Type checking privacy policies in the π-calculus

Anna Philippou
University of Cyprus

Joint work with Dimitrios Kouzapas
Imperial College and University of Glasgow

What is Privacy?

- No single definition
  - Different definitions for privacy are subject to philosophy, legal systems
  - Different definitions in different societies
- From a legal point of view privacy can be seen as a collection of individual's rights.

Why privacy?

- Technology giving rise to new privacy concerns
- New practices relating to the handling of personal information
  - Databases allow the aggregation of personal information
  - Electronic health care record systems
  - Social networks
  - Cloud computing
- Challenges
  - Propose methodologies to protect individuals from violation of their right to privacy
  - Provide solid foundations for a rigorous understanding of privacy rights, threats and violations

Privacy and Formal Methods

- M. C. Tschantz and J. M. Wing. *Formal methods for privacy*. In Proceedings of FM'09, LNCS 5850, pages 115. Springer, 2009.
- A study that discusses the need for formal methods for understanding privacy in the context of information handling.

Privacy and Formal Methods

- The arguments follow a taxonomy of privacy violations from Solove [Sol06]:
  - Invasion
  - Information collection
  - Information processing
  - Information dissemination
- Model of three entities
  - The data subject
  - The data holder
  - The environment (authorized/unauthorized adversaries)

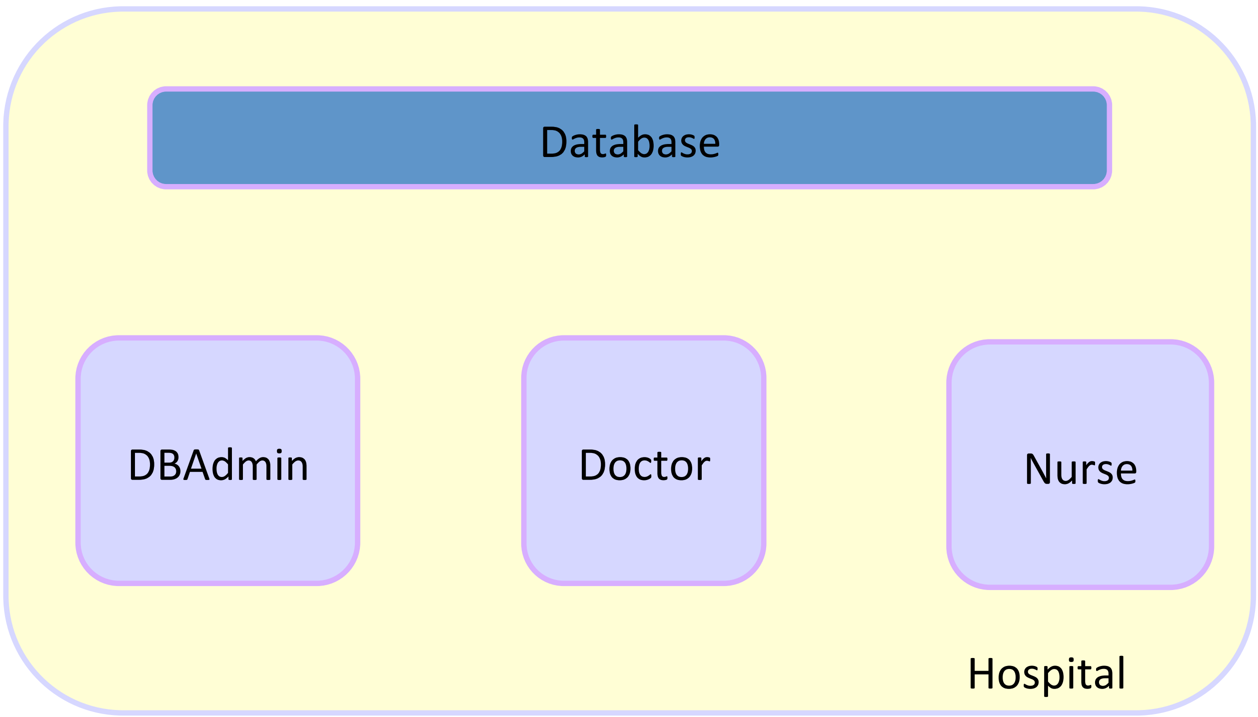
[Sol06] D. J. Solove. A Taxonomy of Privacy. *University of Pennsylvania Law Review*, 154(3):477-560, 2006

Privacy and Behavioral Types

- The π-calculus
- Rich theory in operational, behavioral and typing semantics.
- Use the π-calculus machinery to model privacy concepts.

