



Trust for Ubiquitous, Transparent Collaboration

5th May 2004

Nathan Dimmock, Opera Group

Computer Laboratory

University of Cambridge

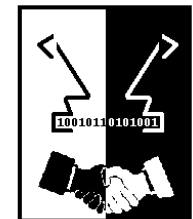
`Nathan.Dimmock@cl.cam.ac.uk`

Overview

- The need for unobtrusive security in ubiquitous computing
- Unifying trust assessments and access control
 - Using risk analysis to determine “how much” trust is required
- User interface design
- Conclusions and future work
- Acknowledgements
 - EU-funded SECURE Project

SECURE

Secure Environments for Collaboration
among Ubiquitous Roaming Entities

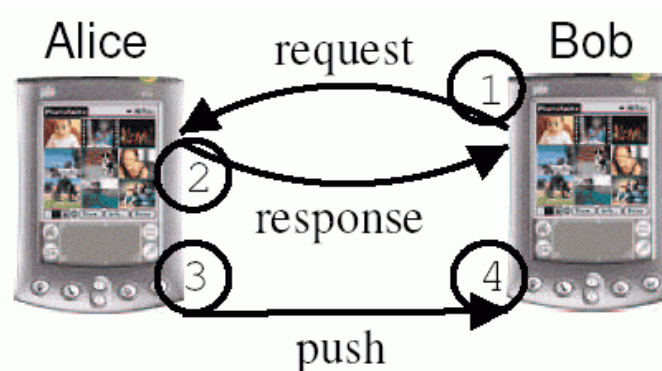


Motivation

- Ubiquitous computing requires trust between participants to support collaborative tasks
 - Example: scheduling a meeting between employees of two different companies
- Sensitive information must be protected, but:
 - User's attention is a scarce resource in pervasive computing
 - Too many existing security systems fail due to high administrative overhead
- Therefore security measures must be proportional to the risk involved

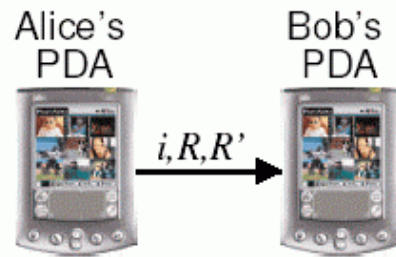
Phone Book Example

- Homogenous recommendation system to allow exchange of privileged information



- Principals can be associated with categories
 - Membership of a category may confer certain access control capabilities
 - Extend RBAC roles by associating trust assessment with category assignment

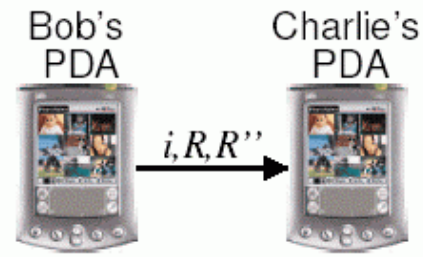
Trust Infrastructure



$$i = \{\text{Phone \#}\}_{\text{Alice}}$$

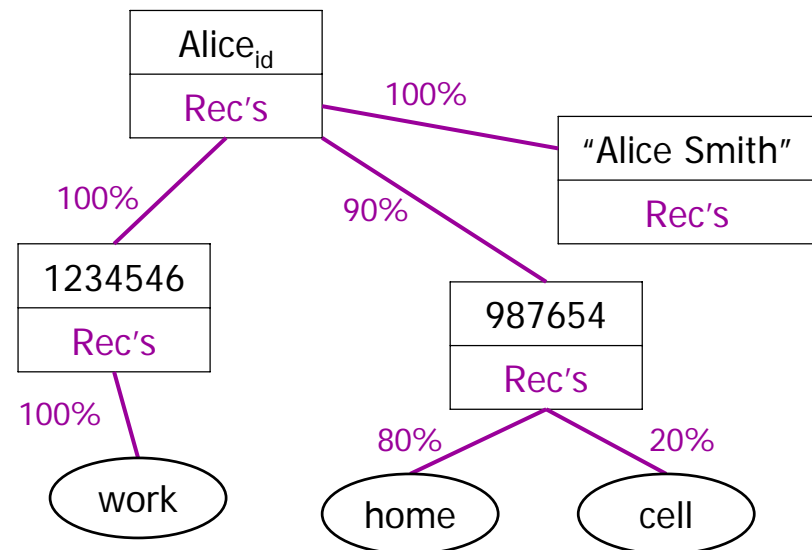
$$R = \{\text{Rec}(i, \text{Alice}, t)\}_{\text{Alice}}$$

