A Framework for Secure Data Collection and Management for Internet of Things

Maribel Fernández,
Murat Kantarcioglu,
Bhavani Thuraisingham

King's College London, University of Texas at Dallas

ICSS, December 2016

Internet of Things

Main notions:

- sensors
- aggregators (devices)
- communication channels
- external utilities (e.g., web services)
- decision triggers

Problem:
Sensitive data transmitted from IoT devices to external utilities.
+ Data collected from IoT devices can be used to provide better services to users.
- Users may become vulnerable as a result of using an IoT device (security, privacy risks).

A Layered System

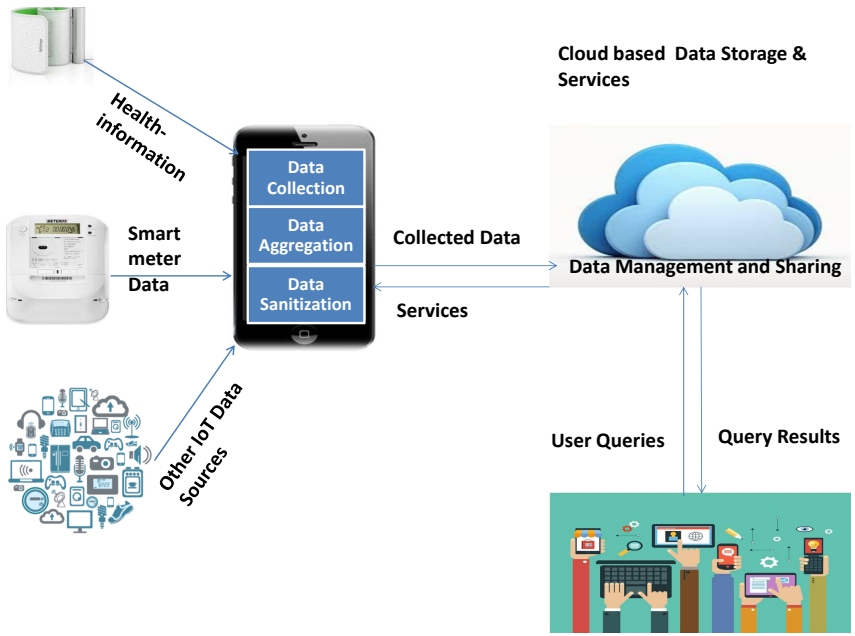

Figura : Layered System

Data Collection: Control

Users must be able to control which/when data is tramitted by their devices.

Dual to access control: Push/Pull.

$\Longrightarrow$ **Integrated data collection, storage and management model + policy languages, enforcement mechanisms, reasoning techniques.**

Challenges:
Variety of devices (hardware/software limitations), policy specification and enforcement, policy composition/update...

Contributions

Main contribution:

Layered metamodel that specifies the essential principles underlying the specification, analysis and enforcement of policies for secure management of data generated by IoT devices.

This paper focuses on data collection: Top layer.

Axiomatic, category-based metamodel for data collection control.
Inspired by the CBAC (category-based access control) metamodel.

Preliminaries: Access Control Policies

Recall:

- principals (e.g., human users, software agents)
- resources (e.g., files, directories)
- privileges: actions on resources (e.g., read, write, execute)

Access control policies associate privileges to principals, depending on properties such as ownership, role in an organisation, historical information, etc.

Access requests are granted or denied (in the general case other answers are also possible) according to the policy specification.

A variety of access control models...

each with specific languages, techniques, properties.

- Role-Based
- Mandatory (e.g., [Bell-Lapadula] military applications)
- Event-Based (e.g., DEBAC in banking applications)
- ...

$\Rightarrow$ An Access Control MetaModel [Barker,Sacmat09] based on the primitive notion of a category.

- core set of principles of access control
- abstracts away many of the complexities of specific models

Operational Semantics for the metamodel

Rewriting semantics [Bertolissi and Fernández, ESSOS 2010]:

- provides techniques to prove properties of policies
- languages and tools for rapid prototyping/policy analysis: MAUDE, TOM, Haskell, CiME,

Results:

- Conditions to ensure totality and consistency of policies
- Encoding well-known access control models: RBAC, MAC, DAC and DEBAC
- A distributed/federative metamodel: distributed resources, individual policies combined to make collective decisions.