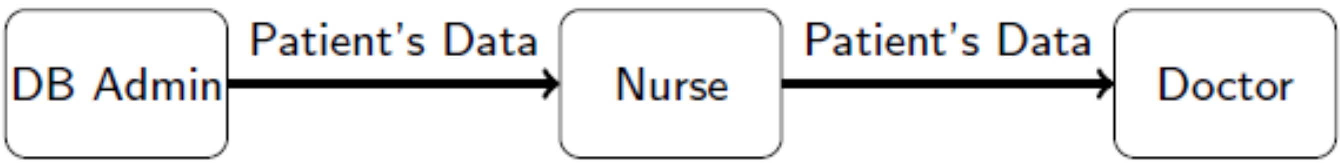
Presentation through example

- A medical database where patient data is stored and accessed by a Database Administrator, a Doctor and a Nurse.



The System

- A Data Base Administrator (data holder) sends Patient's (data subject) data to a Doctor (authorised adversary), using a Nurse (unauthorised adversary) as a delegate.



DBA

$$DBAdmin = tonurse\,\langle c \rangle.0$$

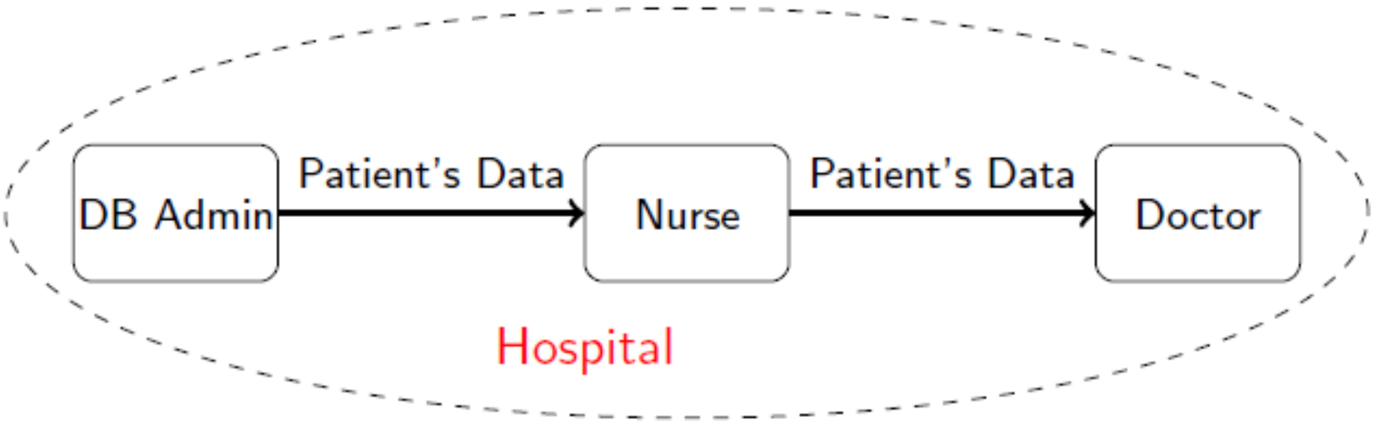
$$Nurse = tonurse(x).\; todoc\,\langle x \rangle.0$$

$$Doctor = todoc(y).y(z).\; y\,\langle data \rangle.0$$

Information Collection

- Requirement 1: No external adversary will be able to access the patient's data.