$$R' = \{\text{Rec}(i, \text{work}, t')\}_{\text{Alice}}$$



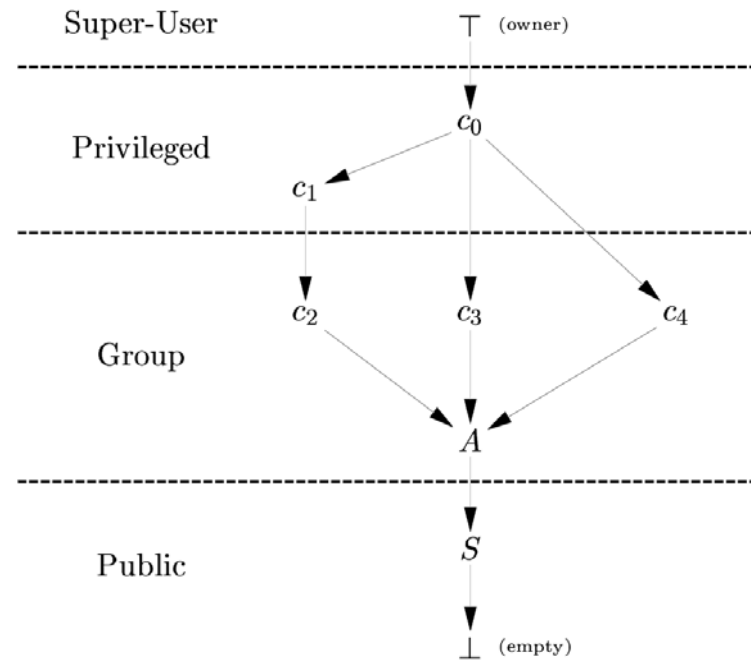
$$R'' = \{\text{Rec}(i, \text{work}, t'')\}_{\text{Bob}}$$

- Phone book example:
 - Alice gives her number to Bob
 - Later, Bob forwards it to Charlie



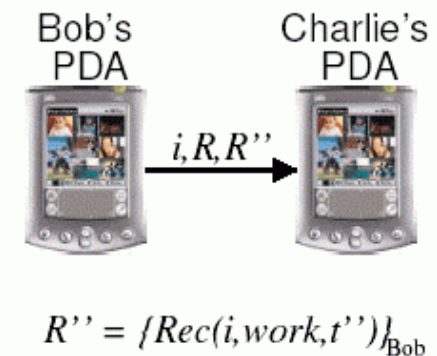
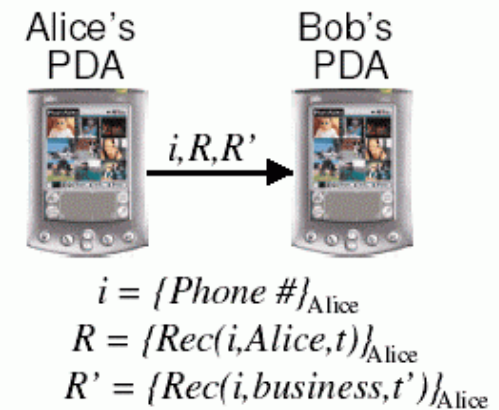
Categories and Roles

- Categories arranged in a partial order
 - Privilege lattice
 - c_1 extends privileges of c_2
 - c_1 may not read c_4
 - Lattice divided into bands
 - Dictate extra privileges for categories within them
- Same category structure used for data and principals
 - Association between names and numbers re-used for access control



Trust Model (1)

- Recommendations naturally express participants' trust beliefs.
- Recommendations factorise policy.
 - Incremental distribution
 - Recommendations can be made to expire
- Chained recommendations allow delegation and transfer of trust.