Category-Based metamodel for Data Collection CBDC

Category: class, group, or domain, to which entities belong

Entities:

- A countable set $\mathcal{D}ev$ of IoT devices $d_1, d_2, \ldots$: to represent data sources and channels; e.g. individual sensors, aggregators, clocks, etc.
- A countable set $\mathcal{DI}$ of data items $di_1, di_2, \ldots$: to represent data emanating from sensors and also contextual information (such as location, time, identifier, etc.)
- A set $\mathcal{A}$ of actions: e.g., send, receive, block, encrypt, decrypt, etc.
- A countable set $\mathcal{C}$ of categories c_0, $c_1, \ldots$
- A countable set $\mathcal{S}$ of services: to represent actual services or users that own/process data (e.g., e-utilities [J. Voas: NoT-NIST]).

Relationships between entities

- *Data-Item Category Assignment:* for each $d \in \mathcal{D}ev$, $\mathcal{DICA}_d \subseteq \mathcal{DI}_d \times \mathcal{C}$, such that $(di, c) \in \mathcal{DICA}_d$ iff the data item $di \in \mathcal{DI}_d$, generated by the device $d \in \mathcal{D}ev$, is assigned to the category c.
- *Action Category Assignment:* $\mathcal{ACA} \subseteq \mathcal{A} \times \mathcal{C}$, such that $(a, c) \in \mathcal{ACA}$ iff action $a \in \mathcal{A}$ can be performed on data items assigned to the category c.
- *Banned-Action Category Assignment:* $\mathcal{BACA} \subseteq \mathcal{A} \times \mathcal{C}$, such that $(a, c) \in \mathcal{BACA}$ iff action $a \in \mathcal{A}$ is banned for data items assigned to the category $c \in \mathcal{C}$.

Relationships between entities

- *Authorisations:* for each $d \in \mathcal{D}ev$, $\mathcal{ADI}_d \subseteq \mathcal{A} \times \mathcal{DI}_d$, such that $(a, di) \in \mathcal{ADI}_d$ iff action $a \in \mathcal{A}$ is authorized on the data item di generated by $d \in \mathcal{D}ev$.
- *Prohibitions:* for each $d \in \mathcal{D}ev$, $\mathcal{BADI}_d \subseteq \mathcal{A} \times \mathcal{DI}_d$, such that $(a, di) \in \mathcal{BADI}_d$ iff action $a \in \mathcal{A}$ is banned on data item di generated by $d \in \mathcal{D}ev$.
- An additional relation $\mathcal{UNDET}_d \subseteq \mathcal{A} \times \mathcal{DI}_d$ contains the tuples (a, di) such that the action a is neither authorized nor banned on the data item di emanating from d.

Example

Truck with sat nav transmitting GPS locations (data-item loc_i).

Tracking service.
Goal: transmit location if truck stolen.

Categories: NormalLoc, AbnormalLoc

Only abnormal locations should be transmitted, so
$(send, AbnormalLoc)$ in $\mathcal{ACA}$
$(send, NormalLoc)$ in $\mathcal{BACA}$.

Axioms

$(dc1)$ $\forall d \in \mathcal{D}ev,\ \forall a \in \mathcal{A},\ \forall di \in \mathcal{DI}_d,$
$(\exists c, c' \in \mathcal{C}, (di, c) \in \mathcal{DICA}_d \wedge c \subseteq c' \wedge (a, c') \in \mathcal{ACA})$
$\Leftrightarrow (a, di) \in \mathcal{ADI}_d$

$(dc2)$ $\forall d \in \mathcal{D}ev,\ \forall a \in \mathcal{A},\ \forall di \in \mathcal{DI}_d,$
$(\exists c, c' \in \mathcal{C}, (di, c) \in \mathcal{DICA}_d \wedge c' \subseteq c \wedge (a, c') \in \mathcal{BACA})$
$\Leftrightarrow (a, di) \in \mathcal{BADI}_d$

$(dc3)$ $\forall d \in \mathcal{D}ev,\ \forall a \in \mathcal{A}, \forall di \in \mathcal{DI}_d,$
$((a, di) \notin \mathcal{ADI}_d \ \wedge \ (a, di) \notin \mathcal{BADI}_d)$
$\Leftrightarrow (a, di) \in \mathcal{UNDET}_d$

