
  - the consequent should be true for each variables and wildcards instantiation for which a statement match the antecedent.

1. Finalize semantics.
2. Write a proper real-world but small example.
3. Submit paper to CSF 2016 privacy track (deadline: Feb. 12)
4. Develop real implementation with nice interface and output report.
5. Write more examples.
6. Automatic modeling?
   (from application/network behavior, logs, etc.)

Privacy as Control
- Challenges

1. Control over Personal Data
   - Capacities
   - Operations
   - Purposes
   - Locations
   - Exposure
2. Control Modeling Language
   - System Modeling
     - Example: newsletter service subscription
   - System Policy
     - Example: commercial use of email address needs consent
3. Semantics
   - Model
   - Policy

Roadmap