- Proposed Solution: Use of groups – π-calculus with groups [CGG05]



(ν Hospital) (DBAdmin| Nurse | Doctor) | External

[CGG05] L. Cardelli, G. Ghelli and A. D. Gordon. Secrecy and Group Creation. *Information and Computation*, 196(2): 127-155, 2005

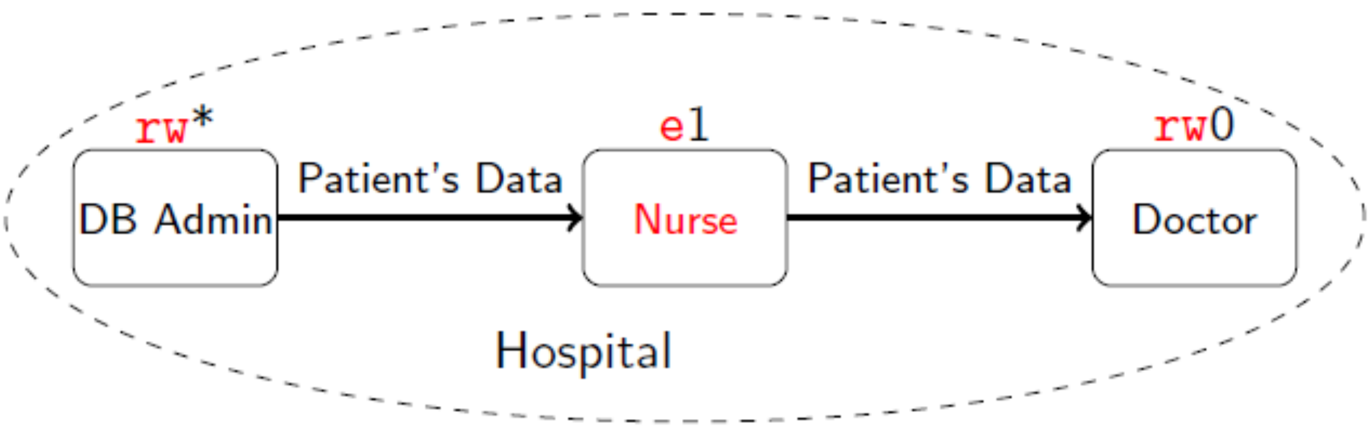
Information Processing

- Requirement 2: A doctor may read and write patient data and a nurse may neither read nor write patient data.

- Proposed solution:
  1. Assign group memberships to distinguish between different "roles"

     (ν Hosp) ((ν DA) DBAdmin| (ν N) Nurse | (ν D) Doctor)

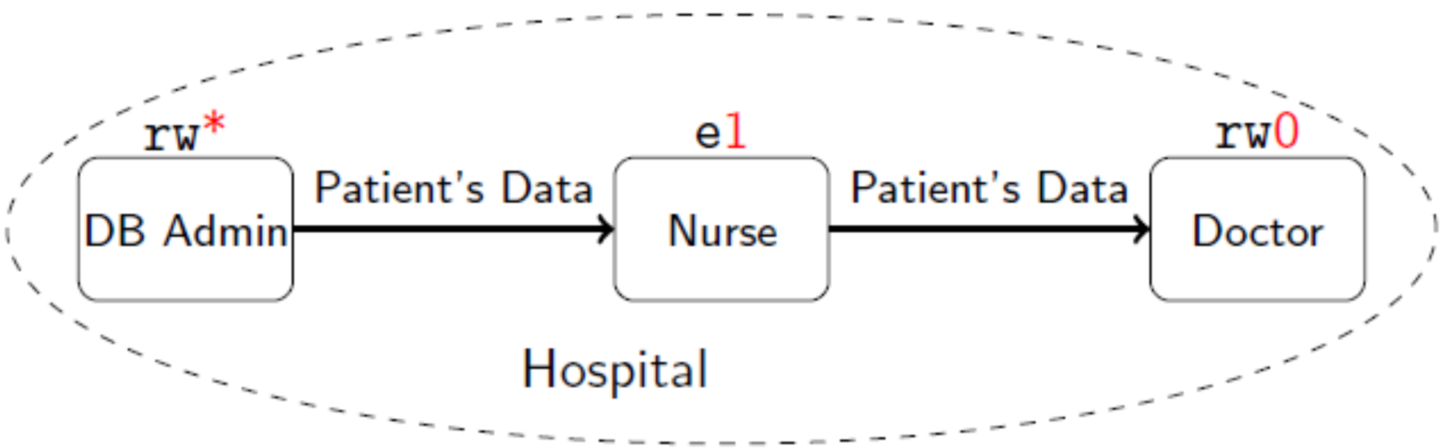
  2. Use i-o types for the π-calculus to prevent access from unauthorised adversaries.



Information Dissemination

- Requirement 3: An administrator may forward the address of a patient's file for an unlimited number of times. A nurse may forward such data once but a doctor must not forward such data.

