Security and Privacy in a Ubiquitous World

Clemens H. Cap
University of Rostock
clemens.cap@computer.org
Smart Labels

Object Identity
Interaction Type
Interaction Circumstances
  Object Location & Orientation
  Time of interaction
  Additional parameters

Absolute: eg. Geographical coordinates
Relative: eg. To known object
Semantic: eg. Contextual interpretation

Absolute: eg. UTC
Relative: eg. Simultaneously, After
Semantic: eg. Contextual interpretation

Environment: eg. Temperature
Object properties: eg. Size, Ownership
Object dynamics: eg. History

Read only fields
Writeable fields
Associative fields
Sensor fields
<SUITCASE>
  bought-by: Clemens Cap
  bought-at: Kaufhof
  location: 49° 33' 22'', 23° 23', 34''
  location: Rostock
  location: Car with license plate HRO-XC7
  content: 1 blue jeans, 5 shirts, ...
  value: 500.- USD
</SUITCASE>
Shadow World Assumption

We shall assume

- Every object carries a label
- High density of readers

Realistic assumption?

- Costs
- Standards & Interoperability
- Benefits
# Shadow World Assumption

**We shall assume**
- Every object carries a label
- High density of readers

**Realistic assumption?**
- Costs
- Standards & Interoperability
- Benefits

<table>
<thead>
<tr>
<th>Capacitive coupling</th>
</tr>
</thead>
<tbody>
<tr>
<td>- No copper coils</td>
</tr>
<tr>
<td>- Printed antenna</td>
</tr>
<tr>
<td>- Defect tolerance</td>
</tr>
<tr>
<td>- Motorola Bistatix</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Polymer based logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Easier process</td>
</tr>
<tr>
<td>- Promising examples</td>
</tr>
<tr>
<td>- Infineon / Erlangen / Ulm</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Economies of Scales</th>
</tr>
</thead>
</table>
Shadow World
Assumption

We shall assume
- Every object carries a label
- High density of readers

Realistic assumption?
- Costs
- Standards & Interoperability
- Benefits
Example

If my shoes leave my house without my umbrella and there is a forecast for rain, then inform me accordingly.

Reactive system models
- Higher order Petri net
- Chemical Abstract Machine
- Enhanced linear logic

Tokens from Smart Labels

Reactive bowl

Reactions

Actuators

Actuator coupling
forall loc, t1:
    shoes (loc, t1) -o arm (loc, t1) * shoes (loc, t2)
forall loc, t1, t2: if t2 - t1 < C then
    arm (loc, t1) * umbr (loc, t2) -o 1
forall loc, t1, t2: if t2 - t1 > C then
    arm (loc, t1) -o beep (loc, t2) @ t2

- Language to describe intended behaviour of system
- Logic to reason about behaviour of the system
- Implementation straight forward
- Limited control on garbage collection via resource destruction
A Short Story

Diagram:
- Home
- Bank
- Casino
- Red Light District
- Prison
So what is the story?

- A family tragedy?
- A policeman on his daily tour?
- A mafia boss caught on his daily tour?
- A medical doctor called in for an emergency?
- A taxi driver at work?
- . . .
**Lessons learned so far**

| Lesson 1: | Raw sensor data is practically meaningless |
| Lesson 2: | Derivation of semantics is (very) difficult  
      | Additional info may be required |
| Lesson 3: | Mining in raw sensor data can be misleading |
| Lesson 4: | Must protect raw sensor data |
Technical Approaches (1)

No security
- Everyone can read / write / access label
- Attack: Buy compatible reader / label

Password protection
- Password used to read / write / access label
- Structure: Several passwords & access areas
- Attack: Crack password
  (but: blocking mechanism) [but: DOS attack] {but: reader auth}
- Attack: Replay password
- Attack: Sniff the password
  (but: encrypt it) [but: replay attack]
Technical Approaches (2)

**Rolling code system**
- Get a new password every time
- Synchronize time of generating device (SecureID token)
- Synchronize state of generating device (car alarm)
  But: out-of-synch, state replication

**Challenge response**
- Reader provides a challenge
- Label calculates a response
- Attack: Man-in-the-middle
  (but: reader must provide proper challenge)
Overall Situation

Label

Reader

Processor

Actuator

secure

secure

secure

secure
Requirements

Processor must be implemented as a
- distributed
- multiparty protocol
- between sensors (and maybe computing nodes)
- with input privacy
- and resilience against cheating participants

Basic result (Yao; Chaum et al; Goldreich et al.)
- can be done if not too many cheaters are present

Example for equality of owner of shoes and umbrella
Some observations
(user interviews in the FASME project)

Observation 1: The privacy & most security issues are mainly in our minds and hence must be treated accordingly

Observation 2: Privacy must be enforced by technology, not by regulations

Observation 3: Privacy must be visible to the user

Observation 4: User must be able to check what is stored about him