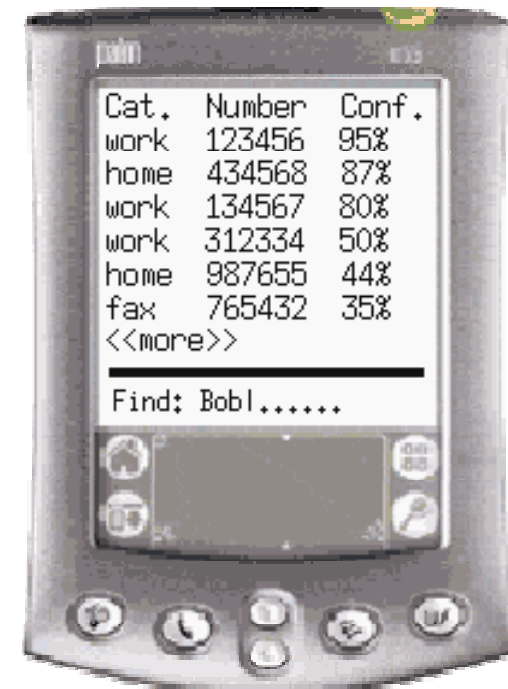


Risk Assessment

- Security must be proportional to risk involved
 - Risk of interacting with a person is total risk of all possible outcomes from the interaction
 - Risk of an outcome is (*maximum potential cost, likelihood of outcome*)
 - Potential cost depends on value of data involved
- Address book example:
 - Position of data item in category lattice assigns it an implicit value
 - Values of categories are user assigned
 - Principal's category assignment determines their likely behaviour

User Interface

- Trust-model also invoked when owner wishes to view information
 - Search for “Bob” gives 10 numbers linked to him with varying degrees of confidence
 - Too many for PDA screen
 - Display ordered by (category value) × (belief in category membership)
- Feedback from owner
 - User interface designed to allow quick feedback when owner e.g. discovers a number is invalid
 - Implemented as recommendation from owner
 - Implicitly highly trusted



Conclusions and Future Work

- Unobtrusive and mostly automated security model for ubiquitous devices
 - Trust-evaluated recommendations highly applicable in pervasive environment
 - Explicit risk analysis gives appropriate security
 - Address book application prototype
- Future work
 - User acceptance and evaluation study
 - Automatic detection of untrustworthy principals
 - See <http://secure.dsg.cs.tcd.ie/>

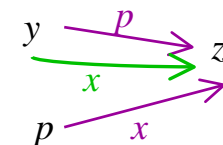
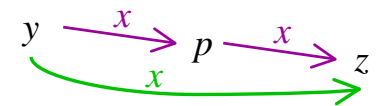
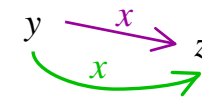
Trust Model (2)

- Recommendations associate one permission with another in this trust model
 - Actors, Categories, Data entries, Read/Write actions, Links
 - Trust (*belief, disbelief*) pairs: $0 \leq \text{belief} + \text{disbelief} \leq 1$
 - (0,0) is unknown, (1,0) is full trust
- Trust policy computation
 - The extent to which x believes y holds permission z:

$$Pol_x(T, y, z) = \bigoplus \left\{ d_x(y, z) \right.$$

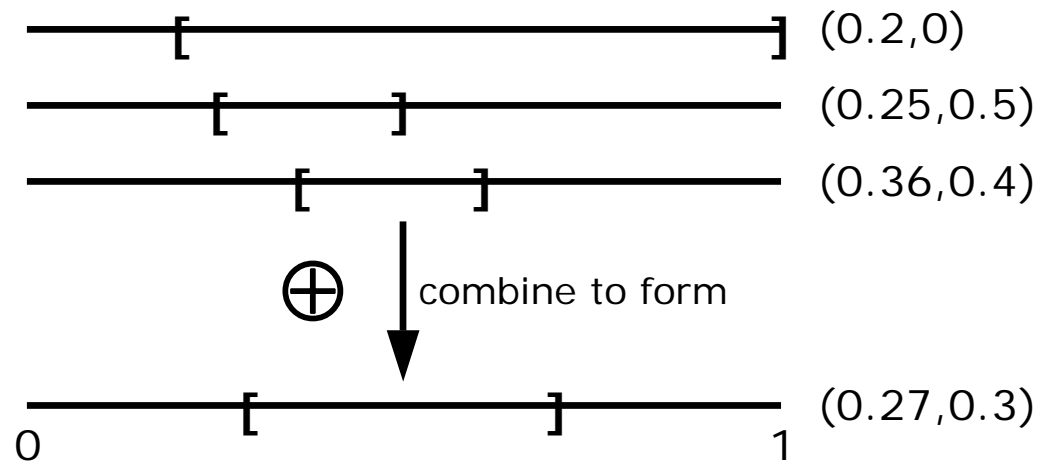
$$\cup \bigcup_{p \in \mathcal{P}} d_x(y, p) \otimes T(x, p, z)$$

$$\cup \bigcup_{p \in \mathcal{P}} d_x(p, z) \otimes T(p, y, z) \left. \right\}$$



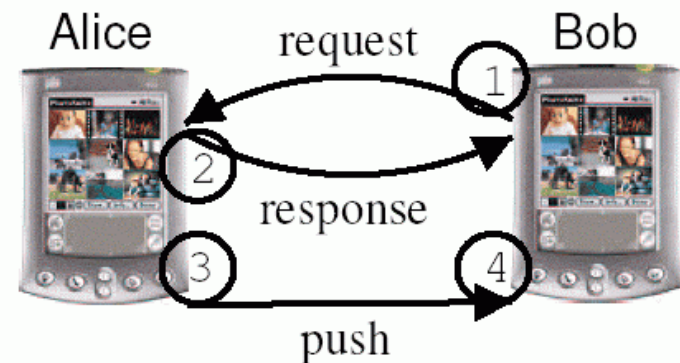
Example of Trust Policy

- Given three recommendations:
 - $(0.2,0)$, $(0.25,0.5)$ and $(0.36,0.4)$
 - The compound recommendation deduced by \oplus is $(0.27,0.3)$
- Trust engine combines all recommendations in a consistent manner

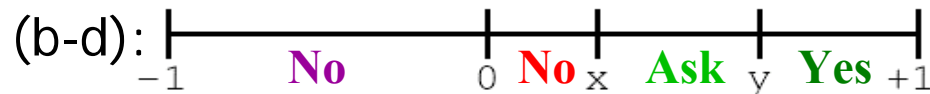


Address Book Decision Making

- Two users may interact in two different ways
 - User must decide whether to participate or not
 - Cost / benefit analysis, incl. value of user's time



- "Is *Bob* a sufficiently strong member of *c* to read *c'*?"



- Sufficiently strong* determined by risk assessment
- Equivalent to: **Answer** = **if** $\text{Benefit}_{no} \geq 0$ **then** "No"
else if $\text{Benefit}_{yes} > 0$ **then** "Yes"
else if $\text{Benefit}_{ask} > 0$ **then** "Ask"
else "No"

Implementation Issues

- Inherent risks in second-hand information
 - Out of date information
 - Validity periods
 - Decay
 - Disinformation – false recommendations
 - Double counting of original information
 - Especially in chained recommendations
- Monotonicity requirement constrains choice of \oplus
- Automatic detection of untrustworthy peers is difficult

Limitations

- No hierarchy of recommendations
 - arbitrarily long chains can be formed
- Monotonicity constrains choice of \oplus
 - Recommendation chains are followed backwards ...
- Recommendation subsets may be misleading

Illustration: Address Book Application



$i = \{Phone\ \#\}_{Alice}$
 $R = \{Rec(i, Alice, t)\}_{Alice}$
 $R' = \{Rec(i, business, t')\}_{Alice}$

$R'' = \{Rec(i, work, t'')\}_{Bob}$

(a) Alice sends Bob her work phone number.

(b) Some time later, Bob forwards Alice's number to Charlie.