- Proposed solution:
  - Use of the notion of linear usage of names



Policy Compliance

- Does the system comply with Requirements 1-3?
- Methodology:
  - Infer a type interface of the system
  - Express requirements in a formal language of policies
  - Compare type interface with policy – compatibility
- Main result:

If $\Gamma \vdash Sys \rhd \Theta$ and $\mathcal{P}$ is compatible with Θ then then Sys satisfies policy $\mathcal{P}$.

π-calculus with groups

- Syntax

$P ::= x(y{:}T).P \mid x\langle z\rangle.P \mid (\nu\, a{:}T)P \mid P{\downarrow}1 \mid P{\downarrow}2 \mid\ !P \mid 0$

$S ::= (\nu\, G)P \mid (\nu\, G)S \mid (\nu\, a{:}T)P \mid S{\downarrow}1 \mid S{\downarrow}2 \mid 0$

- Group membership central in defining privacy-related properties
  1. They impose a boundary on the use of names
  2. They characterize the "roles" of processes

- Structural congruence respects this fact:
  - We disallow equivalence

$$(\nu\, G)(S_1 \mid S_2) \equiv (\nu\, G)S_1 \mid S_2 \text{ if } G \notin \mathrm{fg}(S_2)$$

- Operational semantics defined accordingly

Types and Subtyping

- Types:

$$T ::= \mathrm{BT} \mid G[T]\uparrow p\lambda$$

$$p ::= \mathrm{e} \;/\; \mathrm{r} \;/\; \mathrm{w} \mid \mathrm{rw}$$

$$\lambda ::= * \mid i \qquad\qquad i \geq 0$$

- $x : G[T]^{p\lambda}$:
  - name x can be used within group G in input/output position according to p to communicate objects of type T and up to λ times in object position.
  - e.g. $x{:}\mathrm{Hosp}[\mathrm{Pdata}]^{rw0}$

- Subtyping:
  - input co-variance and output contra-variance
  - coinductive definition

The typing system

- Type environment

  $\Gamma,\Delta ::= \emptyset \;\; / \;\; \Gamma\cdot x{:}T \;\; / \;\; \Gamma\cdot G$

- Type interface

$\Theta ::= \varepsilon \quad | \quad \langle G\downarrow 1 \cdot\ldots\cdot G\downarrow n : \Gamma\rangle\cdot\Theta$

- Typing judgments
  - $\Gamma \vdash x \rhd T$

    In typing environment Γ name x has type T
  - $\Gamma \vdash P \rhd \Delta$

    In typing environment Γ process P is well typed and produces type environment Δ
  - $\Gamma \vdash S \rhd \Theta$

    In typing environment Γ system S is well typed and produces type interface Θ

Typing Rules

- Subsumption

$$(\mathsf{SubsP})\quad \frac{\Gamma \cdot x : T' \vdash P \triangleright \Delta \qquad T' \leq T}{\Gamma \cdot x : T \vdash P \triangleright \Delta}$$

- Input

$$(\mathsf{In})\quad \frac{\begin{array}{c}\Gamma \cdot y : T \vdash P \triangleright \Delta \qquad \Gamma \vdash x : G_x[T']^{r0} \\ (\Delta \uplus y : \mathsf{iperm}(T))(y) = T'\end{array}}{\Gamma \vdash x(y : T).P \triangleright \Delta \uplus y : \mathsf{iperm}(T) \uplus x : G_x[T']^{r0}}$$

Typing Rules

- Group restriction on processes

$$\text{(ResGP)} \quad \frac{\Gamma \cdot G \vdash P \triangleright \Delta}{\Gamma \vdash (\nu\ G)P \triangleright \langle G : \Delta \rangle}$$

- Group restriction on systems

$$\text{(ResGS)} \quad \frac{\Gamma \cdot G \vdash S \triangleright \{\langle \tilde{G}_i, \Delta_i \rangle\}_{i \in I}}{\Gamma \vdash (\nu\ G)S \triangleright \{\langle G, \tilde{G}_i : \Delta_i \rangle\}_{i \in I}}$$

Policies

- Policies assign a set of permissions (positive and negative) to each group for each base type.