$(dc4)$ $\forall d \in \mathcal{D}ev,\ \mathcal{ADI}_d \cap \mathcal{BADI}_d = \emptyset$

Policy

A category-based data collection policy is a tuple

$$\langle \mathcal{E}, \quad \{\mathcal{DICA}_d\}_{d\in\mathcal{D}ev}, \ \mathcal{ACA}, \ \mathcal{BACA}, \\ \{\mathcal{ADI}_d\}_{d\in\mathcal{D}ev}, \{\mathcal{BADI}_d\}_{d\in\mathcal{D}ev}, \{\mathcal{UNDET}_d\}_{d\in\mathcal{D}ev}\rangle$$

where $\mathcal{E} = (\mathcal{D}ev, \mathcal{DI}, \mathcal{A}, \mathcal{C}, \mathcal{S}, \subseteq)$, such that axioms $(dc1)$-$(dc4)$ are satisfied.

Operational semantics: axioms $(dc1)$-$(dc4)$ can be realized through a set of functions, defined by rewrite rules.

$$(dc1') \quad \text{adi}_d(A, Di) \quad \rightarrow \quad \textit{if} \quad A \in \text{aca}^*(\text{below}(\text{dica}_d(Di))) \\ \textit{then accept else} \\ \textit{if } A \in \text{baca}^*(\text{above}(\text{dica}_d(Di))) \\ \textit{then forbid else undetermined}$$

Example: Truck rental company for industrial plant

Rental prices vary depending on whether the truck is taken out of the country or not.
Drivers who are not planning to leave the country can benefit from a discount.

Trucks fitted with tracking devices able to transmit GPS locations.
Goal: tracking information transmitted only if truck crosses border.

Example: Truck rental company for industrial plant

Policy specification:
Assume one truck ($\mathcal{D}ev$ has just one element).
$\mathcal{A} = \{transmit\}$.
$\mathcal{C} = \{home, abroad\}$
$(transmit, abroad) \in \mathcal{ACA}$ and $(transmit, home) \in \mathcal{BACA}$.

$$
\begin{array}{lcl}
\text{dica}_d(Coord) & \rightarrow & \textit{if } \text{inUK}(Coord) \\
 & & \textit{then home else abroad} \\
\text{aca}(abroad) & \rightarrow & [transmit] \\
\text{baca}(home) & \rightarrow & [transmit]
\end{array}
$$

Axioms ensure truck coordinates are only transmitted if truck is abroad.

Example: Chemical production plants

Sensors located in the plants transmit encrypted data (location, identification, quantity of various chemicals produced, etc.) to central monitoring site, where data is decrypted before processing.

Goal: transmit only sanitised versions of sensitive data, e.g., instead of the exact quantity of a specific chemical produced in the plant, a suitably altered (differential privacy) version is transmitted.

Example: Chemical production plants

Policy specification:

$$\mathcal{C} = \{\mathit{topSecret}, \mathit{secret}, \mathit{public}\}$$

$$\mathcal{A} = \{\mathit{tr}, \mathit{average\text{-}tr}, \mathit{altered\text{-}tr}\}$$

$$\begin{array}{lcl}
\text{aca}(\mathit{public}) & \rightarrow & [\mathit{tr}] \\
\text{aca}(\mathit{secret}) & \rightarrow & [\mathit{average\text{-}tr}] \\
\text{aca}(\mathit{topSecret}) & \rightarrow & [\mathit{altered\text{-}tr}] \\
\text{baca}(\mathit{topSecret}) & \rightarrow & [\mathit{tr}, \mathit{average\text{-}tr}] \\
\text{baca}(\mathit{secret}) & \rightarrow & [\mathit{tr}]
\end{array}$$

Secret and top secret data items will be safely transmitted:
Property:
If *di* is a sensitive data item, that is, a data item in the category secret or in the category topSecret, then it will not be transmitted in raw form.

Conclusions - Future work

- Policy languages
- Policy enforcement
- Usability, Risk-Benefit Analysis
- Policy composition
- Architecture/Techniques (differential privacy, encryption) ...