- Permissions

  Per = {read, write, forward λ, exclude, nondisclose}

- Policies

$$\mathcal{P} ::= BT \gg H \mid \mathcal{P};\mathcal{P}$$

$$H ::= G{:}P[H{\downarrow}i]{\downarrow}i \in I \qquad \text{where } P \subseteq \text{Per}$$

Policies vs types/processes

Definition (Compatibility)

A policy $\mathcal{P}$ is compatible with a type interface Θ if any permission exercised by the type interface is allowed by the policy.

Definition (Error Process)

A system S is an error process with respect to policy $\mathcal{P}$ if it exercises actions that violate the requirements of the policy.

Results

Theorem 1 (Subject Reduction)

Suppose $\Gamma \vdash \mathcal{S} \rhd \Theta$ and $\mathcal{S} \longrightarrow \mathcal{S}'$ then $\Gamma \vdash \mathcal{S}' \rhd \Theta'$ and $\Theta \leq \Theta'$.

Theorem 2 (Safety)

If $\Gamma \vdash \mathcal{S} \rhd \Theta$, interface Θ is compatible with policy $\mathcal{P}$ and $\mathcal{S} \longrightarrow\!\upharpoonright * \ \mathcal{S}'$ then $\mathcal{S}'$ is not an error process with respect to policy $\mathcal{P}$.

Example

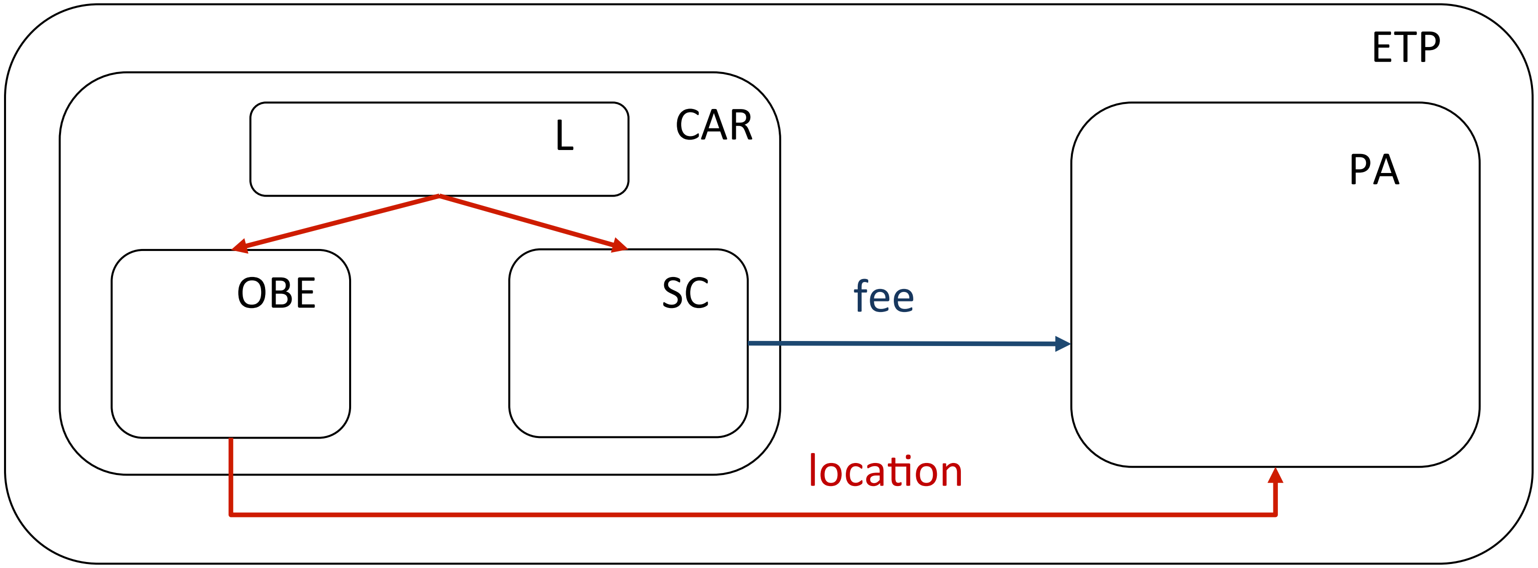
- Electronic traffic pricing
  - Toll collection scheme where the fee to be paid depends on road usage
  - Location information must be collected and processed in order to compute fee
  - Privacy and security threats

- Approaches
  - Centralized: all information is communicated to the Pricing Authority
  - Decentralized:
    - Fee is computed locally (on car) with the aid of a third trusted entity (e.g. smart card).
    - *Some* location information must be communicated to the Pricing Authority to ensure that information provided to TTC is not tampered with.
  - …

The decentralized approach

- SC: The smart card
  - It receives all information about whereabouts of the car and computes the fee to be paid which it communicates to the Pricing Authority
- OBE: The on-board equipment
  - It responds to spot checks performed by the Pricing Authority
- L: the component responsible for computing the current location of the car
- PA: The pricing authority:
  - It communicates with the SC to obtain the fee to be paid and it performs spot checks to confirm that the SC is provided with correct information

The model



The model

$$
\begin{aligned}
S &= !read(loc : T_l).loc(l : \mathsf{Loc}).(\nu\ newval : Fee)\overline{fee}\langle newval\rangle.\overline{send}\langle fee\rangle.\mathbf{0} \\
O &= spotcheck(s_1 : T_x).read(ls_1 : T_l).\overline{s_1}\langle ls_1\rangle.spotcheck(s_2 : T_x).read(ls_2 : T_l).\overline{s_2}\langle ls_2\rangle.\mathbf{0} \\
L &= !(\nu\ newl : T_l)\overline{read}\langle newl\rangle.\mathbf{0} \\
A &= !(\nu\ x : T_x)\overline{spotcheck}\langle x\rangle.x(y : T_l).y(l_s : \mathsf{Loc}).\mathbf{0} \\
&\quad \mid send(fee).fee(v : \mathsf{Fee}).\mathbf{0} \\
\mathsf{System} &= (\nu\ \mathsf{ETP})(\nu\ spotcheck : T_{sc})(\nu\ topa : T_{pa}) \\
&\quad [\ (\nu\ \mathsf{PA})A \ \mid\ (\nu\ \mathsf{Car})((\nu\ read : T_r)((\nu\ \mathsf{OBE})O \mid (\nu\ \mathsf{GPS})L) \mid (\nu\ \mathsf{SC})S)\ ]
\end{aligned}
$$

The policy

- Two types of basic types: Location and Fee

- Policy for locations:

$$
\begin{aligned}
&\mathsf{Loc} \gg \mathsf{ETP} : \mathsf{nondisclose}\ [\\
&\qquad \mathsf{Car} : [\\
&\qquad\qquad \mathsf{OBE} : \{\mathsf{forward}\ 2\}\\
&\qquad\qquad \mathsf{GPS} : \{\mathsf{forward}\ *\}\\
&\qquad\qquad \mathsf{SC} : \{\mathsf{read}\}],\\
&\qquad \mathsf{PA} : \{\mathsf{read}\}\\
&\qquad]
\end{aligned}
$$

Analysis

- We may
  - show that that $\Gamma \vdash S \rhd \Theta$ where Θ exercises the following rights on base type Loc

$$\{\mathsf{ETP} \cdot \mathsf{PA} : \{\mathsf{read}\}, \mathsf{ETP} \cdot \mathsf{Car} \cdot \mathsf{OBE} : \{\mathsf{forward}\,2\},$$
$$\mathsf{ETP} \cdot \mathsf{Car} \cdot \mathsf{GPS} : \{\mathsf{forward}\,*\}, \mathsf{ETP} \cdot \mathsf{Car} \cdot \mathsf{SC} : \{\mathsf{read}\}\}$$

  - And confirm that Θ is compatible with the policy.

Concluding remarks (1)

- A type system for reasoning about basic instances of *information collection*, *information processing* and *information dissemination.*
  - Contextual integrity
  - Privacy-aware role-based access control (P-RBAC)

- Extend theory to handle
  - Dynamicity
  - Pre- and post- obligations of P-RBAC
  - Policy composition
  - Other forms of privacy

Concluding remarks (2)

- Privacy poses new challenges
  - Models, logics, languages, analyses, tools

- Concurrency Theory has the potential of addressing these challenges (behavioral relations, type systems) and it is already proposing solutions (secrecy, anonymity, unlinkability, differential privacy).

- Challenges:
  - Foundations for privacy-related concepts and their interconnections
  - Methodologies for transferring results to real systems